Information and Software Technology 116 (2019) 106175

Contents lists available at ScienceDirect

Information and Software Technology

journal homepage: www.elsevier.com/locate/infsof

Software process line as an approach to support software process reuse: A systematic literature review

Eldânae Nogueira Teixeira[a,*], Fellipe Araújo Aleixo[b], Francisco Dione de Sousa Amâncio[c], Edson OliveiraJr[d], Uirá Kulesza[e], Cláudia Werner[a]

[a] Federal University of Rio de Janeiro, Rio de Janeiro, RJ, Brazil
[b] Federal Institute of Rio Grande do Norte, Natal, RN, Brazil
[c] University of Fortaleza, Fortaleza, CE, Brazil
[d] State University of Maringá (UEM), Maringá, PR, Brazil
[e] Federal University of Rio Grande do Norte, Natal, RN, Brazil

ARTICLE INFO

Keywords:
Systematic review
Software process
Process reuse
Software process line
Process variability management

ABSTRACT

Context: Software Process Line (SPrL) aims at providing a systematic reuse technique to support reuse experiences and knowledge in the definition of software processes for new projects thus contributing to reduce effort and costs and to achieve improvements in quality. Although the research body in SPrL is expanding, it is still an immature area with results offering an overall view scattered with no consensus.

Objective: The goal of this work is to identify existing approaches for developing, using, managing and visualizing the evolution of SPrLs and to characterize their support, especially during the development of reusable process family artefacts, including an overview of existing SPrL supporting tools in their multiple stages; to analyse variability management and component-based aspects in SPrL; and, finally, to list practical examples and conducted evaluations. This research aims at reaching a broader and more consistent view of the research area and to provide perspectives and gaps for future research.

Method: We performed a systematic literature review according to well-established guidelines set. We used tools to partially support the process, which relies on a six-member research team.

Results: We report on 49 primary studies that deal mostly with conceptual or theoretical proposals and the domain engineering stage. Years 2014, 2015, and 2018 yielded the largest number of articles. This can indicate SPrL as a recent research theme and one that attracts ever-increasing interest.

Conclusion: Although this research area is growing, there is still a lack of practical experiences and approaches for actual applications or project-specific process derivations and decision-making support. The concept of an integrated reuse infrastructure is less discussed and explored; and the development of integrated tools to support all reuse stages is not fully addressed. Other topics for future research are discussed throughout the paper with gaps pointed as opportunities for improvements in the area.

1. Introduction

In order to address the demands of a highly competitive market and generate high-quality software products, development organizations require definition, management and continuous improvement of their software processes [1]. However, this is not a trivial task.

Software process is a key-concept of software and systems engineering as it encompasses models, methods and techniques for defining and managing software and IT projects. Thus, high-quality products might be reached from a high-quality process [1]. A process engineer usually defines a specific process for each project in an ad-hoc fashion, which is expensive, unrepeatable and error prone [2]. A common strategy is to tailor a software process by adjusting the definitions and/or customizing the terms of a general process description, thus deriving a new process, which is applicable to a specific environment [3].

A systematic and intentional reuse of software process specifications can be used as a way to reuse past experiences aiming: (i) to reduce the time required for customizing software processes to meet the demands of specific projects; and (ii) to facilitate the management of variability of software process families, seeking to reach maturity by

* Corresponding author.
E-mail address: danny@cos.ufrj.br (E. Nogueira Teixeira).

https://doi.org/10.1016/j.infsof.2019.08.007
Received 1 February 2019; Received in revised form 9 August 2019; Accepted 12 August 2019
Available online 13 August 2019

incorporating knowledge produced in successive executions. The knowledge on software product reuse, such as architectures populated with reusable components, is essential for the success of software process reuse [4]. To this end, software product line [5,6] techniques have been adapted to Software Process Line (SPrL) approaches [7–9].

However, as far as we know, there is no unified approach or consensus regarding how to perform software process reuse using SPrL technique. This can be considered an early research area and it is important to identify the solutions that are already in place, as well as the challenges that still need to be addressed in the field. Aiming to have a broader and more consistent view on software process reuse approaches, this paper presents a systematic literature review (SLR) of SPrL approaches. The goal is to identify existing SPrL approaches and to characterize the state-of-the-art of the area. This analysis involves: (i) an overview of the area in terms of publication venues, citations, demographic distribution, SPrL engineering stages, process engineering stages and tool support; (ii) the identification of process variability representations and process component definitions applied in SPrL approaches; and (iii) evaluation of proposed SPrL approaches, such as, qualitative and quantitative studies, case studies and surveys. Perspectives and gaps of research are also discussed. All analysis can help in identifying the need for new solutions or improvements to the ones already provided.

This study contributes to academia and industry as: it provides researchers and practitioners discussions on state-of-the-art of software process reuse; it points out research opportunities to be developed in both academic and industrial sets; it produces a synthesis on variability representation approaches for SPrLs, as well as component based definitions commonly applied to software process reuse; it discusses on current evaluation techniques of SPrL approaches performed in academia and industry; and it was conceived based on main principles of Open Science and Openness as an opportunity to reproduce it based on its availability of data.

The remainder of this paper is organized as follows. Section 2 presents the main concepts needed to understand the topics discussed in this study and related work that combines five systematic literature reviews in process reuse. Section 3 describes the main goal of this literature review, its planning and execution stages. Section 4 presents and Section 5 discusses obtained results. Section 6 lists threats to validity and action to mitigate them, as well as limitations of this work. Finally, conclusions and future work are outlined in Section 7.

2. Background and related work

2.1. Software process lines

Based on valid analogies between software products and processes, software reuse techniques are being applied in the process area aiming at contributing to improvements in the process definition task. Software Process Line (SPrL) is one technique proposed for systematic process reuse. This can be defined as ‘a set of processes in a particular domain or for a particular purpose, having common characteristics and built based upon common, reusable process assets’ [7]. This ‘prepares the processes of an organization to suit future anticipated (process) needs by determining a common process core and variable process parts that satisfy specific needs’ [9].

According to Rombach [8], SPrL engineering includes two separate development processes: (i) the domain engineering, which creates processes for reuse; and (ii) the application engineering process, which develops project-specific processes based on the selection of process components developed in the domain engineering and on the customization of their respective variability.

Variability refers to the variable aspects of the specific processes of a SPrL [10]. In this point, the SPrL representation is a relevant topic. Process modelling languages, and therefore the metamodels which support them, need to include variability constructors [11]. The notation expressiveness and understandability are considered as some of the most relevant factors that affect the adoption of SPrL by companies [12]. Variability representation has concerns about which software processes elements and relationships can vary and how these variations can occur and be classified.

Component based development (CBD) [13] aspects have also been applied by approaches that conduct the definition of software processes using components [14]. These process components are organized by an architecture, a structure that establishes the process elements involved, their interfaces, ordering, interdependencies, and other relationships amongst them [15]. However, it is not clear into which extent this product reuse concept has been applied in process lines approaches and how process domain knowledge can be organized into a process component based architecture.

Another important SPrL aspect is the process reuse infrastructure. The development, customization and evolution of SPrL without supporting tools may be a very difficult task [16]. The definition of supporting tools can enable the practical use of process reuse approaches, providing a major possibility of their application in real scenarios. This aspect was pointed out in a survey performed in [16] as the biggest concern of the participants. Subjects considered that process reuse can be seriously threatened or can face the same difficulties presented in software reuse in case of inadequate supporting tools[16].

This can be considered an early research area and it is important to identify the solutions that are already in place, as well as the challenges that still need to be addressed in the field. A literature review can bring an understanding of the conceptual and practical development of the area and the extension to which this is being addressed. Initial discussions are presented in the related work identified in next section.

2.2. Related work

In searching for a detailed description of the area and its proposed solutions and applications, only a few works were identified as dealing with the analysis of the state-of-the-art in software process reuse. Four SLRs were previously selected as related work [17–20] (Table 1), although only one addressed the SPrL area directly. More recently, a fifth review was published [21] and included as a related work. This was done in conjunction with two authors of this review and used the set of papers collected until June 2016. The focus was on the specific analysis of the SPrL application engineering stage, including the identification of available tools and techniques applied to support the solving of variability issues in the process domain.

At this point it is important to highlight that the intent of this review is to present more detailed results on both the SPrL engineering stages support and their related process reuse concepts. This involves quantitative and qualitative analysis, including different publication distributions and a detailed characterization for each approach identified as its research groups. The relations found among the studies we identified were presented as a differential aspect of this work as the frequency of citations, along with a detailed description of the tool support and practical examples and evaluations performed. It also presents updates of the findings in the research area until May 2019 and proposes paths for future research.

Pedreira et al. [17] present a SLR discussing the main issues for the software process tailoring area, and providing an up-to-date framework in which to position new research activities. Five search engines were used: IEEE Digital Library, ACM Digital Library, Wiley InterScience (computer science area), Science Direct (computer science area) and SpringerLink. A total of 28 works were analysed. This represents a broader scope than SPrLs.

The SLR conducted by Martínez-Ruíz et al. [18] also deals with the process tailoring area, covering techniques other than the SPrLs. The search string used was “’software process’ AND (‘variability’ OR ‘variant’ OR ‘variation point’ OR ‘tailoring’ OR ‘flexibility’ OR ‘customization’ OR ‘process instantiation’ OR ‘adaptation’)”. Five search engines were used:

Table 1
Related work.

SLR	Period	Primary studies	Goal	Investigated Area
Current SLR	Until May 2019	49	The main goal of this SLR is to characterize the state-of-the-art in SPrL approaches.	SPrLs
Costa et al. (2018) [21]	Until June 2016	26	The main goal was to characterize the SPrL application engineering phase.	SPrL application engineering phase
De Carvalho et al. (2014) [20]	1996–2013	40	"SLR that aims to characterize existing approaches on SPrL", focusing on variability representation	SPrLs
Rocha and Fantinato (2013) [19]	2003–2012	63	SLR analyzing software product line approaches applied in BPM.	Software Product Line approaches applied to Business Process Management
Martínez-Ruíz et al. (2012) [18]	1991–2009	32	Main goal was "to describe the state-of-the-art in process tailoring, stating a complete set of requirements that need to be fulfilled by a tailoring notation, according to the particular needs identified in relevant tailoring initiatives" and "to analyze those tailoring mechanisms that are currently in existence and that consistently support process tailoring".	Process Tailoring
Pedreira et al. (2007) [17]	Up to 2006	28	SLR investigating software process tailoring area	Process Tailoring

Science Direct, Wiley InterScience, Springer-Link, IEEE Digital Library and ACM Digital Library. This SLR focused on analysing the process variation that can occur during process adaptation and the process tailoring mechanisms. A total of 32 works were analysed.

The SLR conducted by Rocha and Fantinato [19] analysed the application of software product line approaches in the business process management (BPM) domain, not addressing SPrLs specifically. The search string used was: "((('product* line*' OR 'product*famil*' OR pl OR spl OR sple OR spf) AND (bpm* OR (process* AND software*) OR (process* AND business*) OR workflow*)) OR ('process famil*') OR ('process line*'))". It was applied to seven search engines: Scopus, ISI Web of Science (WoS), IEEE Xplore, ACM Digital Library, SpringerLink, ScienceDirect and Engineering Village. This SLR focused on business processes, analysing which PM stages are supported per product line approaches. A total of 63 works were analysed.

De Carvalho et al. work [20] presents an SLR that aims to characterize existing approaches of SPrL, focusing on variability representation. The search string used was "('process line' OR 'process lines' OR 'process family' OR 'process families' OR 'variant rich process' OR 'variant-rich process') OR (('business process' OR 'business processes' OR 'software process' OR 'software processes' OR 'process model' OR 'process models') AND (variability OR variabilities OR variation OR variations OR variant OR variants OR commonality OR commonalities OR similarity OR similarities OR 'feature model' OR 'feature models' OR 'feature modelling' OR 'feature-based'))". It was applied to three search engines: Scopus, IEEE Xplore and Engineering Village. A total of 40 works were analysed.

The most recent related work [21] focused on SPrL application engineering stage. The review was done in conjunction with two authors of this review and used the 42 studies collected in this review until June 2016 as the primary source. Sixteen studies were excluded because they did not report activities of the application engineering stage, ending with a total of 26 papers analysed.

The first four related SLRs were published within a period of seven years. Amongst them, the oldest one was published in 2007 and the most recent in 2014. The reviews covered primary studies between 1987 and 2013. This current SLR covers a 13-year period from 1996 to 2019, analysing 49 primary studies. In analysing the data sources, each review chose a different number of search engines from 3 to 7, including the most cited digital libraries as relevant publishers and organizations in Software Engineering (IEEE Xplore (5 SLRs) – ACM Digital Library (3 SLRs) – Wiley InterScience (2 SLRs) – Science Direct (3 SLRs) – SpringerLink (3 SLRs) – Scopus (3 SLRs) – Engineering Village (3 SLRs) and ISI Web of Science (WoS) (1 SLR)).

It is important to point out the difference in results found as related to the domain considered in the search. The population and intervention are different among the SLRs, although the main goal was to identify approaches that support process definition by applying process tailoring mechanisms.

The first SLR [17] analysed different techniques in the process tailoring area, including component-based and generator approaches. The authors concluded that "there was little research available in this field to date and that there is still some work to be done in software process tailoring" [17]. Also, the degree of formality in the process tailoring activity was pointed as a topic to be addressed, indicating the need for a more general framework for process tailoring considering a broad range of situations. The works identified cover the period from 1995 to 2006 (only one was identified in 1987) and only two authors presented in the current SLR were identified. This can reinforce the notion that SPrL is a more recent approach to process reuse (the major approaches found in this current SLR were published from 2006 to 2018, peaking in 2014, 2015 and 2018). This first review was pointed as the only existing related work by the second related SLR [18]. This one considered the concept of process variability as a tailoring mechanism that was currently in existence focusing on the search by terms related to this concept and adaptation. The works range from 1991 to 2009 and have some convergences with this SLR (two research work and more than eight authors of the identified papers). SPrL is directly mentioned as one of the tailoring approaches that appeared around 2002 by employing product-based initiatives as a product line paradigm for software process tailoring. Also, the elements and operations of process variability are discussed and requirements for building a variability notation are listed which, as pointed earlier, is a real need.

Only the last two SLRs and this one aim to characterize existing approaches specifically in the SPrL area. All started with works from 1996, but De Carvalho et al. [20] applied a cut in time to determine a search period and this one handled with the searches performed to produce returns with no time restriction imposition. Although having the same main scope, the search string applied was different and the result differs in 24 papers in converging period (until 2013) and 24 papers added in this review (2014 to 2018). The current SLR produced a more detailed explanation for each approach with an analysis of relations among them (Section 4.1), not found in De Carvalho et al. (2014) review. Different topics are addressed in each SLR: stage(s) of the software process lifecycle and component based strategy of software reusable process elements are discussed in the review of this paper. On the other hand, Process Quality Models and Standards and Software Engineering Paradigms were discussed in De Carvalho et al. (2014) review. Topics as tool support, process variability management and notations, approach evaluations and practical examples can be found with more details and appointments in this work.

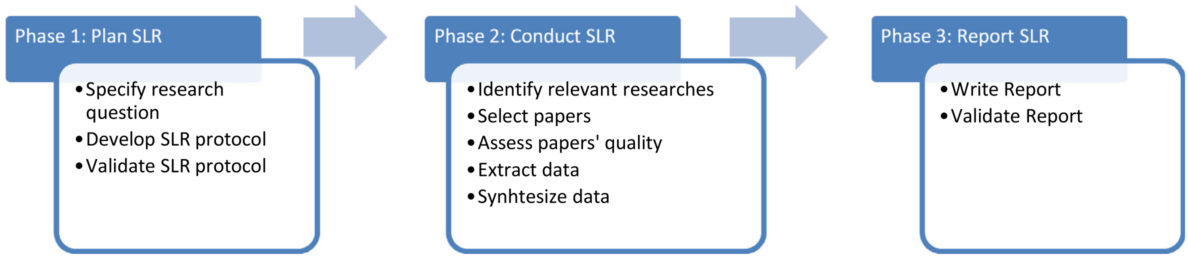


Fig. 1. Systematic Review Process [22].

Although the Costa et al. (2018) review used the 42 studies identified in this review until June 2016 as the primary source, it is important to highlight the differences in scope between them. Costa et al. focused on the application engineering stage, selecting 26 studies that address this specific stage. Because of this, that work addresses different topics as techniques used in variability resolution and decision-making during process definition and the use of context information. Five techniques were identified and all the approaches provide some mechanism to represent context information even though only a few of them indicated the real predefined context information sets that can be used.

In this current SLR, a total of 49 works are identified as SPrL approaches for the 1996–2018 period. The aim is to provide a broader characterization of the area, including a more detailed analysis as per set of different criteria, including: (i) a more detailed description of the works, (ii) indications of research groups working in the area, (iii) analysis of the level of impact of the approaches after an identification of reference and citation relations amongst them, (iv) tool support, (v) variability management support, (vi) component concept application, and (vii) approach evaluations, amongst others.

It is possible to conclude that the reviews have supplementary aspects that can be investigated in conjunction to produce a more general understanding of the process reuse area, its implications and possible future agenda.

3. SPrL systematic literature review

A SLR can be identified as a secondary study type, i.e., a study based on aggregating evidence published in primary studies. The main reason for performing a SLR instead of an ad-hoc literature selection is to improve the quality of scientific studies collected on a specific research area by conducting the review as a well-defined sequence of steps (Fig. 1).

The protocol of our study was defined based on reference research work in the field [22–24]. It was structured after the definition of research questions, SLR execution procedures (data collection procedures, data analysis and data synthesis) and quality assessment (validity procedures). StArt [25], a tool developed at the Federal University of São Carlos (UFSCar) to partially support the SLR process, was used to support the conducting of this SLR, organizing the information retrieved.

The research team responsible for conducting the SLR consisted of 6 members, one MSc student and five Ph.D. researchers. The team consists of two members of the Software Reuse Group at COPPE/UFRJ, two members of the Collaborative and Automated Software Engineering Laboratory (CASE Lab) from the Federal University of Rio Grande do Norte (UFRN), one member from the University of Fortaleza and one member of the Research Group on Systematic Software Reuse and Continuous Experimentation (GReater) at State University of Maring'a (UEM).

3.1. Literature review planning stage

The first step of our study consists of defining a review protocol. The protocol specifies the SLR objective, research questions, and criteria to support the definition for selecting the studies. The search strategy used is established by defining the search string, the data sources and the selection process.

3.1.1. Formulation of objectives and research questions (RQ)

The main goal of this SLR is to characterize the *state-of-the-art* in SPrL approaches. This goal involves gathering and analysing scientific publications on software processes with the purpose of characterizing approaches related to the use of a SPrL from a researchers' point-of-view in the academic and industrial context. To this end, the main research question (RQ) in this SLR was '*Which software process line approaches have been proposed and how do they support software process reuse?*'. Secondary questions (SRQs) were investigated (Table 2) to structure the relevant information acquisition to answer the main research question.

The first secondary question (SQR1) implies an overview of SPrL approaches as a research area. The SPrL approaches identified are presented and a quantitative analysis is provided indicating some distributions such as: (1) number of publications per year; (2) citations and references among them; (3) publishing countries; (4) publishing venues; and (5) main contributions identified. A supplementary qualitative analysis was done and descriptions with important aspects of each approach and related research groups were also presented.

Other main aspects include: (i) which SPrL engineering stage was dealt with by each approach, (ii) which stage(s) of the software process lifecycle was (were) addressed, and (iii) which tool support was provided. The percentage of support for each SPrL engineering stage (DE - Domain Engineering and AE - Application Engineering) was presented and could indicate if there was some concentration of research works in one of the stages. The support to the stages of the software process lifecycle was analysed in order to understand how the reuse process is already contributing to activities in engineering processes such as definition, enactment, simulation, execution and improvement. The definition of some software tools can reflect the practicality and degree of maturity of the SPrL approach. So, considering the importance of the tool support

Table 2
Main Research Question (RQ) and Related Secondary Questions (SRQs).

Id.	Research questions
RQ	**Which SPrL approaches have been proposed and how do they support software process reuse?**
SRQ1	What are the overview of the area and the main characteristics of the proposed approach?
SRQ2	Does the approach apply software reuse concept as process variability treatment and/or a componentization strategy?
SRQ3	How was the described approach evaluated?

for SPrL implementation, the implementation or the adaptation of the software tools was analysed.

The analysis of the second set of secondary questions (SQR2) is the application of software reuse concepts in SPrL approaches such as variability modelling and component definition. Software process variability representation is a relevant aspect in SPrL development and management [11,26]. Therefore, it is important to identify what the existing variability modelling proposals are and how they provide support to desirable requirements for a software process variability representation [27] such as: (i) graphical representation; (ii) support to model configurable process elements (variation points), their alternatives (variants) and artefact optionality; (iii) support to specify relationships among elements and behaviour variation designed by alternative or optional control flows; and (iv) support to specify relations of dependency and mutual exclusivity among elements.

In addition, the process definition by composition based on smaller units called process components is pointed out as a relevant aspect by reference models and standards, such as MPS.BR [28][1]. This structuring of the component domain information can increase the modularity of the knowledge on the reusable process domain. The application of this concept was analysed to identify whether there was some general agreement on the definition of process component, domain organization in architectures and the existence of methodologies to develop these artefacts.

The third secondary question (SRQ3) can be used as a quality assessment aspect in each approach. It indicates a concern to reveal the pros and cons of each approach by analysing how they were evaluated. This question seeks to identify: (i) types of evaluation of SPrL approaches applied; (ii) practical examples used; and (iii) specific criteria or metrics presented to guide the evaluation.

3.1.2. Review scope

PICO (Population, Intervention, Comparison and Outcomes) structure [29] was used to define the review scope: (P) Population: Software process lifecycle research works. (I) Intervention: SPrL approaches. (C) Comparison: Not applied. The SLR's objective is to characterize the approaches. (O) Output: Techniques, Approaches, Methods, Methodologies, Tools and Processes that support software process lifecycle using SPrL.

3.1.3. Control papers

An initial understanding of the area can be identified by control papers. The set of papers selected is based on a previous ad-hoc literature review and on previous SLRs in the software process reuse area. The three papers used as control were:

- ROMBACH, H.D., 2006, 'Integrated Software Process and Product Lines'. In: International Software Process Workshop, pp. 83–90, Beijing, China, May.
- WASHIZAKI, H., 2006, 'Building Software Process Line Architectures from Bottom Up'. In: Product-Focused Software Process Improvement, pp. 415–421, Amsterdam, Netherlands, June.
- ARMBRUST, O., KATAHIRA, M., MIYAMOTO, Y., et al., 2009, 'Scoping Software Process Lines', Software Process: Improvement and Practice, v. 14, n. 3, pp. 181–197.

3.1.4. Search string definition

The keywords that form the scope of the research questions were defined considering the PICO structure, the SLR's objectives and the knowledge of the experts. Terms in population, intervention and outcomes were applied associated with alternative spellings and synonyms connected by logical operators. The set of control publications was also used to refine the initial strings. The final search string used was:

('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** 'tailing of processes' **OR** 'process development' **OR** 'processes development' **OR** 'process engineering' **OR** 'processes engineering' **OR** 'process design' **OR** 'software process modelling' **OR** 'software process modelling' **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')

3.1.5. Data source definition

The selection strategy for search engines was defined by analysing their ability to use logic expressions, full text recovery, automated searches of paper content and searches using specific fields (for example, title, abstract, keywords). The electronic databases used were (1) Scopus[2]; (2) IEEEXplore (IEEE)[3]; and (3) Compendex[4]. These are some of the most relevant databases in the software engineering area. ACM Library overlaps the IEEEXplore Library and its content is indexed by the Scopus Library. Thus, its content can be considered covered by the search.

The search string was syntactically adapted to the different search engines and applied on the title, abstract and keywords.

3.1.6. Inclusion and exclusion criteria

Inclusion and exclusion criteria were defined to support the selection activity of primary studies considered relevant to this review. The study had to meet the following Inclusion Criteria (IC):

- IC: The full reading of the paper should allow the identification of at least one aspect that supports a SPrL approach, which can be identified as the presence of keywords found in a search string, SPrL definitions and main characteristics described in control papers.

Six Exclusion Criteria (EC) were defined to support the exclusion of papers considered outside the review scope, which should meet at least one of the following ECs:

- EC1: Papers that do not partially or fully define a SPrL approach should not be included;
- EC2: Papers not addressing software process reuse by applying a SPrL approach but using other reuse techniques should be excluded;
- EC3: Papers that are duplicated;
- EC4: Papers described and/or presented as keynote speech, tutorials, courses and similar items;
- EC5: Papers whose full contents cannot be accessed; and
- EC6: Papers that are not written in English.

[1] MPS.BR - Brazilian Software Process Improvement Reference Model.

[2] http://www.scopus.com.

[3] http://ieeexplore.ieee.org.

[4] http://www.engineeringvillage2.org.

Table 3
Number of papers returned by search.

Search engine	Retrieved studies in November 2014	Retrieved studies in June 2016	Retrieved studies in May 2019	Total retrieved studies
Scopus	170	136	12	318
IEEEXplore	32	448	704	1184
Compendex	59	16	8	83
Total of papers returned by search	**261**	**600**	**724**	**1585**

3.1.7. Papers selection process

The paper selection process was done in three steps:

1. For each publication returned, its title, authors, abstract, keywords, year of publication were catalogued with the StArt tool for later analysis.
2. Relevant paper selection: first, the duplicated papers were extracted. Later, the remaining papers were classified as included or excluded by two researchers by applying the inclusion and exclusion criteria. A third researcher solved the divergences raised from assessments conflicts. To reduce the risk of excluding a possibly relevant paper at an early stage, when a doubt appeared amongst the researchers, the publication was included.
3. Full paper recovery and reading stage: the full content of each paper selected in the second step was read. The set of papers whose full contents could not be retrieved was excluded from the set of possible reading papers by applying EC5 as described in Section 3.1.6. During the reading stage, papers that did not define a SPrL approach were removed. To analyse this, the following topics were considered: (1) the presence of keywords identified in the search string, and (2) SPrL definitions and their main characteristics as described in the control papers. Additionally, the authors' knowledge on the process reuse area contributed to the exclusion of the papers. After defining the final list of relevant publications, four of the authors read and extracted information using the data extraction form (see Appendix A). One of the authors was responsible for summarizing the data extracted for all the papers selected for later analysis.

3.2. Literature review execution stage

This stage applies the review protocol. The main activities of the SLR execution stage were: (1) search execution and study identification; (2) study selection; and (3) data extraction.

3.2.1. Search execution

This activity was first done in November 2014. To update the findings, new searches were made in June 2016 and May 2019. The number of papers returned in each round of search are shown in Table 3. In the first search round, a total 261 papers were returned. In the second round, which covers searches from 2014 until 2016, a total of 600 papers were returned. In the third round, 724 papers were returned, covering the period from 2016 until 2019.

3.2.2. Studies selection

First, the paper duplication found in both searches was identified and a total 168 duplicated papers were eliminated from a total 1585 papers returned, with 78 duplications from 261 papers returned in the first search, 56 duplications from 600 papers returned in the second search and 34 duplications from 724 papers returned in the third search.

In the first search round, from a total 183 papers analysed (261 papers returned without 78 duplications), a refined group of 41 studies initially picked after a first assessment conducted by two researchers. In this stage, nine divergences were identified, indicating that it was unclear whether any studies matched the inclusion criteria. After the divergences were solved by a third researcher, 36 studies were selected for analysis in the next stage, i.e., paper recovery and reading (Fig. 2), with a total amount of 147 papers excluded.

In the paper recovery and reading stage, the full contents of 3 papers could not be retrieved, so they were excluded from the set of possible papers to be read (EC5 - Section 3.1.6). In the reading stage eight papers were excluded for not defining a SPrL approach. Finally, 25 papers were selected to be analysed after the first search round.

The same procedure applied to the first search was used in the second one. From a total 544 papers analysed, a refined group of 43 studies was initially selected. Twenty-nine analysis divergences were identified and solved by a third researcher, after an analysis of the title and abstract. After the solution of the divergences, 22 studies were selected for analysis in the next stage, i.e., paper recovery and reading (Fig. 3), with a total 522 papers excluded.

During the reading stage, five papers were excluded for not defining a SPrL approach (EC1 - Section 3.1.6). Finally, 17 papers were selected to be analysed after the second round of search.

In the third round of searches, a total 690 papers were analysed after exclusion of 34 duplications. From this, a refined group of 9 studies were initially selected. During the reading stage, two papers were excluded for not defining a SPrL approach (EC1 - Section 3.1.6). A total of seven papers were selected to be analysed (Fig. 4).

It should be pointed here that the IEEE library yielded a high number of papers in the second and third search rounds. This difference can be explained by the change in the search engine that occurred after the first search. A strategy needed to be applied in other to guarantee the same search string application.

From 1585 papers returned in the searches, 168 duplicated papers were eliminated, 1350 papers were excluded in the first filter, as they were clearly outside the scope of this review. A total of three papers were excluded because their full contents could not be retrieved. And finally, fifteen papers were excluded after the reading stage for not defining a SPrL approach. To summarise, a total 49 papers were identified as relevant studies to be analysed in the SPrL area supporting software process reuse (Table 4).

3.3. SLR data sharing

We followed main principles of Openness based on Open Science in this work, mainly for collecting and sharing data [30,31].

Package with data and metadata of main results of this SLR is available at Zenodo.org https://doi.org/10.5281/zenodo.3247655. In such package there is:

- one bibtex-format file with entries of the 49 papers final selected;
- several CSV-format (";" separate mark) files with data of this SLR. Each DATA file has its respective METADATA file, which describes the structure of the DATA CSV file. All CSV METADATA files has two lines:
  - in the first line is the description of each DATA file columns; and
  - in the second line there is a description of the content format of each column from the DATA file.

In addition, this study available data is under Creative Commons Licence (CC-BY), accordingly to the following terms at https://creativecommons.org/licenses/by/4.0, thus one is free to:

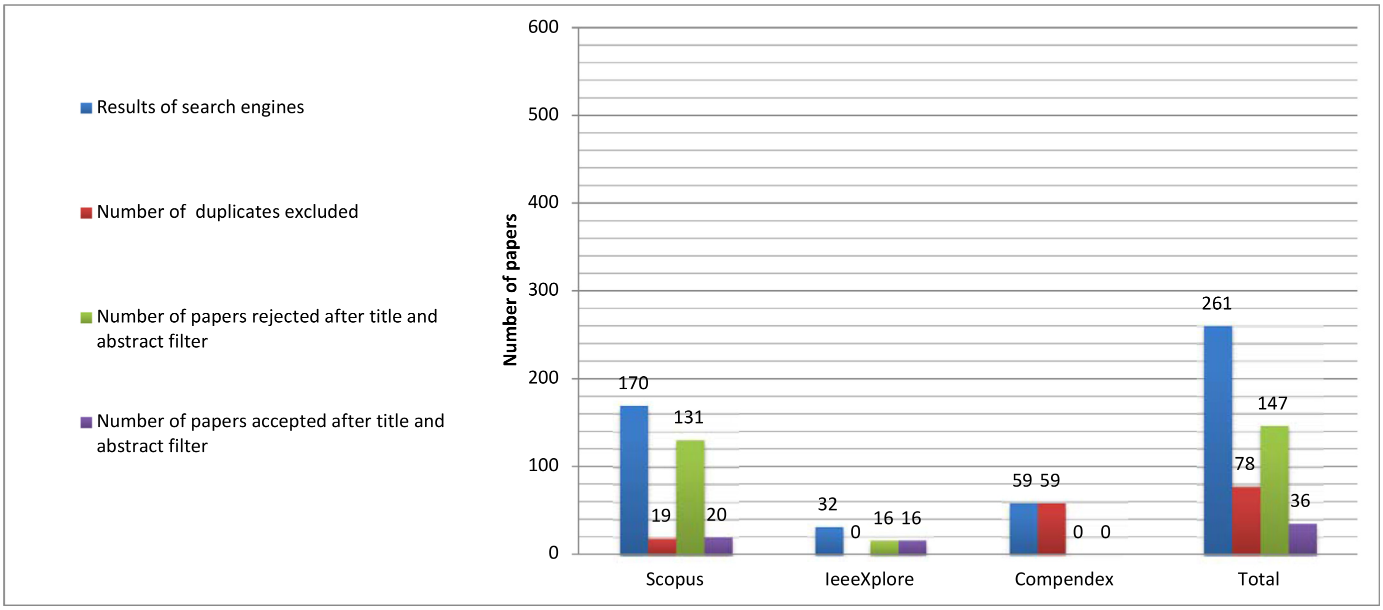


Fig. 2. Filtering process – First Search.

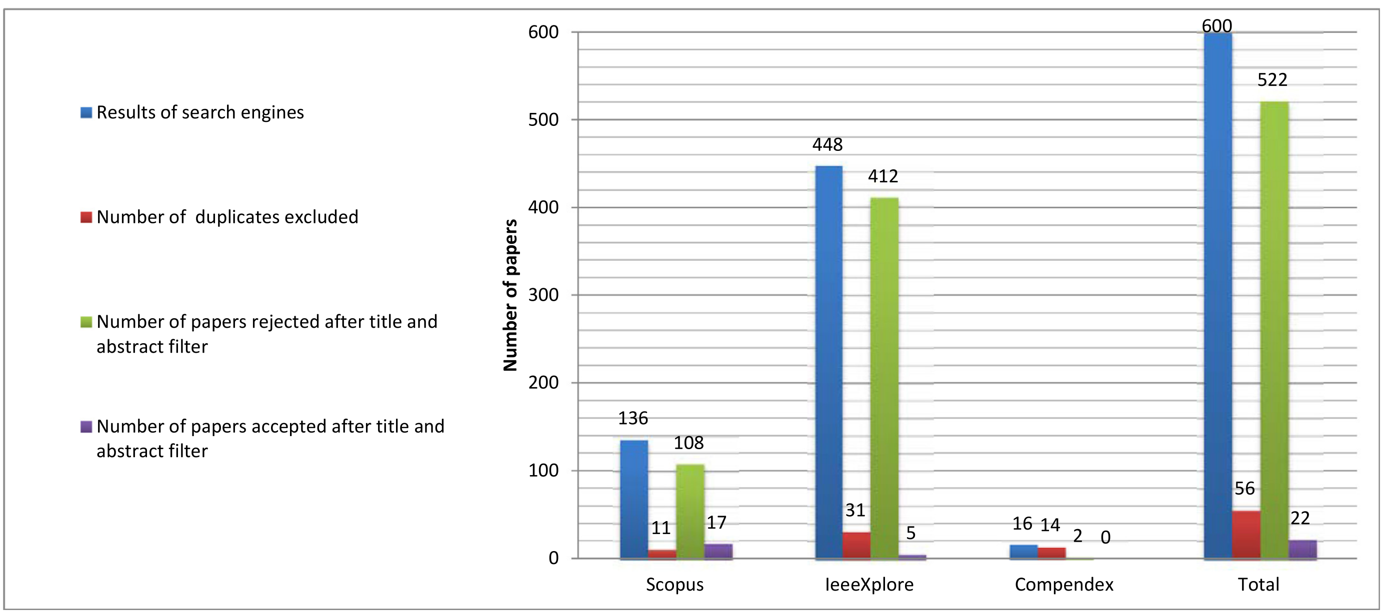


Fig. 3. Filtering process – Second Search.

Table 4
Final number of search execution and studies selection.

Search engine	Retrieved studies in November 2014	Retrieved studies in June 2016	Retrieved studies in May 2019	Total retrieved studies
Scopus	170	136	12	318
IEEEXplore	32	448	704	1184
Compendex	59	16	8	83
Total of papers returned by search	**261**	**600**	**724**	**1585**
Numbers of excluded papers				
Duplicated papers	78	56	34	168
Exclusion criteria application in title and abstract	147	522	681	1350
Recovery stage (full content could not be retrieved)	3	-	-	3
Full reading stage (Exclusion criteria application in full content)	8	5	2	15
Total of exluded papers	**236**	**583**	**717**	**1536**
Numbers of final selected papers				
Final numbers (Total of papers returned Total of exluded papers)	**25**	**17**	**7**	**49**

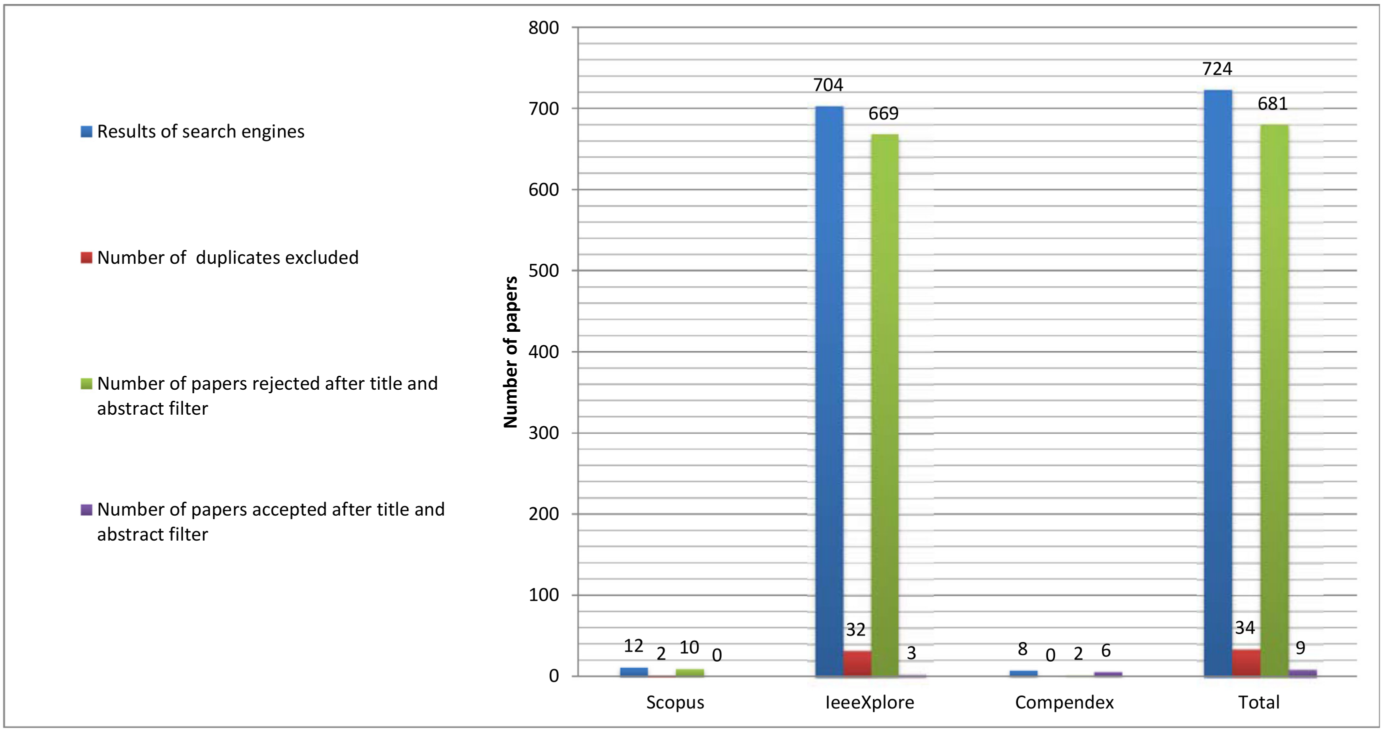


Fig. 4. Filtering process – Third Search.

- Share: copy and redistribute the material in any medium or format; and
- Adapt: remix, transform, and build upon the material for any purpose, even commercially.

4. SLR Results

This section describes the results obtained after the SLR data analysis. Quantitative analysis was used to assess and calculate, in relation to the total number of publications selected, the percentages to derive publication trends. Tables and charts are used as a way of summarizing the results in order to easily interpret them. The qualitative analysis focused on answering the secondary research questions addressed in the SLR. The analysis was done considering different aspects separately, addressed by each specific sub-question. This was meant to make the results more understandable.

4.1. Overview of the area and main characteristics of proposed approaches (SRQ1)

The total number of primary studies selected was 49, distributed between 1996 and 2018 (see Appendix B).

The largest group of publications can be observed from 2004 to 2018. Only two papers were identified before then, both in 1996. Years 2014 and 2015 merit some attention, with the highest number of articles, seven and nine, respectively, followed by 2018 with six papers. This may suggest SPrL is a recent research theme, one that has started to attract greater interest in the field of systematic process reuse, as shown in Fig. 5. It is important to point that 2019 characterizes the area in four months of the year, as the search only included papers until May.

Table 5 shows a graphical representation of the distribution of citations for each paper. This view helps to identify the relation amongst the approaches. The most influential publications can be identified by the number of citations, as 15 papers out of the 49 selected were not cited by others.

The most cited paper identified was that of Rombach (2006) (Table 5, Line 6), cited by 26 papers (Table 5, Line 6). This paper introduces the concept of Integrated Software Product and Process Lines, stressing the variation amongst processes and the benefits of using systematic reuse management for software products with SPLs.

The most cited second individual author, with a total 18 citations, was Washizaki with 14 mentions in the first paper (Wasizaki, 2006a) (Table 5, Line 7) and 4 citations in the other (Washizaki, 2006b) (Table 5, Line 8). The approach describes how to develop Process Lines and represent them in association with feature models.

The third most cited paper was the one by Armbrust et al. (2009) (Table 5, Line 11) with 13 citations. The authors referenced a previous work of 2008 named ‘Scoping Software Process Models - Initial Concepts and Experience from Defining Space Standards’ [32]. Their approach - SCOPE - focuses on an initial activity towards the definition of the SPrL Scope, describing a technique to identify processes and products demands of current and future projects, and legacy process analysis to identify the group with a high level of benefit for organizational goals.

Martínez-Ruíz et al. (2009) (2011a) (2011b) (Table 5, Lines 12/19/20) have a total 15 citations. This work describes the paradigm of rich-variant processes. More recent papers apply this paradigm to the Global Software Development scenario [33]. Four authors are involved in this research work: three work in the ALARCOS Research Group[5] from University of Castilla in La Mancha (Spain) and partnered with a researcher from University of Helsinki, J. Münch. The first paper by Münch in this area was published in 2005, in association with Jaufman and proposes an emergent acquisition process, focusing on process derivation with support to adaptation according to the goals and features of a project’s context (Table 5, Line 5). This author also partnered with the Armbrust group in the 2009 paper mentioned above.

Two papers from 1996 present introductory concepts in the area. The notion of process family was introduced by Sutton and Osterweil

[5] ALARCOS http://alarcos.esi.uclm.es/index_en.

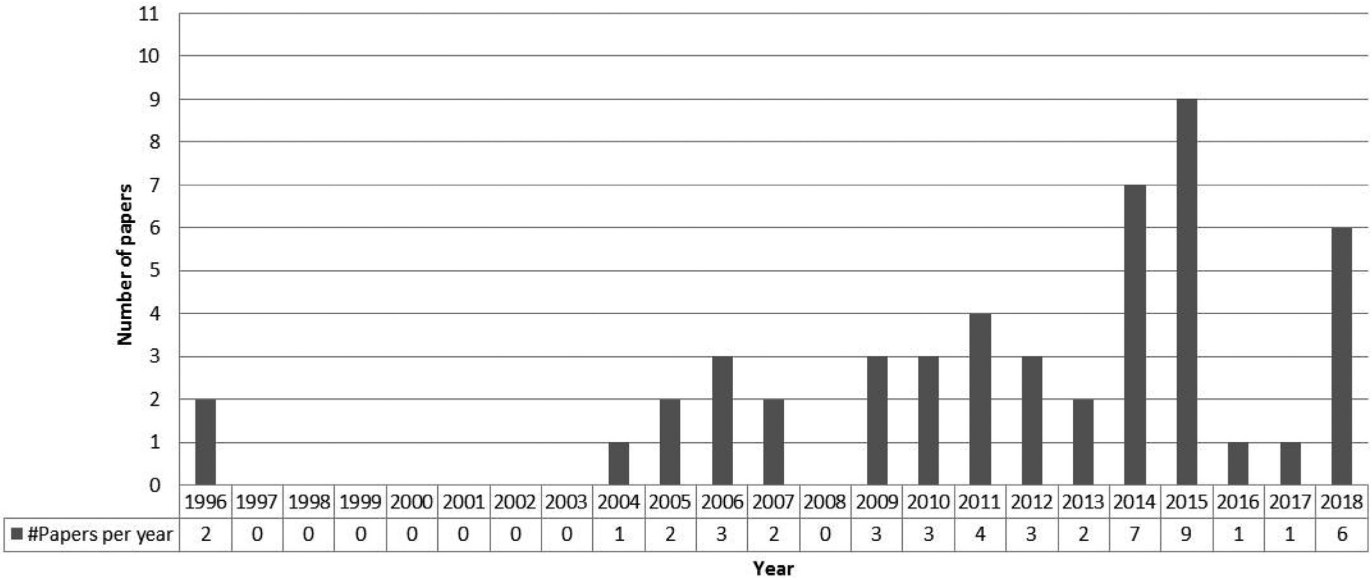


Fig. 5. Number of publications per year.

(Table 5, Line 2), in which relations and analogies amongst product and process were outlined. Hollenbach (Table 5, Line 1) presents an initial practical application of a process domain engineering method in one organization named PRC Inc., a software and systems integrator in Washington DC, in the US. This paper referenced a previous work on the topic in that same year named ‘Software Process Reuse In An Industrial Setting’ [34]. Sutton and Osterweil (Table 5, Line2) were cited by other 5 papers and Hollenbach’s (1996) (Table 5, Line 1) was cited by one paper. A reference to Hollenbach and Frakes (1996) [34] was made in at least 2 other papers.

Simidchieva et al. (2007) (Table 5, Line 9) deal with process variations by specifying behaviour changes of agents, tasks and artefacts as variants of a base process. This approach is cited by 7 papers. However, this research work did neither evolved along the years, nor developed any partnership with other authors.

The approach proposed by Ternité (2009) (Table 5, Line 13) presents the definition of process variability by means of adaptation operations such as the addition or removal of optional elements. This work was cited by eight others, especially by papers describing the partnership with two other authors: Kuhrmann and Fern’andez. Kurhman et al. (2014)(2016) (Table 5, Lines 28/42) and Schramm et al. (2015) (Table 5, Line 38) work with variability operations to support the development of software process lines in the context of German V-Modell XT.

Aleixo et al. (2010) (Table 5, Lines 14/15) present a model-driven and integrated approach to software processes variability management, customization, and execution. These works were cited by five other papers. Recently, the main author worked in partnership with Oliveira Jr to develop the SMartySPEM approach for process variability modelling [10,26,35]. The SMartySPEM approach leads with the identification and representation of process element variability and the derivation of customized processes from a SPrL. More recently, a group of authors involved in the SMartySPEM approach published a paper comparing the modelling of software process variability with SMartySPEM and vSPEM [36]. This paper was returned by the third search of this SLR (Table 5, Line 47).

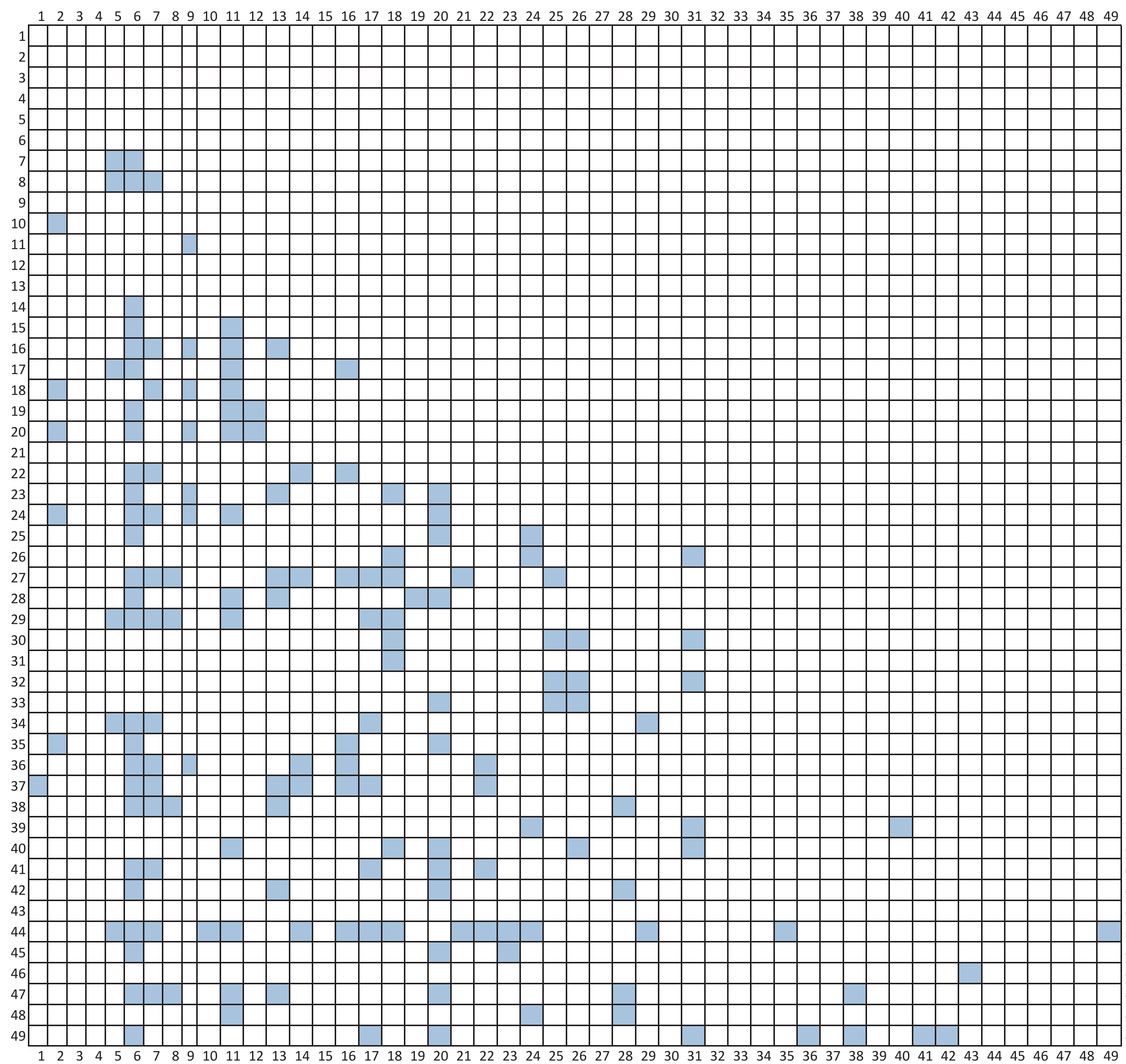
Barreto et al. (2010)(2011) (Table 5, Lines 16/17) propose a software process definition approach based on reuse techniques such as process components, architectures, process lines and features. These works had a high number of citations, totalling 14 citations.

A set of nine papers (Table 5, Lines 18/24/25/26/30/31/32/39/40) were published by a group of ten researchers from Adapte Project[6] - UChile, PUCV and USM. They have 28 citations. Alegría et al. (2011) (2013) (Table 5, Lines 18/24) propose software process tailoring using a model-based approach that automatically generates project-specific processes using organizational process and project contexts. Recently, the group started developing tool support (Silvestre et al., 2014) (Table 5, Line 30) and applying the megamodelling concept to practical approach application (Bastarrrica et al., 2014) (Simmonds et al., 2015) (Table 5, Lines 26/40). This is the research group with the highest number of publications in the area. Some of these authors worked in an approach for the discovery of SPrL (Blum at al., 2015 – Table 5, Line 33). In the third search, a new publication was returned involving one of the authors mentioned in a comparative study of SPrL scoping approaches (Ruiz et al., 2018)(Table 5, Line 48).

Magdaleno et al. (2012) (Table 5, Line 22) present the COMPOOTIM approach to support the decisions of project managers on the selection and combination of process components towards process definition in the context of a particular project. An optimization mechanism is proposed, to provide better suggestions in a solution space of processes. This approach has four citations. Two more recent research works were collected in this SLR second round of searches and one of them has one citation. One presents a case study of a real process line creation in the context of an oil and gas company in Brazil (Magdaleno et al., 2015a) (Table 5, Line 37) and the other one describes an optimization approach to maximize collaboration in the composition of a software process (Magdaleno et al., 2015b) (Table 5, Line 36). Both papers are COMPOOTIM approach extensions and application. These works use a first version of *OdysseyProcess-FEX* [27], a process domain metamodel and notation for SPrL variability representation, which defines an abstract syntax for SPrL feature modelling. Evolutions in *OdysseyProcess-FEX* were performed including the use of cardinality, the component level development, specification of control and data flows, and treatment of variations in processes behaviours, as described in Nogueira Teixeira et al. (2018) (Table 5, Line 49).

[6] ADAPTE (Adaptable Domain And Process Transformation Engineering) http://www.adapte.cl.

Table 5
Selected papers and references among them.



	Authors	# citations
1	Hollenbach, 1996	1
2	Sutton and Osterweil, 1996	5
3	Durán et al., 2004	0
4	Jaufman, 2005	0
5	Jaufman and Münch, 2005	6
6	Rombach, 2006	26
7	Washizaki (a), 2006	14
8	Washizaki (b), 2006	4
9	Simidchieva et al., 2007	7
10	Thränert and Werner, 2007	1
11	Armbrust et al., 2009	13
12	Martínez-Ruiz et al., 2009	2
13	Ternité, 2009	8
14	Aleixo et al. (a), 2010	5
15	Aleixo et al. (b), 2010	0
16	Barreto et al., 2010	7
17	Barreto et al., 2011	7
18	Hurtado Alegría et al., 2011	8
19	Martínez-Ruíz et al. (a), 2011	1
20	Martínez-Ruíz et al. (b), 2011	12
21	Jafarinezhad and Ramsin, 2012	2
22	Magdaleno et al., 2012	4
23	Rouillé et al., 2012	2
24	Hurtado Alegría et al., 2013	5
25	Simmonds et al., 2013	4
26	Bastarrica et al., 2014	4
27	de Carvalho et al., 2014	0
28	Kuhrmann et al., 2014	4
29	Lorenz et al., 2014	2
30	Silvestre, 2014	0
31	Silvestre et al. (a), 2014	6
32	Silvestre et al. (b), 2014	0
33	Blum et al., 2015	0
34	Brondani et al., 2015	0
35	Garcia et al., 2015	1
36	Magdaleno et al. (a), 2015	1
37	Magdaleno et al. (b), 2015	0
38	Schramm et al. , 2015	2
39	Silvestre et al., 2015	0
40	Simmonds et al., 2015	1
41	Teixeira et al., 2015	1
42	Kuhrmann et al., 2016	1
43	Varkoi et al., 2017	1
44	Costa et al., 2018	0
45	Delgado et al., 2018	0
46	Gallina and Iyer, 2018	0
47	Pazin et al., 2018	0
48	Ruiz et al., 2018	0
49	Nogueira Teixeira et al., 2018	1

Some papers were cited by low numbers of citations and no additional research work of SPrL of these authors was recently found in the literature: (1) Durán et al. (2004) (Table 5, Line 3) (zero citation); (2) Thränert and Werner (2007) (Table 5, Line 10) (one citation); (3) Jafarinezhad and Ramsin (2012) (Table 5, Line 21) (two citations); (4) Rouillé et al. (2012) (Table 5, Line 23) (two citations); (5) De Carvalho et al. (2014) (Table 5, Line 27) (zero citation); (6) Garcia et al. (2015) (Table 5, Line 35) (one citation); and (7) Brondani et al. (2015) (Table 5, Line 34) (zero citation).

The first one proposed a process family approach to define a central process with variability in some elements that can be adapted to the specific needs of organizations using a manner of control. The approach focuses on the area of requirements engineering. The second one presents an introductory approach to the knowledge of reuse processes, using the process family concept. The third describes a process factory approach based on features applied to requirements engineering. The fourth approach analyses the Common Variability Language (CVL) application to processes for the modelling of variability.

De Carvalho et al. (Table 5, Line 27) present a SPrL considering Project Planning and Project Monitoring and Control areas using the Scrum agile methodology together with the CMMI maturity model. Garcia et al. (Table 5, Line 35) define a SPrL for engineering service-oriented products. Brondani et al. (Table 5, Line 34) propose a quality evaluation framework, including a process to evaluate software artefacts generated from a SPrL. This approach is related to Lorenz et al. (2014) work (Table 5, Line 29), whose objective was to define a systematic approach to software process tailoring based on SPrL and data on the characteristics of projects. All of these four papers are Brazilian research works. Three are not cooperating with any other groups and none cites other existing Brazilian approaches.

The more recent papers, returned in the last search, were published in 2017 and mainly 2018, indicating recent research efforts that can evolve and generate more papers over the next years.

Teixeira et al. (2015), Costa et al. (2018), and Nogueira Teixeira et al. (2018) (Table 5, Lines 41/44/49) (two citations) are works from the same research group. Teixeira et al. (2015) proposed a checklist-based on an inspection technique (PVMCheck) to support the detection of defects on SPrL models, supporting quality assurance activities in SPrL engineering. Costa et al. (2018) focused on the Application Engineering stage of SPrL with an incremental learning approach for SPrL named Odyssey-ProcessCase. This deals with decision-making support to solve SPrL variability applying techniques such as Case-Based Reasoning (CBR) and Rule-Based System. This approach has the OdysseyProcess-Reuse approach (Nogueira Teixeira et al., 2018) as the basis for the SPrL Domain Engineering stage. These works in conjunction address both SPrL Engineering stages in the same process reuse environment, called Odyssey. A SPrL variability modelling metamodel and its notation were developed with two levels of abstraction (feature and component model). In Costa et al. (2018) (Table 5, Line 44), a Odyssey Repository was developed, which is responsible for centralizing the access to software process artifacts generated through the Odyssey-ProcessCase's systematics.

Varkoi et al. (2017) and Gallina and Iyer (2018) (Table 5, Lines 43/46) (one citation) work with Safety-oriented Process Line Engineering (SoPLE). These papers presented some evaluation in this kind of SPrL scenarios.

Delgado et al. (2018) (Table 5, Line 45) proposed an extension of the v-SPEM support for final users. An automated mechanism is provided to help during the process tailoring. The paper presents the steps needed to use the approach and the tool support provided.

4.1.1. Studies per country distribution

The distribution of research work per country can be seen in Fig. 6. Note that more than one publication per country might occur, thus totalling more than 49 studies. Brazilian and German researchers have the highest number of papers published (15 and 11, respectively),

Fig. 6. Studies Publishing Countries.

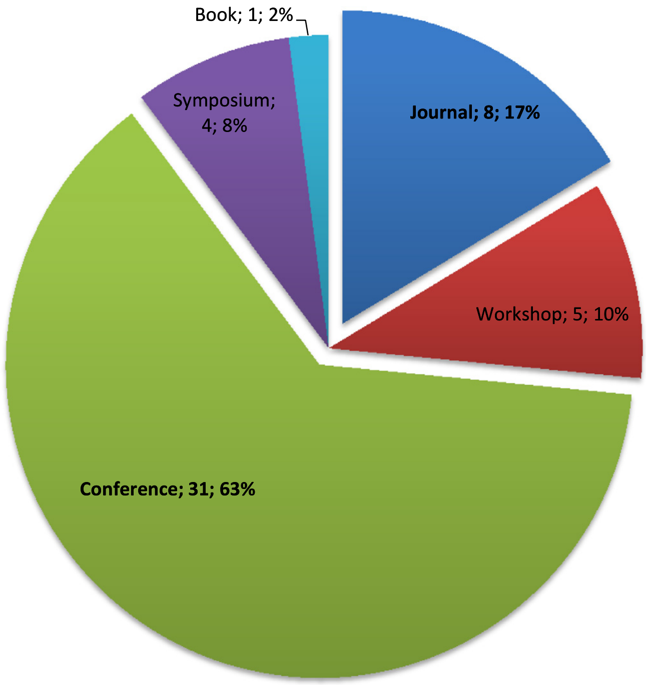


Fig. 7. Studies Publishing Venue Distribution.

followed by Chile (10) and Spain (5). From the five papers from Spanish researchers, two of them have a partnership with a German researcher, J. Münch from Fraunhofer Institute for Experimental Software Engineering (IESE), and one of them has a partnership with Uruguayan researchers. The same German researcher mentioned has also partnered with Japanese research groups. The two approaches proposed by Denmark researchers have a partnership with German researchers from two different institutions. One of the Swedish researchers has a partnership with Finnish researchers from two different institutions.

4.1.2. Studies publishing venue distribution

Most of the SPrL papers were published in conferences (31 out of 49 - 63%) as shown in Fig. 7. Only 17% of papers (8 out of 49) were published in journals (Table 5, Lines 11/17/20/24/29/36/42/47), indicating it is a maturing area and that there are few consolidated works. Five of them published in workshops (Table 5, Lines 1/2/26/39/48). Only one study was published in a book (Table 5, Line 10) and four were published in symposiums (Table 5, Lines 15/30/35/44).

4.1.3. Distribution of the main contributions

Fig. 8 shows the distribution of the main contributions. The topic of method definition was the one that received the highest number of paper contributions, totalling 35 approaches. Variability modelling is another important topic of investigation with 8 graphical notation papers and 13 papers presenting metamodels. Only four comparative studies were identified, indicating a lack of systematic empirical analysis amongst the approaches proposed.

4.1.4. SPrL engineering stages

Considering the two main stages of SPrL Engineering (Domain Engineering - DE - and Application Engineering - AE), most of the SPrL approaches selected in this systematic review provide support to the DE stage, as shown in Fig. 9, with a total of 42 publications considering this stage; 28 papers address only one of the main stages: only the DE stage was considered in 43% of research papers (21 out of 49), while only the AE stage was addressed in 14% (7 out of 49). The combined support for the two stages is provided by 43% of studies (21 out of 49 papers).

DE provides the process reuse infrastructure. This can be an explanation why it is the stage with more relevance in the approaches analysed. Without reusable process artefacts and domain representation, it is more difficult to propose support to use and apply strategies to better combine and adapt the reusable artefacts in new process definitions during AE. We also found that approaches that deal only with AE were detected from the second round of paper searching when looking at research work published from 2014 to 2019. These AE-related and more recent research works were based on the knowledge from the first publications that are more focused on the DE stage.

4.1.5. Process engineering stages

Considering the different stages in Process Engineering, i.e., definition, enactment, simulation, execution and improvement, the major contribution identified in the papers selected in the systematic review is related to the process definition activity. All the papers pointed out this activity (Fig. 10). This analysis indicates missing approaches for the subsequent activities after process definition. However, the other process engineering activities imply different challenges that can contribute to better process and product results. Fig. 10 shows how each of the process engineering activities was mentioned in the selected paper:

- 6 papers mentioned the enactment activity (Table 5, Lines 2/12/13/23/24/45);
- 5 papers mentioned execution (Table 5, Lines 12/13/14/15/35);
- 4 papers mentioned improvement (Table 5, Lines 12/28/34/46); and
- None of the research work mentions simulation.

This analysis indicates missing approaches for the subsequent activities after process definition.

Execution activities were supported in some works (Table 5, Lines 12/13/14/15/35). Three of them (Table 5, Lines 14/15/35) discuss the usage of workflow engines in the application of BPM suites for software process execution and monitoring. Model-to-model transformations (M2M) are performed to derive workflow specification language from process specification derived from a process line. The approach of Garcia et al. (2015) (Table 5, Line 35) used part of the tools specified by Aleixo et al. (Table 5, Lines 14/15). One (Table 5, Lines 14/15), which converts an Eclipse Process Framework (EPF) process to a jPDL workflow specification language using ATLAS Transformation Language (ATL) on the Eclipse platform. Another research work (Table 5, Line 35) transforms a Unified Method Architecture (UMA) model to a Business Process Modeling Notation (BPMN) version 2.0 using Extensible Style Sheet Language Transformation (XLST).

As regards the improvement activity, the research focused on improvements to process line contents as organization process institutionalization or metamodel evolutions, and quality attribute analyses. Martínez-Ruíz et al. (2009) (Table 5, Line 12) pointed some concerns during the process life cycle to achieve improvements in the institutionalization of an organization's process. A cycle of software process institutionalization is presented where after each iteration adjustment made during the process execution is analysed to discover new variations needed to generate the processes that best fit the projects and evolutions in the process lines. Kuhrmann et al. (2014) (Table 5, Line 28) identified variability operations in the context of V-Modell XT. They state that 'the concept of variability operations not only achieves the requirement to build process variants, but also serves the evolution of the metamodel which inherently appears in a long-term development'. As new processes are derived from these metamodels, improvements in new process instances can be achieved. Brondani et al. (2015) (Table 5, Line 34) presented an approach of quality evaluation for software artefacts generated from a tailored SPrL. This quality framework provides support to a quality evaluation plan definition for each artefact selected, in order to previously detect problems during process executions and final products.

Gallina and Iyer (2018) (Table 5, Line 46) deal with the topic of Safety-oriented Process Line Engineering (SoPLE). In this paper, the

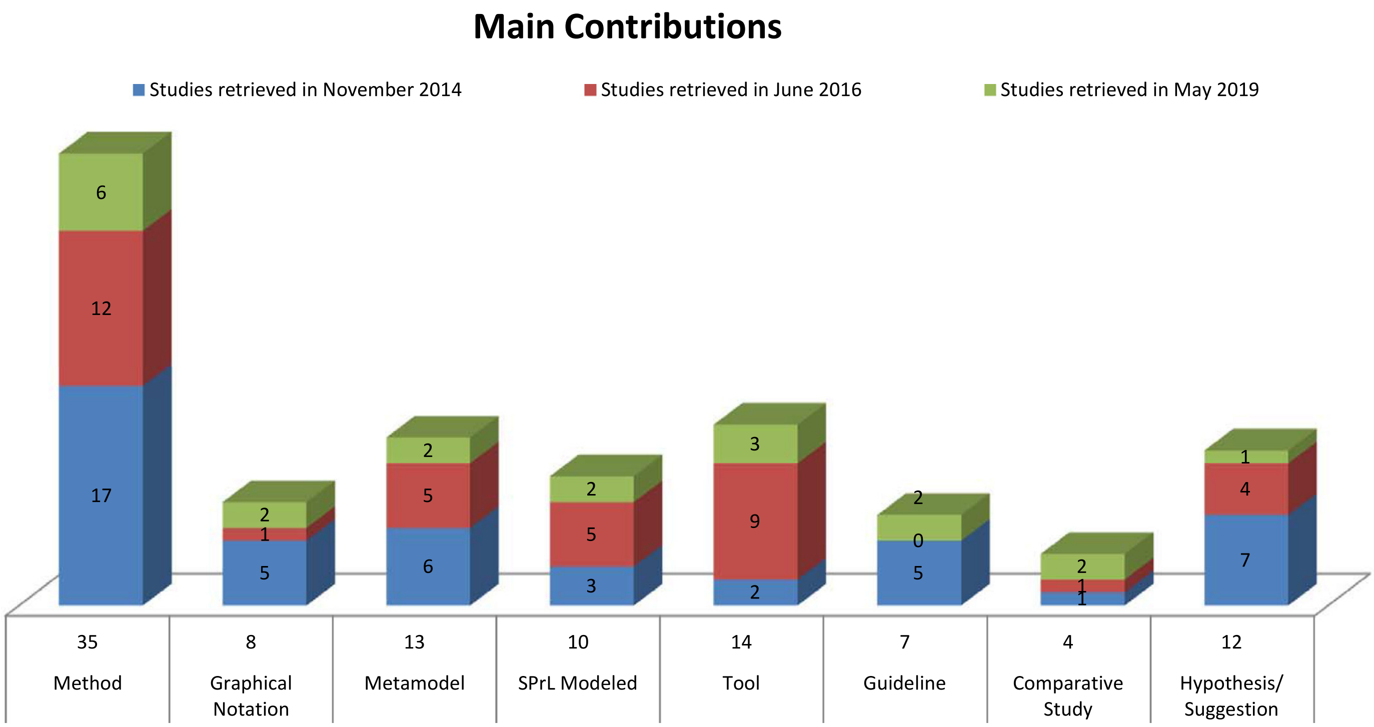


Fig. 8. Main Contributions identified.

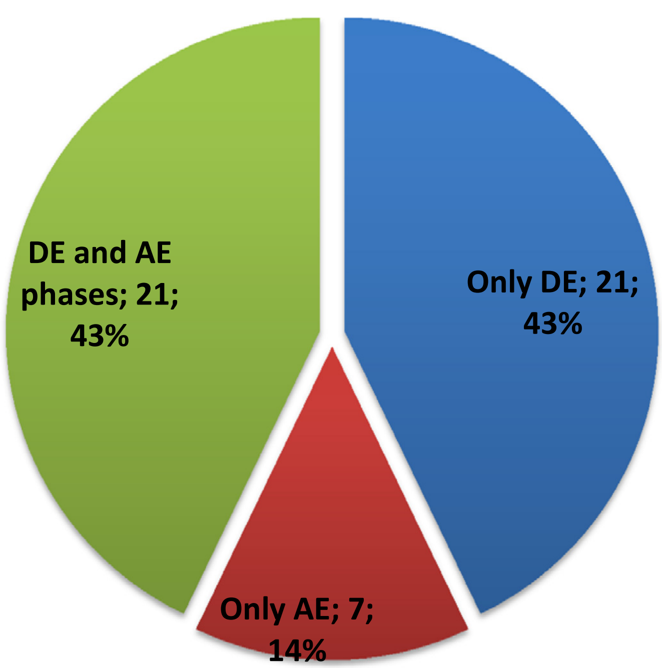


Fig. 9. Studies Support for SPrL Engineering stages (DE - Domain Engineering and AE – Application Engineering).

authors presented a measurement methodology called SoPL-targeted GQMPS model, which uses top reuse metrics. The goal is to enable the measurement of SoPLE effectiveness, which should enable process improvement.

Although not directly addressing process improvements, other approaches treat SPrL improvement through verification mechanisms as peer reviews or checklist application, as Table 5, Lines 17/41/49. This

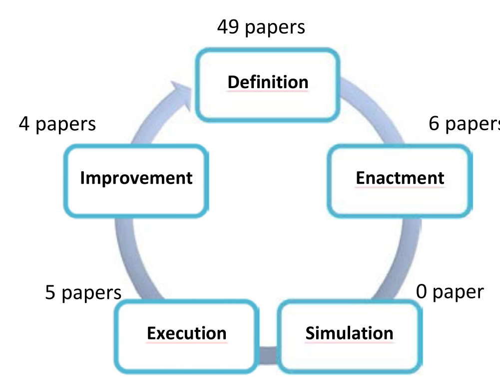


Fig. 10. Process Engineering activities and number of papers related to each stage.

can affect the level of the improvements of process definitions. Also, SPrL improvements through evolutions are briefly discussed such as: mechanism to adapt transformation rules for process derivation over time; strategies to support adjustments need collection after process execution; quality evaluation of artefacts generated during process execution; and reusability evaluation through support to add information on the level of compliance with the process and remarks after project execution.

The major contribution appears during the definition activity where project-specific processes are derived from the defined SPrLs. This includes the selection of process elements and resolutions regarding variability. During the enactment, the adequate allocation of resources can affect the goals related to cost and time constraints. Also, the process stages should be defined. The execution supported by process automation requires precise process models, represented in a relatively deep level of details by an executable modelling language. This implies technique development to define the model transformations needed to derive process representations from process models described on a high

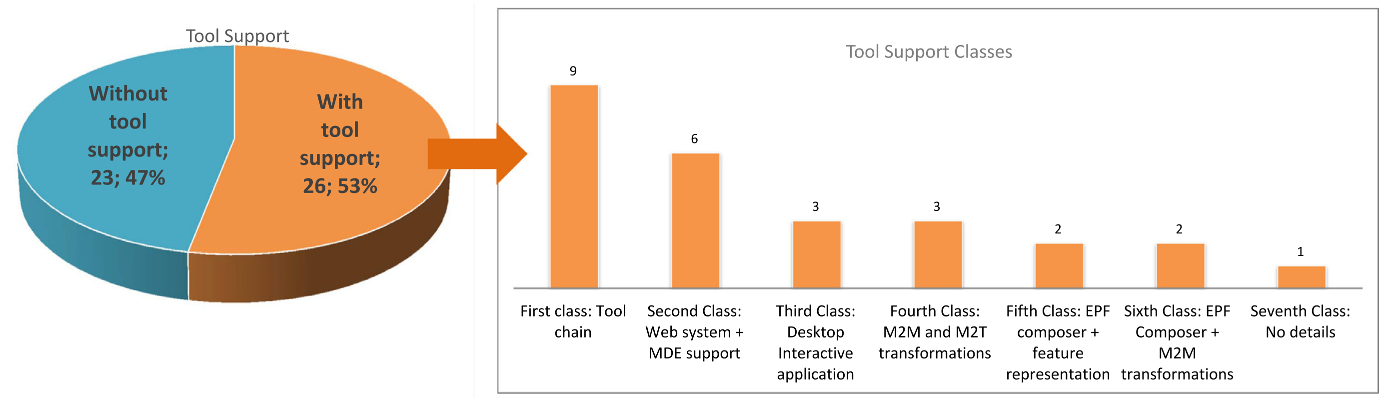


Fig. 11. Tool support for the SPrL approaches.

level of abstraction or without the specifics of execution engine languages. In addition, it is important to point process occurrences to later analyse the adjustments made and reuse this information in new process definitions. Lessons learned and process metrics should be registered and maintained in reuse repositories. Simulations can be applied to support process evolution considering the assessment of process changes prior to actual application. All these topics could be analysed in a reuse perspective and studies need to be developed to improve each one.

4.1.6. Tool support

One important aspect analysed in the studies selected was the implementation or adaptation of software tools to support the proposed SPrL approach. Fig. 11 presents the contrast of the selected studies that describe the use of some software tool to support the SPrL approach and those that do not mention any tool. The majority of the selected studies (26 out of 49 - 53%) mentioned the implementation or the adaptation of some software tool to help the approach application. On the other hand, a sizeable number of studies (23 out of 49 - 47%) does not mention the usage of a support software tool. The definition of some software tools reflects the practicality and the degree of maturity of the SPrL approach. This result brings into focus the fact that SPrL approaches are becoming more practical.

The 26 studies mentioning tool support could be distributed into seven major classes (Fig. 11):

- The first class includes 9 out of 26 studies ($\approx$35%) that presented a tool chain, where each individual tool is responsible for a step of the SPrL life cycle (Table 5, Lines14/22/26/30/36/37/40/44/49);
- The second class includes 6 out of 26 studies ($\approx$23%) presenting a tool with a Web interface with the support of model-to-model transformations (M2M) and model-to-text transformations (M2T) [37] (Table 5, Lines 1/16/27/29/32/34);
- The third class includes 3 out of 26 studies ($\approx$11%) presenting desktop interactive applications with the mission of assisting a process engineer in the generation of tailored processes (Table 5, Lines 17/28/31);
- The fourth class includes 3 out of 26 studies ($\approx$11%) presenting the definition of M2M and M2T transformations using some contextual information of a specific software project in a tailored software process (Table 5, Lines 18/24/39);
- The fifth class includes 2 out of 26 studies ($\approx$8%) presenting the use of infrastructure of the Eclipse Process Framework (EPF) combined with a process feature representation notation, such as feature model or SPLOT. In this class the processes' assets are defined in the EPF Composer and related to the process features, which are selected to tailor a specific process (Table 5, Lines 15/25);
- The sixth class includes 2 out of 26 studies ($\approx$8%), presenting the use of the EPF Composer in process definition with the support of M2M transformations to generate a tailored software process. In these studies, the tool also supports the generation of a workflow corresponding to the generated software process (Table 5, Lines 35/45); and
- The seventh class includes the last specific study (Table 5, Line 23), which mentions the use of a supporting tool but does not provide further details about it.

4.2. Proposed approaches application of reuse concept: Variability representation and component based definitions (SRQ2)

This section deals with the topics related to reuse concepts such as variability representation and process components and SPrL architecture definitions. It is important to understand how these concepts from the software product area are applied to SPrL approaches.

4.2.1. Process variability representation

All the research papers selected were analysed in order to identify process variability modelling proposals. Eight publications do not mention any notation for modelling the SPrL. The main modelling notations identified were SPEM based models (22 out of 49); feature models (18 out of 49); and UML (3 out of 49). Eighteen approaches present other SPrL representations, such as component models, V-Model XT metamodel, and their own notations, such as Little-JIL and WebAPSEE language. BPMN based notations were pointed out by only two approaches. Thirteen approaches present a combination of representation proposals.

The most cited process elements involved in variations are: activities (41 citations), tasks (33 citations), work products (29 citations) and roles (26 citations). Tools were mentioned by eleven approaches and steps only by three approaches. Each of the selected studies presented more than one type of process element involved in the variations. Additional elements cited were guidance, discipline, stages, iterations, and practices.

A total of 25 approaches provide some representation for control flow, generally using SPEM or UML (activity diagram) notations. Although in some cases control and data flows can be classified as optional, they are not treated in the approaches identified as variable elements. In general, the approaches do not focus on the behavioural aspect of processes (flows), addressing variability only in process elements, especially in work units (activities and tasks).

There should be an explicit dependency and mutually exclusive relationships of process elements, describing the composition of them that should be selected together (dependence relationship cited by 21 approaches) or do not have a meaning to be combined with (exclusive relation cited by 18 approaches). A total of 21 approaches did not mention support for representing constraints or rules for consistency checking. Contextual models were cited as a supplementary representation by 15 approaches.

Two main types of variability classification were observed: optional (32 out of 49) or variation point (34 out of 49). Optional refers to elements that may or may not be present in the process and variation point refers to elements in which several alternatives are placed.

In brief, all the representations proposed have not fully addressed the needs for process variability modelling. Some issues that are missing in current proposals are: (i) the whole set of process elements that are indicated as variables in a process family (i.e., activity, task, role, work product, tool, and relations) are generally not completely addressed in the same approach; (ii) the variation in control flows is rarely addressed; (iii) the variability of process elements is not always defined together with their optionality; and (iv) special notations to represent variability in different process perspectives were only presented in a few works.

4.2.2. Process component definitions

The topic of component based strategy for reusable software process elements was analysed to identify how the approaches define the concepts of process component and SPrL architecture. The use of architectures populated with reusable components is a technique for process reuse proposed based on the knowledge of the topics of software product reuse [4]. A process component can be considered an encapsulation of process information and behaviours at a given level of granularity [14]. These process components are organized by an architecture, a structure that establishes the process elements involved, their interfaces, ordering, interdependencies, and other relationships amongst them [15].

When analysing the SPrL approaches, only five of them mention the two concepts of SPrL Architecture and Process Component (Table 5, Lines 16/17/22/29/34/36/37/49). Beyond these five approaches, other two mention the concept of SPrL Architecture (Table 5, Lines 7/8/14/15). In Washizaki (2006a) (Table 5, Line 7), a Process Line Architecture is defined as 'a process structure that reflects the commonality and variability in a collection of processes that make up a process line from the perspective of overall optimisation'.

Magdaleno et al. (2012) (2015a) (Table 5, Lines 22/36) define process architecture as "the skeleton" a process should have, establishing the main components, their internal properties and interfaces, and how they relate to each other. In this approach, process components are "reusable items that can be connected to create new and more complex processes" (Magdaleno et al., 2015b) (Table 5, Line 37). A component organizes the common and variant parts within a specific domain that can be reused and combined, according to composition rules, to dynamically compose and adapt processes' (Magdaleno et al., 2012) (Table 5, Line22).

Barreto et al. (2011) (Table 5, Line 17) define a component as a process element with 'an internal architecture, consisting of other components that may not be completely defined, to allow some kind of customization (variability). In this approach, a SPrL is considered 'a kind of process architecture, when it is able to generate different process definitions, having abstract components or optional elements' (Barreto et al., 2010) (Barreto et al., 2011) (Table 5, Lines 16/17).

For Lorenz et al. (2014) (Table 5, Line 29), process architecture is described using optional, mandatory, concrete, or abstract components. The architecture provides rules for connecting process elements to process lines. Brondani et al., (2015) (Table 5, Line 34) apply the concepts described by Lorenz et al. (2014) (Table 5, Line 29).

Nogueira Teixeira et al. (2018) (Table 5, Line 49) define a process component as 'a process fragment abstraction based on the "black box" principle, representing a modular part of a process that encapsulates its content and communicates with its environment by interfaces'. A process domain component model is represented as a modular view of the domain knowledge, in order to improve the domain's comprehension, maintainability and reusability. The components can be defined with the use of a mapping mechanism, described as heuristics. These heuristics assist the mapping of properties from a feature model to the corresponding component model, guaranteeing consistency and traceability of variability and structural aspects.

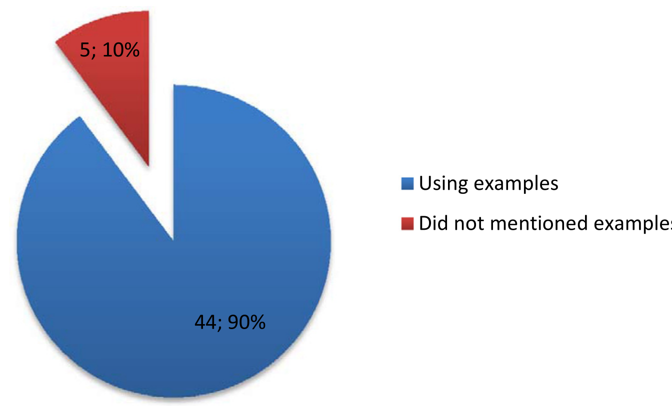


Fig. 12. Application of the proposed SPrL approaches using examples.

It is possible to conclude that although different approaches contribute to the componentization strategy for reusable process elements, they generally do not provide support for component organization in the architectures or methodologies to construct them. As stated by Barreto et al. (2011) (Table 5, Line 17) and reinforced by Magdaleno et al. (2015b) (Table 5, Line 37), "there is no general agreement on which information has to exist in a process component or which level of detail it should have. These decisions are usually based on the use intended for a component in each approach'.

4.3. Evaluation of proposed approaches (SRQ3)

This section studies the quality assessment aspects of the approaches identified by evaluation as practical examples used; the types of evaluation used for SPrL approaches; and specific criteria or metrics presented to guide the evaluation.

The results of this research question indicate that most of the publications selected, 44 out of 49 (90%) discuss the application of the proposed SPrL approaches using examples, as shown in Fig. 12. Some of these 44 studies that contextualize the SPrL approaches through practical examples are mentioned below.

These practical examples illustrate the application of SPrL approaches in a variety of scenarios, managing the variability of the processes in specific contexts such as: (i) software development processes, (ii) service-oriented software process, (iii) requirement engineering processes, (iv) hardware and software co-design processes, (v) focus on project management discipline, (vi) template-oriented processes, (vii) design-oriented processes, (viii) software or service acquisition, (ix) restrictions to the quantitative analysis of variability operators, (x) safety-oriented and nuclear domains, amongst others.

The evaluations of the SPrL approaches could be classified as: (i) examples, (ii) case study, (iii) comparison, (iv) survey, (v) expert assessment, (vi) other, and (vii) not mentioned. Fig. 13 shows the distribution of types of evaluations used in the selected studies. Some of the studies selected were classified into more than one category. Evaluations based on examples (32%) and case studies (21%) were mostly used in selected studies - illustrating the application of the approaches proposed in specific and controlled contexts. The third most used type of approach evaluation (11%) traced comparisons to evaluate two or more SPrL approaches based on some specific criteria. The least used, but not necessarily less important, were evaluations through surveys (2%) and expert assessments (2%).

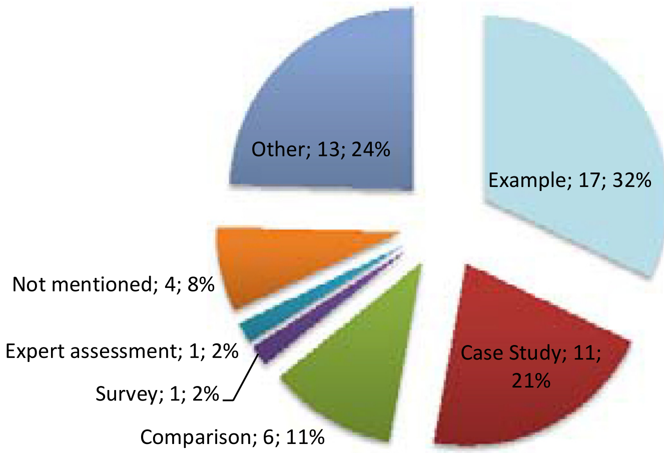


Fig. 13. distribution of the type of SPrL approach evaluation.

Other types of evaluation were cited, such as: (i) definition of a formal plan of applications and assessments (Table 5, Line 30), (ii) conducting of controlled experiments (Table 5, Lines 4/36/48), (iii) use of illustrative examples (Table 5, Lines 5/10/15), (iv) qualitative analysis with the application of questionnaires (Table 5, Line 20), (v) quantitative analysis of the use of variability operators (Table 5, Line 28/38/42), and (vi) a proof of concept through a quasi-experiment (Table 5, Line 41/49).

Only four of the studies selected (8%) do not mention any specific type of evaluation for the proposed approach. This reinforces the evidence of maturity for the SPrL approaches proposed which naturally leads to a concern regarding the assessment of the approaches to reveal the pros and cons of each one, and how they can be improved.

Another important result is that 21 out of the 49 studies selected mention and discuss their application in the industry context. They present examples or studies applied to real scenarios or using information collected from actual processes and projects.

Eight studies (Table 5, Lines 18/24/26/31/32/33/39/40) report the application of a MDE SPrL approach to small and medium-size Chilean software companies as part of Adapte Project - a large government funded project. These studies are related to the group of ten researchers from Chile - UChile, PUCV and USM - described in Section 4.1. The study related to Table 5, Line 24 reports the use of the proposed approach in two Chilean software companies – RI and Amisoft – tailoring software processes based on RUP and OpenUP, respectively. The authors mention that the results were presented and discussed with the company's employees, especially the process engineers. The studies related to Table 5, Lines 26/31/32 illustrate the application of the proposed MDE approach in a small Chilean software company, called Rhiscom. The authors present and discuss the models and transformations used. The study related to Table 5, Line 39 cites the work done again with Amisoft (small Chilean software company) with focus on the activities of the process engineer in the definition stage and in the activities of the project manager in the execution stage. The study related to Table 5, Line 33 illustrates the application of an algorithm responsible for discovering a SPrL in a small Chilean software company called Mobius. The study related to Table 5, Line 40 cites the work again done with Mobius (small Chilean software company), now with focus on a megamodel developed as the result of two years of work. This work discusses how the megamodel developed supports the modelling process, tailoring and evolution, in a scenario of integrated hardware and software solutions for a public transportation system.

Durán et al. (2004) (Table 5, Line 3) used information from a Spanish company to illustrate their approach in applying system families concepts to derive requirement engineering processes with variability.

Simidchieva et al. (2007) (Table 5, Line 9) collected information from a research collaboration with the National Mediation Board (NMB) to apply to process variation representations of processes that define how online dispute resolution is done by mediation professionals.

Armbrust et al. (2009) (Table 5, Line 11) use information collected from the Japan Aerospace Exploration Agency (JAXA) in the case study performed. Martínez-Ruíz et al. (2011) (Table 5, Line 19) also used this same JAXA Process Line to illustrate their approach.

Rouillé et al. (2012) (Table 5, Line 23) report on the application of the proposed approach in a French company called Sodifrance – to manage the variations of a simplified software development process.

Aleixo et al. (2010) (Table 5, Lines 14/15) used information processes identified in a research context of a technical educational organization development project. Barreto et al. (2011) (Table 5, Lines 16/17) applied the knowledge acquired from projects done in the Software Engineering Laboratory of an academic institution.

Magdaleno et al. (2015) (Table 5, Lines 36/37) applied their optimization approach to the context of a large oil company in Rio de Janeiro, Brazil. The focus of the case study was on the analysis of feasibility to support a process manager in the creation of the organization's process line. Part of the models applied in this study was represented using a first version of *OdysseyProcess-FEX* [27], which was evolved after further studies, as described in Nogueira Teixeira et al. (2018)(Table 5, Line 49).

Pazin et al. (2018) (Table 5, Line 47) use different excerpts extracted from the COMPETISOFT project. This project presents a reference framework for the improvement and certification of Latin American software processes based on proven solutions as four Mexican software companies.

Ruiz et al. (2018) (Table 5, Line 48) applied their study to GreenSQA, a software company located in south-west Colombia. The focus was the testing process model implemented by GreenSQA.

These specific results show some important evidence. The first element is that the common base to SPrL is mature enough to allow the proposed approaches to become more practical than just concepts and hypotheses. The second element regards the general purpose of SPrL concepts and techniques. The third one is that even with more practical examples for the approaches proposed, there is still a lack of exploration of their application in real scenarios. The reports on the SPrL application in software development companies could reveal the true strengths and drawbacks of each approach and could offer the directions to consistently evolve them.

Finally, considering the specific criteria or metrics presented to evaluate the approach, only 10 out of 49 studies had some criteria and metrics (Table 5, Lines 5/25/28/36/38/41/42/47/48/49). These specific studies perform evaluations based on quantitative metrics through the analysis of an application of the proposed approach. In three of these studies, qualitative aspects are also considered to analyse the users" impressions on the application of the approaches proposed. Table 6 shows the criteria and metrics adopted, the study that proposed them (identified by its reference in Table 5), their type (quantitative or qualitative), their main goal and how they were measured.

The evolution of the SPrL approaches proposed is directly linked to the analysis of the evaluation results and the subsequent identification of the disadvantages and strengths of each approach. With these analyses it can be defined what can be streamlined in each approach to reach better results. The definition of consistent criteria and metrics that can be reused and applied to a variety of contexts is also a key element for the streamlining of the SPrL approaches.

5. Discussion on results

This section discusses results mainly in the perspective of: domain and application engineering stages, variability management, process visualization, software process engineering reuse, industry sets, combi-

Table 6
Criteria and metrics for evaluation of the proposed SPrL approaches.

Metric/Criterion	Work ID (Table 5)	Type	Main goal	Measurement
Development effort	Lines 5/36/41	Quantitative	Estimate the effort of approach application	Necessary time
Product quality	Line 5	Qualitative	Result analysis for the SPrL approach	Quality analysis of products
Team satisfaction	Line 5	Qualitative	Estimate the satisfaction of approach usage	Survey with involved team
Expressiveness	Line 25	Qualitative	Estimate what can be done by the approach	Survey with users
Understandability	Line 25	Qualitative	Estimate the clearness of the constructs	Survey with users
Tool availability and support	Line 25	Qualitative	Estimate the amount of available tools and the access to the use instructions	Survey with users
Tool usability	Line 25	Qualitative	Estimate how easy were the use of tools	Survey with users
Interoperability with standards	Line 25	Qualitative	Estimate the interoperability with the industry patterns	Survey with users
Use of the variability operators	Lines 28/38/42	Quantitative	Practical use of the proposed variability operators	Counting each operator use
Experience level	Line 41	Qualitative	To characterize the experience of the subject responsible for SPrL inspection	Responses in a questionnaire
Number of defects	Line 41	Quantitative	To find as many defects as possible through an inspection of a SPrL models	Checklist-oriented inspection
Efficiency	Lines 41/47/49	Quantitative	1. To identify the efficiency of the checklist helping to find defects in SPrL	(Defects found / time and (Number of corrected elements/time
			2. Efficiency as the ratio between the total correctness and time spent (continuous variable	
			3. Efficiency as average time that the subject needed to create a correct component model element	
Effectiveness	Lines 41/49	Quantitative	1. To identify the effectiveness of the checklist helping to find defects in SPrL	(Defects found / existing defects and (number of correct component model elements created/ total amount of original component model elements
			2. To identify the effectiveness of the mapping heuristics	
Time spent	Line 47	Quantitative	To measure the time spent in the proposed activity	Time
Correctness	Line 47	Quantitative	To analyze the correctness of the answers and correctness of the modifications proposed	Discrete variable, ranging from 0 to 100 is totally incorrect, 0.5 is partially correct, and 10 is totally correct
Advantages and Disadvantages	Line 48	Qualitative	To identify advantages and disadvantages of scope determination in an industrial context by using two methods reported in the literature	Discussion among all participants through workshop

nation of component and SPrL approaches, and potential prospective research topics.

5.1. Domain engineering activities

To promote the reuse of process assets, it is first necessary to develop them and determine their representation. Major approaches focus on defining how to develop an SPrL and the reuse infrastructure, which means more research work in the domain engineering stage. Forty-two out of 49 publications consider this stage, as discussed in Section 4.1. However, our systematic review showed that:

- the concept of an integrated reuse infrastructure is less discussed and explored; and
- the development of integrated tools to support all reuse stages is not fully addressed (see Section 4.1).

The early activities of domain engineering can be more discussed. Domain Identification includes the definition of the scope that involves a domain feasibility study. Few works discuss the definition of the scope stage (Table 5, Lines 11/48). However, procedures and strategies should be developed to guide the investigation of reuse opportunities, considering an organization's potential and capacity to develop process reuse.

Techniques to capture and store information and knowledge within the software process domain need to be improved. This involves the development of process re-engineering techniques applied to the Organizational Base to analyse legacy processes, i.e., existing organizational process definitions or knowledge retrieved from prior projects executed.

The analysis of Domain Similarity and Variability can be supported by a semi-automatic approach to determine the points where the domain processes are similar (mandatory elements) and points where they diverge (optional and alternative elements), which represent adaptation points during specific process derivation.

5.2. Application engineering activities

The Application Engineering stage can be explored with improvements or the adaptation of techniques applied to software reuse. This stage started to be discussed back in 2014 (Section 4.1). During the software process definition, decision support, simulation and optimization mechanisms can be provided as means for the resolution to support variability.

The existing approaches do not discuss how to identify and deal with the impacts of decisions made during this stage. Concepts of context-aware process lines can be improved to determine how to develop mechanisms to explicitly represent contextual information and describe how it affects process variability decisions. Contextual models were cited as supplementary representation by 15 approaches. In addition, mechanisms to validate composition rules and domain restrictions and check consistency should be specified.

Adaptations to select reusable artefacts should be considered and some support needs to be provided in order to assist this activity.

5.3. Variability management

The representation and management of variability was one of the most frequently commented topics (see Section 4.2). Different SPrL modelling languages were proposed based on the process modelling field and software variability representations. Therefore, as no process language has been adopted in a practical sense yet and process modelling standardization is still a hurdle to overcome, the representation for process line variability has room for more improvements and efforts to reach consensus on a set of topics that should be treated and represented in this modelling. However, instead of proposing new representations, studies should be conducted to understand the barriers of their use in practice and to invest in improvements to existing proposals. Training and guidelines for usage can help in a first attempt in practical application, considering that some concepts are not familiar to process engineers.

Furthermore, as the process variability modelling can be represented with different proposed notations, in which one might be chosen considering the higher adequacy to the modelling requirements, greater knowledge of the process team, popularity, amongst other factors, where the flexibility within the variability modelling activity can be a topic of research. This can be achieved by the possibility of transitioning amongst the proposals for modelling of variability and technical support to transform the final process defined into a specific process modelling language with a broad use, such as BPMN or SPEM. This may be a first step towards understanding the needs of representation and mapping for the existing representations, which can culminate with the unification of the proposals for process variability and a consensus for the supporting process in tailoring representations.

5.4. Process visualization

Process visualization can also be explored to support the understanding of the process line and the communication of process alternatives to be derived from the line. Studies in the software visualization and computer-human interaction areas could be made to support the development of an infrastructure to better support reuse-related tasks, considering each stakeholder involved and one's specific needs.

5.5. Software process engineering reuse

Endeavors should be made on how reuse can improve the support for different activities of software process engineering. Relationships amongst SPrL and the execution of software processes, and their simulation and improvement should be explored (see Section 4.1). Only three papers (two approaches, actually) mention some support of using workflow engines of BPM suites and model-to-model transformations. The automatic transformation of the software process defined to an execution specification supported by a process engine can be streamlined considering these few existing proposals and other works focused on the process execution supported. Software process technology and process-centred environments can be studied in order to identify integration opportunities with software process reuse tools and to take advantage of each specific support.

Another topic is the guarantee of compliance between process definition and process execution via the monitoring and controlling of its results. This information can be considered as feedback for domain specification and registered as reusable information. In addition, if deviations are detected, appropriate actions can be taken considering the replacement amongst other process components that are more appropriate according to the domain information analysis. At this point, the topic of Dynamic SPL can be explored to analyse the possibility of the dynamic adaptation of the processes. The specification of SPrL covering widely adopted process models, standards or references can help to improve the capability and maturity of the processes.

5.6. Industry set

Industry application was pointed by a small number of approaches in the sample studied. Only 21 of the selected 49 studies mention and discuss its application in an industry context (see Section 4.3).

This can indicate a lack of proposals applied to real software organizations scenarios. It is expected that a proposal for practical application demonstrates usage results, evidences its main issues and leads to an alignment between the current and future needs of industry and academics interests in the area. In addition, industry application provides real and more complex examples to be used by the approaches, boosting their potential and scalability for real scenarios.

5.7. Combining component and SPrL approaches

It would be interesting to explore the relations amongst the studies and the possibility of blending different techniques for the reuse of software processes.

Few existing approaches try to combine techniques as SPrL and process component. Only five of the approaches mention the concepts of a SPrL Architecture and Process Component (see Section 4.2). Therefore, there is room for further research and inspiration in the field of software reuse where these approach combinations have already taken place.

5.8. Potential prospective topics

Topics not frequently discussed by the SPrL approaches are related to: (i) software process repository development and management; (ii) techniques to calculate the costs associated to the adaptation of reusable artefacts; (iii) strategies to promote reuse skills combining concepts of software reuse strategies and software process knowledge; and (iv) techniques to guarantee reusable artefacts quality, including techniques for the verification and validation of reusable artefacts developed and the final processes derived from the line.

6. Threats to validity and SLR limitations

An important topic is the coverage of the primary studies identified. The treatment of the search string definition and the use of different engines were considered to maximize the number of primary studies collected. The search string was defined using a PICO structure and combined a set of keywords and their synonyms. It was reviewed by the researchers involved and refined using a set of control publications that were returned during the string application in the different digital libraries. However, the number of engines used (only three) and the lack of access to some papers could affect in the completeness of the identification of the primary studies. In future works other databases and engines can be used to possibly increase the results, such as ScienceDirect, SpringerLink, and Wiley InterScience. Also, other terms can be applied in the search string to enlarge the SLR scope and consider related areas which can contribute with process reuse research, such as terms and synonyms common in Method Engineering area as 'method families', 'method variants' and 'methodology engineering'.

In order to avoid the exclusion of relevant papers during the selection stage, a systematic procedure was defined including the definition of SLR main goal and specific questions. The inclusion and exclusion criteria were also defined. Three researchers were involved to reach consensus on the divergences and different interpretations regarding paper suitability for the goal of the SLR.

The reliability of the analysis of the data extraction categories and results is directly related to the correct interpretation of the research works involved in the execution of the SLR analysis. The lack of information of some papers and the existence of conflicting and overlapping concepts drive the need of interpretation by the researchers what could have introduced some bias or an erroneous result description. In order to avoid this, the data extraction, classification and result summarizing were conducted by different researchers and reviewed by others.

Despite the systematic review process used, it is possible to get different conclusion results if the study is replicated by other researchers, as some steps involve individual interpretation, perspective, and experience.

7. Conclusion

Software Process Line (SPrL) is an emergent approach that aims to promote software process reuse in a systematic way, towards the same benefits already supported by software product line engineering. This includes: (1) the definition of a process family or process domain concept, (2) the definition of variability in processes, its representation and management procedures, and (3) the adaptation of the domain engineering and the application engineering development stages.

Beyond these elements, the hurdles already faced by the implementation of a software reuse programme, as related to organizational, economic, political and psychological issues should also be addressed by the area of software process reuse.

The phases and procedures adopted in this systematic review were described in detail in order to enable its replication. The timely replication of this study is extremely encouraged. Obtaining historical series of results may evidence evolution path of the research field, besides revealing trends of standardization.

The presented results allowed to identify existing approaches for developing, using, managing and seeing the evolution of SPrLs and to characterize their support, especially during the development of reusable process family artefacts, including an overview of existing SPrL supporting tools in their multiple stages; to analyse variability management and component-based aspects in SPrL; and to identify practical examples and evaluations applied. The results also allowed to perceive some potential prospective topics that can be addressed in future research.

One main challenge for future research in SPrL is how to apply the SPrL benefits to the software development industry, besides which would be the safest way to conduct this process. Only in the software industry scenario, the benefits of each approach can be proofed effective or ineffective. The systematic review results indicated that only a few approaches are reaching this maturity level. The results of applying these approaches in software industry real contexts will bring new directions for research work in SPrL.

Finally, during the systematic review execution it was noticed that many works facing the customization of business process models were rejected because they were out of the intended scope. Among the rejected works there were those that applied the software product line engineering in the context of business process models. Future works may explore the intersections between these two different contexts. Considering business process models execution in BPM engines, it can be investigated if there are potential benefits of executing software processes derived from SPrLs in the same engines.

Declaration of Competing Interest

The authors declare no conflict of interest.

Acknowledgment

This research was partially funded by CNPq, CAPES and FACEPE. CAPES and INES 2.0 - FACEPE grant APQ-0399-1.03/17, CNPq grant 465614/2014-0 and CNPq grant 304282/2018-9.

Appendix A. Extraction Form

The information extraction template is presented in Table A1.

Table A1
Extraction items.

Extraction form
Reference identification
Title
Author(s)
Year
Institution/Country
Type of publishing vehicle: Journal/Workshop/Conference/Symposium
Type of Contribution
Method
Graphical Notation /Metamodel/Process Line Modeled/Variability requirements
Tool
Guideline
Comparative Study
Hypothesis or suggestion
Phases of SPrL Engineering supported by the approach
Domain Engineering (development for reuse)
Project Engineering (development *by*reuse)
Phases of the software process lifecycle activity supported by the approach
Process Definition Process Simulation Process Enactment Process Execution Process Improvement
Software process engine is used to support process execution? Yes /No
Which engine?
Execution data is collected? **Yes/**No
Some specific representation is used? State diagram; BPMN; Petri net; Other
SPrL improvement/evolution is supported? Yes/No
SPrL verification and/or validation is supported? Yes/No
Type of supporting tool provided by the approach
Process Variability
SPrL Representation:
Modeling Language: Feature Model SPEM or Derivation of SPEM UML (Activity Diagram) BPMN or Derivation of BPMN Not mentioned **Other**
Process Variability Management:

Appendix B. Selected Primary Studies

Tables B1 and B2 present the final set of selected primary studies.

Table B1
Final set of selected studies - Part 1.

ID	Publication
1	Hollenbach, C. R. 1996. Experiences in process domain engineering at PRC Inc. *In Proceedings of the 10th International Software Process Workshop, Process Support of Software Product Lines. IEEE Computer Society* (1996), 78–79.
2	Sutton, S. M. and Osterweil, L. J. 1996. Product families and process families. *In Proceedings of the 10th International Software Process Workshop, Process Support of Software Product Lines. IEEE Computer Society* (1996), 109–111.
3	Durán, A., Benavides, D., and Bermejo, J. 2004. Applying system families concepts to requirements engineering process definition. In *5th International Workshop on Software ProductFamily Engineering (PFE 2003). Springer Berlin Heidelberg Revised Papers* (2004), 140–151.
4	Jaufman, O. 2005. Emergent process design. In *Proceedings of the 27th International Conference on Software Engineering (ICSE 2005). IEEE* (2005), 653-653.
5	Jaufman, O. and Münch, J. 2005. Acquisition of a projectspecific process. In *6th International Conference Product Focused Software Process Improvement (PROFES 2005). Springer Berlin Heidelberg* (2005), 328–342.
6	Rombach, D. 2006. Integrated software process and product lines. In *Unifying the Software Process Spectrum: International Software Process Workshop (SPW 2005). Springer Berlin Heidelberg Revised Selected Papers* 3840 (2006), 83–90.
7	Washizaki, H. 2006. Building software process line architectures from bottom up. In *Proceedings of 7th International Conference on Product Focused Software Process Improvement (PROFES 2006). Springer Berlin Heidelberg* (2006), 415–421.
8	Washizaki, H. 2006. Deriving projectspecific processes from process line architecture with commonality and variability. In *4th IEEE International Conference on Industrial Informatics (INDIN06). IEEE* (2006), 1301–1306.
9	Simidchieva, B. I., Clarke, L. A., and Osterweil, L. J. 2007. Representing process variation with a process family. In *Software Process Dynamics and Agility: International Conference on Software Process (ICSP 2007). Springer Berlin Heidelberg* (2007), 109–120.
10	Thränert, M. and Werner, A. 2007. A Process Family Approach for the reuse of development processes. In *Innovations and Advanced Techniques in Computer and Information Sciences and Engineering. Springer Netherlands* (2007), 309–313.
11	Armbrust, O., Katahira, M., Miyamoto, Y., Münch, J., Nakao, H., and Ocampo, A. 2009. Scoping software process lines. In *Software Process: Improvement and Practice*, 14 (3) (2009), 181–197.
12	Martínez-Ruiz, T., García, F., and Piattini, M. 2009. Process Institutionalization using Software Process Lines. In *International Conference on Enterprise Information Systems (ICEIS)*, 3 (2009), 359–362.
13	Ternité, T. 2009. Process lines: a product line approach designed for process model development. In *Proceedings of the 35th Euromicro Conference on Software Engineering and Advanced Applications (SEAA'09). IEEE Computer Society* (2009), 173–180.
14	Aleixo, F. A., Freire, M. A., dos Santos, W. C., and Kulesza, U. 2010. A Model-driven Approach to Managing and Customizing Software Process Variabilities. In *International Conference on Enterprise Information Systems (ICEIS)*, 3 (2010), 92–100.
15	Aleixo, F. A., Freire, M. A., dos Santos, W. C., and Kulesza, U. 2010. An Approach to Manage and Customize Variability in Software Processes. In *Proceedings of the 2010 Brazilian Symposium on Software Engineering (SBES 2010). IEEE Computer Society* (2010), 118–127.
16	Barreto, A., Duarte, E., Rocha, A. R., and Murta, L. 2010. Supporting the Definition of Software Processes at Consulting Organizations via Software Process Lines. In *Proceedings of Seventh International Conference on the Quality of Information and Communications Technology (QUATIC). IEEE* (2010), 15–24.
17	Barreto, A., Murta, L. G. P., and da Rocha, A. R. C. 2011. Software Process Definition: a Reuse-based Approach. In *Journal of Universal Computer Science (J. UCS)*, 17 (13) (2011), 1765–1799.
18	Hurtado Alegría, J. A., Bastarrica, M. C., Quispe, A., and Ochoa, S. F. 2011. An MDE approach to software process tailoring. In *Proceedings of the 2011 International Conference on Software and Systems Process. ACM (2011)*, 43–52.
19	Martínez-Ruíz, T., García, F., Piattini, M., and Munch, J. 2011. Applying AOSE concepts to model crosscutting variability in variant-rich processes. In *37th EUROMICRO Conference on Software Engineering and Advanced Applications (SEAA 2011). IEEE Computer Society* (2011), 334–338.
20	Martínez-Ruiz, T., García, F., Piattini, M., and Münch, J. 2011. Modelling software process variability: an empirical study. *IET Software*, 5, 2 (2011), 172–187.
21	Jafarinezhad, O., and Ramsin, R. 2012. Development of Situational Requirements Engineering Processes: A Process Factory Approach. In *IEEE 36th Annual Computer Software and Applications Conference (COMPSAC). IEEE Computer Society* (2012), 279–288.
22	Magdaleno, A. M., de Araujo, R. M., and Werner, C. 2012. COMPOOTIM: An Approach to Software Processes Composition and Optimization. In *Congresso Ibero-Americano em Engenharia de Software (CIbSE 2012)* (2012), 42–55.
23	Rouillé, E., Combemale, B., Barais, O., Touzet, D., and Jézéquel, J. M. 2012. Leveraging CVL to manage variability in software process lines. In *19th Asia-Pacific Software Engineering Conference (APSEC). IEEE* (2012), 148–157.
24	Hurtado, J. A., Bastarrica, M. C., Ochoa, S. F., and Simmonds, J. 2013. MDE software process lines in small companies. *Journal of Systems and Software*, 86, 5 (2013), 1153–1171.
25	Simmonds, J., Bastarrica, M. C., Silvestre, L., and Quispe, A. 2013. Variability in software process models: Requirements for adoption in industrial settings. In *Proceedings of 4th International Workshop on Product Line Approaches in Software Engineering (PLEASE). IEEE* (2013), 33–36.

Table B2
Final set of selected studies - Part 2.

ID	Publication
26	Bastarrica, M. C., Simmonds, J., and Silvestre, L. 2014. Using megamodeling to improve industrial adoption of complex MDE solutions. In *Proceedings of the 6th International Workshop on Modeling in Software Engineering (MiSE). ACM* (2014), 31–36.
27	de Carvalho, D. D., Chagas, L. F., AND Reis, C. A. L. 2014. Definition of Software Process Lines for Integration of Scrum and CMMI. In *XL Latin American Computing Conference (CLEI). IEEE* (2014), 1–12.
28	Kuhrmann, M., Fernández, D. M., and Ternité, T. 2014. Realizing software process lines: Insights and experiences. In *Proceedings of the 2014 International Conference on Software and System Process (ICSSP). ACM* (2014), 99–108.
29	Lorenz, W. G., Brasil, M. B., Fontoura, L. M., and Vaz Pereira, G. 2014. Activity-Based Software Process Lines Tailoring. In *International Journal of Software Engineering and Knowledge Engineering*, 24(09), 1357–1381.
30	Silvestre, L. 2014. A Domain Specific Transformation Language to Support the Interactive Definition of Model Transformation Rules. In *Doctoral Symposium at 17th International Conference on Model Driven Engineering Languages and Systems, (MoDELS2014), CEUR-WS, vol. 1321* (2014).

(*continued on next page*)

Table B2 (*continued*)

ID	Publication
31	Silvestre, L., Bastarrica, M. C., and Ochoa, S. F. 2014. A model-based tool for generating software process model tailoring transformations. In *2nd International Conference on Model-Driven Engineering and Software Development (MODELSWARD). IEEE* (2014), 533–540.
32	Silvestre, L., Bastarrica, M. C., and Ochoa, S. F. 2014. Reducing Complexity of Process Tailoring Transformations Generation. In *2nd International Conference on Model-Driven Engineering and Software Development. Revised Selected Papers, Springer International Publishing, vol. 506* (2015), 171–182.
33	Blum, F. R., Simmonds, J., and Bastarrica, M. C. 2015. Software process line discovery. In *Proceedings of the 2015 International Conference on Software and System Process (ICSSP). ACM* (2015), 127–136.
34	Brondani, C., Bertuol, G., and Fontoura, L. M. 2015. Quality Evaluation of Artifacts in Tailored Software Process Lines" In: *Proceedings of the International Conference on Software Engineering and Knowledge Engineering* (SEKE). (2015), 223–226. DOI: 10.18293/SEKE2015-052
35	Garcia, C., Malucelli, A., Paludo, M., and Reinehr, S. 2015. A software process line for service-oriented applications. In *Proceedings of the 30th Annual ACM Symposium on Applied Computing (ACM/SIGAPP). ACM* (2015), 1680–1687.
36	Magdaleno, A. M., de Oliveira Barros, M., Werner, C. M. L., de Araujo, R. M., and Batista, C. F. A. 2015. Collaboration optimization in software process composition. In *Journal of Systems and Software*, 103 (C), 452–466.
37	Magdaleno, A.M., Araujo, R.M., Werner, C.M.L., and Batista, C.F.A. 2015. A Practical Experience of a Software Process Line Creation In *XVIII Ibero-American Conference on Software Engineering (CIbSE)*. (2015), 95–108.
38	Schramm, J., Dohrmann, P., and Kuhrmann, M. 2015. Development of flexible software process lines with variability operations: a longitudinal case study. In *Proceedings of the 19th International Conference on Evaluation and Assessment in Software Engineering. ACM* (2015), 13. DOI: 10.1145/2745802.2745814
39	Silvestre, L., Bastarrica, M.C., and Ochoa, S.F. 2015. A Usable MDE-based Tool for Software Process Tailoring. In *Posters and Demos at 18th International Conference on Model Driven Engineering Languages and Systems* (MoDELS2015), CEUR-WS, vol. 1554 (2015), 36–39.
40	Simmonds, J., Perovich, D., Bastarrica, M. C., and Silvestre, L. 2015. A megamodel for software process line modeling and evolution. In *18th International Conference on Model Driven Engineering Languages and Systems (MODELS). ACM/IEEE* (2015), 406–415.
41	Teixeira, E. N., de Mello, R. M., Motta, R. C., Werner, C. M. L., Vasconcelos, A. 2015. Verification of Software Process Line Models: A Checklist-based Inspection Approach. In *XVIII Ibero-American Conference on Software Engineering (CIBSE)*. (2015), 81–91.
42	Kuhrmann, M., Méndez Fernndez, D., and Ternité, T. 2016. On the use of variability operations in the V-Modell XT software process line. In *Journal of Software: Evolution and Process*, 28(4), 241–253. DOI: 10.1002/smr.1751.
43	Varkoi T., Mäkinen T., Gallina B., Cameron F., Nevalainen R. (2017) Towards Systematic Compliance Evaluation Using Safety-Oriented Process Lines and Evidence Mapping. In: *Stolfa J., Stolfa S., O'Connor R., Messnarz R. (eds) Systems, Software and Services Process Improvement* (EuroSPI 2017). Communications in Computer and Information Science, vol 748, pp 83–95. Springer, Cham
44	Costa, D. M., Teixeira, E. N., & Werner, C. M. (2018, October). Odyssey-processcase: a case-based software process line approach. In *Proceedings of the 17th Brazilian Symposium on Software Quality*, Curitiba, Brazil (2018), 170–179. ACM. DOI: 10.1145/3275245.3275263
45	Delgado, A., Calegari, D., & García, F. (2018) Modeling of software process families with automated generation of variants. In: *The 30th International Conference on Software Engineering and Knowledge Engineering* (2018), DOI: 10.18293/SEKE2018-019
46	Gallina B., Iyer S. (2018) Towards Quantitative Evaluation of Reuse Within Safety-Oriented Process Lines. In: *Larrucea X., Santamaria I., O'Connor R., Messnarz R. (eds) Systems, Software and Services Process Improvement* (EuroSPI 2018). Communications in Computer and Information Science, vol 896, pp. 469–479. Springer, Cham
47	Pazin, M. G., Allian, A. P., & OliveiraJr, E. (2018). Empirical study on software process variability modelling with SMartySPEM and vSPEM. In: *IET Software*, 12(6), 536–546. DOI: 10.1049/iet-sen.2017.0061
48	Ruiz, P. H., Camacho, C., & Hurtado, J. A. (2018, November). A Comparative Study for Scoping a Software Process Line. In:*2018 ICAI Workshops* (ICAIW), Bogota, Colombia (pp. 1–6). IEEE. DOI: 10.1109/ICAIW.2018.8554998
49	Nogueira Teixeira, E., Vasconcelos, A., & Werner, C. (2018). OdysseyProcessReuse-A Component-based Software Process Line Approach. In: *Proceedings of the 20th International Conference on Enterprise Information Systems* (ICEIS 2018) - (2) (pp. 231–238). DOI: 10.5220/0006672902310238

References

[1] J. Mnch, O. Armbrust, M. Kowalczyk, M. Soto, Software Process Definition and Management, Springer Publishing Company, Incorporated, 2012.

[2] J.A. Hurtado Alegría, M.C. Bastarrica, A. Quispe, S.F. Ochoa, An mde approach to software process tailoring, in: Proceedings of the 2011 International Conference on Software and Systems Process, in: ICSSP '11, ACM, New York, NY, USA, 2011, pp. 43–52, doi:10.1145/1987875.1987885.

[3] M. Ginsberg, L. Quinn, Process tailoring and the software capability maturity Model, Technical Report. CMU/SEI-94-TR-024, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, 1995.

[4] M.I. Kellner, Connecting reusable software process elements and components, in: Proceedings 10th International Software Process Workshop, 1996, pp. 8–11, doi:10.1109/ISPW.1996.654356.

[5] P.C. Clements, L. Northrop, Software Product Lines: Practices and Patterns, SEI Series in Software Engineering, Addison-Wesley, 2001.

[6] L.M. Northrop, Sei'S software product line tenets, IEEE Softw. 19 (4) (2002) 32–40, doi:10.1109/MS.2002.1020285.

[7] H. Washizaki, Building software process line architectures from bottom up, in: J. Münch, M. Vierimaa (Eds.), Product-Focused Software Process Improvement, Springer Berlin Heidelberg, Berlin, Heidelberg, 2006, pp. 415–421.

[8] D. Rombach, Integrated software process and product lines, in: M. Li, B. Boehm, L.J. Osterweil (Eds.), Unifying the Software Process Spectrum, Springer Berlin Heidelberg, Berlin, Heidelberg, 2006, pp. 83–90.

[9] O. Armbrust, M. Katahira, Y. Miyamoto, J. Münch, H. Nakao, A. Ocampo, Scoping software process lines, Softw. Process 14 (3) (2009) 181–197, doi:10.1002/spip.v14:3.

[10] E. OliveiraJr, M.G. Pazin, I.M.S. Gimenes, U. Kulesza, F.A. Aleixo, SMartySPEM: a SPEM-based approach for variability management in software process lines, in: J. Heidrich, M. Oivo, A. Jedlitschka, M.T. Baldassarre (Eds.), Product-Focused Software Process Improvement, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013, pp. 169–183.

[11] T. Martinez-Ruiz, F. Garcia, M. Piattini, J. Munch, Modelling software process variability: an empirical study, IET Softw. 5 (2) (2011) 172–187, doi:10.1049/iet-sen.2010.0020.

[12] J. Simmonds, M.C. Bastarrica, L. Silvestre, A. Quispe, Variability in software process models: requirements for adoption in industrial settings, in: 2013 4th International Workshop on Product LinE Approaches in Software Engineering (PLEASE), 2013, pp. 33–36, doi:10.1109/PLEASE.2013.6608661.

[13] J. Sametinger, Software Engineering with Reusable Components, Springer-Verlag, Berlin, Heidelberg, 1997.

[14] K.A. Gary, T.E. Lindquist, Cooperating process components, in: Proceedings. Twenty-Third Annual International Computer Software and Applications Conference (Cat. No.99CB37032), 1999, pp. 218–223, doi:10.1109/CMPSAC.1999.812704.

[15] M. Chrissis, M. Konrad, S. Shrum, CMMI: Guidelines for Process Integration and Product Improvement, SEI series in software engineering, Addison-Wesley, 2003.

[16] A.S. Barreto, L.G.P. Murta, A.R.C. da Rocha, Software process definition: a reuse-based approach, J. Univers. Comput. Sci. 17 (13) (2011) 1765–1799.

[17] O. Pedreira, M. Piattini, M.R. Luaces, N.R. Brisaboa, A systematic review of software process tailoring, SIGSOFT Softw. Eng. Notes 32 (3) (2007) 1–6, doi:10.1145/1241572.1241584.

[18] T. Martínez-Ruiz, J. Münch, F. García, M. Piattini, Requirements and constructors for tailoring software processes: a systematic literature review, Softw. Qual. J. 20 (1) (2012) 229–260, doi:10.1007/s11219-011-9147-6.

[19] R. dos Santos Rocha, M. Fantinato, The use of software product lines for business process management: a systematic literature review, Inf. Softw. Technol. 55 (8) (2013) 1355–1373, doi:10.1016/j.infsof.2013.02.007.

[20] D.D. de Carvalho, L.F. Chagas, A.M. Lima, C.A.L. Reis, Software process lines: a systematic literature review, in: A. Mitasiunas, T. Rout, R.V. O'Connor, A. Dorling (Eds.), Software Process Improvement and Capability Determination, Springer International Publishing, Cham, 2014, pp. 118–130.

[21] D.M. Costa, E.N. Teixeira, C.M.L. Werner, Software process definition using process lines: asystematic literature review, in: Latin American Computing Conference, 2018, pp. 1–10.

[22] P. Brereton, B.A. Kitchenham, D. Budgen, M. Turner, M. Khalil, Lessons from applying the systematic literature review process within the software engineering domain, J. Syst. Softw. 80 (4) (2007) 571–583, doi:10.1016/j.jss.2006.07.009.

[23] B. Kitchenham, Procedures for Performing Systematic Reviews, Technical Report TR/SE-0401, Keele University, Department of Computer Science, Keele University, UK, 2004.

[24] B. Kitchenham, R. Pretorius, D. Budgen, O. Pearl Brereton, M. Turner, M. Niazi, S. Linkman, Systematic literature reviews in software engineering - a tertiary study, Inf. Softw. Technol. 52 (8) (2010) 792–805, doi:10.1016/j.infsof.2010.03.006.
[25] E. Hernandes, A. Zamboni, S. Fabbri, A.A.D. Thommazo, Using GQM and TAM to evaluate start - a tool that supports systematic review, CLEI Electron. J. 15 (2012). 3–3
[26] J.W. Dias, E. OliveiraJr, Modeling variability in software process with EPFcomposer and SMartySPEM: an empirical qualitative study, in: Proceedings of the 18th International Conference on Enterprise Information Systems, in: ICEIS 2016, SCITEPRESS - Science and Technology Publications, Lda, Portugal, 2016, pp. 283–293, doi:10.5220/0005771502830293.
[27] E. Teixeira, A component-based software process line engineering with variability management in multiple perspectives, in: 18th International Software Product Line Conference Doctoral Symposium, 2014, pp. 1–10.
[28] Softex, Association for promoting the brazilian software excellence - Mps.br - brazilian software process improvement, general guide: 2016 [online], Available: https://softex.br/mpsbr/guias/ - ISBN 978-85-99334-84-3,Â 2016.
[29] M. Pai, M. McCulloch, J.D. Gorman, N. Pai, W.T.A. Enanoria, G.E. Kennedy, P. Tharyan, J.M. Colford, Systematic reviews and meta-analyses: an illustrated, step-by-step guide., Natl. Med. J. India 17 2 (2004) 86–95.
[30] E. National Academies of Sciences, Medicine, Open Science by Design: Realizing a Vision for 21st Century Research, The National Academies Press, Washington, DC, 2018, doi:10.17226/25116.
[31] FOSTER, Foster open science, 2019. URL https://www.fosteropenscience.eu.
[32] O. Armbrust, M. Katahira, Y. Miyamoto, J. Münch, H. Nakao, A. Ocampo, Scoping software process models - initial concepts and experience from defining space standards, in: Q. Wang, D. Pfahl, D.M. Raffo (Eds.), Making Globally Distributed Software Development a Success Story, Springer Berlin Heidelberg, Berlin, Heidelberg, 2008, pp. 160–172.
[33] T. Martínez-Ruiz, F. García, M. Piattini, F. De Lucas-Consuegra, Process variability management in global software development: a case study, in: Proceedings of the 2013 International Conference on Software and System Process, in: ICSSP 2013, ACM, New York, NY, USA, 2013, pp. 46–55, doi:10.1145/2486046.2486056.
[34] C. Hollenbach, W. Frakes, Software process reuse in an industrial setting, in: Proceedings of Fourth IEEE International Conference on Software Reuse, 1996, pp. 22–30, doi:10.1109/ICSR.1996.496110.
[35] J.W. Dias, E. OliveiraJr, M.A.G. Silva, Preliminary empirical evidence on SPrL variability management with EPF and SMartySPEM, in: Proceedings of the 30th Brazilian Symposium on Software Engineering, in: SBES '16, ACM, New York, NY, USA, 2016, pp. 133–142, doi:10.1145/2973839.2973850.
[36] M.G. Pazin, A.P. Allian, E. OliveiraJr, Empirical study on software process variability modelling with SMartySPEM and vSPEM, IET Softw. 12 (6) (2018) 536–546, doi:10.1049/iet-sen.2017.0061.
[37] M. Brambilla, J. Cabot, M. Wimmer, Model-Driven Software Engineering in Practice, 1st, Morgan & Claypool Publishers, 2012.