The Journal of Systems & Software 186 (2022) 111189

Contents lists available at ScienceDirect

The Journal of Systems & Software

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

Software product line scoping: A systematic literature review[☆]

Luciano Marchezan[a,*], Elder Rodrigues[b], Wesley Klewerton Guez Assunção[a,c], Maicon Bernardino[b], Fábio Paulo Basso[b], João Carbonell[b]

[a] Institute for Software Systems Engineering, Johannes Kepler University, Linz, Austria
[b] Laboratory of Empirical Studies in Software Engineering, Federal University of Pampa, Alegrete, Brazil
[c] Opus Research Group – Informatics Department, Pontifical Catholic University of Rio de Janeiro, Rio de Janeiro, Brazil

ARTICLE INFO

Article history:
Received 28 March 2021
Received in revised form 20 September 2021
Accepted 8 December 2021
Available online 21 December 2021

Keywords:
Software product lines
Software product line scoping
Systematic literature review
Snowballing

ABSTRACT

Software product line (SPL) scoping aids companies to define the boundaries of their resources such as products, domains, and assets, the target of reuse tasks scoping technical and organizational aspects. As scoping guides the management of the resources in SPL development, it becomes one of the core activities in this process. We can find in the literature several approaches on this topic, proposing techniques and methodologies to be applicable in different organizational scenarios. However, no work comprehensively reviews such approaches and describes the advances in state of the art in the last years. In this context, we look into identifying, analyzing, and extracting detailed characteristics from SPL scoping proposals found in the literature. These characteristics allowed us to compare these approaches, reason about their applicability, and identify existing limitations and research opportunities. Thus, we conducted a systematic literature review alongside snowballing, following a well-defined protocol to retrieve, classify and extract information from the literature. We analyzed a total of 58 studies, identifying 41 different approaches in the field, highlighting their similarities and differences, and establishing a generic scoping process. Furthermore, we discuss research opportunities in the SPL scoping field.

1. Introduction

Software Product Line (SPL) is a well-known approach for systematically creating reusable software assets and customized software products for a specific domain or market segment (Pohl et al., 2005; Apel et al., 2013). To achieve this goal, SPL engineering (SPLE) employs a process composed of two phases, namely domain engineering and application engineering (van Der Linden et al., 2007). Domain engineering focuses on higher-level aspects considering the whole scenario in which the SPL will be developed and used, while application engineering generates concrete artifacts by using composition and merging techniques (Kang et al., 1998).

Defining the boundaries of an SPL is one of the core planning activities performed during domain engineering, which is known as *SPL scoping* (Schmid, 2000; de Moraes et al., 2009). John and Eisenbarth (2009) define scoping as "the planning to apply SPL-based techniques considering business alignment and goals". Thus, SPL scoping may deal with both technical and organizational aspects, depending on the level on which they are addressed. Accordingly to Schmid (2000), these levels of scoping can be: product, domain and asset.

SPL scoping aids companies to define appropriate bounds regarding the domain its products will serve, which may determine the success or failure of the whole product line effort (deBaud and Schmid, 1999a). Scoping has an impact on the overall SPL development since it guides the management of the resources in a way that promises a high return on investment (Balbino et al., 2011). Although not deeply discussed in technical SPL contributions, activities to support the definition of development budgets and cost estimations are essential. SPLE requires a high upfront investment but has proven practical benefits in the long term (van Der Linden et al., 2007). This high upfront investment can hinder the wide adoption of SPLs. However, a proper SPL scoping can provide companies a clear view related to these long-term advantages from a technical perspective, such as systematic reuse, and also from the business perspective, as for example an optimal return on investment in terms of time to market (Schmid, 2002a). In this sense, SPL scoping may aid companies to better understand how and when to start developing SPLs instead of relying on single-product development or *ad-hoc* software reuse (Assunção et al., 2017).

[☆] Editor: W.K. Chan.

[*] Correspondence to: Johannes Kepler University Linz, Computer Science SP3 Room SP3 240 - Altenberger Strasse 69 4040, Linz, Austria.

E-mail addresses: lucianomarchp@gmail.com (L. Marchezan), elderrodrigues@unipampa.edu.br (E. Rodrigues), wesleyklewerton@gmail.com (W.K.G. Assunção), bernardino@acm.org (M. Bernardino), fabiopbasso@gmail.com (F.P. Basso), joaocarbonellpc@gmail.com (J. Carbonell).

https://doi.org/10.1016/j.jss.2021.111189
0164-1212/

As stated, there are many reasons for performing SPL scoping during the development of SPLs. Thus, activities associated with SPL scoping processes should be discussed in detail. For example, it may be difficult for companies with flexible/agile organizational contexts to adopt a rigid SPL scoping approach. Hence, discussing how an approach handles differences among companies by allowing the adaptation/customization of its activities, artifacts, roles, and mechanisms can ease the choosing and application of SPL scoping approaches (Mærsk-Møller and Jørgensen, 2010a). Although there are some pieces of work analyzing and comparing SPL scoping approaches (John and Eisenbarth, 2009; Schmid, 2000), they are non-systematic studies. Only one study found in the literature reports a systematic review on the topic (de Moraes et al., 2009), which analyzed a small set of 13 studies. Despite the important contribution of these pieces of work, there is a need for a systematic study on this topic to provide an in-depth review of existing practical and theoretical contributions. In addition, existing studies have become out of date due to the increasing number of publications in this particular field (Garcés et al., 2017; Mendes et al., 2019).

To fill the gap regarding the need for a comprehensive systematic study covering the existing body of knowledge on SPL scoping approaches, in this work we present a Systematic Literature Review (SLR). An SLR is a secondary study aiming at investigating a particular research topic or area (Kitchenham et al., 2010). Based on the analysis of 58 studies, the results of this SLR on SPL scoping are used to answer our research questions related to (i) similarities and differences among existing approaches, (ii) how these approaches were evaluated, and (iii) the research gaps still open and opportunities for future studies.

From the primary sources, we identified 41 different approaches related to SPL scoping. By analyzing the characteristics of such approaches, we establish a scoping concept map categorizing the main similarities and differences among them. Furthermore, we defined a generic SPL scoping process with common activities identified. We also identified 26 case studies used for evaluating the proposed approaches. To motivate new studies, we also identified research opportunities that may be further investigated in the future, such as the proposition of new metrics, flexibilization of existing mechanisms, development of tool support, investigating scoping during the re-engineering of legacy systems, and reducing the effort of SPL scoping activities.

The main contributions of the study reported in this SLR are:

i A comprehensive analysis of the current literature on the topic of SPL scoping. Our work encompasses more than 10 years of advance in this research topic with 42 additional papers in relation to previous mappings and surveys on this topic.

ii For practitioners our work contributes by providing a scoping concept map and a generic scoping process. Both are defined based on common activities found on existing approaches, to guide those companies envisaging the adoption or migration towards SPLs.

iii This SLR supports researchers in understanding the current body of knowledge on SPL scoping in terms of existing approaches and their evaluations, as well as the scoping concepts extracted from them. Furthermore, we describe identified open gaps, challenges, and research opportunities to conduct new studies.

The remainder of this work is structured as follows: Section 2 presents the main concepts and definitions related to our SLR; Section 3 describes the design and execution of the SLR; Section 4 reports the results and answers the research questions; Section 5 describes the related work; Section 6 presents our conclusions and future works with regard to the SLR results.

2. Background

In this section, we present the definition of the main concepts used in our SLR.

2.1. Software product lines

SPL is as "a set of software-intensive systems that share a common, managed set of features satisfying the specific needs of a particular market segment or mission" (Clements and Northrop, 2002). Some of the main benefits of implementing an SPL are the long-term cost reduction, significant quality improvement of the products, and reduction of the time-to-market (Pohl et al., 2005). As it happens with regular software products, SPL also follows a development process (Kang et al., 2002). The field dedicated to the study of this development is Software Product Line Engineering (SPLE). SPLE relies on platforms and mass customization concepts (Davis, 1989) to enable variability management and systematic reuse. Different technologies are used in SPLE as facilitators, making the SPL creation process easier. From these technologies, we can highlight the component-based architecture, software project patterns, and the object-oriented paradigm (van Der Linden et al., 2007).

A development model for SPLE was proposed by Pohl et al. (2005), composed of domain engineering and application engineering. The life-cycle of domain engineering focuses on the definition of the SPL scope and common assets generation, from requirements to testing, that compose the SPL platform. During this life-cycle, as illustrated in Fig. 1, product management is performed, with the goal of defining common and variable features for the products, generating a product roadmap. Then, the domain requirements engineering phase is used to elicit, document, validate and verify requirements, generating a set of common variables and an SPL variability model. In the third phase, domain design, the requirements are used alongside the variability model to defined the SPL architecture. The next two phases, domain realization, and domain testing represent the implementation and test of domain assets respectively. During application engineering, in the bottom part of Fig. 1, the common and variable assets developed in the domain engineering are combined with specific assets to create an SPL product. The phases composing this life-cycle are similar to those composing domain engineering, with the exception of product management. In each phase, however, the artifacts are produced for each individual product, rather than for the SPL (the goal of the domain engineering life cycle).

Furthermore, Krüger et al. (2020) defined a round-trip engineering process with activities such as analyze domain, scope platform, and analyze commonalities and variabilities that resembles similar goals with the domain engineering life-cycle from Pohl et al. (2005). These goals drive the direction on which SPL scoping follows.

2.2. SPL scoping

As defined by Clements (2002), SPL scope is "the set of products in, or envisioned to be in, the family". Hence, the scoping of SPL aims at defining a scope and generating a product map, that specify the common and variable features of the products (Kang et al., 2002). Scoping the SPL is deciding which products will be "in" and which will be "out" of the family (Clements, 2002). SPL scoping is important to companies as their plans for SPL development should be performed carefully (Balbino et al., 2011). Additionally, deBaud and Schmid (1999a) stated that the initial decisions concerning to which scope should be included in the SPL are fundamentally important.

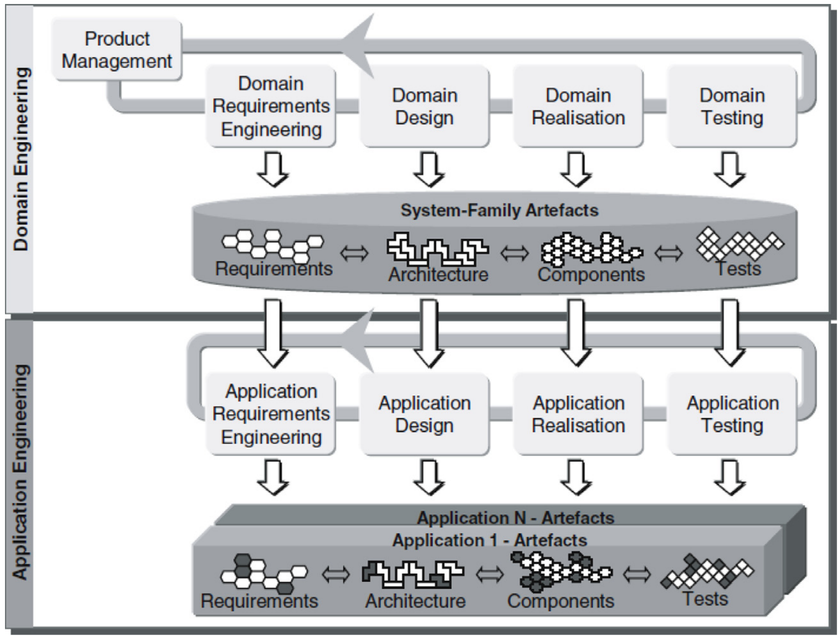

Fig. 1. The two-life-cycle model of SPLE from Pohl et al. (2005).

This importance is given as the scope of the SPL should be optional considering the organization's scenario. The SPL scoping process aims at reducing uncertainty related to these decisions (John and Eisenbarth, 2009).

Furthermore, SPL scoping intends to identify and limit the features and products from a target domain or market segment. The scope of the SPL must be not only of the right size but also targeting the correct products, based on the information gathered related to competitors, market trends, business goals, among others (Clements, 2002). In this sense, we can see the importance that such an activity has in SPLE as well as its potential impact on the organization, as it may directly involve the organization's goals and investments. Given its importance to companies, we can understand why this topic resulted in many approaches being proposed and evaluated over the years (Schmid, 2000; Kang et al., 2002; de Moraes et al., 2009). Despite this, SPL scoping mighty not have a clear and formalized representation (Schmid, 2002a; Clements, 2002). In contrast with the SPLE (Clements and Northrop, 2002), scoping has many approaches, mostly frameworks, that can be adapted and extended to attend to the organization's goals (de Moraes et al., 2009).

These approaches, however, may follow different directions, depending on the goals of the organizations where they were used, or the scenario where they were evaluated. Thus, SPL scoping approaches do not follow or are restricted to a classification of scoping techniques and activities. The most well-known and adopted classification is presented by Schmid (2000), which defines three different types of scoping, namely: (i) *product*, *domain*, and *asset*. Product scoping, or product portfolio scoping, defines a "description of products in a product line" by enumerating the individual products and individual requirements relevant to them (Schmid, 2000). When scoping products, group/team interactions for gathering requirements may be recommended. Domain scoping is a more conceptual definition. It identifies the domains and sub-domains where the SPL may be part. Identifying the domain scope can be described as the process of "identify the information required by using a few standard information sources" (Schmid, 2000). Lastly, asset scoping aims and identifying which assets from the products may have reuse potential for the SPL. This may involve SPL reengineering techniques (Assunção et al., 2017) or a "built from scratch" generic structure (Clements, 2002). As mentioned, however, these three types of scoping might not be always identified in proposed approaches in the field. This lack of clear classification and definition among the approaches can increase the effort of companies when deciding on which approach might be most suited to them (Krüger et al., 2020).

These definitions of SPL, SPL scoping, and the three types of scoping are important as they were used during the classification part of our review to identify and group potential SPL scoping techniques/activities from the primary studies analyzed. Further details about the classification are discussed next.

3. SLR design and execution

An SLR protocol must be well planned and executed to obtain the desired results. We adopted the protocol proposed by Kitchenham et al. (2009). The first step is to define the goal and RQs of our SLR. The main goal of our study is:

> ***Goal:*** *Review the literature on SPL scoping for the purpose of identifying similarities and differences among existing approaches and processes, conceptual characteristics, and research opportunities.*

To achieve this goal, we derive our Research Questions (RQs):

RQ.1. What are the similarities and differences among the approaches? In this question, we investigate the details of the processes followed by each approach. To compare the similarities and differences, we rely on thematic synthesis (Cruzes and Dybå, 2010; Cruzes and Dyba, 2011). To answer this question, we collected and coded the process phases and activities, types of scoping, scoping techniques, and how the approaches are open for adaptation. For reasoning about the similarities and differences, we applied the Formal Concept Analysis technique (Tonella, 2004) to derive the relationship among concepts/codes that was further refined into a concept map (Kane and Trochim, 2007) to represent a visual organization of the concepts related to scoping approaches.

RQ.2. How are existing scoping approaches evaluated? To reason about the applicability of scoping approaches, we investigate the empirical methods used in the studies and their evolution through the years. By investigating how approaches are evaluated, we can have insights of its maturity level and the contexts where the approaches have been applied. More specifically, we classified existing evaluation into case studies, experiments, quasi experiment, survey, proof of concepts, and manual (ad-hoc) comparison (Wohlin et al., 2012; Runeson et al., 2012). In addition to the empirical method described by the authors, we analyzed the protocol and results to confirm if described evaluations follow the definitions and principles reported in this literature (Wohlin et al., 2012; Runeson et al., 2012; Fink, 2003).

RQ.3. What are the open research gaps and opportunities for new studies on the topic of SPL scoping? The objective of this research question is to shed light on new pieces of research on SPL scoping. We considered the findings of our SLR and the observed gaps in the primary sources to compose a list of directions for new studies. As part of the thematic analysis based on the reading, data collection, and coding, we also focus on identifying research gaps and opportunities. This identification took into account limitations and future directions described in the papers, as well as opportunities observed when comparing existing approaches. The concepts (codes) (Cruzes and Dybå, 2010) related to these research gaps and opportunities were grouped in order to answer this RQ.

For the search of primary sources, we defined the relevant terms, and their synonyms, related to the goal and RQs. These terms and synonyms are used to establish our search string. To define this string, we relied on the population, intervention, comparison, outcomes, and context (PICOC) method (Wohlin et al., 2012), as described next:

- **Population**: SPL scoping studies.
- **Intervention**: processes, approaches, techniques, strategies, or frameworks.
- **Comparison**: activities performed, scoping types covered, similarities and differences, approach's adaptation for different scenarios, evaluations applied, and future work.
- **Outcome**: a set of SPL scoping approaches, an overview of the approaches describing their activities, types of scoping, similarities and differences among them, adaptation strategies, evaluation methodologies, gaps, and opportunities in the field.
- **Context**: both academic and industrial context. Considering the industrial context, we aim at opportunities in the field.

Based on the PICOC information, we extracted terms and synonyms to construct the base search string, using *OR* and *AND* operators, as presented in Table 1. Table 2 presents the six digital libraries (DLs) used to search for primary sources and the specific search string for each DL. These specific search strings were defined after executing, analyzing, and refining the base search string until they returned a satisfactory result. Additionally, when allowed by the DL, we included keywords and filters in the strings which are related to the IC/EC, such as to search only papers published in English. These special keywords can be seem at the search strings for Engineering Village, Scopus and We of Science. Finally, for the screening of the studies returned from each DL, according to the protocol of our SLR, we defined one inclusion criterion (IC) and four exclusion criteria (EC) to select only relevant papers, as presented in Table 3.

Table 1
Terms, synonyms and the search string.

Terms	Synonyms
Scope	scoping
Software product line	product line, software family, software product family, software reuse, SPL
Approach	method, methodology, process, technique
Search String	Scope OR Scoping AND Software product line OR product line OR software family OR software product family OR software reuse OR SPL AND approach OR method OR methodology OR process OR technique

The next step in the SLR protocol is to evaluate the quality of the selected studies. In this way, we defined a set of quality assessment questions (QAs) that were used to qualify and classify the studies. We assigned a score for each question, and after answering them, we computed the final score for each study. The possible answers for each question are: "*total*", when the study has information to totally answer the QA, then the study would score 1 point for that question; "*partial*", when we can use the study information only to partially answer the QA, the score, in this case, is 0.5; and "*none*", when the study does not provide any information to answer the question, the score is equal to 0. The QAs and their respective score rules are described next.

Quality Assessment Questions:

QA1. Does the study describe the activities and artifacts regarding SPL scoping?

Total : the study details its activities and artifacts presenting examples of use;

Partial : the study only presents a list of brief description of its activities and artifacts;

None : neither activities nor artifacts are presented.

QA2. Does the study present some kind of process/work-flow?

Total : the study presents a well-documented process containing roles, artifacts and a work-flow among its activities;

Partial : a work-flow for performing the activities is presented;

None : no work-flow is defined.

QA3. Does the study present an evaluation and its results?

Total : the study applied and reported an empirical evaluation discussing its results;

Partial : the study applied and reported a non-empirical evaluation discussing its results;

None : no evaluation is applied or discussed.

Another artifact of an SLR is the data extraction form. The content retrieved when applying this extraction into the studies is crucial for achieving our goals. Our data extraction (DE) form had the following fields: **DE1.** Title; **DE2.** Author; **DE3.** Year of publication; **DE4.** Study goals; **DE5.** Study summary; **DE6.** Scoping Activities; **DE7.** Scope types; **DE8.** Scoping concepts; **DE9.** Adaptation strategy; **DE10.** Proposal evaluation methodology; **DE11.** Study limitations; **DE12.** Research opportunities identified/proposed by the authors. For answering QA3 as well as extracting the information for DE10, we used the evaluation methodologies definitions presented in Table 4. Thus, when extracting this information from the studies, we used the references and definitions from Table 4.

Table 2
Digital libraries and search strings.

Digital Library	Search String
ACM	(Scope Scoping) AND ("Software product line" "product line" "software family" "software product family" "software reuse" SPL) AND (approach method methodology process technique)
Eng. Village	(((Scope OR Scoping) WN KY) AND (("Software product line" OR "product line" OR "software family" OR "software product family" OR "software reuse" OR SPL) WN KY) AND ((approach OR method OR methodology OR process OR technique) WN KY) AND (english WN LA))
IEEE Xplore	(Scope OR Scoping) AND ("Software product line" OR "product line" OR "software family" OR "software product family" OR "software reuse" OR SPL) AND (approach OR method OR methodology OR process OR technique)
Science Direct	(Scope OR Scoping) AND ("Software product line" OR "product line" OR "software family" OR "software product family" OR "software reuse" OR SPL) AND (approach OR method OR methodology OR process OR technique)
Scopus	TITLE-ABS-KEY (Scope OR Scoping) AND ("Software product line" OR "product line" OR "software family" OR "software product family" OR "software reuse" OR SPL) AND (approach OR method OR methodology OR process OR technique) AND (LIMIT TO (SUBJAREA, "COMP"))
Web of Science	(TS=((Scope OR Scoping) AND ("Software product line" OR "product line" OR "software family" OR "software product family" OR "software reuse" OR SPL) AND (approach OR method OR methodology OR process OR technique))) AND LANGUAGE:(English)

Table 3
Inclusion/exclusion criteria.

Inclusion Criteria	Exclusion Criteria
IC1. The primary study must present an SPL scoping process, approach, technique, strategy or framework.	EC1. Studies written in languages other than English.
	EC2. Studies not available online. EC3. Studies without any evaluation of the approach. EC4. Studies with less than 6 pages.

Table 4
Evaluations type.

Id	Evaluation	Description
CS	Case Study	"...an empirical enquiry that draws on multiple sources of evidence to investigate one instance of a contemporary software engineering phenomenon within its real life context..." (Runeson et al., 2012)
Ex	Experiment	"...an empirical enquiry that manipulates one factor or variable of the studied setting. Based in randomization, different treatments are applied to or by different subjects, while keeping other variables constant, and measuring the effects on outcome variables." (Wohlin et al., 2012)
MC	Manual Comparison	Compare the approach with related work manually, describing and comparing main differences, advantages and drawbacks.
PoC	Proof-of-Concept	Produce a conceptual implementation of the approach to demonstrates its use.
QE	Quasi-experiment	"...similar to an experiment, where the assignment of treatments to subjects cannot be based on randomization but emerges from the characteristics of the subjects or objects themselves." (Wohlin et al., 2012)
Su	Survey	"...is a system for collecting information from or about people to describe, compare or explain their knowledge, attitudes and behavior." (Fink, 2003)

Fig. 2 presents the number of studies retrieved from each digital library. After performing the search in the digital libraries,[1] a total of 932 studies were retrieved. From these studies, 289 were duplicated. Thus, when the total of studies that remained was 643. As a result of the application of the I/E Criteria, 36 studies remained. To improve our search protocol, we also applied the snowballing technique (Wohlin, 2014). The first seed used for the snowballing were the three related work surveys discussed in Section 5. We first looked at the primary papers discussed in these surveys and applied our I/E criteria to them. Then, we looked into papers citing these three surveys (forward snowballing). After this, we used the 36 primary studies to start a new snowballing phase where we looked into papers citing these 36 studies. The two phases of the snowballing resulted in the inclusion of additional 22 studies. Thus, the final set of studies analyzed to answer our questions was composed of 58 primary sources. The repository of this SLR is available at Marchezan et al. (2021).

An overview of the studies included, presenting their reference, year of publication, venue, and inclusion source (digital library or snowballing) is presented in Table 5.

4. Results and RQ answers

In this section, we present and discuss the results obtained after the reading, collecting data, and analyzing 58 primary sources. These results are the basis to answer the posed RQs.

4.1. RQ1. What are the similarities and differences among the approaches?

Among the 58 primary sources, we found 41 approaches. Table 6 lists all these approaches sorted by year of their first appearance. In this table we also present their names, indicate in which studies they were present, and describe an overview

[1] Search was carried out on June 2021.

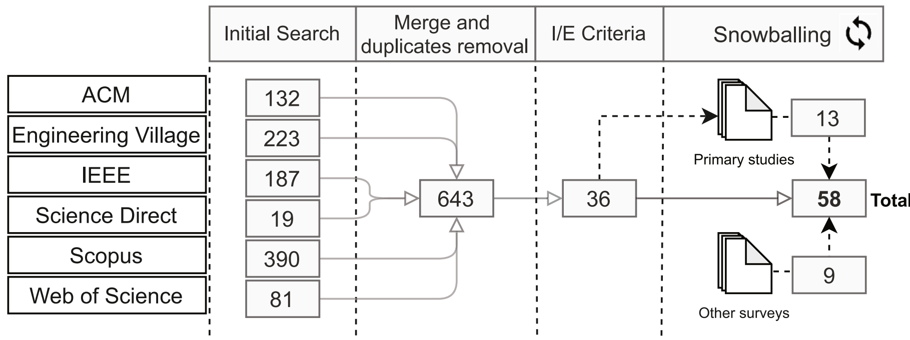

Fig. 2. Numbers of the SLR on Scoping.

about them. PuLSE is the approach with most references, being introduced by Bayer et al. (2000), deBaud and Schmid (1999) in 1999. The approaches most recently introduced CoMeS (Ojeda et al., 2018) in 2018 and Small-SPL (Ojeda et al., 2019) in 2019.

With the intent of deriving a conceptual map and also a generic process, in the following, we analyzed each of these 41 approaches. They were analyzed chronologically to better understand their evolution to accomplish specific scoping issues.

4.1.1. Approach similarities and differences

For an in-depth investigation, we analyzed similarities and differences among the 41 found approaches. More specifically, to collect evidence related to this question, we followed the process illustrated in Fig. 3. This process was defined based on the principles of thematic synthesis (Cruzes and Dybå, 2010; Cruzes and Dyba, 2011). These principles include the identification of specific segments of text found in the primary studies analyzed, which is the Step 1 of our process (Fig. 3). In our case, we analyzed the segments of text describing the activities and techniques related to scoping found in each approach. Such segments are used as a basis for the discussion presented in the following sections. These segments were then labeled using *codes* used for supporting the grouping of scoping activities and techniques that are recurrent in the literature on the field. This is the second step of our process depicted in Fig. 3. During Step 3, we removed some concepts that only appeared in less than 10% of the approaches. These two concepts are risk management and tool support, and they were considered as gaps and research opportunities in the field, discussed when answering RQ3 (see Section 4.3). Then, based on identified scoping activities and techniques of each study, we created groups that describe concepts recurrent in the approaches.

Table 7 describes the activities identified in each approach.[2] We grouped the activities and techniques identified considering their scope types, generating a list of concepts based on definitions from the literature as well as the use of primary sources, as presented in Table 8. After this (Step 5 of the process depicted in Fig. 3), we traced the generated concepts from the primary studies, as presented in Table 9.

With this tracing information, we were able to analyze the variability of the concepts using formal concept analysis and thus generating a concept lattice,[3] which is presented in Fig. 4. This lattice shows how each approach relates to the mapped concepts, and as well as showing which concepts are more common and how both the concepts and the approaches relate to each other. For instance, the approaches from Cruz et al. (2013) and Karimpour and Ruhe (2016) are grouped together as they contain the same concepts, namely cost models, customer needs, and metrics definition. Furthermore, approaches such as RiPLE (bottom left) and Small-SPL (top right) are very far apart in the lattice, as they do not share similar concepts. Fig. 4 can also be used to analyze how the SPL scoping concepts are related to each other. In this sense, we can observe that concepts such as market analysis, candidate analysis, cost models, and customer needs, are linked to a similar set of approaches. This indicates that these concepts are related and can complement each other when used together. Lastly (Step 6 in Fig. 3), we generate the scoping concept map, by classifying and grouping the concepts from the concept lattice, as presented in Fig. 5 and discussed next.

The research and practice in the area of SPL scoping can benefit from a model that, both provides means for decision support and that constraint selection by a well-characterized set of features found in these 41 approaches. The Feature Model is a well-accepted representation for such a need (Seidl et al., 2016), as illustrated by Fig. 5, which describes in a conceptual map the context in which each approach can be inserted. We derived this conceptual map to guide software engineers in the selection process for an approach, as well as to characterize the studies. Our conceptual map is composed of three scoping types, namely asset, domain, and product. For each scoping type, we have or-alternative concepts. Firstly, for the asset scoping type, there are the architecture definition and scoping metamodel concepts. For domain scoping, we have an abstract concept called cost–benefit analysis, which has three or-alternative concepts, cost models, customer needs, and metrics definition. Still, for domain scoping, there is a market analysis and product map. Lastly, for product scoping, we have candidate analysis, evolution planning, and prioritize products. As mentioned earlier, the definition of these concepts is described in Table 8.

Table 9 presents the traceability between each concept from the concept map (depicted in Fig. 5) to the 41 approaches identified in this SLR. Recall that this result was obtained from the results presented in Tables 7 and 8. As shown in Table 9, when considering asset scoping, architecture definition is being used in 17 approaches (41%), while scoping metamodeling is only considered in six approaches (14%). For domain scoping, cost models are established and used for cost–benefit analysis in ten of the approaches (24%), customer needs are considered in 18 (44%), and metrics are defined and used in 11 approaches (27%). Market analysis is performed in 17 approaches (41%), and a product map is used in nine approaches (22%). For product scoping, candidates are analyzed in 13 approaches (32%), eight approaches (19%) have specific activities or artifacts for evolution planning and, 11 approaches (27%) consider prioritize products as part of them. Considering this traceability, an important contrast is presented

2 Although the PuLSE framework is composed of different stages, for this identification of activities we only considered the scoping process within PuLSE, called PuLSE-Eco. More specifically, we extracted the activities from its 2.0 version, presented in Schmid (2002a).

3 The concept lattice was generated using the concept explorer tool http://conexp.sourceforge.net/.

Table 5
Overview of the publications included.

Ref	Year	Title (Venue)	Inclusion by
Bayer et al. (2000)	1999	Customizable Domain Analysis (GCSE)	Digital Library
Bayer et al. (1999)	1999	PuLSE: A methodology to develop software product lines (SSR)	Snowballing
deBaud and Schmid (1999)	1999	A systematic approach to derive the scope of software product lines (ICSE)	Digital Library
Bayer et al. (2000)	2000	PuLSE-I: Deriving instances from a product line infrastructure (ECBS)	Snowballing
Knauber et al. (2000)	2000	Applying Product Line Concepts in Small and Medium-Sized Companies (IEEE Software)	Digital Library
Schmid and Gacek (2000)	2000	Implementation Issues in Product Line Scoping (ICSR)	Digital Library
Kishi et al. (2002)	2002	A Method for Product Line Scoping Based on a Decision-Making Framework (SPLC)	Digital Library
Schmid (2002a)	2002	A Comprehensive Product Line Scoping Approach and Its Validation (ICSE)	Digital Library
Schmid (2002b)	2002	The Product Line Mapping Approach to Defining and Structuring Product Portfolios (ICRE)	Snowballing
Kang et al. (2002)	2002	Using a Marketing and Product Plan as a Key Driver for Product Line Asset (SPLC)	Snowballing
Geppert and Weiss (2003)	2003	Goal-Oriented Assessment of Product-Line Domains (METRICS)	Snowballing.
Kim et al. (2003)	2003	A Domain Analysis Method for Software Product Lines Based on Scenarios, Goals and Features (APSEC)	Snowballing.
Schmid et al. (2005)	2005	Introducing the PuLSE approach to an embedded system population at Testo AG (ICSE)	Snowballing
Shin Young Park and Soo Dong Kim (2005)	2005	A systematic method for scoping core assets in product line engineering (APSEC)	Digital Library
Kim et al. (2005)	2005	Traceability Map: Foundations to Automate for Product Line Engineering (SERA)	Snowballing
Ramachandran and Allen (2005)	2005	Commonality and variability analysis in industrial practice for product line improvement (SPIP)	Digital Library
John et al. (2006)	2006	A practical guide to product line scoping (SPLC)	Digital Library
Her et al. (2007)	2007	A framework for evaluating reusability of core asset in product line engineering (IST)	Snowballing
Noor et al. (2007)	2007	A Collaborative Approach for Product Line Scoping: A Case Study in Collaboration Engineering (SE)	Digital Library
Noor et al. (2008a)	2008	A Collaborative Method for Reuse Potential Assessment in Reengineering-Based Product Line Adoption (CEE-SET)	Digital Library
Noor et al. (2008b)	2008	Agile product line planning: A collaborative approach and a case study (JSS)	Snowballing
Kim et al. (2008)	2008	DRAMA: A framework for domain requirements analysis and modeling architectures in software product lines (JSS)	Snowballing
Carbon et al. (2008)	2008	Providing Feedback from Application to Family Engineering - The Product Line Planning Game at the Testo AG (SPLC)	Snowballing
Estublier et al. (2010)	2010	Software Product Line Evolution: The Selecta System (ICSE)	Digital Library
Jhon (2010)	2010	Using Documentation for Product Line Scoping (IEEE Software)	Digital Library
Ullah et al. (2010)	2010	Decision support for moving from a single product to a product portfolio in evolving software systems (JSS)	Snowballing
Elsner et al. (2010)	2010	Multi-level product line customization (SOMET)	Digital Library
Villela et al. (2010)	2010	Evaluation of a Method for Proactively Managing the Evolving Scope of a Software Product Line (REFSQ)	Digital Library
Mærsk-Møller and Jørgensen (2010)	2010	Experiences Initiating Software Product Line Engineering in Small Teams with PuLSE (IASTED)	Digital Library
Cavalcanti et al. (2011)	2011	Towards Metamodel Support for Variability and Traceability in Software Product Lines (VaMoS)	Digital Library
Balbino et al. (2011)	2011	An Agile Scoping Process for Software Product Lines (SEKE)	Digital Library
Muller (2011)	2011	Value-Based Portfolio Optimization for Software Product Lines (SPLC)	Digital Library

(continued on next page)

Table 5 (*continued*).

Ref	Year	Title (Venue)	Inclusion by
Acher et al. (2012)	2012	On Extracting Feature Models from Product Descriptions (VaMoS)	Digital Library
Bartholdt and Becker (2012)	2012	Scope Extension of an Existing Product Line (SPLC)	Digital Library
Gillain et al. (2012)	2012	Product portfolio scope optimization based on features and goals (SPLC)	Digital Library
O'Leary et al. (2012)	2012	The Pro-PD Process Model for Product Derivation within software product lines (IST)	Digital Library
Lobato et al. (2012)	2012	Risk management in software product lines: An industrial case study (ICSSP)	Digital Library
Abbas and Andersson (2013)	2013	Architectural Reasoning for Dynamic Software Product Lines (SPLC)	Digital Library
Cruz et al. (2013)	2013	Toward a hybrid approach to generate Software Product Line portfolios (CEC)	Digital Library
Alsawalqah et al. (2013)	2013	A method for software product platform design based on features (SPLC)	Snowballing
Oliveira et al. (2013)	2013	A Feature-Driven Requirements Engineering Approach for Software Product Lines (SBCARS)	Snowballing
Nöbauer et al. (2014)	2014	Similarity Analysis within Product Line Scoping: An Evaluation of a Semi-automatic Approach (CAiSE)	Digital Library
da Silva et al. (2014)	2014	Software product line scoping and requirements engineering in a small and medium-sized enterprise: An industrial case study (JSS)	Digital Library
Sierszecki et al. (2014)	2014	Extending Variability Management to the Next Level (SPLC)	Digital Library
Khtira et al. (2014)	2014	Towards a requirement-based approach to support early decisions in Software Product Line Engineering (WCCS)	Digital Library
Alsawalqah et al. (2014)	2014	A method to optimize the scope of a software product platform based on end-user features (JSS)	Digital Library
Vale et al. (2014)	2014	SPLICE: A Lightweight Software Product Line Development Process for Small and Medium Size Projects (SBCARS)	Snowballing
Ianzen et al. (2015)	2015	Scoping Automation in Software Product Lines (ICEIS)	Digital Library
Karimpour and Ruhe (2016)	2016	Evolutionary robust optimization for software product line scoping: An explorative study (JCL)	Digital Library
Neto et al. (2016)	2016	A hybrid approach to suggest software product line portfolios (ASoC)	Digital Library
Koziolek et al. (2016)	2016	Assessing software product line potential: an exploratory industrial case study (EMSE)	Snowballing
Alam et al. (2017b)	2017	A Secure Framework for Software Product Line Development (IJCA)	Snowballing
Alam et al. (2017a)	2017	An Empirical Study of the Improved SPLD Framework using Expert Opinion Technique (IJEACS)	Snowballing
Haidar et al. (2017)	2017	Agile Product Line Engineering: The AgiFPL Method (ICSOFT)	Snowballing
Ojeda et al. (2018)	2018	A Collaborative Method for a Tangible Software Product Line Scoping (ICAIW)	Digital Library
Ojeda et al. (2019)	2019	Identifying Collaborative Aspects During Software Product Lines Scoping (SPLC)	Digital Library
Kiani et al. (2020)	2020	Extending The Scrum to Introduce The Concept of Systematic Reusability (iCoMET)	Digital Library
Kiani et al. (2021)	2021	A dynamic variability management approach working with agile product line engineering practices for reusing features (TJS)	Snowballing

considering the concepts of product map and market analysis as shown in Table 9.

These two concepts are used alongside each other (Schmid and Gacek, 2000; Balbino et al., 2011; Alam et al., 2017b; Ojeda et al., 2018) they are also dependent on each other. This relation is due to a product map containing market-related aspects, such as time for launching the software products or related costs. A similar analysis may be performed considering the cost models and customer needs, which are concepts that can be related to both the product map and market analysis. Besides that, prioritize products is also related to the candidate analysis as some works (Kishi et al., 2002; Balbino et al., 2011; Cruz et al., 2013; Alsawalqah et al., 2014) use the analysis of candidates for prioritizing products. Thus, despite their different strategies, both concepts influence each other. Although the majority of approaches handle all types of scoping, their strategies vary. It is also important to mention that some approaches focus on specific contributions besides discussing scoping activities, such as CADSE (Estublier et al., 2010) focusing on an SPL scoping metamodel. More details about these aspects are discussed in the following sections.

Table 6
Approaches identified in the studies.

References	Approach Name	Overview
Bayer et al. (2000, 1999), deBaud and Schmid (1999), Bayer et al. (2000), Knauber et al. (2000), Schmid and Gacek (2000), Schmid (2002a,b), Schmid et al. (2005), John et al. (2006), Mærsk-Møller and Jørgensen (2010)	PuLSE	PuLSE was developed as a customizable method to support the conception, construction, usage, and evolution of SPL.
Kishi et al. (2002)	Kishi et al.	Proposes a method for SPL scoping as a decision-making activity.
Kang et al. (2002)	FORM	Describes a SPL asset development process that uses marketing and product plan as a key driver.
Geppert and Weiss (2003)	GOA	Introduces a goal-oriented approach for assessing and evaluating domain candidates for SPLE.
Kim et al. (2003)	DRM	Presents a domain requirements model that integrates features with both goals and scenarios.
Shin Young Park and Soo Dong Kim (2005)	Park et al.	Proposes a process for domain analysis and economical analysis of core asset scope.
Ramachandran and Allen (2005)	FARE	Introduces a method for analyzing requirements for their scope and for their potential to be candidate requirements for a product family.
Kim et al. (2005)	Traceability Map	Defines meta-models of SPLE artifacts used for suggesting a traceability system for general SPL process including a traceability map, links and mapping relationships
Her et al. (2007)	Her et al.	Proposes a comprehensive framework for evaluating the reusability of core assets in SPL based on ISO/IEC 9126.
Noor et al. (2007, 2008a,b)	Noor et al.	Proposes a collaborative scoping approach for organizations migrating existing products to SPL.
Kim et al. (2008)	DRAMA	Provides a framework for modeling domain architecture based on domain requirements within SPL.
Carbon et al. (2008)	Planning Game for SPLE	Presents an adaptation of the agile practice "planning game" to a real SPL context.
Estublier et al. (2010)	CADSE	Describes how evolution of associated engineering environment market and SPL scope needs are addressed together.
Jhon (2010)	CAVE	Describes the CAVE approach and its industrial applications as a solution for the domain expert lack of involvement problem.
Ullah et al. (2010)	COPE+	Introduces a method that attempts to address shortcomings for the specific evolution scenario when a single evolving software system is evolved into SPL.
Elsner et al. (2010)	PLiCs	Provides a generic, reusable reference architecture and methodology for implementing such customizable product lines.
Villela et al. (2010)	PLEvo-Scoping	Describes a quasi-experiment performed to characterize PLEvo-Scoping in terms of adequacy and feasibility
Cavalcanti et al. (2011)	Cavalcanti et al.	Presents a metamodel that aims to coordinate SPL activities by managing different SPL phases and their responsibilities and to maintain the traceability and variability among different artifacts.
Balbino et al. (2011), Lobato et al. (2012), da Silva et al. (2014)	RiPLE	Proposes RiPLE, an agile process for SPL requirements, scoping, risk and evolution management.
Muller (2011)	VB Portfolio Opt.	Introduces Value-Based Portfolio Optimization as an addition to common Product Portfolio Scoping approaches that help to decide on what features are most important to realize.
Acher et al. (2012)	Acher et al.	Aims at easing the transition from product descriptions expressed in a tabular format to FMs accurately representing them.
Bartholdt and Becker (2012)	Bartholdt and Becker	Discusses experiences related to the scope extension of an SPL.

(*continued on next page*)

4.1.2. Architecture definition

The study of Bartholdt and Becker (2012) discusses how important the reference architecture is to the SPL scope as it represents the basis for a governance process. Also, they argue that the architecture must be agreed on as it may reduce the complexity of managing the products of the SPL. PuLSE aims at identifying the characteristics that may be supported by the reference architecture. This is achieved by mapping the product candidates for

Table 6 (*continued*).

References	Approach Name	Overview
Gillain et al. (2012)	Gillain et al.	Proposes a mathematical program able to optimize the product portfolio scope of a software product line and sketch both a development and a release planning
O'Leary et al. (2012)	Pro-PD	Defines a systematic process that provides a structured approach to the derivation of products from an SPL based on a set of tasks roles and artifacts.
Abbas and Andersson (2013)	ARF-E	Proposes extensions (E) to an architectural reasoning framework (ARF) with constructs/artifacts to define and model a domain's scope and dynamic variability
Cruz et al. (2013)	Cruz et al.	Presents a hybrid approach that combines fuzzy inference systems and the multi-objective metaheuristics
Alsawalqah et al. (2013, 2014)	PPSMS	Proposes a novel method to find the optimized scope of a software product platform based on end-user features
Oliveira et al. (2013)	FeDRE	Presents an approach to help developers in the requirements engineering activity for SPL development
Nöbauer et al. (2014)	Nobauer et al.	Presents an evaluation of a tool-supported approach that enables the semi-automatic analysis of existing products to calculate their similarity.
Sierszecki et al. (2014)	Sierszecki et al.	Presents an extension of the variability management that goes beyond the scope of software assets reuse previously introduced into the organization.
Khtira et al. (2014)	SPLBench	Proposes a requirement-based framework that capitalizes on the specific products already derived from the product line.
Vale et al. (2014)	SPLICE	Presents a lightweight development process combining SPLE and agile practices, following reactive and extractive approaches to build similar systems.
Ianzen et al. (2015)	Ianzen et al.	Presents a semi-automatic approach for defining scope identification and classification of product features along with an approach for evaluating the variabilities and commonalities between the established line and a new product.
Karimpour and Ruhe (2016)	Karimpour and Ruhe	Proposes to include uncertainty as part of the SPL scoping model. Scoping planning developed in consideration of uncertainty would be more robust against possible fluctuations in estimates.
Neto et al. (2016)	Neto et al.	Presents an improved hybrid approach to solve the feature model selection problem aiming at supporting product portfolio scoping.
Koziolek et al. (2016)	Domain Analysis process	Reports on case studies performed by applied the domain analysis process into different projects.
Alam et al. (2017b,a)	ISPL	Proposes an improved framework for SPL which addresses cross-cutting concerns such as security and configurability.
Haidar et al. (2017)	AgiFPL	Proposes an agile framework for managing evolving PL with the intention to address the deficiencies identified in current methods, while making use of their advantages.
Ojeda et al. (2018)	CoMeS	Proposes a collaborative method for SPL Scoping.
Ojeda et al. (2019)	Small-SPL	Reports an exploratory study aimed to identify problems related to the collaborative work at scoping SPL in practice.
Kiani et al. (2020, 2021)	APLE	Proposes an approach that extends Scrum to introduce the concept of systematic reusability.

the SPL. The reference architecture is then improved using the results of benefit functions applied in the features. In Kishi et al.'s approach (Kishi et al., 2002), the candidates are analyzed in terms of architecture so that the reference architecture can be build based on them. The DRAMA framework (Kim et al., 2008) gives support to domain architecture modeling based on different architectural styles. One of these styles is later adopted, then evolved according to components and quality attributes analyzed in other steps of the framework. The ARF-E framework (Abbas and Andersson, 2013) deals with architecture reasoning using constructs and artifacts to define and model the scope of the domain as well as the variability of dynamic SPL. Their framework also uses architectural patterns and tactics to establish a knowledge foundation for the architecture. Similarly, in the traceability map approach (Kim et al., 2005), an architecture specification is defined to describe the composition of software components and their functionalities at the conceptual level. This specification includes information such as the style, view, elements, and inter-element relationships.

Other studies, handle the architecture to make it generic to increase its reusability. In Her et al. (2007)'s metamodel, an architecture that is generic to all products is considered. This architecture can be used as a reference architecture when establishing the scope of the SPL. The FORM (Kang et al., 2002) approach have specific activities for conceptualizing the design of the SPL architecture, which is later refined into an architecture model. The PLiCs approach presents a generic reference architecture for customizing SPL. Their reference architecture provides a way of integrating their approach into an SPL that already exists. However, it is limited to SPLs that are generic and consider separation of concerns as a key architectural principle. ISPL (Alam et al., 2017a) uses a generic architecture for developing customer-specific products. This generic architecture is defined during the

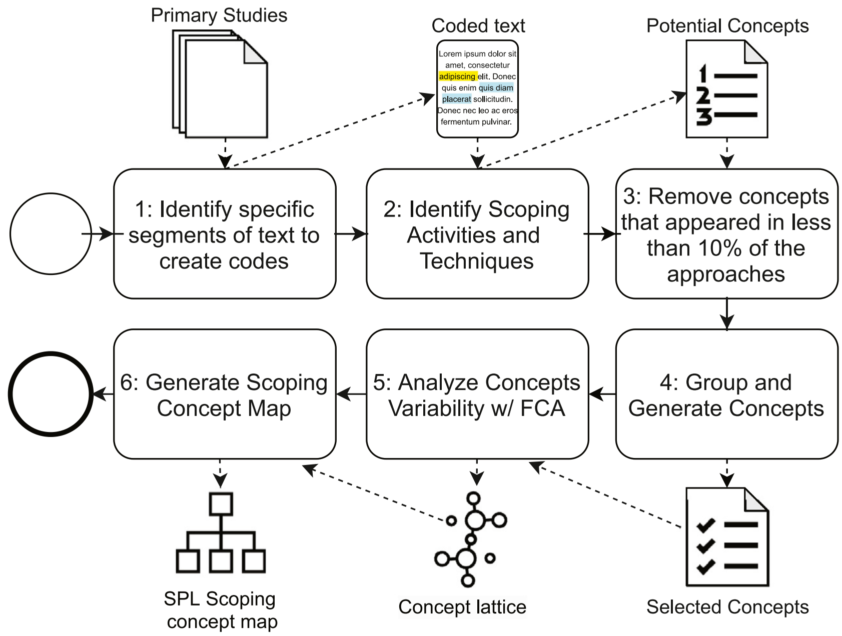

Fig. 3. Process for generating the SPL scoping concept map.

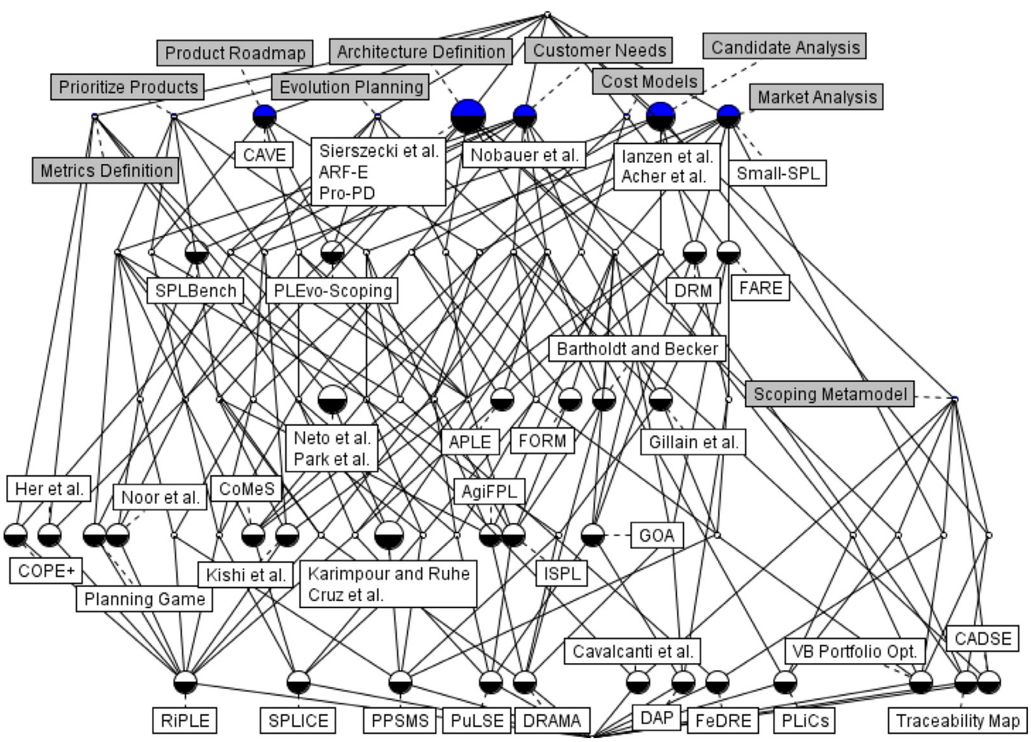

Fig. 4. SPL scoping lattice.

product line design and architecting activity, later evolved into the customer-specific version in the application design and architecting activity. A reference architecture is also used in the AgiFPL approach (Haidar et al., 2017). This architecture is created by using requirements models as input and represents the common assets from the products. Similarly, an architecture sketch is defined in the domain analysis process (Koziolek et al., 2016). This sketch describes the similarities among the products' architecture patterns and styles. This information is used for identifying commonalities and variabilities between products. In the APLE approach proposed in Kiani et al. (2020, 2021), the SPL platform architecture is developed based on the artifact called *SPL backlog*. This architecture can be tailored to derive product architectures for each product of the SPL.

Some approaches have specific roles for dealing with architecture perspectives. RiPLE has the *architect* role, responsible for indicating features that will define the reference architecture. This definition is performed by analyzing non-functional features associated with quality attributes. These attributes have an impact when developing the architecture alongside the asset scoping performed in a specific phase of the process. Pro-PD (O'Leary et al., 2012) defines the *product architect* role which is

Table 7
Approaches activities.

Approach	Phases	Activities
PuLSE	(A) Preparation	1. Pre-assessment meeting 2. Initialization 3. Assessment Team Identification 4. Planning and scheduling 5. PL-Mapping
	(B) Execution	1. Opening Briefing 2. Domain Assessment 3. Preliminary Results 4. Interviewee Feedback
	(C) Analysis	1. Final Report Preparation 2. On-site final meeting
Kishi et al.	NM	1. Identify the requirements 2. Define the design policy 3. List the architectural candidates 4. Determine the Preference of Each Architectural Candidate 5. Examine the Architectural Candidate's Applicability for Each Product 6. Examine the Candidates for the SPL Scope 7. Determine Preferences among the Candidates 8. Define Scope
FORM	NM	1. Marketing and Product Plan 2. Product Plan Refinement 3. Feature Modeling 4. Conceptual Architecture Design 5. Design Object Modeling 6. Architecture Refinement 7. Component Design
GOA	NM	NM
DRM	(A) SPL Scoping	1. Business Investigation 2. Candidate Products Selection 3. Development of SPL scenarios 4. Feature Attachment
	(B) Domain Requirements Analysis	1. Goal Discovery 2. Scenario Authoring 3. Feature Attachment
	(C) Commonality and Variability Analysis	1. Commonality and Variability Analysis 2. Feature Modeling
Park et al.	NM	1. Commonality analysis 2. Variability analysis 3. Variability Dependency Analysis 4. Domain Model Refinement 5. Economical Evaluation of Core Asset Scope.
Traceability Map	(A) Core Asset Engineering	1. SPL Scoping 2. Domain Analysis 3. Core Analysis
FARE	(A) Prepare	1. Establish Analysis Scope 2. Carry out Feasibility Study
	(B) Plan	1. Prepare checklists for assessment 2. Explain Checklists and Processes to Participants 3. Identify Domain Boundaries
	(C) C&V Analysis	1. Identify Commonalities 2. Identify Reuse Opportunities
	D) Quantify	1. Generate Variation Parameters 2. Carry out Cost–Benefit Analysis
	E) Review	1. Apply Checklists 2. Check consistency of market requirements 3. Highlight areas of improvement
Her et al.	NM	NM
Noor et al.	NA	1. Review process objectives and reuse focus 2. Review SPL feature map 3. Identify logical components 4. Map technical solution packages to logical components 5. Map features to the logical components 6. Review reusability metrics of logical components 7. Evaluate the reuse potential of logical components 8. Prioritize logical components for reuse.
DRAMA	NM	1. Identifying components 2. Calculating the priority of components 3. Calculating the priority of quality attributes 4. Modeling domain architectures
Planning Game for SPLE	(A) Exploration	1. Customer creates User Stories (US) with prioritization 2. Developers add estimated time and effort in the US 3. Generate prioritized set of US
	(B) Commitment	1. Sort US by value and risk 2. Derive the scope from selected US
	(C) Steering	1. Conduct planning game for iterations
CADSE	NM	NM
CAVE	(A) Preparation	1. Collect user documentation 2. Divide documents into manageable parts 3. Check manageable parts
	(B) Analysis	1. Apply patterns 2. Produce invalidated SPL artifacts
	(C) Validation	1. Validate and change invalidated artifacts 2. Generate product map
COPE+	(A) Voice of the customer analysis	1. Customer voting on features 2. Clustering of Customers 3. Identification of product variants for each cluster configuration
	(B) Structural impact analysis	1. Identification of features impact 2. Setting up the Genetic Algorithm (GA) 3. Features sequence generation using GA
	(C) Similarity Analysis	1. Conformance of product variant implementation
PLiCs	NM	1. Specify Customized Product Lines (CPL) 2. Set up CPL 3. Specify CPL Product 4. Generate CPL Product
PLEvo-Scoping	(A) Preparation for volatility analysis	1. Establish the time-frame restriction 2. Identify/update system components related to SPL products
	(B) Environment Change Anticipation	1. Identify the actors that play a role in the PL's environment 2. Identify and characterize facts that may be caused or realized by the identified actors 3. Verify the perspective of new actors playing a part in the SPL's environment 4. Classify facts according to their relevance
	(C) Change impact analysis	1. Identify adaptation needs 2. Characterize adaptation needs 3. Classify adaptation needs according to relevance
	D) SPL evolution planning	1. Determine when and which adaptations are expected to be introduced 2. Analyze alternative solutions for dealing with adaptation needs 3. Select alternatives for dealing with the adaptation needs 4. Revise the SPL Evolution Map
Cavalcanti et al.	NM	NM

(*continued on next page*)

responsible for tailoring the platform architecture into the product architecture. The product architecture is composed of a based product configuration and handles variations points based on the requirements. In the Goal-oriented approach (GOA) (Geppert and Weiss, 2003), the architects are involved in the interviews that are carried out to collect data about the products. In this

Table 7 (*continued*).

Approach	Phases	Activities
RiPLE	(A) Pre-Scoping	1. Pre-scoping meeting 2. Analyze market
	(B) Domain Scoping	1. Analyze domains 2. Review domains 3. Identify sub-domains 4. Analyze sub-domains 5. Prioritize domains and sub-domains
	(C) Product Scoping	1. Construct user stories 2. Identify features 3. Features review meeting 4. Identify products 5. Construct product map 6. Validate product map
	D) Assets Scoping	1. Create metrics 2. Apply metrics 3. Prioritize product map
VB Portfolio Opt.	NM	1. Select Preliminary Features 2. Analyze Customer 3. Analyze Cost 4. Analyze Competitors 5. Optimize 6. Decide 7. Realize
Acher et al.	NM	NM
Bartholdt and Becker	NM	NM
Gillain et al.	NM	1. Determine the relevant customers and what their needs are 2. Defining what the products are constituted of 3. Identify conditions for the product to realize the tasks
Pro-PD	(A) Initiate project	1. Translate Customer Requirements 2. Coverage analysis 3. Customer negotiation 4. Create the product requirements 5. Verify the product requirements 6. Define role and task structures
	(B) Identify and refine requirements	1. Find and outline requirements 2. Create the product test cases 3. Allocate requirements 4. Create guidance for decision makers
	(C) Derive products	1. Component development 2. Component testing 3. Component integration 4. Integration testing
ARF-E	NM	NM
Cruz et al.	NM	1. Inferring the cost of each asset 2. Calculating the asset relevance for each segment 3. Calculating candidate products for each segment 4. Qualifying candidate products 5. Grouping the best product of each segment
FeDRE	NM	1. Scoping 2. Requirements Specification for Domain Engineering 3. Requirements Specification for Application Engineering.
Nobauer et al.	NM	1. Select products for analysis 2. Define the scope of the analysis 3. Define how similarity between selected configuration settings are calculated 4. Perform similarity analysis 5. Draw conclusions
Sierszecki et al.	NM	1. Portfolio 2. Requirements management 3. Design and implementation 4. Testing
SPLICE	(A) Portfolio planning	1. Select Business Goals and Marketing Strategies 2. Identify products 3. Identify major features 4. Build product map and feature model 5. Prioritize major features
	(B) Release development	1. Release planning 2. Sprint development
SPLBench	(A) Requirements	1. Elicitation 2. Weighting 3. Transformation of requirements to language
	(B) Features	1. Transformation of domain FM to XML 2. Instantiation of application FM
PPSMS	(A) Analyzing customer needs	1. Classify customer preferences using the Kano's model 2. Prioritize features using the absolute importance values
	(B) Analyzing features	1. Analyzing features for potential commonality and variability
	(C) Optimization	1. Construct mathematical model 2. Optimize with simulated Annealing 3. Analyze non-dominated solutions
Ianzen et al.	(A) Scoping	1. Feature identification 2. Feature Classification
	(B) Product engineering	1. Evaluate variabilities and commonalities 2. Decide to include the features
Karimpour and Ruhe	NM	1. Plan the portfolio scoping based on high profits goals 2. Incorporate uncertainty into SPL scope modeling 3. Perform optimization by simulating changes in the environment
Neto et al.	NM	1. Calculate features' cost 2. Calculate features' relevance 3. Generate candidate products 4. Calculate products' suitability 5. Select best products
Domain analysis process	(A) Domain analysis	1. Gather products and information sources; 2. Establish criteria for reuse 3. Collect and analyze documentation 4. Prepare initial interview documents 5. On-site interview sessions 6. Evaluate results and identify opportunities
	(B) Economic analysis	1. Create business case calculation
ISPL	(A) Domain engineering	1. Business feasibility study 2. SPL scoping 3. SPL requirements analysis 4. Security policy and modeling 5. SPL design and architecting 6. SPL implementation 7. SPL Testing
AgiFPL	(A) Domain requirement engineering	1. Problem space 2. Upgrade domain requirements
	(B) Domain design	1. Domain scoping 2. Domain modeling
	(C) Implementation	1. Planning meeting 2. Production flow 3. Review 4. Retrospective
CoMeS	NM	1. Initial meeting 2. Explore existing products 3. Identify features 4. Identify products sub-domains 5. Specify product map 6. Establish objectives 7. Quantify product map and domains 8. Closure meeting
Small-SPL	(A) Scoping	1. Study the objective domain 2. Identify needs 3. Explore existing solutions 4. List possible solutions and Identify features 5. Establish common features 6. Recognize variable features 7. Diagram feature model
APLE	(A) Planning Meeting	1. Identify Future Configuration 2. Select Base Configuration
	(B) Map use requirements	1. Customer evaluation 2. Customer negotiation
	(C) Sprint backlog	1. Implementation 2. Daily meeting
	(C) Sprint review	1. Review 2. Retrospective

NM - Not Mentioned.

Table 8
SPL scoping concepts identified.

Id	Concept	Definition
AD	Architecture Definition	Define a high-level structure to be used for all products (Pohl et al., 2005)
SM	Scoping Metamodel	Make use of a metamodel to define the structure and constraints of SPL scoping
CM	Cost Models	Define/Use mathematical models for calculating costs related with the SPL development (de Moraes et al., 2009)
CN	Customer Needs	Understand and consider the needs of customers when scoping the SP (Lee et al., 2010)
MD	Metrics Definition	Define metrics to be used for measuring SPL scoping tasks or artifacts (de Moraes et al., 2009)
MA	Market Analysis	Analyze the market to understand the domain and identify competitor products
PM	Product Map	Maps the relation between features and products, usually represented as a matrix (Balbino et al., 2011)
CA	Candidates Analysis	Analyze candidate products or assets to be reused by the SPL (Schmid, 2000)
EP	Evolution Planning	Plan the evolution of the SPL based on the demand from new customer requirements (Krüger et al., 2020)
PP	Prioritize Products	Give higher/lower prioritization to products during SPL scoping.

Table 9
Traceability of scoping concepts on each approach.

Approach	AD	SM	CM	CN	MD	MA	PM	CA	EP	PP
PuLSE	✓			✓		✓	✓	✓	✓	
Kishi et al.	✓							✓		✓
FORM	✓					✓			✓	
GOA	✓					✓		✓		
DRM						✓		✓		
Park et al.			✓		✓					
Traceability Map	✓	✓						✓		
FARE			✓			✓				
Her et al.	✓				✓					
Noor et al.				✓	✓		✓			✓
DRAMA	✓		✓	✓		✓				✓
Planning Game				✓					✓	✓
CADSE		✓							✓	
CAVE							✓			
COPE+					✓				✓	
PLiCs	✓	✓		✓						
PLEvo-Scoping							✓		✓	
Cavalcanti et al.		✓			✓					
RiPLE	✓			✓	✓		✓	✓	✓	✓
VB Portfolio Opt.		✓	✓	✓		✓		✓		
Acher et al.								✓		
Bartholdt and Becker	✓			✓		✓				
Gillain et al.			✓	✓		✓				
Pro-PD	✓									
ARF-E	✓									
Cruz et al.			✓	✓	✓					✓
FeDRE		✓				✓	✓			✓
Nobauer et al.				✓						
Sierszecki et al.	✓									
SPLICE				✓		✓	✓	✓		✓
SPLBench				✓	✓					
PPSMS			✓	✓	✓	✓		✓		✓
Ianzen et al.								✓		
Karimpour and Ruhe			✓	✓	✓					
Neto et al.			✓		✓					
Domain analysis process	✓		✓			✓		✓		
ISPL	✓			✓		✓	✓		✓	
AgiFPL	✓					✓				✓
CoMeS				✓		✓	✓			✓
Small-SPL						✓				
APLE	✓			✓				✓		
Total	**17**	**6**	**10**	**18**	**11**	**17**	**9**	**13**	**8**	**11**

AD - Architecture Definition; SM - Scoping Metamodel; CM - Cost Models;
CN - Customer Needs; MD - Metrics Definition; MA - Market Analysis;
PM - Product map; CA - Candidates Analysis;
EP - Evolution Planning; PP - Prioritize Products.

interview process, the interviewees are asked questions about the documentation related to the architecture of the products.

4.1.3. Scoping metamodel

In terms of scoping metamodel, Cavalcanti et al. (2011) proposes a metamodel developed as a UML profile, which was defined based on a set of SPL scoping requirements identified in the literature. This proposed metamodel may be used for different scoping tasks found in other approaches, namely, feature modeling, product scoping, variability management, and asset definition. In VB Portfolio Opt. (Muller, 2011) a model is defined as an ontology, which considers costs and market-related factors that can be addressed while scoping an SPL. With similar goals, the FeDRE approach (Oliveira et al., 2013) utilizes a metamodel to establish guidelines for the SPL requirements specification. This metamodel represents concepts involved when specifying use cases and their relationships. The metamodel is also used for defining the features of the SPL corresponding to the use cases.

With different goals than the previous two approaches, CADSE (Estublier et al., 2010) uses metamodel in different SPL scoping

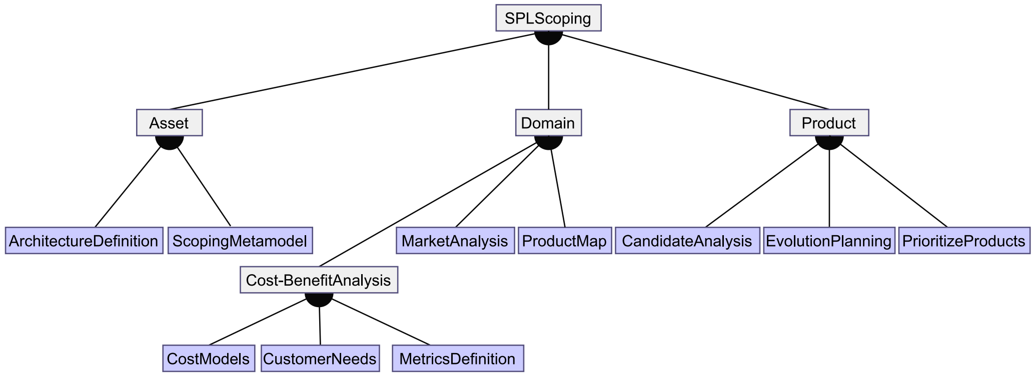

Fig. 5. SPL scoping concept map.

activities, such as for planning the scope and market evolution. These metamodels are used by the CADSE generator for generating an associated engineering environment dedicated to the production of the products. PLiCs uses problem space metamodels, which describe the scope domain. These metamodels are instantiated for specifying the products of the SPL. In the traceability map approach (Kim et al., 2005), authors argue to use metamodels for defining SPL artifacts in a "semi-formal manner". The approach proposes metamodels for variability analysis, core asset design, and decision resolution. All these models are used for the traceability aspect of the approach, as their structure is used as the basis for generating the related artifacts.

4.1.4. Cost models

When dealing with cost management, approaches intend to focus on the profit of the organization. The RiPLE process considers the economical and social perspectives for dealing with costs. In this sense, the process defined business goals to deal with these perspectives, including reducing the costs, improve productivity, improve the time-to-market, among others. Karimpour and Ruhe (2016) considers profit as a goal for planning scoping portfolio. The portfolio provides metrics such as costs for calculating profit using specific functions. The Value-Based Portfolio Optimization approach (Muller, 2011) considers a cost-revenue for defining the profit related to the SPLE.

Additional strategies are used considering the features/assets cost estimation. For instance, PPSMS et al. (Alsawalqah et al., 2014) considers the estimation of feature costs that are calculated according to the scenario where the approach is being applied. Cruz et al. (2013) formalized the inferences of development costs using mathematical equations. This is considered during the scoping process. Gillain et al. (2012) makes use of cost functions from other studies. These cost functions are integrated within their mathematical model for generating cost-related results. Neto et al. (2016) uses the results of feature cost metrics for generating candidate products for the SPL aiming at a higher return of investment. In Noor et al. (2007, 2008a) several strategies are used to provide initial cost/effort estimation for adapting logical components while modeling scoping.

Cost models are also important due to their relation with decision-making. In Kishi et al. (2002), costs related to the SPL are used as a quality attribute for decision making during the scoping process. There is also the Pro-PD approach (O'Leary et al., 2012), which considers the cost calculation for taking decisions as their approach, thus defining cost-dependent activities. Shin Young Park and Soo Dong Kim (2005) defines metrics for calculating the cost of different aspects of the scoping process. Lastly, FARE (Ramachandran and Allen, 2005) applies cost–benefit analysis by using mathematical expressions. This decision-making based on costs is also present in Koziolek et al. (2016) where the calculation of the return of investment is defined based on several inputs as defined in Bockle et al. (2004). This return of investment is the ratio of the cost savings to the cost of investments. These sets of costs include the costs for maintaining each of the former products without reuse, costs for changing the organization to adopt SPL engineering and for training people in SPL engineering, costs for building the core asset base or platform including product scoping and implementation, among others.

4.1.5. Customer needs

Customer needs are important in any software development strategy as they drive the requirement specifications, thus, defining the features of the software products. In the SPL scoping context, the needs of a customer are associated with which features of the SPL should be maintained and which could be removed. This analysis is related to the domain or market segment. In the PuLSE framework (Schmid, 2002a), for instance, customer needs are taken into account when considering the market analysis. These needs are used for deciding which features will be part of the products. This result is achieved by applying a weight-based schema, in which customer preferable features receive a higher weight, thus, receiving a higher priority. In Noor et al. (2007, 2008a) customers are a crucial part of the process as they are responsible for prioritizing features, products, and their needs are considered during marketing decisions. In the DRAMA framework (Kim et al., 2008), customer needs are important for the goal/scenario-based process. This process aims to provide products based on customer preferences and needs. As part of the Planning Game in SPLE (Carbon et al., 2008), customers work alongside developers for formulating the scope of the SPL. Based on their interaction, user stories are defined, containing the information related to the necessities of the customers. The PLiCs methodology (Elsner et al., 2010) aims at generating products based on customer needs. This generation relies on the analysis of customer sub-domains, where the customer requirements are registered and analyzed before generating the product. Furthermore, in the SPLICE approach (Vale et al., 2014), customer needs are defined based on business goals. This information is gathered during the analysis of the market, where market strategies are defined.

In the RiPLE process, the customer is a well-defined role and its feedback is collected during the whole process during review meetings. This feedback is used for refining the requirements, as well as prioritizing the product map. In the APLE approach (Kiani et al., 2020, 2021), the customer is also crucial as it provides continuous feedback, similarly to how this feedback is provided in the Scrum methodology. The importance of the customer is highlighted by the *product owner* role, which represents the needs of customers and users of the products.

In the VB Portfolio Opt. (Muller, 2011), customers are considered an important entity, and their relation with products and assets is analyzed for calculating the possible profit of each candidate product. The needs of the customer are also considered by analyzing the market competitors. Bartholdt and Becker (2012) approach considers the customer needs for establishing common agreements regarding the SPL scope. Also, the decision-making process related to technical solutions considers the customer needs. Gillain et al. (2012) utilize mathematical models for identifying relevant customers and their needs. These models are described as goal models, representing the goals of the customers which are used when deciding upon the inclusion of features in the SPL.

The approach of Cruz et al. (2013) generates product portfolios based on customer needs from different segments. In their approach, the customer needs are based on answers concerning the relevance of each feature according to the customer's perspective. In the study of Nöbauer et al. (2014), their approach compares customer-specific product configurations to perform the variability analysis for generating the SPL. Thus, each product is composed of features defined based on the customer's needs. This comparison is performed considering how each feature is configured for each product. The SPLBench framework (Khtira et al., 2014) uses the level of importance of the requirements for the customer as means to define the scope of the project. The customers' needs are gathered and analyzed through the whole process as they are crucial for achieving SPLBench goals. PPSMS (Alsawalqah et al., 2014) uses Kano's model (Matzler and Hinterhuber, 1998) for understanding customer preferences and classifying customer needs based on survey data. The result of this classification is used for prioritizing the features of the SPL. Karimpour and Ruhe's approach (Karimpour and Ruhe, 2016) considers customer segments to understand how much these customers are willing to pay for the products. This value is used to generate a cost model that aids the definition of the SPL costs. The ISPL framework (Alam et al., 2017a) proposes the development of customer-specific products based on the generic architecture, reusing core assets from the domain engineering. The CoMeS method considers the customer as a possible member of the stakeholders' team, which is responsible for scoping the SPL in their approach.

4.1.6. Metrics definition

Considering metrics for defining scoping, in ARF-E (Abbas and Andersson, 2013) authors use quality attribute scenarios (QAS). These QAS are composed of different elements which are related to three variability questions: (i) why does it vary? (ii) what does vary? and (iii) how does it vary? The authors also extended these QAS with additional elements that can be used for specifying the variability in terms of variation points and constraints. PPSMS et al. (Alsawalqah et al., 2014) uses Kano's model (Matzler and Hinterhuber, 1998) of customer satisfaction for classifying and prioritizing customer needs to be based on how these needs affect their satisfaction.

The RiPLE (Balbino et al., 2011; da Silva et al., 2014; Lobato et al., 2012) approach presents a specific phase where metrics are created based on business goals and categorized as development benefits metrics and characterization metrics. These metrics are applied to prioritize a product map, selecting the features with more potential for the SPL. A similar strategy is used by Kishi et al. (2002), where the metrics are defined using a decision-making method, the analytic hierarchy process (AHP). The AHP method considers the most desirable architectural candidates according to some decision criteria. The results of this selection are used for calculating the applicability of the architectural candidates for each product of the SPL.

The metamodel presented by Cavalcanti et al. (2011) consider the importance of metrics in SPL scoping. The metamodel includes the abstraction of these metrics allowing their instantiation based on a specific scenario. As their approach considers metrics at a modeling level, the work of Neto et al. (2016) uses metrics in a source code level. The first metric is related to the level of interdependence between scoping assets. The second metric represents the number of lines of code associated with each asset. The last metric considered is the number of flows represented by each asset of the SPL. The studies of Noor et al. (2007, 2008a) also consider metrics in a source code level such as the size of files and methods of a file. Other metrics such as complexity, dependencies, and understandability of the source code are also measured for the decision-making process regarding the SPL development.

In Cruz et al. (2013), the cyclomatic complexity and the size of product assets are calculated considering metrics from their metric tool. In SPLBench (Khtira et al., 2014), metrics are used for performing the bench-marking of the SPL. These metrics are the number of products per requirement, requirements per product, number of requirements per product, an average of requirements in products, requirements of the best product, new requirements, percentage of new requirements, non-implemented requirements, and percentage of non-implemented requirements.

The work of Her et al. (2007) is completely designed around quality metrics. These metrics are defined based on quality attributes from the ISO/IEC 9126. As their set of metrics is extensive, three of these are specific designed for scoping: functional coverage, non-functional commonality, and variability richness. Besides, Shin Young Park and Soo Dong Kim (2005) merges metrics with cost models, using a set of metrics to analyze the economical aspects and value of the SPL. Lastly, the COPE+ approach (Ullah et al., 2010) uses a small set of metrics to measure similarities among product portfolios.

4.1.7. Market analysis

Analyze the market opportunities is important for any software company. In the SPL scoping context, market analysis is done by several approaches, using different strategies. The PuLSE approach has a specific step where marketing experts can contribute with information on what features of the SPL may be helpful to gain market share. The main goal of this marketing phase is to optimize the number of features that can be included in each product, thus addressing a larger range of customers. The FARE method uses the market analysis results to assign priorities to each variability and commonality of the SPL. Also, according to the authors, "market analysis is the key to the assessment of potential benefits and risks" (Ramachandran and Allen, 2005). The DRAMA framework uses a goal and scenario-based analysis technique to elicit, analyze, and construct domain requirements gathered from analyzing the markets and customers. Similarly, the GOA approach (Geppert and Weiss, 2003) uses the analysis of the market for understanding the revenue potential of the SPL. This potential is obtained by analyzing how the products may impact the market domain, considering factors such as high market share and low development costs.

The FORM approach proposed in Kang et al. (2002) focuses on the analysis of the market as the main driven force for designing the SPL products. Thus, the approach has specific activities that are market-related, such as *marketing and product planning* and *product plan refinement*. The Domain Requirements Model (DRM) approach (Kim et al., 2003) is also driven by marketing analysis. This analysis is used when scoping the SPL when "business investigation" is performed. This investigation aims at identifying and understanding the organization's business goals.

The RiPLE process has a role defined as *market analyst*, which can provide information from the market analysis, aiding the identification of the most relevant domains and products for the SPL market segment. The process has a specific phase called *analyze market* which intends to "obtain information of the market segments in which the domains are inserted for identifying issues that can determine their success in the marketplace" (Balbino et al., 2011). The CoMeS framework also defines a role called *market expert*, which represents business concerns, is aware of customers, competitors, and their products, and provides information regarding sales. All this information provided can aid the decision-making when developing the SPL. A similar role, with similar goals, is present in the Small-SPL process. Similarly, in the FeDRE approach (Oliveira et al., 2013), the *market expert* is defined as a role which is responsible for performing the scoping activity. In the AgiFPL approach (Haidar et al., 2017), the role called *Software Vendor* is responsible for building a business strategy based on market trends, potential investment opportunities, and core-business improvements. This role has similarities with the *market analyst/expert* mentioned in the previous approaches.

In the VB Portfolio Opt. (Muller, 2011), there is a specific activity for analyzing the competitors defined in the market, as understanding the competitors' products is important for providing customers with a better-suited product. In Gillain et al. (2012), authors use a mathematical model to generate sets of features satisfying each market segment relevant to the company. This generation varies based on the type of the market segment, leading to different marketing strategies obtaining different results. The PPSMS (Alsawalqah et al., 2014) method uses input from domain and market experts to identify indicators for commonality and variability analysis. This information is also used for classifying and prioritizing each feature of the SPL and how they affect customer satisfaction for each market segment. Karimpour and Ruhe (2016) use a different approach as the authors incorporate uncertainty into the market analysis to plan the portfolio scoping based on high profits goals using robust analysis algorithms. In the ISPL (Alam et al., 2017b,a) framework, the market is analyzed to understand the feasibility of the business strategies related to the SPL development.

Market analysis is crucial in the SPLICE approach (Vale et al., 2014). The approach defines the role of the *business expert*, which is responsible for aligning goals and market strategies. These strategies are defined during the *portfolio planning* phase. In SPLICE, these strategies will define how the products of the SPL may be released to customers considering their needs. Market analysis is also applied in Koziolek et al. (2016) to check if the scope of the SPL is expected to increase or decrease in the future. This analysis defines a criterion in their approach called *market fit*.

4.1.8. Product map

A product map is important for managing the progression in which the SPL is developed and evolved. In the PuLSE framework (Bayer et al., 2000; deBaud and Schmid, 1999; Bayer et al., 2000; Knauber et al., 2000; Schmid and Gacek, 2000; Schmid, 2002a; Schmid et al., 2005; John et al., 2006; Mærsk-Møller and Jørgensen, 2010), a product map can be used to facilitate the identification and description of potential products, reducing the future effort required when developing the SPL. This map might be processed in advance to prepare the several activities of their process, such as identify products, identify existing assets, and plan product releases. In Noor et al. (2007, 2008a), participants use thinkLets (Briggs et al., 2001) to navigate through the product map and suggest possible changes for the products. In the CAVE approach (Jhon, 2010), a product map may be extended by experts by including new features, domains, among other aspects. This product map is used for documenting the SPL scoping.

PLEvo-Scoping (Villela et al., 2010) identifies products and features of the SPL and registers them in the product map. RiPLE (Balbino et al., 2011; Lobato et al., 2012; da Silva et al., 2014) defined the product map as a matrix of features and products discovered. Their product map also differentiates current features in the products with possible future features. Their product map is later validated with customers, domain experts, and market experts. In the FeDRE approach, the product map is also defined as a matrix of *products X features*. Their product map describes the relationship among SPL features, guiding the construction of the SPL. This product map structure is similar to the one presented in SPLICE (Vale et al., 2014). In their approach, the product map is defined during the *portfolio planning* phase, and it is used for specifying which features are present in each product. Other studies use the product map as input/output for documenting SPL scoping activities (Alam et al., 2017b,a; Ojeda et al., 2019).

4.1.9. Candidates analysis

The analysis of candidates is important for defining the product portfolio. This analysis is performed by seven approaches following different directions. For instance, PuLSE's goal is related to developing a mechanism to represent different product candidates for an SPL, detailing them in a specific granularity level. The approach has a step called "map out product candidates" where these candidates are identified and classified as existing products, future products, and potential products (Schmid, 2002a). Kishi et al.'s approach (Kishi et al., 2002) has a similar goal. In their approach candidates are analyzed to identify the most appropriate one. This analysis is based on the requirements present in each candidate, however, also applying an analytic hierarchy process (Satty et al., 1980) on the quality attributes present in each candidate. In the VB Portfolio Opt. (Muller, 2011) a candidate is selected, and compared in terms of overall performance according to a profit function, which considers prices, product assignments, and other products of the portfolio. This candidate is then compared to the other candidates, and the one with a higher result in the profit function is selected. Cruz et al. (2013) applies candidates analysis intending to prioritize products to be part of the portfolio. This analysis is performed during two stages. Firstly, candidate products for the portfolio are generated based on minimizing feature's cost and maximizing features relevance. Secondly, these candidates are evaluated in terms of quality. This quality is calculated by applying a fuzzy inference system which is guided by the intuition of customers and developers. This result is used for ranking the generated candidates. Neto et al. approach's (Neto et al., 2016) generates candidate products based on their cost and relevance. This calculation is performed based on the cost and relevance of the features present in each product. Furthermore, their approach also calculates a suitability score for each candidate product based on each customer segment. A selection of possible candidate products is also performed in the DRM approach (Kim et al., 2003). This selection determines what products should be included in the SPL being developed. The decisions related to this selection is based on the market analysis performed in previous steps of their process.

Candidates are also analyzed in the domain analysis process (Koziolek et al., 2016) in the asset level. These assets are identified based on the architecture reconstruction performed in their process. Acher et al. (2012) analyzes candidates in the feature level instead of product level. The features are analyzed considering their classification (optional or mandatory) and relations (OR and XOR) to determine their preference. Also considering the candidate feature-level analysis, the RiPLE process uses market experts' knowledge for analyzing the features list and business plan (Balbino et al., 2011). The result is a list of candidate products that have more potential to become SPL. Furthermore, COPE+ (Ullah et al., 2010) analyzes different aspects of the features' source code to determine decisions related to adding or removing features from the candidates. These aspects analyzed include the level of impact of a given feature considering classes and packages in the source code. The approach filters the possible candidates using these aspects until select those that present less structural impact if transformed into SPL. PPSMS (Alsawalqah et al., 2014) also deals with candidate features. Based on the feedback obtained from customers, candidate features are classified as common (mandatory) or variable (optional). The result of this analysis is used when building a candidate product platform based on the features. In the GOA approach (Geppert and Weiss, 2003), candidates are analyzed at the domain level. More specifically, candidate domains are analyzed and the results of this analysis are used for deciding if such domains which domains will be developed first.

Ianzen et al. (2015) deals with feature candidates by analyzing the documentation of the system to be engineered. This analysis is automated, where textual analysis is performed for identifying the candidate features in multiple documents. The result of this analysis are artifacts describing the classified features, that can be used as input for stating the SPL scoping. The traceability map approach (Kim et al., 2005) handles the analysis of candidates when specifying a *decision resolution model*. This model includes the information related to why appropriate candidates were selected. In SPLICE (Vale et al., 2014) the *scope owner* and the *product expert* define a list of candidate features that are used for defining the product map. In the APLE approach (Kiani et al., 2021), the candidates are analyzed as components that can be adapted to meet required needs.

4.1.10. Evolution planning

The plan for evolving the SPL based on the demand of new requirements is important for maintaining the products updated to the market needs. This plan is adopted following distinct strategies. The PuLSE framework, for instance, has a specific component for handling the evolution of the SPL infrastructure over time. It relies on an element called *maturity scale*, which provides a path for integrating and evolving SPL adoption. The SPL scoping generated by one of the PuLSE's modules is taken into account when planning the evolution as the scope of the SPL might evolve as well.

Concerning decision-making aspects, the Planning Game in SPLE (Carbon et al., 2008) considers evolution planning to be a decision that should be made from an organizational point of view. Thus, the evolution planning is performed by considering both application's and domain's engineers' feedback concerning SPL components. Furthermore, COPE+ (Ullah et al., 2010) is a customer-oriented product evolution method. It considers different criteria for making evolution decisions, namely: organizational, financial, technical, and customer-related. Thus, evolution planning is performed throughout the whole process, as it relies on evolution-based decisions.

Differing from those, the CADSE approach proposes to use a different type of artifact versioning that provides more flexibility and reliability for SPL components. This versioning strategy relies on applying versioning and control not only to individual source files but also to different models created for the SPL definition. In this model, each attribute, and each relationship definition have their evolution characteristic. PLEvo-Scoping (Villela et al., 2010) focus on evolution planning that may be applied to different scoping approaches. The approach gives this support by helping the SPL scoping team anticipate emergent features and distinguish unstable (features impacted from future adaptations) from stable ones. More specifically, the evolution planning is performed by establishing when and how important adaptation needs should be introduced in the SPL. Also, the accommodation of these needs is prepared beforehand.

Not so related to the approaches discussed previously, the ISPL (Alam et al., 2017b,a) considers evolution during the application engineering phase. Although evolution planning is not specifically defined, the authors recommend the documentation of SPL evolution aspects when applying their approach. Similarly, in the FORM approach (Kang et al., 2002) an evolution planning activity is not defined. Their approach, however, aims at planning how the products can evolve based on the market evolution. Thus evolution planning is a core aspect of their approach.

4.1.11. Prioritize products

The prioritization of products is used in the process of planning the SPL development. In Kishi et al. (2002) product candidates are analyzed in terms of architecture to determine their preference in comparison to the other applicable architectural candidates. This preference is calculated based on the required architectural changes that each candidate may need to become an actual product. Similarly, the CoMeS method (Ojeda et al., 2018) applies prioritization when building the product map. This prioritization must be agreed upon by all team members before closing the SPL scope.

In the PPSMS method (Alsawalqah et al., 2014) prioritization is performed in terms of features to provide decision capabilities regarding which features should be part of the products. This prioritization is calculated using Kano's model (Matzler and Hinterhuber, 1998) based on customer needs, as discussed earlier. Also, RiPLE handles product prioritization during the product map refinement. While refining the product map, the customer, manager, marketing team, and domain experts can prioritize features with more potential for the SPL. This process is similar to how features are prioritized in the FeDRE and in the SPLICE approaches. Additionally, Cruz et al. (2013) uses the candidate analysis for prioritizing the products to be part of the portfolio. This prioritization is based on the cost and relevance of the features of each product.

Some approaches consider prioritizing different aspects of the products. In Noor et al. (2007, 2008a), prioritization is handled in terms of logical components. This prioritization is carried out by stakeholders considering the reuse value, and the effort required to reuse each logical component. The Planning Game in SPLE (Carbon et al., 2008) uses prioritization in user stories (also referred in their work as *reuse stories*) to establish development effort estimation. This prioritization is later used when selecting the user stories to be implemented. In the AgiFPL approach (Haidar et al., 2017) features are prioritized similar to users stories in Scrum, by defining an *app backlog* that represents the list of most important features to be developed in a *sprint*.

4.1.12. Generic process and approaches adaptability

By analyzing the similar activities amongst these works, we were able to establish a generic SPL scoping process, illustrated in Fig. 6. *Pre-Scoping* is the first task, where supporting concepts from the scoping concept map may be used, such as metrics definition.

The other activities are related to each scoping type. For instance, an analysis of the market may be performed in the *domain scoping* activity. As presented in the BPMN model, the activities related to the scoping type are not mandatory, however, at least one must be performed. Lastly, the *scoping closure* activity is executed. This activity is generic as several ways to close the scoping process may be performed, such as the evolution planning, also extracted from the concept map.

When analyzing the approaches to define the generic process, we had to understand how they handle the capability of adaptation to different organizational contexts/scenarios. In this sense, we can observe that the topic is covered using different strategies as presented next. PPSMS (Alsawalqah et al., 2014) uses a set of rules for defining possible scenarios. These scenarios are responsible for the result of the cost calculations. Thus, their approach handles the scoping costs according to the scenario of the organization. Cavalcanti et al. (2011) metamodel also considers organization scenario. The metamodel was designed for modeling the scoping according to different scenarios. In addition, PLiCs (Elsner et al., 2010) authors discuss their approach applicability in other contexts, besides the one shown in their case study. Similarly, CAVE (Jhon, 2010), may be applicable in different situations based on existing documentation, as it was designed as a generic process. COPE+ (Ullah et al., 2010) also is customizable according to the organization's business and technical parameters. Her et al. (2007) goes further in that area, as their framework contains guidelines to aid its application in different projects.

Gillain et al. (2012) is a context-aware approach, and might be instantiated in different contexts, as the authors show in their study. Their mathematical model considers market strategies as one of the aspects for defining the instantiation. Neto et al. (2016) presents a different strategy using fuzzy sets associated with a fuzzy inference system for customizing their approach. Lastly, PuLSE (deBaud and Schmid, 1999; Bayer et al., 2000; Bayer et al., 2000; Knauber et al., 2000; Schmid and Gacek, 2000; Schmid, 2002a; Schmid et al., 2005) presents a sub-process called PuLSE-BC. This sub-process defines the adaptation/customization of the framework according to the context where it will be applied. This customization ensures that the process and the products are appropriate. Three works have reported a customized PuLSE process (John et al., 2006; Mærsk-Møller and Jørgensen, 2010; Schmid et al., 2005) as a case study.

Answering RQ1: What are the similarities and differences among the approaches? *By analyzing the variability among SPL scoping strategies we were able to define an SPL scoping concept map (Fig. 5). This map is composed of three distinct scoping types, namely asset, domain and product. Each type contains sub-concepts such as architecture definition and scoping metamodel for asset scoping; cost–benefit analysis, market analysis and product map for domain scoping; and, candidate analysis, evolution planning and prioritize products for product scoping; By analyzing the traceability of each approach to these concepts, we could understand how the approaches applied these concepts and how the concepts related with each other. This analysis gave us evidence to establish a generic SPL scoping process (Fig. 6) which represents the different scoping types. The process is composed of five activities, namely pre-scoping, domain scoping, product scoping, asset scoping, and scoping closure. By instantiating the concepts from the map into the generic process, researchers and companies may establish a scoping process adapted to a specific scenario.*

4.2. RQ.2 How are existing scoping approaches evaluated?

The selection of an approach may also be affected by its maturity level, which is directly related to the contexts where studies have been applied. In this section, we present the evaluation aspects of the analyzed studies. As discussed in Section 3, to classify the evaluations, we used the definitions presented in Table 4. For instance, despite authors mentioning "case studies" as their evaluation, their protocol and results were not what the literature defines as a case study (Wohlin et al., 2012). Thus, we classified their evaluation following the definitions and principles reported in this literature (Wohlin et al., 2012; Runeson et al., 2012; Fink, 2003).

4.2.1. Evaluations applied

As defined in our protocol, all studies included in this SLR present some evaluation. Fig. 7 presents an overview of the types of evaluations per primary study. There were 26 case studies, four experiments, 14 manual comparisons, 15 proofs-of-concept, three quasi-experiments, and three surveys. Considering the approach perspective, Table 10 also shows how those different evaluation methodologies were used for the same approach, such as Cruz et al. that implemented a proof-of-concept and applied a survey. We also compared the methodology used with the publication year, giving an overview of the number of different evaluations among the years. The most applied empirical evaluation was the case study, performed in 26 studies. The reason would be the context of a case study protocol, as we understand that applying SPL approaches in real organizations gives more reliability to the proposal. We noticed, however, a lack of controlled experiments as only four (Fig. 7) were present in the studies.

Although case studies are more indicated to evaluate in real scenarios, experiments are still important when the researchers intend to monitor and compare their proposals with similar ones.

There were also non-empirical evaluations performed (proof of concepts, manual comparisons, and surveys). Although these strategies for evaluating an approach may provide important results, they lack the reliability of empirical evaluations. Fig. 7 also shows that the number of publications related to SPL scoping has decreased in the last five years. This decreasing number may be the result of the organizations' lack of understanding of the real benefits obtained from adopting well-defined SPL scoping approaches.

Answering RQ2: How are existing scoping approaches evaluated? *We concluded that most of the evaluations applied by the approaches focused on measuring how these approaches may benefit the organizations in real projects, as 26 studies (45%) applied case studies. We also understand that controlled experiments are not common in the field as only four studies (7%) applied them. Also, the first reported experiment was published in 2012, thirteen years ahead of the first identified approach (1999). Furthermore, we could see that the number of publications has been decreasing in the last five years.*

4.3. RQ.3. What are the open research gaps and opportunities for new studies on the topic of SPL scoping?

In this section, we present and discuss research gaps and opportunities identified from our findings. These opportunities are categorized according to SPL scoping concepts.

Decision-Making: As we identified, many aspects of SPL scoping are used for decision-making. In this context, some works are planned to further investigate how decisions may be addressed based on different elements of the SPL. In Alsawalqah et al. (2014), the authors discuss that business factors must be addressed to provide more comprehensive decision support for their framework. These factors may aid the team to decide how to prioritize the SPL features. In Bayer et al. (2000), authors stated that they will work on a guidebook, containing lessons learned from their experience. Such a guidebook will be incorporated into their process. A similar gap is identified in Estublier et al. (2010), where the authors argue that heuristics could be used to improve their process. In John et al. (2006), authors cite that in the future, they plan to create a model for scoping and SPL goals that may be integrated into their approach and used for introducing the SPL concept into an organization.

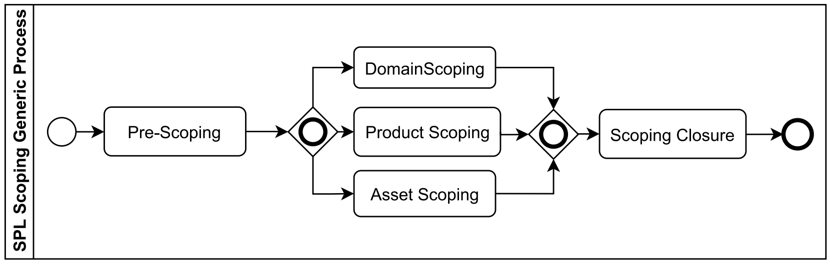

Fig. 6. A generic SPL scoping process.

Table 10
Approaches evaluations.

Approach	CS	Ex	MC	PoC	QE	Su
PuLSE	⊕		✓	✓		
Kishi et al.	✓					
FORM			✓	✓		
GOA	✓					
DRM			✓	✓		
Park et al.				✓		
Traceability Map		✓				
FARE			✓	✓		
Her et al.	✓					
Noor et al.	⊕					
DRAMA			✓		✓	
Planning Game	✓					
CADSE			✓	✓		
CAVE	✓					
COPE+				✓		
PLiCs	✓		✓			
PLEvo-Scoping					✓	
Cavalcanti et al.				✓		
RiPLE	⊕		✓			
VB Portfolio Opt.				✓		
Acher et al.		✓				
Bartholdt and Becker			✓			
Gillain et al.	✓					
Pro-PD			✓			
ARF-E				✓		
Cruz et al.				✓		✓
FeDRE	✓					
Nobauer et al.	✓					
Sierszecki et al.	✓					
SPLICE	✓					
SPLBench				✓		
PPSMS	✓			✓		
Ianzen et al.		✓				
Karimpour and Ruhe		✓				
Neto et al.	✓					
Domain analysis process	✓					
ISPL			✓			
AgiFPL				✓		
CoMeS				✓		
Small-SPL					✓	
APLE		✓				✓

⊕ - Multiple Cases
CS - Case Study; Ex - Experiment; MC - Manual Comparison;
PoC - Proof-of-Concept; QE - Quasi-Experiment; Su - Survey.

Decision-making is also related to risk management. This concept however, was only found in three approaches, namely RiPLE (Lobato et al., 2012), Cavalcanti et al. (2011) and Planning game for SPL (Carbon et al., 2008). Despite its low appearance in the approaches, risk management is an important scoping concept as its application may improve the SPL development process and avoid potential problems (Lobato et al., 2012). Its importance can be also seen as few works discuss including risk management in future work (Gillain et al., 2012; Alsawalqah et al., 2013, 2014). Future work discussed by the authors also includes investigating how to integrate customer decisions based on the analysis of competitors' products (Gillain et al., 2012). As the market analysis is important, so it is the consideration of cost–benefit from the SPL perspective. Thus, mathematical models capable of calculating costs are important when considering SPL scoping such as those presented in Gillain et al. (2012). In this sense, guidelines are important to ease the complexity of this calculation. Besides, the authors of Gillain et al. (2012) intend to extend their cost functions to further integrate them into mathematical models. Thus, integrating risk management based on these cost models. Improving risk management is also discussed in future work of Lobato et al. (2012). In their case, however, the idea is to combine their experience obtained for executing different case studies. Another work that mentions the improvement of cost models as future work is Schmid (2002a), as authors intend to improve market aspects of their proposal.

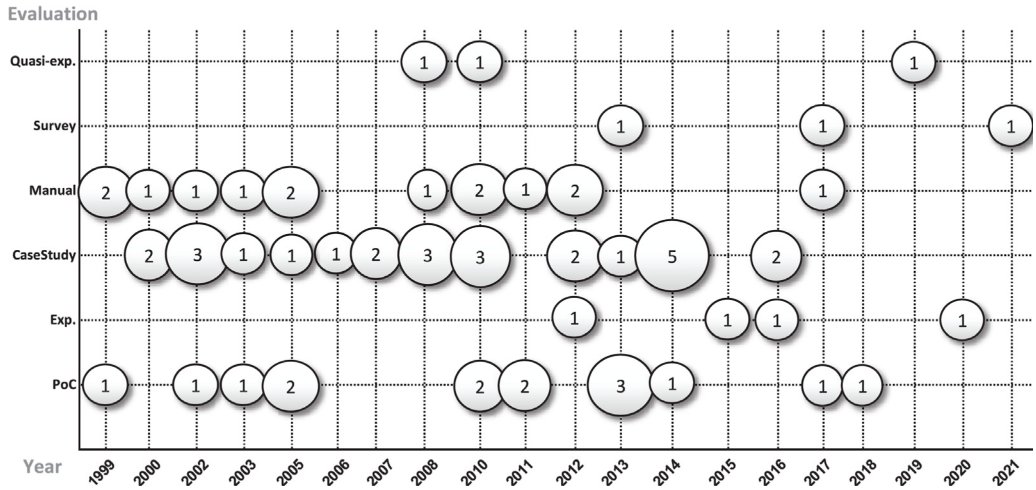

Fig. 7. Study evaluations by year.

Another aspect concerning decision-making is the use of metrics. When analyzing the metrics proposed or used by the studies, we noticed that this aspect of the approaches needs more formalization. As reported in Cavalcanti et al. (2011), the management of metrics is important for detailing SPL scoping at a technical level. Using the ISO/IEC for defining such metrics may aid to formalize this process as presented in Her et al. (2007). In this sense, the authors from Acher et al. (2012) plan, in future work, to adapt existing metrics to further characterize properties of the FMs generated by their approach. Another future work discussing the use of metrics is presented in Cavalcanti et al. (2011), where the authors plan to extend their metamodel for supporting metrics management. In Khtira et al. (2014), the use of metrics is also cited as future work. In this case, the authors plan to add new metrics related to cost estimation, which is related to the use of cost models.

Another important result obtained is illustrated in Fig. 4, showing that four concepts, namely, customer needs, market analysis, metrics definition, and cost models are related to each other. As discussed in this section, such concepts are also related to the decision-making process considering the domain and market aspects of the SPL. Moreover, works have handled these aspects in a more intra-organizational way (internal). However, the decision-making could benefit from considering inter-organization (external) decisions. These decisions may include partnerships with other organizations as well as outsourcing the SPL development. Such decisions could directly impact the cost–benefit of the SPL scoping process. Dealing with such aspects, however, would require the use of different techniques from the business field, such as business intelligence (Cheng et al., 2020), a technique not explored by the SPL scoping approaches.

Adaptation and Evaluation: Although different adaptions strategies are used in different approaches, this is still a challenge, as mentioned in da Silva et al. (2014). According to them, the SPL scoping process should consider the organizational aspects of the company. A similar problem is discussed in Knauber et al. (2000), where authors argued that in small companies, where project resources and data are short, the team would not devote their time for gathering additional data to perform Scoping. However, the authors also stated that their proposal, PuLSE-Eco, handles this problem by only requiring little effort on the side of the company.

As presented in Fig. 7, empirical evaluations have being performed in most studies. However, we also identified that several authors argued that their proposals needed additional evaluations to collect more evidence about their capabilities and maturity (Acher et al., 2012; Alam et al., 2017b; Balbino et al., 2011; Carbon et al., 2008; Cruz et al., 2013; da Silva et al., 2014; Estublier et al., 2010; Her et al., 2007; Ianzen et al., 2015; Kim et al., 2008; Mærsk-Møller and Jørgensen, 2010; Muller, 2011; Neto et al., 2016; Noor et al., 2008a; Schmid and Gacek, 2000; Villela et al., 2010; Sierszecki et al., 2014; Ojeda et al., 2019). The most common problem stated is related to the lack of sufficient evidence to consider their proposal to be reliable. Thus, many approaches still need more solid evaluations.

Other research opportunities: When analyzing the open research opportunities in the field, one aspect that was clear in the works is the lack of supporting tools. We conclude this as only a few tools were found in the proposals. For instance, DRAMA (Kim et al., 2008) is a process with an automation supporting tool. Other processes use tools for specific tasks, such as PuLSE-BEAT in John et al. (2006) and an untitled tool in da Silva et al. (2014). We notice, however, that tools are cited as future works of several studies (Gillain et al., 2012; deBaud and Schmid, 1999; Noor et al., 2008a; Acher et al., 2012; Alsawalqah et al., 2014; Jhon, 2010; Khtira et al., 2014; Shin Young Park and Soo Dong Kim, 2005). This may indicate that developing and evaluating such tools in different organizational aspects is still an open research opportunity in the field. Hence, researchers may guide their future works by trying to answer the following RQs: **How to develop a method-based tool to support the SPL scoping process? and which features of the tool are important for handling adaption in different organizations?**

A similar conclusion was achieved based on the domains for which the proposals were proposed to. To the best of our knowledge, SPL re-engineering is important as we understand that it is a common strategy used by organizations when migrating to an SPL context (Krüger et al., 2020). In this sense, only Noor et al. (2007, 2008a) have covered this aspect, leaving opportunities for further investigation in this area when considering SPL scoping. Thus, future works may try to answer the following RQs: **How can SPL re-engineering and SPL scoping be integrated? Which are the main benefits and drawbacks?**

Another aspect discussed among the works is the effort required to perform scoping-related activities as this process demands much effort from the companies. In this sense, authors of da Silva et al. (2014) argued that reduce the effort of the SPL scoping is an important challenged in the field. A similar conclusion is presented in Karimpour and Ruhe (2016), where authors stated that their approach effort should be reduced. Similar future work is stated by the authors in Ullah et al. (2010). One of the possible strategies for overcoming this challenge, as mentioned in da Silva et al. (2014), is the use of agile practices for requirements engineering. This strategy is being used in the RiPLE framework (Balbino et al., 2011). However, additional strategies may be used for handling the issue related to the demanding effort of scoping the SPL. These possibilities make a possible RQ emerges: **Which strategies and techniques may be used for reducing the effort of scoping an SPL?**

> ***Answering RQ3: What are the open research gaps and opportunities for new studies on the topic of SPL scoping?*** *As a result of answering our RQs, we identified several aspects for improvement in the SPL scoping field. Decision-making using costs models and metrics is still not well formalized, further investigation should be performed and even guidelines may be proposed. Adaptation of approaches and their evaluation are aspects that may also require further investigation. For the latter, evaluations considering different organizational aspects may collect evidence about how the adaptation may benefit companies. Considering open research opportunities, we concluded that supporting tools, SPL re-engineering, and strategies for reducing the SPL scoping effort may be investigated and even combined to increase the reliability and appeal of approaches in the field.*

4.4. Threats to validity

In this section, we discuss the main threats to validity related to our SLR and present how we mitigated them based on Wohlin et al. (2012), Ampatzoglou et al. (2019).

Conclusion validity: Researchers are usually not aware of their own bias during the analysis and classification of studies. This bias could negatively impact the results of the SLR. An additional threat is a possible inaccuracy during data extraction. Therefore, trying to mitigate both threats, two researchers independently performed the QA and data extraction from the studies. In case of divergences, an additional researcher was assigned for discussing divergent points. Lastly, the fishing and error rate problem may impact the conclusions of an SLR. We handle this problem by achieving our conclusions and answering our RQs after collecting and analyzing the results of all primary studies.

Internal validity: Publication bias refers to SPL scoping approaches that were not selected due to research results not being satisfactory. To mitigate this threat, the analysis of the studies on this field was performed considering a large sample of studies. An additional threat is the inclusion of studies considered to have low quality, thus, negatively impacting the RQs' answers. For mitigating this threat, we defined and applied QAs, classifying and guaranteeing that each study selected contained at least the minimum information required for answering our RQs. Another possible threat is the low number of studies analyzed, which may not represent the field. We mitigated this problem by selecting and analyzing a large set of studies. When compared to related reviews (see Section 5), we analyzed 30 additional studies.

Construct validity: Not retrieving studies due to their absence from a certain database is an additional major threat. We mitigated this problem by using different digital libraries which index a large number of conference proceedings and journals. Also, we conducted snowballing, which is a technique independent of DLs. To mitigate the issue related to the inclusion/exclusion of relevant works, two researchers applied independently a set of well-defined exclusion and inclusion criteria, as well as QA for qualification and classification of studies. In case of divergences, a third researcher would present its opinion. These strategies give our SLR reliability regarding the studies retrieved.

External validity: A possible external threat is related to the coherence of our results. In this sense, as we based our SLR protocol on well-defined SLR guidelines (Kitchenham et al., 2010) we believe that we achieved satisfactory results for answering our RQs. Considering the RQs' answers, we tried to perform a generic analysis of the data collected, considering both academic and practice points of view. An additional threat is related to the incorrect identification of the research opportunities discussed in RQ3. To mitigate this threat, we extracted the limitations and future work described by the authors in each study. After documenting this information, we analyzed whether the limitations of one study were covered by another or not. We also considered open research opportunities that were pointed out as future work by several studies, such as the development of supporting tools.

5. Related work

When considering secondary studies, we may find several in the SPL field. For the past decade, more than 60 relevant secondary studies, including systematic reviews were conducted in this field (Marimuthu and Chandrasekaran, 2017). Among these, we have reviews focusing on SPL requirements (Alves et al., 2010; Sepúlveda et al., 2016), SPL quality attributes (Montagud et al., 2012), SPL reengineering (Assunção et al., 2017; Laguna and Crespo, 2013), SPL testing (Machado et al., 2014; Engström and Runeson, 2011; da Mota Silveira Neto et al., 2011) as well as empirical SPL engineering (Chacon-Luna et al., 2020). There are also systematic studies comparing tools and technologies focused on domain analysis (Khurum and Gorschek, 2009; Lisboa et al., 2010). Despite their important contributions to the field, however, there are few studies in the literature comparing SPL scoping approaches. Table 11 summarizes these contributions: three studies identified as related to this SLR. In this section, we describe these studies in comparison with our SLR, discussing their focus in analyzing the state of the art for SPL scoping approaches.

Schmid (2000) reported a survey analyzing a set of scoping approaches. The survey analyzed technological approaches considering four dimensions: scoping tasks, the object of scoping, scoping product, and scoping process. The author used a framework to structure the scoping field with four goals: organize and structure the scoping field; analyze scoping approaches considering their benefits and drawbacks to provide an overview of the field; assist the selection of existing approaches; and assist the improvement of existing methods and development of new ones. The approaches analyzed were categorized into three scoping categories: product line scoping, domain scoping, and asset scoping. When conducting our SLR, we apply the same classification. This classification was combined with the goal of the approaches which could be for identification, evaluation, or optimization of scope.

The survey results may be used to better understand the SPL scoping activity, by organizing its results among the proposed dimensions it was possible to identify considerable differences among the proposals. Although we have similar goals, our SLR will focus on more technical aspects of the proposals, such as scoping types, metrics, and cost models.

John and Eisenbarth (2009) presented the results of another survey aiming to investigate some aspects in SPL scoping: the goal of the approaches; variability management; inputs and outputs of the approaches; roles; effort to perform scoping activities; and maturity and benefits of the approaches analyzed. To investigate these aspects the authors formulated three main goals: identify the connection between scoping and requirements engineering; identify the connection between scoping and architecture; how the approaches handle the production of quantifiable results. These two works Schmid (2000), John and Eisenbarth (2009), did not apply any systematic review protocol, thus their reviews

Table 11
Summary of related work.

Ref.	Year	Protocol	Studies Analyzed	Main Goal
Schmid (2000)	2000	Survey	13	Analyze technological proposals for SPL scoping considering scoping tasks, object of scoping, scoping products and scoping process.
John and Eisenbarth (2009)	2009	Survey	16	Investigate SPL scoping approaches among their goals, variability management; inputs and outputs among other characteristics.
de Moraes et al. (2009)	2009	SLR	13	Investigate SPL scoping approaches and identify scope definition techniques, analyzing their properties, in addition to strong points and drawbacks.
Our SLR	2021	SLR	58	Identify similarities and differences among SPL scoping approaches and processes, business aspects, conceptual characteristics and research opportunities.

are not repeatable. The SR presented in de Moraes et al. (2009), however, applied such a protocol allowing users to reuse its protocol.

The main goal of the SR presented in de Moraes et al. (2009) is to investigate the existent SPL scoping approaches to identify scope definition techniques, analyzing their properties, in addition to strong points and drawbacks. Their strongest contribution is to serve as a guide to practitioners to identify the more appropriate approach to be used in an industrial context. These goals are similar to our SLR goals. However, we intend to identify information that was not analyzed by them, such as re-factoring strategies, commonalities, and variabilities and domains. Despite these differences, as their work was conducted in 2009, our SLR included SPL scoping approaches published in the last ten years. Their work reported the result of an analysis performed in eleven primary studies, which were used to extract information used in our protocol, such as keywords, and terms for the search strings definition.

Another comparison work was presented in Lee et al. (2010) where the authors compared and analyzed three called "mainstream approaches". The authors presented a framework to perform such comparisons. The result was the extraction of their essential components and the creation of a unified approach.

6. Conclusion

To best manage their assets, companies may adopt different practices when establishing Software Product Line (SPL) processes. Configuring a process requires knowledge to deal with organizational contexts and application domains. Accordingly, a process instance is a result of the selection of specific SPL scoping activities, designed to deal with different reusable artifacts, team contexts, budgets, and other inputs. The absence of comparative studies in SPL scoping makes hard the scoping selection. Thus, professionals would be benefited from a systematic study that allows identifying those activities that best meet specific needs in SPL contexts.

Aiming at helping professionals along a selection process, in this work we presented the design and results of a Systematic Literature Review (SLR) on SPL scoping approaches. Likewise, we compared existing proposals in the area. The results were achieved by applying a well-defined protocol, searching in six different Digital Libraries, narrowing the results, and applying the snowballing technique to increase the set of studies. In total, we analyzed 58 distinct studies, identifying 41 different approaches. As a contribution to the research in SPL scoping, we present an updated state of the art, encompassing more than 10 years of advance in comparison to prior studies (Schmid, 2000; de Moraes et al., 2009; John and Eisenbarth, 2009), thus providing a comprehensive analysis on the SPL scoping topic.

Through four research questions, it was identified and discussed the similarities and differences of the approaches, establishing a generic process for SPL scoping and an SPL scoping concept map. The generic process may be used to guide the adoption of SPL scoping in different organizational scenarios, while the concept map presents the variability that may aid the adoption of this process. Thus, this work is also a contribution to carrying out a selection process for those elements that best meet specific needs in the area.

Along with this study, it was reasoned that the selection for one or another approach may depend on the reliability and maturity of the research. For this reason, it was analyzed how the SPL scoping proposals were evaluated. In a conclusion, most of the proposals focused on evaluating their performance in real projects, as 26 studies presented case studies. These studies are distributed in a bubble plot and, together with their evaluation by the approach in Table 10, allows professionals to direct their selection according to the research maturity as well.

Lastly, we observed open research opportunities regarding SPL scoping. The decision-making is still not formalized and additional investigation should focus on identifying or proposing guidelines for these decisions. Considering the evaluations, only a few approaches were evaluated in different organizational scenarios, showing the results of their adaptability. Additional opportunities identified include the possibility of establishing a set of features for SPL scoping supporting tools, as well as a protocol for evaluating these tools; how to integrate SPL scoping and SPL re-engineering for different scenarios; and how to reduce the effort of SPL scoping activities in organizations that possesses low project resources.

In conclusion, as the adoption of SPLs requires a high upfront investment, SPL scoping is paramount for achieving success. Our study showed that there exists several approaches for SPL scoping, describing how they are applied and how flexible they are to be customized to company scenarios. Finally, we could observe that, despite being a topic receiving attention for many years, with several types of empirical studies, there are still open gaps to be filled with new studies and opportunities to make SPL scoping more robust for supporting its use in practice. Therefore, research opportunities identified may motivate the increase of research in the SPL scoping field, making software companies aware of the several benefits and importance of this field for management and business aspects during SPL development.

CRediT authorship contribution statement

Luciano Marchezan: Conceptualization, Methodology, Investigation, Data curation, Writing – original draft, Writing – review & editing, Visualization. **Elder Rodrigues:** Conceptualization, Methodology, Validation, Project administration, Supervision, Writing – review & editing. **Wesley Klewerton Guez Assunção:** Conceptualization, Methodology, Validation, Supervision, Visualization, Writing – original draft, Writing – review & editing. **Maicon Bernardino:** Conceptualization, Methodology, Project administration, Supervision, Writing –

review & editing. **Fábio Paulo Basso:** Conceptualization, Methodology, Validation, Project administration, Supervision, Writing – review & editing. **João Carbonell:** Conceptualization, Methodology, Investigation, Data curation, Writing – original draft, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This research was partially funded by Carlos Chagas Filho Foundation for Supporting Research in the State of Rio de Janeiro (FAPERJ) PDR-10 Fellowship, grant: 202073/2020.

References

Alves, V., Niu, N., Alves, C., Valença, G., 2010. Requirements engineering for software product lines: A systematic literature review. Inf. Softw. Technol. 52 (8), 806–820.
Ampatzoglou, A., Bibi, S., Avgeriou, P., Verbeek, M., Chatzigeorgiou, A., 2019. Identifying, categorizing and mitigating threats to validity in software engineering secondary studies. Inf. Softw. Technol. 106, 201–230.
Apel, S., Batory, D., Kstner, C., Saake, G., 2013. Feature-Oriented Software Product Lines: Concepts and Implementation. Springer Publishing Company, Incorporated.
Assunção, W.K.G., Lopez-Herrejon, R.E., Linsbauer, L., Vergilio, S.R., Egyed, A., 2017. Reengineering legacy applications into software product lines: A systematic mapping. Empir. Softw. Eng. 22 (6), 2972–3016.
Balbino, M., Almeida, E., Meira, S., 2011. An agile scoping process for software product lines. In: International Conference on Software Engineering and Knowledge Engineering (SEKE), pp. 717–722.
Bockle, G., Clements, P., McGregor, J., Muthig, D., Schmid, K., 2004. Calculating ROI for software product lines. IEEE Softw. 21 (3), 23–31. http://dx.doi.org/10.1109/MS.2004.1293069.
Briggs, R.O., De Vreede, G.., Nunamaker, J.F., Tobey, D., 2001. Thinklets: achieving predictable, repeatable patterns of group interaction with group support systems (GSS). In: Proceedings of the 34th Annual Hawaii International Conference on System Sciences. p. 9. http://dx.doi.org/10.1109/HICSS.2001.926238.
Chacon-Luna, A.E., Gutiérrez, A.M., Galindo, J.A., Benavides, D., 2020. Empirical software product line engineering: A systematic literature review. Inf. Softw. Technol. 128, 106389. http://dx.doi.org/10.1016/j.infsof.2020.106389, URL https://www.sciencedirect.com/science/article/pii/S0950584920301555.
Cheng, C., Zhong, H., Cao, L., 2020. Facilitating speed of internationalization: The roles of business intelligence and organizational agility. J. Bus. Res. 110, 95–103. http://dx.doi.org/10.1016/j.jbusres.2020.01.003, URL https://www.sciencedirect.com/science/article/pii/S0148296320300035.
Clements, P.C., 2002. On the importance of product line scope. In: van Der Linden, F. (Ed.), Software Product-Family Engineering. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 70–78.
Clements, P., Northrop, L., 2002. Software Product Lines. Addison-Wesley,.
Cruzes, D.S., Dyba, T., 2011. Recommended steps for thematic synthesis in software engineering. In: 2011 International Symposium on Empirical Software Engineering and Measurement. pp. 275–284. http://dx.doi.org/10.1109/ESEM.2011.36.
Cruzes, D.S., Dybå, T., 2010. Synthesizing evidence in software engineering research. In: Proceedings of the 2010 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement. In: ESEM '10, Association for Computing Machinery, New York, NY, USA, http://dx.doi.org/10.1145/1852786.1852788, URL https://doi.org/10.1145/1852786.1852788.
da Mota Silveira Neto, P.A., Carmo Machado, I.d., McGregor, J.D., de Almeida, E.S., de Lemos Meira, S.R., 2011. A systematic mapping study of software product lines testing. Inf. Softw. Technol. 53 (5), 407–423. http://dx.doi.org/10.1016/j.infsof.2010.12.003.
Davis, S., 1989. From "future perfect": Mass customizing. Plan. Rev. 17 (2), 16–21, Available in: https://doi.org/10.1108/eb054249. Access in: 10 jun 2020.
de Moraes, M.B.S., de Almeida, E.S., Romero, S., 2009. A systematic review on software product lines scoping. In: 6th Experimental Software Engineering Latin American Workshop (ESELAW 2009), pp. 63.
deBaud, J., Schmid, K., 1999a. A systematic approach to derive the scope of software product lines. In: International Conference on Software Engineering. pp. 34–43. http://dx.doi.org/10.1145/302405.302409.
Engström, E., Runeson, P., 2011. Software product line testing - A systematic mapping study. Inf. Softw. Technol. 53 (1), 2–13. http://dx.doi.org/10.1016/j.infsof.2010.05.011.
Fink, A., 2003. The Survey Handbook. sage.
Garcés, L., Felizardo, K.R., Oliveira, L.B.R., Nakagawa, E.Y., 2017. An experience report on update of systematic literature reviews. In: International Conference on Software Engineering and Knowledge Engineering (SEKE), pp. 91–96.
John, I., Eisenbarth, M., 2009. A decade of scoping: A survey. In: 13th International Software Product Line Conference, SPLC '09, pp. 31–40.
Kane, M., Trochim, W.M., 2007. Concept Mapping for Planning and Evaluation.. Sage Publications, Inc.
Kang, K.C., Jaejoon Lee, Donohoe, P., 2002. Feature-oriented product line engineering. IEEE Softw. 19 (4), 58–65. http://dx.doi.org/10.1109/MS.2002.1020288.
Kang, K.C., Kim, S., Lee, J., Kim, K., Shin, E., Huh, M., 1998. FORM: A feature-oriented reuse method with domain-specific reference architectures. Ann. Softw. Eng. 5 (1), 143. http://dx.doi.org/10.1023/A:1018980625587.
Khurum, M., Gorschek, T., 2009. A systematic review of domain analysis solutions for product lines. J. Syst. Softw. 82 (12), 1982–2003. http://dx.doi.org/10.1016/j.jss.2009.06.048, URL https://www.sciencedirect.com/science/article/pii/S016412120900154X.
Kitchenham, B., Brereton, O.P., Budgen, D., Turner, M., Bailey, J., Linkman, S., 2009. Systematic literature reviews in software engineering - a systematic literature review. Inf. Softw. Technol. 51 (1), 7–15. http://dx.doi.org/10.1016/j.infsof.2008.09.009, Special Section - Most Cited Articles in 2002 and Regular Research Papers.
Kitchenham, B., Pretorius, R., Budgen, D., Brereton, O.P., Turner, M., Niazi, M., Linkman, S., 2010. Systematic literature reviews in software engineering – A tertiary study. Inf. Softw. Technol. 52 (8), 792–805. http://dx.doi.org/10.1016/j.infsof.2010.03.006.
Krüger, J., Mahmood, W., Berger, T., 2020. Promote-Pl: A round-trip engineering process model for adopting and evolving product lines. In: Proceedings of the 24th ACM Conference on Systems and Software Product Line: Volume a - Volume a. In: SPLC '20, Association for Computing Machinery, New York, NY, USA, pp. 1–12. http://dx.doi.org/10.1145/3382025.3414970.
Laguna, M.A., Crespo, Y., 2013. A systematic mapping study on software product line evolution: From legacy system reengineering to product line refactoring. Sci. Comput. Program. 78 (8), 1010–1034.
Lee, J., Kang, S., Lee, D., 2010. A comparison of software product line scoping approaches. Int. J. Softw. Eng. Knowl. Eng. 20 (05), 637–663. http://dx.doi.org/10.1142/S021819401000489X.
Lisboa, L.B., Garcia, V.C., Lucrédio, D., de Almeida, E.S., de Lemos Meira, S.R., de Mattos Fortes, R.P., 2010. A systematic review of domain analysis tools. Inf. Softw. Technol. 52 (1), 1–13. http://dx.doi.org/10.1016/j.infsof.2009.05.001, URL https://www.sciencedirect.com/science/article/pii/S0950584909000834.
Machado, I.D.C., Mcgregor, J.D., Cavalcanti, Y.a.C., de Almeida, E.S., 2014. On strategies for testing software product lines: A systematic literature review. Inf. Softw. Technol. 56 (10), 1183–1199. http://dx.doi.org/10.1016/j.infsof.2014.04.002.
Mærsk-Møller, H.M., Jørgensen, B.N., 2010a. Experiences initiating software product line engineering in small teams with pulse. In: IASTED International Conference on Software Engineering, Vol. 7. p. 9. http://dx.doi.org/10.2316/P.2010.677-042.
Marchezan, L., de Macedo Rodrigues, E., Assunção, W.K.G., Bernardino, M., Basso, F.P., Carbonell, J., 2021. Software Product Line Scoping: A Systematic Literature Review. Zenodo, http://dx.doi.org/10.5281/zenodo.5776419.
Marimuthu, C., Chandrasekaran, K., 2017. Systematic studies in software product lines: A tertiary study. In: 21st International Systems and Software Product Line Conference - Volume a. In: SPLC '17, Association for Computing Machinery, New York, NY, USA, pp. 143–152. http://dx.doi.org/10.1145/3106195.3106212.
Matzler, K., Hinterhuber, H.H., 1998. How to make product development projects more successful by integrating kano's model of customer satisfaction into quality function deployment. Technovation 18 (1), 25–38. http://dx.doi.org/10.1016/S0166-4972(97)00072-2.
Mendes, E., Felizardo, K., Wohlin, C., Kalinowski, M., Search strategy to update systematic literature reviews in software engineering. In: 2019 45th Euromicro Conference on Software Engineering and Advanced Applications (SEAA), pp. 355–362.
Montagud, S., Abrahão, S., Insfran, E., 2012. A systematic review of quality attributes and measures for software product lines. Softw. Qual. J. 20 (3–4), 425–486.
Pohl, K., Böckle, G., van Der Linden, F., 2005. Software Product Line Engineering: Foundations, Principles and Techniques. Springer Science & Business Media.
Runeson, P., Host, M., Rainer, A., Regnell, B., 2012. Case Study Research in Software Engineering: Guidelines and Examples. John Wiley & Sons.
Satty, T.L., et al., 1980. The Analytic Hierarchy Process. McGraw-Hill, New York.
Schmid, K., 2000. Scoping software product lines: An analysis of an emerging technology. In: First Conference on Software Product Lines : Experience and Research Directions: Experience and Research Directions. Kluwer Academic Publishers, Norwell, MA, USA, pp. 513–532.

Schmid, K., 2002a. A comprehensive product line scoping approach and its validation. In: 24th International Conference on Software Engineering. In: ICSE '02, ACM, New York, NY, USA, pp. 593–603.
Seidl, C., Winkelmann, T., Schaefer, I., 2016. A software product line of feature modeling notations and cross-tree constraint languages. In: Oberweis, A., Reussner, R. (Eds.), Modellierung 2016. Gesellschaft für Informatik e.V., Bonn, pp. 157–172.
Sepúlveda, S., Cravero, A., Cachero, C., 2016. Requirements modeling languages for software product lines: A systematic literature review. Inf. Softw. Technol. 69, 16–36.
Tonella, P., 2004. Formal concept analysis in software engineering. In: Proceedings. 26th International Conference on Software Engineering. pp. 743–744. http://dx.doi.org/10.1109/ICSE.2004.1317515.
van Der Linden, F., Schmid, K., Rommes, E., 2007. Software Product Lines in Action: The Best Industrial Practice in Product Line Engineering, 2007th Springer Science & Business Media.
Wohlin, C., 2014. Guidelines for snowballing in systematic literature studies and a replication in software engineering. In: 18th International Conference on Evaluation and Assessment in Software Engineering. In: EASE '14, ACM, New York, NY, USA, pp. 38:1–38:10. http://dx.doi.org/10.1145/2601248.2601268.
Wohlin, C., Runeson, P., Höst, M., Ohlsson, M.C., Regnell, B., Wesslén, A., 2012. Experimentation in Software Engineering, Vol. 1. Springer Science & Business Media.

Primary Sources

Abbas, N., Andersson, J., 2013. Architectural reasoning for dynamic software product lines. In: 17th International Software Product Line Conference Co-Located Workshops. In: SPLC '13 Workshops, ACM, New York, NY, USA, pp. 117–124. http://dx.doi.org/10.1145/2499777.2500718.
Acher, M., Cleve, A., Perrouin, G., Heymans, P., Vanbeneden, C., Collet, P., Lahire, P., 2012. On extracting feature models from product descriptions. In: Sixth International Workshop on Variability Modeling of Software-Intensive Systems. In: VaMoS '12, ACM, New York, NY, USA, pp. 45–54. http://dx.doi.org/10.1145/2110147.2110153.
Alam, M.M., Khan, A.I., Zafar, A., 2017a. An empirical study of the improved SPLD framework using expert opinion technique. IJEACS Int. J. Eng. Appl. Comput. Sci. 2 (03).
Alam, M.M., Khan, A.I., Zafar, A., 2017b. A secure framework for software product line development. Int. J. Comput. Appl. 975, 8887.
Alsawalqah, H., Kang, S., Lee, D., 2013. A method for software product platform design based on features. In: Proceedings of the 17th International Software Product Line Conference Co-Located Workshops. In: SPLC '13 Workshops, Association for Computing Machinery, New York, NY, USA, pp. 18–25. http://dx.doi.org/10.1145/2499777.2500723.
Alsawalqah, H.I., Kang, S., Lee, J., 2014. A method to optimize the scope of a software product platform based on end-user features. J. Syst. Softw. 98, 79–106. http://dx.doi.org/10.1016/j.jss.2014.08.034.
Balbino, M., Almeida, E., Meira, S., 2011. An agile scoping process for software product lines. In: International Conference on Software Engineering and Knowledge Engineering (SEKE), pp. 717–722.
Bartholdt, J., Becker, D., 2012. Scope extension of an existing product line. In: 16th International Software Product Line Conference - Vol. 1. In: SPLC '12, ACM, New York, NY, USA, pp. 275–282. http://dx.doi.org/10.1145/2362536.2362573.
Bayer, J., Flege, O., Knauber, P., Laqua, R., Muthig, D., Schmid, K., Widen, T., DeBaud, J.-M., 1999. Pulse: A methodology to develop software product lines. In: Proceedings of the 1999 Symposium on Software Reusability. pp. 122–131.
Bayer, J., Gacek, C., Muthig, D., Widen, T., 2000. PuLSE-I: Deriving instances from a product line infrastructure. In: Proceedings Seventh IEEE International Conference and Workshop on the Engineering of Computer-Based Systems (ECBS 2000). pp. 237–245. http://dx.doi.org/10.1109/ECBS.2000.839882.
Bayer, J., Muthig, D., Widen, T., 2000. Customizable domain analysis. In: First International Symposium on Generative and Component-Based Software Engineering. In: GCSE '99, Springer-Verlag, London, UK, UK, pp. 178–194.
Carbon, R., Knodel, J., Muthig, D., Meier, G., 2008. Providing feedback from application to family engineering - the product line planning game at the testo AG. In: 12th International Software Product Line Conference. pp. 180–189. http://dx.doi.org/10.1109/SPLC.2008.21.
Cavalcanti, Y.C., do Carmo Machado, I., da Mota, P.A., Neto, S., Lobato, L.L., de Almeida, E.S., de Lemos Meira, S.R., 2011. Towards metamodel support for variability and traceability in software product lines. In: 5th Workshop on Variability Modeling of Software-Intensive Systems. In: VaMoS '11, ACM, New York, NY, USA, pp. 49–57. http://dx.doi.org/10.1145/1944892.1944898.
Cruz, J., Neto, P.S., Britto, R., Rabelo, R., Ayala, W., Soares, T., Mota, M., 2013. Toward a hybrid approach to generate software product line portfolios. In: IEEE Congress on Evolutionary Computation. pp. 2229–2236. http://dx.doi.org/10.1109/CEC.2013.6557834.
deBaud, J., Schmid, K., 1999. A systematic approach to derive the scope of software product lines. In: International Conference on Software Engineering. pp. 34–43. http://dx.doi.org/10.1145/302405.302409.
Elsner, C., Schwanninger, C., Schroder-Preikschat, W., Lohmann, D., 2010. Multi-level product line customization. In: 9th International Conference on New Trends in Software Methodologies, Tools and Techniques, SoMeT_10. Yokohama, Japan, Japan Society for the Promotion of Science (JSPS); SANGIKYO Co..
Estublier, J., Dieng, I.A., Leveque, T., 2010. Software product line evolution: The selecta system. In: 2010 ICSE Workshop on Product Line Approaches in Software Engineering. In: PLEASE '10, ACM, New York, NY, USA, pp. 32–39. http://dx.doi.org/10.1145/1808937.1808942.
Geppert, B., Weiss, D., 2003. Goal-oriented assessment of product-line domains. In: Proceedings. 5th International Workshop on Enterprise Networking and Computing in Healthcare Industry (IEEE Cat. No.03EX717). pp. 180–188. http://dx.doi.org/10.1109/METRIC.2003.1232466.
Gillain, J., Faulkner, S., Heymans, P., Jureta, I., Snoeck, M., 2012. Product portfolio scope optimization based on features and goals. In: 16th International Software Product Line Conference, Vol. 1. ACM Digital Library; New York, pp. 161–170.
Haidar, H., Kolp, M., Wautelet, Y., 2017. Agile product line engineering: The agifpl method. In: ICSOFT. pp. 275–285.
Her, J.S., Kim, J.H., Oh, S.H., Rhew, S.Y., Kim, S.D., 2007. A framework for evaluating reusability of core asset in product line engineering. Inf. Softw. Technol. 49 (7), 740–760. http://dx.doi.org/10.1016/j.infsof.2006.08.008.
Ianzen, A., Mantovani Fontana, R., Paludo, M.A., Malucelli, A., Reinehr, S., 2015. Scoping automation in software product lines. In: 17th International Conference on Enterprise Information Systems - Vol. 2. In: ICEIS 2015, SCITEPRESS - Science and Technology Publications, Lda, Portugal, pp. 82–91. http://dx.doi.org/10.5220/0005372400820091.
Jhon, I., 2010. Using documentation for product line scoping. IEEE Softw. 27 (3), 42–47. http://dx.doi.org/10.1109/MS.2010.34.
John, I., Knodel, J., Lehner, T., Muthig, D., 2006. A practical guide to product line scoping. In: 10th International Software Product Line Conference. pp. 3–12. http://dx.doi.org/10.1109/SPLINE.2006.1691572.
Kang, K.C., Donohoe, P., Koh, E., Lee, J., Lee, K., 2002. Using a marketing and product plan as a key driver for product line asset development. In: International Conference on Software Product Lines. Springer, pp. 366–382.
Karimpour, R., Ruhe, G., 2016. Evolutionary robust optimization for software product line scoping: An explorative study. Comput. Lang. Syst. Struct. 47, 189–210. http://dx.doi.org/10.1016/j.cl.2016.07.007.
Khtira, A., Benlarabi, A., El Asri, B., 2014. Towards a requirement-based approach to support early decisions in software product line engineering. In: Second World Conference on Complex Systems (WCCS). pp. 152–157. http://dx.doi.org/10.1109/ICoCS.2014.7060993.
Kiani, A.A., Hafeez, Y., Aqib, M., Ali, S., Javed, J., Anwar, N., 2020. Extending the scrum to introduce the concept of systematic reusability. In: 2020 3rd International Conference on Computing, Mathematics and Engineering Technologies (ICoMET). pp. 1–6. http://dx.doi.org/10.1109/iCoMET48670.2020.9074066.
Kiani, A.A., Hafeez, Y., Imran, M., Ali, S., 2021. A dynamic variability management approach working with agile product line engineering practices for reusing features. J. Supercomput. 1–42.
Kim, S.D., Chang, S.H., La, H.J., 2005. Traceability map: foundations to automate for product line engineering. In: Third ACIS Int'L Conference on Software Engineering Research, Management and Applications (SERA'05). pp. 340–347. http://dx.doi.org/10.1109/SERA.2005.66.
Kim, J., Park, S., Sugumaran, V., 2008. DRAMA: A framework for domain requirements analysis and modeling architectures in software product lines. J. Syst. Softw. 81 (1), 37–55. http://dx.doi.org/10.1016/j.jss.2007.04.011.
Kim, M., Yang, H., Park, S., 2003. A domain analysis method for software product lines based on scenarios, goals and features. In: Tenth Asia-Pacific Software Engineering Conference, 2003.. pp. 126–135. http://dx.doi.org/10.1109/APSEC.2003.1254365.
Kishi, T., Noda, N., Katayama, T., 2002. A method for product line scoping based on a decision-making framework. In: Chastek, G.J. (Ed.), Software Product Lines. Springer Berlin Heidelberg, pp. 348–365.
Knauber, P., Muthig, D., Schmid, K., Widen, T., 2000. Applying product line concepts in small and medium-sized companies. IEEE Softw. 17 (05), 88–95. http://dx.doi.org/10.1109/52.877873.
Koziolek, H., Goldschmidt, T., de Gooijer, T., Domis, D., Sehestedt, S., Gamer, T., Aleksy, M., 2016. Assessing software product line potential: An exploratory industrial case study. Empir. Softw. Eng. 21 (2), 411–448.
Lobato, L.L., da Mota Silveira Neto, P.A., do Carmo Machado, I., de Alemida, E.S., de Lemos Meira, S.R., 2012. Risk management in software product lines: An industrial case study. In: International Conference on Software and System Process (ICSSP). pp. 180–189. http://dx.doi.org/10.1109/ICSSP.2012.6225963.
Mærsk-Møller, H.M., Jørgensen, B.N., 2010. Experiences initiating software product line engineering in small teams with pulse. In: IASTED International Conference on Software Engineering, Vol. 7. p. 9. http://dx.doi.org/10.2316/P.2010.677-042.

Muller, J., 2011. Value-based portfolio optimization for software product lines. In: 15th International Software Product Line Conference. pp. 15–24. http://dx.doi.org/10.1109/SPLC.2011.18.

Neto, P.A.S., Britto, R., Rabêlo, R.A.L., Cruz, J.J.A., Lira, W.A.L., 2016. A hybrid approach to suggest software product line portfolios. Appl. Soft Comput. 49, 1243–1255. http://dx.doi.org/10.1016/j.asoc.2016.08.024.

Nöbauer, M., Seyff, N., Groher, I., 2014. Similarity analysis within product line scoping: An evaluation of a semi-automatic approach. In: Jarke, M., Mylopoulos, J., Quix, C., Rolland, C., Manolopoulos, Y., Mouratidis, H., Horkoff, J. (Eds.), Advanced Information Systems Engineering. Springer International Publishing, Cham, pp. 165–179.

Noor, M.A., Grünbacher, P., Briggs, R.O., 2007. A collaborative approach for product line scoping: A case study in collaboration engineering. In: 25th Conference on IASTED International Multi-Conference: Software Engineering. In: SE'07, ACTA Press, Anaheim, CA, USA, pp. 216–223.

Noor, M.A., Grünbacher, P., Hoyer, C., 2008a. A collaborative method for reuse potential assessment in reengineering-based product line adoption. In: Meyer, B., Nawrocki, J.R., Walter, B. (Eds.), Balancing Agility and Formalism in Software Engineering. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 69–83.

Noor, M.A., Rabiser, R., Grünbacher, P., 2008b. Agile product line planning: A collaborative approach and a case study. J. Syst. Softw. 81 (6), 868–882. http://dx.doi.org/10.1016/j.jss.2007.10.028, Agile Product Line Engineering, URL https://www.sciencedirect.com/science/article/pii/S0164121207002531.

Ojeda, M.C.C., Alegría, J.A.H., Rodriguez, F.J.A., Melenje, P.H.R., 2018. A collaborative method for a tangible software product line scoping. In: International Conference on Applied Informatics: Workshops (ICAIW). pp. 1–6. http://dx.doi.org/10.1109/ICAIW.2018.8554999.

Ojeda, M.C.C., Rodriguez, F.A., Collazos, C.A., 2019. Identifying collaborative aspects during software product lines scoping. In: 23rd International Systems and Software Product Line Conference - Vol. B. In: SPLC '19, ACM, New York, NY, USA, pp. 98–105. http://dx.doi.org/10.1145/3307630.3342420.

O'Leary, P., de Almeida, E.S., Richardson, I., 2012. The pro-PD process model for product derivation within software product lines. Inf. Softw. Technol. 54 (9), 1014–1028. http://dx.doi.org/10.1016/j.infsof.2012.03.008.

Oliveira, R.P.d., Insfran, E., Abrahão, S., Gonzalez-Huerta, J., Blanes, D., Cohen, S., Almeida, E.S.d., 2013. A feature-driven requirements engineering approach for software product lines. In: 2013 VII Brazilian Symposium on Software Components, Architectures and Reuse. pp. 1–10. http://dx.doi.org/10.1109/SBCARS.2013.11.

Ramachandran, M., Allen, P., 2005. Commonality and variability analysis in industrial practice for product line improvement. Softw. Process: Improv. Pract. 10 (1), 31–40. http://dx.doi.org/10.1002/spip.212.

Schmid, K., 2002a. A comprehensive product line scoping approach and its validation. In: 24th International Conference on Software Engineering. In: ICSE '02, ACM, New York, NY, USA, pp. 593–603.

Schmid, K., 2002b. The product line mapping approach to defining and structuring product portfolios. In: Proceedings IEEE Joint International Conference on Requirements Engineering. pp. 219–226. http://dx.doi.org/10.1109/ICRE.2002.1048532.

Schmid, K., Gacek, C., 2000. Implementation issues in product line scoping. In: Frakes, W.B. (Ed.), Software Reuse: Advances in Software Reusability. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 170–189.

Schmid, K., John, I., Kolb, R., Meier, G., 2005. Introducing the PuLSE approach to an embedded system population at testo AG. In: 27th International Conference on Software Engineering (ICSE). pp. 544–552. http://dx.doi.org/10.1109/ICSE.2005.1553600.

Shin Young Park, Soo Dong Kim, 2005. A systematic method for scoping core assets in product line engineering. In: 12th Asia-Pacific Software Engineering Conference (APSEC'05), Vol. 1. pp. 1–8. http://dx.doi.org/10.1109/APSEC.2005.18.

Sierszecki, K., Steffens, M., Hojrup, H.H., Savolainen, J., Beuche, D., 2014. Extending variability management to the next level. In: 18th International Software Product Line Conference - Vol. 1. In: SPLC '14, ACM, New York, NY, USA, pp. 320–329. http://dx.doi.org/10.1145/2648511.2648548.

da Silva, I.F., da Mota Silveira Neto, P.A., O'Leary, P., de Almeida, E.S., de Lemos Meira, S.R., 2014. Software product line scoping and requirements engineering in a small and medium-sized enterprise: An industrial case study. J. Syst. Softw. 88, 189–206. http://dx.doi.org/10.1016/j.jss.2013.10.040.

Ullah, M.I., Ruhe, G., Garousi, V., 2010. Decision support for moving from a single product to a product portfolio in evolving software systems. J. Syst. Softw. 83 (12), 2496–2512. http://dx.doi.org/10.1016/j.jss.2010.07.049, TAIC PART 2009 - Testing: Academic & Industrial Conference - Practice And Research Techniques.

Vale, T., Cabral, B., Alvim, L., Soares, L., Santos, A., Machado, I., Souza, I., Freitas, I., Almeida, E., 2014. SPLICE: A lightweight software product line development process for small and medium size projects. In: 2014 Eighth Brazilian Symposium on Software Components, Architectures and Reuse. pp. 42–52. http://dx.doi.org/10.1109/SBCARS.2014.11.

Villela, K., Dörr, J., John, I., 2010. Evaluation of a method for proactively managing the evolving scope of a software product line. In: Wieringa, R., Persson, A. (Eds.), Requirements Engineering: Foundation for Software Quality. Springer Berlin Heidelberg, pp. 113–127.

Luciano Marchezan is currently a Ph.D. student and Junior Researcher at the Institute for Software Systems Engineering (ISSE) at the Johannes Kepler University Linz, Austria. He received his M.Sc in Software Engineering (2019) from the Federal University of Pampa (Unipampa), Brazil. His research interests include Empirical Software Engineering, Software Reengineering, Software Reuse, Software Product Lines, and Model-Driven Engineering.

Elder de Macedo Rodrigues is currently an Adjunct Professor at the Federal University of Pampa (Unipampa), Brazil. His research interests include Software Testing, Software Product Lines, Software Reuse, and Experimental Software Engineering.

Wesley Klewerton Guez Assunção is currently a Post-Doctoral researcher at Pontifical Catholic University of Rio de Janeiro - Brazil, and a University Assistant at Johannes Kepler University Linz - Austria. Wesley received his M.Sc. in Informatics (2012) and Ph.D. in Computer Science (2017) from the Federal University of Paraná - Brazil. His areas of interest are Software Modernization, Variability Management, Collaborative Engineering, and Software Testing. He published research papers in conferences like ICSME, SANER, MSR, EASE, SPLC, as well as in journals such as EMSE, IST, and JSS. Wesley has served as a reviewer at many conferences and journals. Further information: https://wesleyklewerton.github.io/

Maicon Bernardino da Silveira is currently an Adjunct Professor at the Federal University of Pampa (Unipampa), Brazil. He is associated in the undergraduate program in Computer Science and Software Engineering, and an assessor of conditions of teaching by the Ministry of Education (MEC/INEP). Experience with Project Management, Performance Testing, Model-Based Testing, Domain-Specific Language, Software Product Lines, Game Programming, Database Administrator, Software Testing, and Application Developer in a wide variety of business applications. Member of the Association for Computing Machinery (ACM).

Fábio Paulo Basso is currently an Adjunct Professor at the Federal University of Pampa (Unipampa), Brazil. His research interests include Software Architecture and Coopetition Business Models performed through services, in particular when applied to software factory contexts.

João Carbonell is a Master Student at the Federal University of Pampa (Unipampa), Brazil. His research interests include Model-drivel Engineering, Software Testing, Domain-Specific Languages, Empirical Software Engineering, and Software Reuse.