

Software Process Definition using Process Lines: A Systematic Literature Review

Diogo Matheus Costa
Systems Engineering and Computer
Science Department
PESC/COPPE/UFRJ
Rio de Janeiro, Brazil
dmc@cos.ufrj.br

Eldânae Nogueira Teixeira
Systems Engineering and Computer
Science Department
PESC/COPPE/UFRJ
Rio de Janeiro, Brazil
dannyc@cos.ufrj.br

Claudia Maria Lima Werner
Systems Engineering and Computer
Science Department
PESC/COPPE/UFRJ
Rio de Janeiro, Brazil
werner@cos.ufrj.br

Abstract—Software processes have been the focus of discussion in literature, but defining a software process remains a challenge. The Software Process Line (SPrL) technique offers a systematic process reuse approach that identifies processes' similarities and variability. This study aims to characterize the state-of-the-art of the software process definition using SPrL. A Systematic Literature Review (SLR) was conducted and 26 papers were identified to be studied in depth. The results indicate a concentration of approaches that use mapping/rules techniques to support variability solution and the need for integrated supporting tools. In addition, three problems that affect the software process definition using SPrL were observed: (1) low understanding of SPrL models complexity and their impact, (2) lack of understanding about the impacts of the defined software process during the project execution, and (3) experts' knowledge dependence issues in decision-making to solve variability.

Keywords—Software Process Line, Process Definition, Tools, Technologies, Literature Review.

I. INTRODUCTION

Faced with a scenario of high competitiveness in industry, software producers are constantly searching for approaches and development methods to increase their productivity and flexibility to provide services and products.

Considering the fact that the process by which software is developed influences the final product quality, companies are concerned with their process definitions. It is considered a complex activity that requires experience and knowledge from a variety of Software Engineering disciplines [1][2]. Besides that, defining a software process from scratch is risky and requires much time and effort [3].

In this scenario, Software Reuse studies become relevant, contributing to effort reduction and quality enhancement in software development. Although it emerged in the 60's, the Software Reuse area continues to aggregate new research topics. Currently, studies in this area cover topics such as Software Ecosystems [4], Software Product Line (SPL) [5], Software Process Line (SPrL) [6], etc.

Software Process Line (SPrL) offers a systematic approach to reuse software processes, identifying their similarities and variabilities. SPrL represents a set of processes in a particular area or for a specific purpose that shares characteristics, being built upon reusable process assets [3]. Some of SPrL's expected benefits are: (1) increase in reuse potential, (2) process definition productivity, (3) increase in processes quality and suitability, and (4) risk reduction.

The SPrL term has been proposed by several approaches that apply the concept of SPL in software processes [6][18][20], aiming to understand the relationship among organizational processes and project-specific software processes. In this scenario, commonalities should be captured in the company-wide process and the controlled variability should be specified with guidelines for tailoring [6]. SPrL approaches are identified as a proactive reuse initiative where an organization prepares its software processes for a set of expected needs [17].

In its engineering vision, this technique is divided into two phases [6]: (1) Software Process Domain Engineering (SPDE), and (2) Project-Specific Process Engineering (PSPE). In the SPDE phase, development *for* reuse, process engineers define the SPrL reusable artifacts. On the other hand, in the PSPE phase, development *by* reuse, reusable process elements are selected for a project-specific software process considering the process definition context.

The main activities of the PSPE phase are: (1) project characterization, identifying its requirements, (2) reusable process elements selection, solving configuration points, and (3) remaining software process adaptation, to meet specific needs that may not be contemplated in the process domain. The result is a project-specific software process defined with all the elements that should compose its execution.

In this scenario, each project has its own characteristics that require a particular set of activities, techniques and strategies [30]. During the software process definition, these characteristics can be captured through context information. According to Dey et al. (2001) [29], a context corresponds to any information used to characterize the situation of an entity (e.g., person, place, action, object) considered relevant. Context entities (i.e., dimensions) are described by a specific set of information that should be monitored.

On the other hand, understanding the project characterization and selecting the process elements to solve the SPrL variability represents a decision-making problem. Thus, in order to obtain the expected benefits, the project manager needs a supporting tool to define a project-specific software process using SPrL.

The experience level of the professional can affect the software process definition. Thus, the use of SPrL aims to reduce this dependency providing support for understanding the process domain and decision-making to solve variability during reusable process elements selection.

The SPrL area is recent and not fully consolidated [19]. The area is still immature and needs to be improved in terms of empirical evaluation [13]. There is no consensus approach

to accomplish the software process adaptation in a controlled and consistent manner [11][21].

In order to identify existing software process reuse approaches and their main features, some literature reviews were performed, as presented in Section II. After analyzing these reviews, a lack of specific analysis about SP_rL PSPE phase was observed, which includes the identification of available tools and techniques applied to support process domain variability resolution during software process definition. So, aiming to exploit the PSPE phase analysis, this work presents the major results of a systematic literature review focused in this phase.

In addition to this introduction, in Section II, related works are presented; in Section III, the research protocol is detailed; in Section IV, the study results are presented; in Section V, the results are discussed; in Section VI, threats to validity are analyzed; and finally in Section VII, a conclusion is presented, as well as proposals for future work.

II. RELATED WORK

This section presents some literature review performed in the Software Process Reuse area and analyzes what has been investigated. Systematic Literature Review (SLR) is an exhaustive search method, used to identify all relevant results in a particular research topic [11]. This method offers information of real interest on the investigated topic and increment the scientific value of the obtained results.

As noted in Table I, seven reviews were identified in the literature, four papers and three studies conducted as part of a doctoral thesis research.

TABLE I. SOFTWARE PROCESS REUSE LITERATURE REVIEW

Author	Type	Number of documents	Research target	Year
Pedreira et al. [10]	Paper	28	Process Tailoring	2007
Barreto [15]	Thesis section	57	Process Reuse Techniques	2011
Martínez-Ruíz et al. [11]	Paper	32	Process Tailoring	2013
Aleixo [16]	Thesis section	40	Process Variability Management	2013
Rocha e Fantinato [12]	Paper	63	SPL Applied to Business Process Management	2013
De Carvalho et al. [13]	Paper	40	SP _r L	2014
Teixeira [14]	Thesis section	25	SP _r L	2016

Pedreira et al. [10] investigated Software Process Tailoring, identifying 28 primary studies. The results of the study highlighted a lack of attention to approaches application in small and medium-sized organizations scenarios.

Barreto [15] presented a broader focus identifying reuse techniques applied in software process definition. The results of the study highlighted the following techniques as the main used (60% of the identified approaches): (1) Process Components, (2) Software Process Line, and (3) Process Standards.

Martínez-Ruíz et al. [11] investigated how Software Process Tailoring approaches treat variability, adaptation operations representation and modeling notation. The results indicated the lack of maturity of the notations for application in the industry scenario. Among the results, it highlighted activities, artifacts, and roles as the process elements more treated as variable points by the approaches.

Aleixo [16] focused on software process variability management. The results indicate that 32% of the studies offer supporting tool to manage variability.

Rocha and Fantinato [12] investigated the application of SPL on Business Process Management (BPM). The results indicated a considerable limitation, since the studies do not fully support the BPM life cycle.

De Carvalho et al. [13] investigated the SPDE phase in SP_rL approaches, including their variability representation. The results highlighted the most use of Feature Model and SPEM derivation for SP_rL representation. In addition, it presented the increase of interest in SP_rL research, even if reporting it as an immature area and pointing out many open questions.

Teixeira [14] in her work also aimed to characterize the SP_rL area state-of-the-art. The results indicated that 60% of the approaches offer support for SPDE phase.

In this scenario, only some of these literature reviews have SP_rL as focus and mainly discuss about the SPDE phase. So, considering the initial lack of attention and importance of the PSPE phase, this work investigates specific questions about it.

III. LITERATURE REVIEW PLANNING AND EXECUTION

A. Objective

This work aims to characterize the state-of-the-art of the Project-Specific Process Engineering (PSPE) phase (i.e., software process definition using SP_rL reusable artifacts). This objective involves identifying scientific publications about SP_rL and characterizing their support to the PSPE phase from the researcher's point of view.

This research was conducted based on the set of papers identified as part of another more general SLR [14], which had the goal of identifying approaches for developing, using, managing and evolving SP_rLs. A research team conducted that review composed by 6 members from four Brazilian universities, one being a master student and five Ph.D. researchers. In the current review, three researchers composes the team. Two of these participated in the original review.

The first search performed in November 2014 was updated in June 2016 by the same team, using three electronic databases (Scopus, IEEE Xplore and Compendex). The following search string was defined based on the research scope (i.e., PICO structure) and SLR objectives. Also, a refinement step was performed based on control documents and experts' knowledge. The search string was applied to the title, keywords and abstract fields of each selected electronic database:

("process adaptation" OR "processes adaptation" OR "customization of processes" OR "software processes customization" OR "software process customization" OR "customizing software processes" OR "process definition" OR "processes definition" OR "process composition" OR "compose processes" OR "processes composition" OR "process tailoring" OR "processes tailoring" OR "tailoring of processes" OR "process development" OR "processes development" OR "process engineering" OR "processes engineering" OR "process design" OR "software process modeling" OR "software process modeling" OR "process implementation" OR "managing processes")

AND

("family of software process" OR "family of software processes" OR "families of software process" OR "families of software processes" OR "software process line" OR "software process lines" OR "software processes line" OR "software processes lines" OR "process-line" OR "process-lines" OR "processes-line" OR "processes-lines" OR "software process family" OR "software processes family" OR "software process families" OR "software processes families" OR "process-family" OR "processes-family" OR "process-families" OR "processes-families" OR "software process variability" OR "software process variabilities" OR "software processes variability" OR "software processes variabilities" OR "variabilities in software processes" OR "process domain engineering" OR "processes domain engineering" OR "process feature" OR "process features" OR "processes feature" OR "processes features" OR "process asset reuse")

From 861 papers returned, 134 duplicate documents were removed, 669 documents were eliminated in a first filter (title and abstract analysis), three documents were excluded because their full content could not be retrieved and thirteen documents were removed after the reading phase for being out of the SLR scope. Summarizing, a total of 42 papers were identified as relevant SPPrL studies supporting software process reuse.

The rest of this section describes the planning and execution stages of the current SLR focused in the PSPE phase. The 42 papers selected from the original SLR were considered as the initial set to identify SPPrL approaches that had some specific support for the PSPE phase.

B. Scope

Table II presents the scope of this literature review according to the PICO structure (Population, Intervention, Comparison and Outcomes) [7].

TABLE II. RESEARCH SCOPE (PICO)

PICO	Description
Population	Software Process Line researches.
Intervention	Approaches that support the Project-Specific Process Engineering phase.
Comparison	Not applied. The SLR's objective is to characterize the approaches.
Outcomes	SPPrL approaches, techniques, tools and industry involvement.

C. Research Question and Information Extraction Form

Based on the study objective and challenges of the PSPE phase, Table III presents the research questions of this literature review. These questions were defined to characterize the PSPE phase as well as to analyze the involvement of industry in the SPPrL approaches.

TABLE III. RESEARCH QUESTIONS

Type	Question
Primary	Which Software Process Line approaches with support to the Project-Specific Process Engineering phase are available in the literature?
Secondary (SQ1)	Is there evidence of context information use?
Secondary (SQ2)	Which techniques are applied in the Project-Specific Process Engineering phase?
Secondary (SQ3)	Which supporting tools are reported?
Secondary (SQ4)	Which evaluations are reported to validate the SPPrL approach?
Secondary (SQ5)	Is there evidence of industry involvement?

To answer the defined questions, an information extraction form was defined to enable a systematic analysis, as presented in Table IV.

TABLE IV. INFORMATION EXTRACTION FORM

Information	Question
Title, abstract, author(s) and year of publication	N/A
Context representation	SQ1
- Number of suggested context informations - Suggested context informations	SQ1
Supporting technique applied in the PSPE phase	SQ2
Supporting tool applied in the approach	SQ3
- Supporting tool applied in the PSPE phase	SQ3
Approach evaluation type	SQ4
Industry involvement	SQ5

D. Control Documents

Two control documents were defined (i.e., ad-hoc search) to analyze the papers identification step. In this scenario, the control documents were applied in the set of papers adopted to verify the effectiveness of the search. These documents are presented below:

- Magdaleno, A. M., Araujo, R. M., & Werner, C. M. L. "COMPOOTIM: An Approach to Software Processes Composition and Optimization." Ibero-American Conference on Software Engineering (CIbSE). 2012. pp. 42-55.
- Hurtado, J. A., Bastarrica, M. C., Ochoa, S. F., & Simmonds, J. "MDE software process lines in small companies." Journal of Systems and Software, Volume 86, Issue 5. 2013. pp. 1153-1171.

E. Studies Selection Criteria

Table V presents the inclusion and exclusion criteria used to guide the selection of relevant documents.

TABLE V. SELECTION CRITERIA

Type	Criteria
Inclusion	The approach should use the Software Process Line concept and present some support to the PSPE phase.
Exclusion	Publications that do not define partially or entirely a Software Process Line approach should be removed.
Exclusion	Publications with Software Process Line approaches that do not support the Project-Specific Process Engineering phase should be removed.
Exclusion	Publications of the presentation type, tutorial, workshop or similar type should be removed.
Exclusion	Publications without access to the full content should be removed.

F. Execution

Based on the SLR planning and applying the selection criteria for the 42 papers identified in the previous SLR (described in Subsection III.A), 16 documents were eliminated because they did not report activities of PSPE phase.

Summarizing, a total of 26 documents were selected and analyzed in this review. These documents are listed in Appendix 1 of this paper.

IV. STUDY RESULTS

A. Studies Distribution Per Year

Through the studies distribution per year (Fig. 1), it is possible to observe the emergence of the support to the PSPE phase in SPRL approaches since 2005, as well as the growth of the area over the years. In this figure, it is verified that the PSPE phase activities were more discussed in the years 2014 and 2015 with 6 primary studies each.

No publication niche among publication venues (e.g., journals, conferences) were observed based on the selected documents. At this point, the International Conference on Model Driven Engineering Languages and Systems (MoDELS) was the publication venue with the highest number of papers selected, three documents.

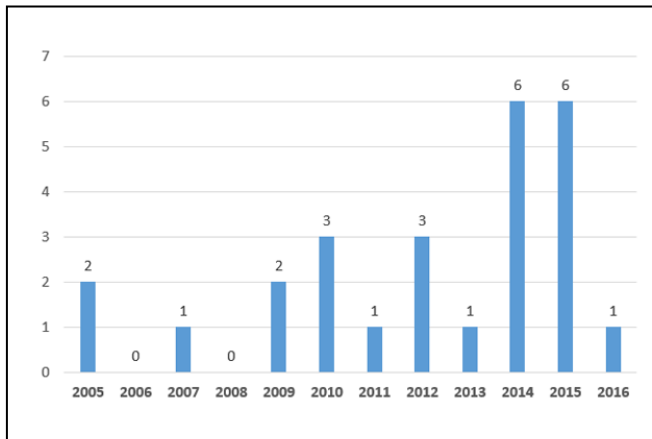


Fig. 1. Studies distribution per year

B. Identified Approaches

After analyzing the 26 selected documents, thirteen approaches were identified (Table VI). Hurtado et al. stand out as the group with the highest number of papers selected in this literature review, seven documents.

Each selected document was analyzed for information extraction according to the form defined in the research protocol (Subsection III.C). In cases of approaches that presented multiple documents (6 out of 13), all the related documents were analyzed together to group the information and derive an aggregated analysis of the support provided by each approach to the PSPE phase.

The results and discussions presented in this article are described considering the 13 identified approaches (Table VI).

TABLE VI. IDENTIFIED APPROACHES

Approach	Number of researchers	Number of documents	Documents
Aleixo et al.	4	2	[A.1] [A.2]
Barreto et al.	4	2	[A.3] [A.4]
Garcia et al.	4	1	[A.5]
Hurtado et al.	6	7	[A.6] [A.7] [A.8] [A.9] [A.10] [A.11] [A.12]
Jafarinezhad and Ramsin	2	1	[A.13]
Jaufman and Münch	2	1	[A.14]
Lorenz et al.	6	2	[A.15] [A.16]
Magdaleno et al.	5	2	[A.17] [A.18]
Martínez-Ruiz et al.	3	1	[A.19]
Rombach	1	1	[A.20]
Rouillé et al.	5	1	[A.21]
Ternité et al.	5	4	[A.22] [A.23] [A.24] [A.25]
Thränert and Werner	2	1	[A.26]

C. Context Representation

All the identified approaches have indicated the use of context information to support the definition of project-specific software processes. However, little information has been presented, limiting the information extraction.

The approaches of Aleixo et al., Garcia et al., Jaufman and Münch, Martínez-Ruiz et al., Rouillé et al., Ternité et al., Thränert and Werner, reported the influence of project *features*, but did not present entities or information suggestions.

Barreto et al. reported the use of process features as an aspect, quality, or characteristic that the process has to be compliant with. Features are applied to guide process elements selection, constraining the available choices throughout process derivation. The authors pointed out that the user could define the semantics of a feature to be applied as a high-level mechanism for process component selection.

Hurtado et al. indicated that project context could vary according to different project variables (i.e., context attributes as size, complexity, knowledge, type) along specific entities, such as: product, project, and development team. The authors defined a Software Process Context Metamodel (SPCM) as a way to represent context model for each project, which will be defined by its user. There is not a predefined set of context entities and context information.

Jafarinezhad and Ramsin use a situation model to describe a project context. This model defines a project situation in terms of situation factors and process criteria. The authors suggested that this information can be represented informally, or by defining a list of characteristic-value pairs using a specific notation, such as *i** modeling language. The situation model permits a trade-off analysis and influences a feature model resolution. A set of 23 context information were elicited to be used in a Requirement Engineering SPRL: Project Type, Application Domain, Project Size, Complexity, Management

Commitment, Degree of Resistance, Requirements Volatility, Level of Criticality, Scarcity of People and Resources, Team Size, Familiarity with the Domain, Requirements Engineering Team Knowledge, Degree of Knowledge about the Requirements, Availability of Skilled Facilitators, Potential for Conflict, Innovation Level of the Project, Customer Availability, Degree of Reusability, Degree of Implicit Knowledge, Degree of Outsourcing Required, Capability Maturity Level, Organizational Impact, and Strategic importance.

Lorenz et al. define project contextualization as a way to specify the main characteristics of projects that need to be considered in the software process tailoring. The context is defined by factors (process characteristics as size, stable architecture, business model, team distribution, rate of change, age of system, criticality, and control/governance) with associated attributes (values). The authors indicate three models as possibilities to represent the content of project contexts: (1) Boehm and Turner [31], (2) Octopus Model [32], and (3) Cockburn [33]. These three studies suggest the use of a total of 16 context information. In the case studies performed, the Octopus Model was chosen.

Magdaleno et al. define a context model composed by context entities and information. Context rules are defined to suggest process selection based on context definitions (situations where context information assume specific values). These are elements of a context representation proposed in [34]. They report the use of three context entities: (1) Organizational, (2) Project, and (3) Team. In addition, 17 context information was suggested: Organizational Structure, Organizational Culture, Knowledge Management, Business Objective, Customer Relationship, Size, Complexity, Novelty, Criticality, Duration, Requirements Stability, Team Size, Domain Experience, Technical Experience, Work Together Experience, Proximity, and Stability. This set of context information was obtained from a literature review and validated through a survey with experts [35].

Finally, Rombach indicates three entities in his research: (1) Project, (2) Product, and (3) Process.

D. Techniques Applied in PSPE Phase

Five techniques to support variability solution were identified in the analyzed approaches (Table VII).

TABLE VII. TECHNIQUES APPLIED IN PSPE PHASE

Technique	Approach
Feature Mapping	Aleixo et al., Barreto et al., Garcia et al., Rouillé et al.
Rule-Based System	Hurtado et al., Thränert and Werner, Magdaleno et al.
Fuzzy Inference System	Jafarinezhad and Ramsin
Analytic Hierarchy Process (AHP)	Lorenz et al.
Genetic Algorithm	Magdaleno et al.
N/A	Jaufman and Münch, Martínez-Ruiz et al., Rombach, Ternité et al.

The frequency of use of each technique (Fig. 2) emphasizes greater use of Feature Mapping and Rule-Based System to support the PSPE phase activities. In four of the

approaches, it was not possible to identify a technique supporting the PSPE phase based on the selected papers.

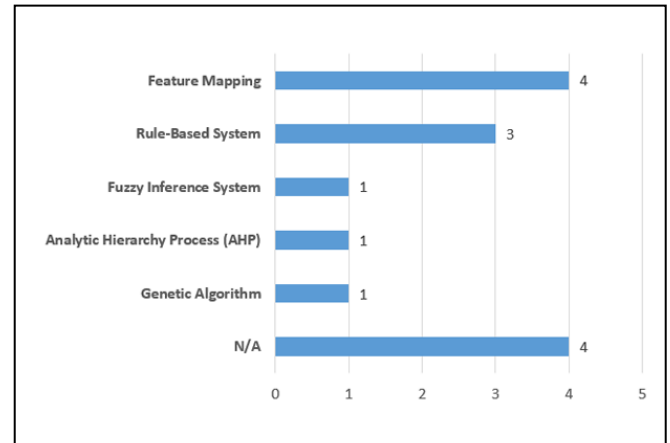


Fig. 2. Frequency of the techniques applied in the PSPE phase

The Feature Mapping technique, in this research, represents a direct mapping involving a context information, feature or goal that determines the inclusion or exclusion of reusable process elements. This was the technique with the highest frequency in the analyzed approaches. However, defining complex context situations does not seem trivial in this technique (e.g., multiple context information impacting multiple process elements). This aspect may represent a limitation to the decision-making support.

Rule-Based Systems [37] are used to store and manipulate knowledge to interpret information, i.e., an artificial intelligence technique used to develop expert systems [22][23]. On the other hand, an expert system is a computer system that emulates the decision-making ability of a human expert. In this research scenario, Rule-Based Systems have been applied for rules definition for variability solution in software process involving context information. The goal of the Rule-Based System technique is to make the critical information explicit. The disadvantage of this technique is the knowledge acquisition, because of the lack of availability of domain experts.

Fuzzy Inference Systems [24][25] are used to formulating the mapping from a given input to an output using fuzzy logic. The mapping then provides a basis from which decisions can be made. This technique is commonly used for handling uncertainty and imprecision of judgment in multi-objective decision-making processes. The advantage of this technique is that the solution can be cast in terms that human operators can understand [24]. In Jafarinezhad and Ramsin research [A.13], this technique was applied to support process elements classification, but the decision about process elements selection is the process engineer's responsibility.

Analytic Hierarchy Process (AHP) is a method applied to support decision-making in complex scenarios. AHP works with alternatives and an overall goal. In this scenario, the numerical probability of each alternative is calculated and the higher the probability, the better the chances that the alternative has to satisfy a final goal [26]. AHP can be applied to decision situations [27], such as: (1) Choice, the selection of one alternative from a given set of alternatives, usually where there are multiple decision criteria involved, (2) Ranking, putting a set of alternatives in order from the most to the least desirable one, and (3) Prioritization,

determining the relative merit of members of a set of alternatives, as opposed to selecting a single one or merely ranking them. In Lorenz et al., research [A.15], the AHP technique was applied to support the software process elements prioritization.

Genetic Algorithms are search algorithms based on the mechanisms of natural selection and natural genetics [28]. This technique is a metaheuristic that belongs to the larger class of evolutionary algorithms. It combines survival of the fittest, fitness-oriented reproduction, and random mutation to evolve a population of candidate solutions. However, it does not guarantee a globally optimal solution. A Genetic Algorithm requires a genetic representation of the solution domain (e.g., array of bits) and a fitness function to evaluate the solution domain. This technique cannot effectively solve problems that the only fitness measure is a single right/wrong measure. However, if the situation allows the success/failure trial, then the ratio of successes to failures provides a suitable fitness measure. In Magdaleno et al. research [A.18], the GA technique was applied to optimize collaboration in software processes, where collaboration is used as a mono-objective fitness function. Before that, the Rule-Based System technique is applied to support the project-specific software process definition.

E. Supporting Tool

Table VIII presents the supporting tools reported in the approaches, highlighting the tools applied in the PSPE phase.

TABLE VIII. SUPPORTING TOOL

Approach	Supporting tools	PSPE tools
Aleixo et al.	EPF Composer, GenArch and jBPM	GenArch
Barreto et al.	Web-Based Prototype	Web-Based Prototype
Garcia et al.	EPF Composer, GenArch, UMA2BPMN, Yaoqiang BPMN and Activiti	GenArch
Hurtado et al.	EPF Composer, Exeed, ATR, Web-Based Prototype and AM3	ATR, Web-Based Prototype and AM3
Jafarinezhad and Ramsin	MATLAB	MATLAB
Jaufman and Münch	N/A	N/A
Lorenz et al.	MfPTt Support Tool	MfPTt Support Tool
Magdaleno et al.	Odyssey and COMPOOTIM	COMPOOTIM
Martínez-Ruiz et al.	N/A	N/A
Rombach	N/A	N/A
Rouillé et al.	CVL tooling	CVL tooling
Ternité et al.	V-Modell XT VMEd and V-Modell XT VMPA	V-Modell XT VMPA
Thränert and Werner	N/A	N/A

Most approaches (i.e., 69%) presented supporting tool to the activities of the PSPE phase (Fig. 3). However, existing tools have been reused by some approaches (e.g., SPL tools). This kind of reuse may represent a limitation to the specific needs of the SPRL concept. Even so, the analysis of this kind of impact or possible limitation it is not discussed in the approaches that adopt existing tools. On the other hand, 31%

of the approaches consider the activities of the PSPE phase, but do not offer supporting tool.

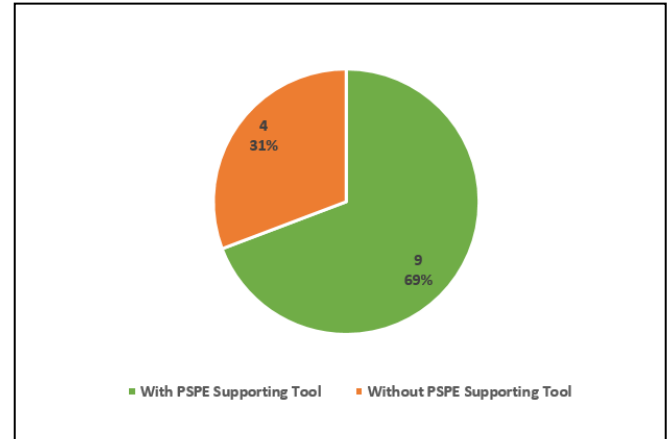


Fig. 3. Percentage of approaches with supporting tools in the PSPE phase

Aleixo et al. indicate the use of three supporting tools (EPF Composer, GenArch and jBPM). Eclipse Process Framework (EPF) was used to define the software process based on the Unified Method Architecture (UMA) notation using the EPF Composer tool (Eclipse Process Framework Composer). The variability management and software process definition are performed using GenArch, an existing product line tool. This tool has been extended to support process fragments from an EPF specification. GenArch tool allows the mapping between variability and the process elements. Based on this mapping, the tool automatically derives a project-specific software process. Resolution of feature constraints and process component dependencies are also computed during this step and the resulting software process is an EPF specification. Finally, the project-specific software process is processed by model-to-model (M2M) and model-to-text (M2T) transformations using ATL and Aceleo languages. These transformations generate a JPDL workflow specification that can be deployed and executed in JBoss Business Process Management (jBPM) workflow engine.

Barreto et al. present a Web-Based Prototype tool developed to support software process implementing organizations (SPIOs). A SPIO is a consultant organization that is hired by other organizations intending to define, deploy, or improve their software processes [A.4]. The Web-Based Prototype tool is used to define process components (i.e., process elements), associate it to process features (e.g., goals, context information), define a SPRL and derive project-specific software process from the SPRL. The project-specific software process definition support is reported as being under development.

Garcia et al. use a similar toolset and based on the study of Aleixo et al., adopting EPF Composer and GenArch to specify the software process, manage the variability and define a project-specific software process. The main difference is the use of BPMN for execution, monitoring and controlling the software process. It required the development of UMA2BPMN to transform the UMA notation to BPMN. In addition, Yaoqiang BPMN tool, an open source modeling tool for BPMN was used to review BPMN. Finally, the Activiti Platform was adopted, a BPM Platform for execution, monitoring and controlling software processes.

Hurtado et al. indicate the use of five tools (EPF Composer, Exeed, ATR, Web-Based Prototype and AM3). The tool chain was built on top of the Eclipse Modeling Framework (EMF) and ATL transformation language. An organization software process is defined using EPF Composer tool. The models were implemented as instances of defined metamodels and edited using Exeed (Extended EMF Editor), EMF reflective editor. After the organization's software process definition, a Web-Based Prototype tool is used to define the organization context model, i.e., a set of context information and their possible values based on Software Process Context Metamodel (SPCM). Then, ATR tool (Architect of Tailoring Rules) is applied to define rules and produces the Variation Decision Model (VDM), based on organization software process and organization context model. VDM formally represents transformation rules using DSL, a high-level representation of the transformation rules using an abstract syntax (metamodel) and a concrete syntax (textual representation). Finally, the Web-based Prototype tool is used to define the project context, which means an organizational context model configuration, where for each context attribute a value has been assigned. Then, the project-specific software process is defined.

Also, Hurtado et al. adopt the Global Model Management (GMM) megamodeling approach with AM3 supporting tool, an Eclipse plugin fully open-source. The megamodeling concept executes a series of model-to-model (M2M), text-to-model (T2M) and model-to-text (M2T) transformations to automate the software process definition, hiding the complexity. AM3 tool was reported as unstable.

Jafarinezhad and Ramsin report the use of a fuzzy logic toolbox, part of MATLAB tool. The tool was applied to represent a Fuzzy Inference System (FIS) that uses a Fuzzy Rule Engine (FRE) for mapping an input to an output based on fuzzy logic. This concept is applied to classify the process elements during feature analysis. In this scenario, each output value can be interpreted as the presence condition for the corresponding feature. Thereafter, the process engineer is responsible for making a decision about process elements selection, interpreting the classification results. The authors indicate that this step can be automated. For this, it would be necessary to define a minimum value for process elements selection.

Lorenz et al. report the development of MfPTt Support Tool to support in the software process definition using SPPrL. The tool covers functionalities from the registration of process elements, such as artifacts, tasks, roles, and activities, up to the SPPrL architectures definition and project-specific software process derivation. To define a project-specific software process, it is necessary that the process elements have been registered and stored in the repository. The recovery of the process elements best suited to a particular process starts from the selection and prioritization of activities from the repository using the Analytic Hierarchy Process (AHP) technique. In the tailoring module of MfPTt Support Tool, a set of steps is offered: (1) Definition of project context, (2) Tailoring requirements selection, (3) SPPrL architecture selection, (4) Activities prioritization, and (5) Project-specific software process definition.

Magdaleno et al. report the use of two tools to support the approach: (1) Odyssey, an environment used to model

and maintain SPPrL models (feature model and context model), and (2) COMPOOTIM, a tool based on JMetal Framework, developed to support the approach on project-specific software process definition and optimization of collaboration in the resulted software processes. COMPOOTIM tool supports the project manager decision about process components selection, combination and optimization based on a collaboration aspect. The tool uses the organization's SPPrL as input. It receives process components, composition rules (represent dependencies or mutually exclusive relationships) and context information, selects the most appropriate components for the context, combines these components in search for a feasible software process, and evaluates the effectiveness of this process. In the end, it suggests a set of alternative processes that maximize collaboration for the project context.

Rouillé et al. report the use of a CVL tooling based on Common Variability Language (CVL), a domain-independent language for specifying and resolving variability from SPL area. CVL tooling is used in the approach to define an SPPrL and automatically derive a project-specific software process. Three models were used to define the SPPrL: (1) Software process model, (2) Variability Abstraction Model (VAM), that captures the requirements variability, and (3) Variability Realization Model (VRM), that specifies the binding between context information and process elements. CVL tooling allows the selection of the requirements of a given project in a Realization Model (RM) to automatically derive a project-specific software process. According to Rouillé et al., these models contain enough information to provide a resolved base model without variability. CVL derivation engine relies on Eclipse Modeling Framework (EMF) API to load, manage and save the models.

Ternité et al. indicate the use of V-Modell XT SPPrL Framework, a comprehensive software process framework (including, e.g., metamodels, tools, reference implementations, and guidelines). In this scenario, two tools are used to support the approach: (1) V-Modell XT VMEd, an editor that supports process authoring, deployment, and management tasks, and (2) V-Modell XT VMPPA, a project assistant that serves the project-specific software process definition including initial planning and the generation of document templates. Although Ternité et al. indicate other tools applied to perform the evaluation study (e.g., merge tool), these tools were not classified as supporting tools of the approach.

F. Evaluation

Four evaluation types were identified during the analysis. They are presented in Table IX.

TABLE IX. APPROACHES EVALUATION

Evaluation type	Approach
Case study	Aleixo et al., Hurtado et al., Lorenz et al., Ternité et al.
Controlled experiment	Jaufman and Münch, Magdaleno et al., Rouillé et al.
Expert assessment	Garcia et al., Magdaleno et al.
Survey	Barreto et al.
N/A	Jafarinezhad and Ramsin, Martínez-Ruíz et al., Rombach, Thränert and Werner

Although some approaches provided an example to facilitate the reader's understanding, this kind of report was not considered an evaluation study.

The frequency of use of the evaluation types in the approaches (Fig. 4) emphasizes greater use of case studies (4) and controlled experiments (3) to validate the results. Four of the approaches did not report any performed evaluation.

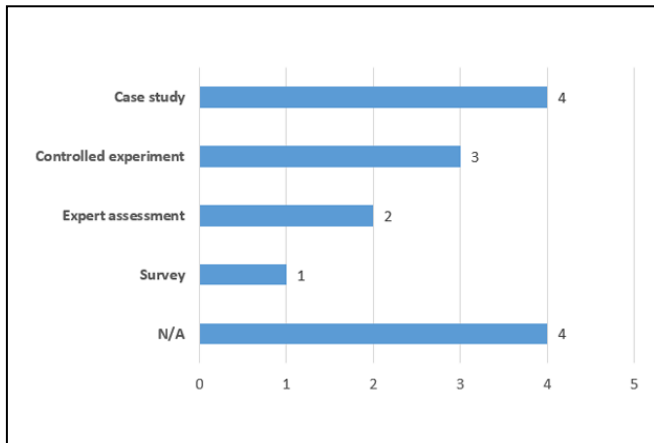


Fig. 4. Frequency of the evaluation types

Magdaleno et al. was the only approach that presented more than one evaluation type (i.e., controlled experiment and expert assessment). However, only the expert assessment reported industry involvement. On the other hand, in the controlled experiment the results of the Genetic Algorithm technique was compared with other search techniques.

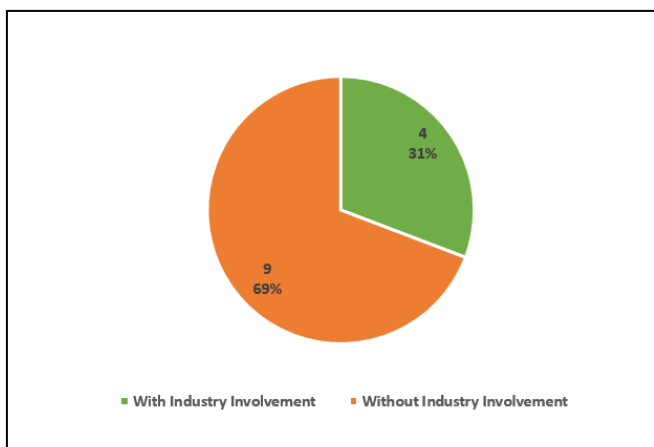


Fig. 5. Percentage of the industry involvement on the approaches

Four approaches stand out by industry involvement in their evaluations: (1) Hurtado et al. conducted multiple case studies with Chilean companies of software development, (2) Magdaleno et al. performed an expert assessment involving a large Brazilian oil and gas company, (3) Rouillé et al. performed a controlled experiment involving a software and computing services company, and (4) Ternité et al. conducted a long-term case study (i.e., 2 years) involving multiple organizations. The approaches of Barreto et al. and Lorenz et al. did not report involvement of the industry; however, projects in academic environment were performed.

The percentage of approaches that reported industry involvement (Fig. 5) indicates that this practice is still under-explored. However, it is necessary to take into account that it is an emergent area.

V. DISCUSSION

In general, the approaches have reported the use of context information and its influence to define a project-specific software process. Many of them provide some mechanism to represent context information to be defined in each specific approach application. Only few of them indicate predefined context information sets (i.e., suggestions).

This fact can make the approach adoption difficult, since in the SPDE phase, in addition to reusable process elements definition, domain knowledge will be necessary to define the context information that can affect the software process definition and variability solution. In this scenario, the most cited context information in the analyzed approaches were: Criticality, Project size, Team Size, Project type, Domain Experience, Organizational Culture, Requirements Stability, Team Distribution and Team Experience.

Although integrated support tools have been observed, the analyzed tools indicated a low-level of integration. Knowing the importance of integrated tools, as well as interfaces and standardized representations in Software Engineering activities support (i.e., Software Engineering Environments) [8], integrated tools are still needed, with simultaneous support to the SPDE and PSPE phases activities.

As presented in the results, five support techniques for variability solution were identified and most of them focused on mapping or rules definition, which may represent a limitation. In this scenario, an issue remains open: "Which techniques can be applied to support variability solution in the PSPE phase?". To answer this question, different areas can be investigated, such as SPL and Multiple Criteria Decision-Making [9].

Finally, three problems that affect the PSPE phase were observed as open research points:

(1) Low understanding of SPRL models complexity and their impact: None of the approaches verified the possible impact of the uncontrolled growth of the domain's variabilities number and its relationship with the effort to define a project-specific software process.

(2) Lack of understanding about the impacts of the defined software process during the project execution: None of the studies presented support or debate to analyze the possible impacts of the defined software process on the software product.

(3) Experts' knowledge dependence issues in decision-making to solve variability: The decision-making support observed in the studies focused on making explicit the experts' knowledge in the SPDE phase (e.g., mapping and rules), which may represent a possible overload at this phase. On the other hand, in the PSPE phase, the understanding of the possible impacts generated by the decision-making to solve SPRL variability in the software process definition depends on the knowledge of the one responsible for the activity.

VI. THREATS TO VALIDITY

A threat to the validity of this research can be the incomplete or inadequate selection of primary studies. To mitigate this threat, a systematic approach was adopted, as well as multiple researchers for isolated document analysis. In addition, debates were held to solve disagreements.

VII. CONCLUSIONS AND FUTURE WORKS

This study aimed to characterize the state-of-the-art of the SP_rL PSPE phase. The importance of developing integrated tools, involving support to both SPDE and PSPE phases, was identified. In addition, more research can be done to variability solution techniques diversification during project-specific software processes derivation from SP_rL models.

The authors are defining an incremental learning approach for SP_rL focusing in the PSPE phase as a continuation of this research. This proposal aims to explore the relationship between Case-Based Reasoning [36] and Rule-Based System [37] (i.e., Artificial Intelligence techniques) to provide a flexible support for defining software process using process lines through complementary mechanisms.

As future work, a review can be conducted in SPL literature to identify techniques to support variability solution applied in the Application Engineering phase that can be adapted to SP_rL PSPE phase. In addition, an experiment can be performed to understand the impacts of the complexity of SP_rL models on the project-specific software process definition. Finally, a study can be performed to analyze similarities and divergences of the process tailoring technique and SP_rL PSPE phase.

ACKNOWLEDGMENT

The authors would like to thank the research agencies CAPES, FAPERJ and CNPq for the financial support.

REFERENCES

- [1] Aleixo, F. A., Freire, M. A., dos Santos, W. C., & Kulesza, U. "A Model-driven Approach to Managing and Customizing Software Process Variabilities." International Conference on Enterprise Information Systems (ICEIS). 2010. pp. 92-100.
- [2] Barreto, A., Duarte, E., Rocha, A. R., & Murta, L. "Supporting the definition of software processes at consulting organizations via software process lines." International Conference on the Quality of Information and Communications Technology (QUATIC). 2010. pp. 15-24.
- [3] Washizaki, H. "Building software process line architectures from bottom up." International Conference on Product Focused Software Process Improvement. Berlin, Heidelberg, 2006. pp. 415-421.
- [4] Dhungana, D., Groher, I., Schludermann, E., & Biffl, S. "Software ecosystems vs. natural ecosystems: learning from the ingenious mind of nature." European Conference on Software Architecture. 2010. pp. 96-102.
- [5] Northrop, L. M. "SEI's software product line tenets." IEEE software, Volume 19 Issue 4. 2002. pp. 32-40.
- [6] Rombach, D. "Integrated Software Process and Product Lines." International Conference on Unifying the Software Process Spectrum. Berlin, Heidelberg. 2005. pp. 83-90.
- [7] Pai, M., et al. "Systematic reviews and meta-analyses: an illustrated, step-by-step guide." The National Medical Journal of India, Volume 17, n. 8. 2004. pp. 86-95.
- [8] Ossher, H., Harrison, W., & Tarr, P. "Software engineering tools and environments: a roadmap." In: Proceedings of the Conference on the Future of Software Engineering. 2000. pp. 261-277.
- [9] Greco, S., Figueira, J., & Ehr Gott, M. "Multiple criteria decision analysis." Springer's International Series. 2005.
- [10] Pedreira, O., Piattini, M., Luaces, M. R., & Brisaboa, N. R. "A systematic review of software process tailoring." ACM SIGSOFT Software Engineering Notes, Volume 32, Issue 3. 2007. pp. 1-6.
- [11] Martínez-Ruiz, T., Münch, J., García, F., & Piattini, M. "Requirements and constructors for tailoring software processes: a systematic literature review." Software Quality Journal, Volume 20, Issue 1. 2012. pp. 229-260.
- [12] Rocha, R. S., & Fantinato, M. "The use of software product lines for business process management: A systematic literature review." Information and Software Technology, Volume 55, Issue 8. 2013. pp. 1355-1373.
- [13] De Carvalho, D. D., Chagas, L. F., Lima, A. M., & Reis, C. A. L. "Software process lines: A systematic literature review." International Conference on Software Process Improvement and Capability Determination (SPICE). 2014. pp. 118-130.
- [14] Teixeira, E. N. "OdysseyProcessReuse: Uma metodologia para engenharia de linha de processos de software baseada em componentes." Doctoral thesis. COPPE/UFRJ, 2016.
- [15] Barreto, A. "Uma Abordagem para Definição de Processos Baseada em Reutilização Visando à Alta Maturidade em Processos." Doctoral thesis. COPPE/UFRJ, 2011.
- [16] Aleixo, F. A. "Uma abordagem anotativa para gerência de variabilidade em linhas de processos de software: concepção, implementação e avaliação." Doctoral thesis. COPPE/UFRJ, 2013.
- [17] Armbrust, O., et al. "Scoping Software Process Models - Initial Concepts and Experience from Defining Space Standards." In: Proceedings of the International Conference on Software Process (ICSP), Berlin, Heidelberg. 2008. pp. 160-172.
- [18] Armbrust, O., et al. "Scoping Software Process Lines." Software Process: Improvement and Practice, Volume 14, Issue 3. 2009. pp. 181-197.
- [19] Barreto, A. S., Murta, L. G. P., & Rocha, A. R. "Componentizando Processos Legados de Software Visando a Reutilização de Processos." Anais do Simpósio Brasileiro de Qualidade de Software (SBQS), Ouro Preto, MG, Brasil. 2009. pp. 189-203.
- [20] Jaufman, O., & Münch, J. "Acquisition of a project-specific process." International Conference on Product Focused Software Process Improvement (PROFES), Berlin, Heidelberg. 2005. pp. 328-342.
- [21] Pillat, R. M., Oliveira, T. C., Alencar, P. S., & Cowan, D. D. "BPMNt: A BPMN extension for specifying software process tailoring", Information and Software Technology, Volume 57. 2015. pp. 95-115.
- [22] Jackson, P. "Introduction to expert systems." Addison-Wesley Longman Publishing Co., Inc. 1998.
- [23] Hayes-Roth, F., Waterman, D. A., & Lenat, D. B. "Building expert system." Addison-Wesley Longman Publishing Co., Inc. 1983.
- [24] Pedrycz, W. "Fuzzy control and fuzzy systems." John Wiley & Sons, Inc. 1993.
- [25] Hájek, P. "Metamathematics of fuzzy logic." Springer Science & Business Media. 1998.
- [26] Vargas, R. V. "Using the analytic hierarchy process (AHP) to select and prioritize projects in a portfolio." In: PMI Global Congress. 2010. pp. 1-22.
- [27] Forman, E. H., & Gass, S. I. "The analytic hierarchy process-an exposition." Operations Research, Volume 49, Issue 4. 2001. pp. 469-486.
- [28] Goldberg, D. E. "Genetic algorithms in search, optimization, and machine learning." Addison-Wesley Longman Publishing Co., Inc. 2009.
- [29] Dey, A. K., Abowd, G. D., & Salber, D. "A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications." Human-Computer Interaction, Volume 16, Issue 2. 2001. pp. 97-166.
- [30] Hurtado Alegría, J. A., Bastarrica, M. C., Quispe, A., & Ochoa, S. F. "An MDE approach to software process tailoring." In: Proceedings of the International Conference on Software and Systems Process (ICSSP). 2011. pp. 43-52.

- [31] Boehm, B., & Turner, R. "Using risk to balance agile and plan-driven methods." *Computer*, Volume 36, Issue 6. 2003. pp. 57-66.
- [32] Kruchten, P. "Contextualizing agile software development." *Journal of Software: Evolution and Process*, Volume 25, Issue 4. 2013. pp. 351-361.
- [33] Cockburn, A. "Crystal Clear: A Human-Powered Methodology for Small Teams." Pearson Education. 2004.
- [34] Fernandes, P., Werner, C. M. L., & Murta, L. G. P. "Feature Modeling for Context-Aware Software Product Lines." *International Conference on Software Engineering and Knowledge Engineering (SEKE)*. 2008. pp. 758-763.
- [35] Leite, A. M. S. "Modelo de Contexto para Adaptação de Processos de Software." Masters dissertation. PPGI-UNIRIO. 2011.
- [36] Aamodt, A., & Plaza, E., "Case-based reasoning: Foundational issues, methodological variations, and system approaches." *AI Communications*, Volume 7, Issue 1. 1994. pp. 39-59.
- [37] Hayes-Roth, F., "Rule-based systems." *Communications of the ACM*, Volume 28, Issue 9. 1985. pp. 921-932.
- International Conference on Model Driven Engineering Languages and Systems (MoDELS). Ottawa, Canada. 2015. pp. 406-415.
- [A.12] Silvestre, L., Bastarrica, M. C., & Ochoa, S. F. "A Usable MDE-based Tool for Software Process Tailoring." In: *Posters and Demos at International Conference on Model Driven Engineering Languages and Systems (MoDELS)*. Ottawa, Canada. 2015. pp.36-39.
- [A.13] Jafarinezhad, O., & Ramsin, R. "Development of situational requirements engineering processes: A process factory approach." *Computer Software and Applications Conference (COMPSAC)*. 2012. pp. 279-288.
- [A.14] Jaufman, O., & Münch, J. "Acquisition of a project-specific process." *International Conference on Product Focused Software Process Improvement (PROFES)*, Berlin, Heidelberg. 2005. pp 328-342.
- [A.15] Lorenz, W. G., Brasil, M. B., Fontoura, L. M., & Vaz Pereira, G. "Activity-based software process lines tailoring." *International Journal of Software Engineering and Knowledge Engineering (SEKE)*, Volume 24, Issue 09. 2014. pp. 1357-1381.
- [A.16] Brondani, C., Bertuol, G., & Fontoura, L. M. "Quality Evaluation of Artifacts in Tailored Software Process Lines." *International Conference on Software Engineering and Knowledge Engineering (SEKE)*. 2015. pp. 223-226.
- [A.17] Magdaleno, A. M., Araujo, R. M., & Werner, C. M. L. "COMPOOTIM: An Approach to Software Processes Composition and Optimization." *Ibero-American Conference on Software Engineering (CibSE)*. 2012. pp. 42-55.
- [A.18] Magdaleno, A. M., Barros, M. O., Werner, C. M. L., Araujo, R. M., & Batista, C. F. A. "Collaboration optimization in software process composition." *Journal of Systems and Software*, Volume 103. 2015. pp. 452-466.
- [A.19] Martínez-Ruiz, T., García, F., Piattini, M., Cordeiro, J., & Filipe, J. "Process Institutionalization using Software Process Lines." *International Conference on Enterprise Information Systems (ICEIS)*. 2009. pp. 359-362.
- [A.20] Rombach, D. "Integrated Software Process and Product Lines." *International Conference on Unifying the Software Process Spectrum*. Berlin, Heidelberg. 2005. pp. 83-90.
- [A.21] Rouillé, E., Combemale, B., Barais, O., Touzet, D., & Jézéquel, J. M. "Leveraging CVL to manage variability in software process lines." *Asia-Pacific Software Engineering Conference (APSEC)*, Volume 1. 2012. pp. 148-157.
- [A.22] Ternité, T. "Process lines: A product line approach designed for process model development." *Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*. 2009. pp. 173-180.
- [A.23] Kuhrmann, M., Fernández, D. M., & Ternité, T. "Realizing software process lines: Insights and experiences." In: *Proceedings of the International Conference on Software and System Process (ICSSP)*. 2014. pp. 99-108.
- [A.24] Schramm, J., Dohrmann, P., & Kuhrmann, M. "Development of flexible software process lines with variability operations: a longitudinal case study." *Proceedings of the International Conference on Evaluation and Assessment in Software Engineering (EASE)*. Nanjing, China. 2015.
- [A.25] Kuhrmann, M., Méndez Fernández, D., & Ternité, T. "On the use of variability operations in the V-Modell XT software process line." *Journal of Software: Evolution and Process*, Volume 28, Issue 4. 2016. pp. 241-253.
- [A.26] Thränert, M., & Werner, A. "A Process Family Approach for the reuse of development processes." *Innovations and Advanced Techniques in Computer and Information Sciences and Engineering*. 2007. pp. 309-313.

APPENDIX 1 - SELECTED PAPERS

- [A.1] Aleixo, F. A., Freire, M. A., Santos, W. C., & Kulesza, U. "A Model-driven Approach to Managing and Customizing Software Process Variabilities." *International Conference on Enterprise Information Systems (ICEIS)*. 2010. pp. 92-100.
- [A.2] Aleixo, F. A., Freire, M. A., Santos, W. C., & Kulesza, U. "An approach to manage and customize variability in software processes." *Brazilian Symposium on Software Engineering (SBES)*. 2010. pp. 118-127.
- [A.3] Barreto, A., Duarte, E., Rocha, A. R., & Murta, L. "Supporting the definition of software processes at consulting organizations via software process lines." *International Conference on the Quality of Information and Communications Technology (QUATIC)*. Porto, Portugal. 2010. pp. 15-24.
- [A.4] Barreto, A., Murta, L. G. P., & da Rocha, A. R. C. "Software Process Definition: a Reuse-based Approach." *Journal of Universal Computer Science*, Volume 17, Issue 13. 2011. pp. 1765-1799.
- [A.5] Garcia, C., Paludo, M., Malucelli, A., & Reinehr, S. "A software process line for service-oriented applications." In: *Proceedings of the Annual ACM Symposium on Applied Computing (SAC)*. 2015. pp. 1680-1687.
- [A.6] Hurtado, J. A., Bastarrica, M. C., Ochoa, S. F., & Simmonds, J. "MDE software process lines in small companies." *Journal of Systems and Software*, Volume 86, Issue 5. 2013. pp. 1153-1171.
- [A.7] Silvestre, L. "A Domain Specific Transformation Language to Support the Interactive Definition of Model Transformation Rules." *Doctoral Symposium at International Conference on Model Driven Engineering Languages and Systems (MoDELS)*. CEUR-WS, Volume 1321. Valencia, Spain. 2014.
- [A.8] Silvestre, L., Bastarrica, M. C., & Ochoa, S. F. "A model-based tool for generating software process model tailoring transformations." *International Conference on Model-Driven Engineering and Software Development (MODELSWARD)*. 2014. pp. 533-540.
- [A.9] Silvestre, L., Bastarrica, M. C., & Ochoa, S. F. "Reducing Complexity of Process Tailoring Transformations Generation." *International Conference on Model-Driven Engineering and Software Development (MODELSWARD)*. 2014. pp. 171-182.
- [A.10] Bastarrica, M. C., Simmonds, J., & Silvestre, L. "Using megamodeling to improve industrial adoption of complex MDE solutions." In: *Proceedings of the International Workshop on Modeling in Software Engineering (MiSE)*. 2014. pp. 31-36.
- [A.11] Simmonds, J., Perovich, D., Bastarrica, M. C., & Silvestre, L. "A megamodel for Software Process Line modeling and evolution." *International Conference on Model Driven Engineering Languages and Systems (MoDELS)*. Ottawa, Canada. 2015. pp. 406-415.