Information and Software Technology 52 (2010) 806–820

Contents lists available at ScienceDirect

Information and Software Technology

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

Requirements engineering for software product lines: A systematic literature review

Vander Alves [a,*], Nan Niu [b], Carina Alves [c], George Valença [c]

[a] Departamento de Ciência da Computação, Universidade de Brasília, Campus Darcy Ribeiro, 70910-900 Brasília, Brazil
[b] Department of Computer Science and Engineering, Box 9637, Mississippi State University, 39762 MS, USA
[c] Centro de Informática, Universidade Federal de Pernambuco, Av. Professor Luís Freire s/n, Cidade Universitária, 50740-540 Recife, Brazil

ARTICLE INFO

Article history:
Received 4 October 2009
Received in revised form 30 March 2010
Accepted 31 March 2010
Available online 10 April 2010

Keywords:
Software product lines
Requirements engineering
Systematic literature review

ABSTRACT

Context: Software product line engineering (SPLE) is a growing area showing promising results in research and practice. In order to foster its further development and acceptance in industry, it is necessary to assess the quality of the research so that proper evidence for adoption and validity are ensured. This holds in particular for requirements engineering (RE) within SPLE, where a growing number of approaches have been proposed.
Objective: This paper focuses on RE within SPLE and has the following goals: assess research quality, synthesize evidence to suggest important implications for practice, and identify research trends, open problems, and areas for improvement.
Method: A systematic literature review was conducted with three research questions and assessed 49 studies, dated from 1990 to 2009.
Results: The evidence for adoption of the methods is not mature, given the primary focus on toy examples. The proposed approaches still have serious limitations in terms of rigor, credibility, and validity of their findings. Additionally, most approaches still lack tool support addressing the heterogeneity and mostly textual nature of requirements formats as well as address only the proactive SPLE adoption strategy.
Conclusions: Further empirical studies should be performed with sufficient rigor to enhance the body of evidence in RE within SPLE. In this context, there is a clear need for conducting studies comparing alternative methods. In order to address scalability and popularization of the approaches, future research should be invested in tool support and in addressing combined SPLE adoption strategies.

Contents

* Corresponding author. Tel.: +55 61 8157 1464; fax: +55 61 3273 3589.
E-mail addresses: valves@unb.br (V. Alves), niu@cse.msstate.edu (N. Niu), cfa@cin.ufpe.br (C. Alves), gavs@cin.ufpe.br (G. Valença).

0950-5849/$ - see front matter
doi:10.1016/j.infsof.2010.03.014

1. Introduction

A software product line (SPL) can be defined as "a set of software-intensive systems sharing a common, managed set of features that satisfy the specific needs of a particular market segment or mission and that are developed from a common set of core assets in a prescribed way" [1]. Software product line engineering (SPLE) exploits the common properties of software systems to increase the level of reuse. The goal of SPLE is to support the systematic development of a family of software systems by identifying and managing their similarities and variations. The successful adoption of SPLE requires a profound organizational mind shift. The whole software engineering process is affected from requirements to maintenance and evolution activities.

In particular, it is argued that the nature of a SPL is to manage the commonality and variability of products by means of a "requirements engineering (RE) — change management" process [1]. RE is concerned with the real-world goals for, functions of, and constraints on software systems [2]. Compared with RE for a single custom-built system, RE for a family of software-intensive systems focuses more on systematic reuse, not only from the technical perspective, but from the organizational, marketing, and process perspectives as well [3]. Specifically, the following aspects must be considered in RE for SPLs:

- Scoping, commonality, variability (SCV) analysis [4] aims to explicate the definition of a domain and the inclusion and exclusion criteria of the domain. Identifying and managing the common and variable requirements are key to the SPL.
- Asset management is unique to product line engineering in that designated personnel, e.g., domain engineers, must design, build, and evolve the reusable set of core assets, so that application engineers can effectively reuse the asset base to derive individual products.
- The stakeholders will include not only a single application's customers, users, developers, testers, maintainers, etc., but also the parties who are involved in asset development and management.
- Marketing plays an important role in launching a SPL or transitioning from single-system development to product line engineering, so factors like reuse ratio and return on investment (ROI) need to be considered when planning the product line requirements.
- Organizational and process changes are expected in product line engineering, e.g., whether the core assets are developed in a centralized or distributed manner, whether the organization chooses proactive, reactive, or extractive adoption strategy [5].
- The techniques, most notably the modeling techniques, are different from single-system RE. Single-system requirements are often modeled from the *use* perspective, e.g., use cases, sequence diagrams, etc., SPL requirements are modeled from the *reuse* perspective by explicitly representing the commonality and variability information, e.g., feature models, orthogonal variability models, etc.

Many important decisions are made during the requirements engineering phase for developing a software product line. For instance, after the SPL's scope definition, the domain engineer determines the common requirements that are shared among family members and the optional ones that are unique to a specific product. Managing SPL requirements is non-trivial because they reflect stakeholders' diverse perspectives, have complex configuration dependencies (e.g., requires, excludes), and are expressed in various forms (e.g., textual, goals) and at different granularities (e.g., features, qualities).

The underlying research and practice area of SPLE has reported positive results for more than a decade, including improved quality, reduced time-to-market and costs in different domains. Nevertheless, its further development requires assessing the quality of the conducted research so that proper evidence for adoption and validity are ensured. Indeed, as reported to occur in software engineering [6], studies conducted are often not clearly reported in a way that allows researchers to obtain scientific conclusions and identify derived lines of research. The frequent lack of reliable evidence regarding the effectiveness of approaches proposed in the literature prevents their wide adoption by practitioners. It is often not clear how much confidence others can place in the proposed elements of a new method nor whether the results obtained with a particular method can be compared with similar approaches. These issues can also occur in RE for SPLs, where a growing number of approaches have been proposed. Thus, a rigorous assessment of RE within SPLE is necessary to increase their acceptance in practice.

Researchers have recently conducted systematic reviews focusing on different aspects of RE for SPLs, including domain analysis solutions for SPLs [7], domain analysis tools [8], requirements frameworks for SPLs [9], and variability management in SPL [10]. As will be discussed in more detail in Section 5, our research questions are different from those posed in the above literature reviews. In particular, we are interested in how SPL RE activities (elicitation, modeling, scoping, commonality and variability analysis, evolution, etc.) are supported by existing approaches, and which SPL adoption strategies (proactive, extractive, and reactive) are followed. None of the related reviews adequately covers these issues. In addition to focusing on the practitioner concerns of the availability of tool support raised in [7,8], we also consider research-oriented issues like the credibility of evidence and the rigor of reporting. Finally, our review's time frame is the broadest among the related literature reviews mentioned above.

Accordingly, this paper presents a systematic literature review (SLR) of RE for SPLs from 1990 to 2009. SLR is gaining popularity in software engineering as it provides a methodologically rigorous review of studies and it is the main method of evidence synthesis [6]. A systematic review of studies can help researchers and practitioners to locate, assess, and aggregate the outcomes from relevant studies, thus providing a balanced and objective summary of the relevant evidence. In particular, the goals of this study are the following:

- Assess the quality of the research in RE for SPLs.
- Synthesize evidence to suggest important implications for practice.
- Identify research trends, open problems, and areas for improvement.

The results of our systematic review reveal that, although there are some empirical studies assessing the effectiveness of methods, the evidence for adoption of the methods is not mature, given the primary focus on toy examples. The proposed approaches still have serious limitations in terms of rigor, credibility, and validity of their findings. Additionally, most approaches still lack tool support addressing the heterogeneity and mostly textual nature of requirements formats as well as address only the proactive SPLE adoption strategy. Therefore, further empirical studies should be performed with sufficient rigor to enhance the body of evidence in RE within SPLE. In this context, there is a clear need for conducting studies comparing alternative methods. In order to address scalability and popularization of SPL RE approaches, future research should be invested in developing reliable tool support, addressing combined SPLE adoption strategies, and providing objective guidance to adopt SPL RE methods.

The remainder of the paper is organized as follows. Section 2 explains our literature review methodology. Section 3 presents and analyzes the results to answer our research questions. Section 4 provides a thorough discussion about our findings. Related work is described in Sections 5 and 6 presents concluding remarks.

2. Method

Informed by the established method of systematic literature review [11], we undertook the review of studies in distinct stages: specifying research questions, developing and validating review protocol, searching relevant studies, assessing quality, analyzing data, and synthesis. In the rest of this section, we describe the details of these stages and the methods used.

2.1. Research questions

This study aims at addressing the following research questions:

RQ1. What SPL RE methods and/or tools are available to the practitioners?
RQ2. How much evidence is available to adopt the proposed methods?
RQ3. What are the limitations of current SPL RE methods?

Answering RQ1 allows us to identify the method's name, its goals, and whether tool support is available to practitioners. In order to provide a fuller context for understanding and applying a particular method, we distill RQ1 by considering a number of issues:

RQ1.1. What types of requirements artifacts does the method handle?
Examining the formats (e.g., textual, use cases, feature models), in terms of input and output of the method, helps the organization to align its current practices and to make future adoption decisions.
RQ1.2. Which RE activities does the method support?
According to [12], the major activities in RE include: plan and elicit, model and analyze, negotiate and agree, and realize and evolve requirements. Addressing this issue guides the organization in identifying to which RE phases the proposed method could be applied.
RQ1.3. Which SPL adoption strategies does the method follow?
This sub-question is designed to examine whether the method follows the proactive, extractive, and reactive adoption strategies [5]. With the proactive strategy, an organization makes an upfront investment to develop reusable assets for the SPL and products are developed using the assets. The extractive approach reuses one or more existing software products for the SPL's initial baseline. In the reactive strategy, an existing SPL is extended to encompass the requirements for a new product.

RQ2 evaluates the evidence level of the proposed method. The results are critical for researchers to identify new topics for empirical studies, and for practitioners to assess the maturity of a particular method or tool. Kitchenham classified five levels of study design in software engineering based on the evidence hierarchy suggested from medical research [11]. Since confounding variables abound in human-centric software development, researchers rarely design and undertake randomized controlled trials (RCTs), as noted in [11]. To make our assessment more practical, we revise Kitchenham's classification and use the following hierarchy (from weakest to strongest) in our study:

1. No evidence.
2. Evidence obtained from demonstration or working out toy examples.
3. Evidence obtained from expert opinions or observations.
4. Evidence obtained from academic studies, e.g., controlled lab experiments.
5. Evidence obtained from industrial studies, e.g., causal case studies.
6. Evidence obtained from industrial practice.

In particular, we add "no evidence" and "demonstration or toy examples" to the weak end of the hierarchy, while in the strong end, we replace RCTs and pseudo-RCTs with "industrial practice", as suggested in [13]. The rating "industrial practice" indicates that the method has already been approved and adopted by some SPL organization. In our opinion, such daily engineering practice shows a convincing proof that something *works*, so we rank it the strongest in the hierarchy.

Table 1
Quality assessment criteria.

No.	Question	Issue
RQ3.1	Is the paper based on research?	Reporting
RQ3.2	Is there a clear statement of the aims of the research?	Reporting
RQ3.3	Is there an adequate description of the context in which the research was carried out?	Reporting
RQ3.4	Was the research design appropriate to address the aim of the research?	Rigor
RQ3.5	Was there a control group with which to compare the treatments?	Rigor
RQ3.6	Was the data collected in a way that addressed the research issue?	Rigor
RQ3.7	Was the data analysis sufficiently rigorous?	Rigor
RQ3.8	Has the relationships between researcher and participants been adequately considered?	Credibility
RQ3.9	Is there a clear statement of findings?	Credibility
RQ3.10	Is there an explicit way to deal with scoping, commonality, and variability?	Relevance
RQ3.11	Are there any practitioner-based guidelines?	Relevance

We adopt the quality assessment criteria outlined in [14] in order to address RQ3, the limitations of current SPL RE methods. Specifically, we refine RQ3 into 11 sub-questions that cover 4 main issues: (1) reporting of the study's rationale, aims, and context, (2) rigor of the research methods employed to establish the validity and the trustworthiness of the findings, (3) credibility of the study methods for ensuring that the findings were valid and meaningful, and (4) relevance of the study for the software industry at large and the research community. Table 1 lists the refined criteria.

2.2. Search process

We aim to find a complete list of SPL RE studies reported since 1990. Our SLR dates back to 1990, since that year is the start of a fertile time period encompassing key SPL approaches, which has also been observed elsewhere [10]. Fig. 1a shows our original search strategy, which involved two-phases: automatic search of electronic databases and manual search of journals and conference proceedings.

During the first phase, we searched the following databases: ACM Digital Library, IEEE Xplore, Science Direct Elsevier, and Wiley Inter Science Journal Finder. The search was configured to match the search string only in the title of the papers. The string used in the automatic search consisted of two parts: RE **AND** SPL. We also identified a number of alternate keywords to form a more expressive query by using **OR** to connect the alternate keywords and **AND** to join the major parts. Hence, the search string employed in all databases is represented in the following box:

> *(Requirement* ***OR*** *requirements* ***OR*** *requirements engineering)* ***AND*** *(software product line* ***OR*** *software product lines* ***OR*** *product lines* ***OR*** *product line engineering* ***OR*** *software product family* ***OR*** *software product families* ***OR*** *product family* ***OR*** *product families* ***OR*** *variability* ***OR*** *variability management* ***OR*** *requirements reuse* ***OR*** *domain analysis* ***OR*** *domain engineering)*

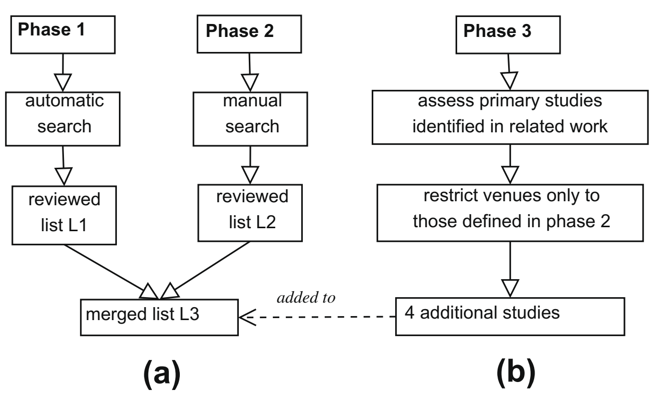

Fig. 1. Search strategy: (a) original two-phase strategy combining automatic and manual search and (b) additional search in related work.

Table 2
Selected journals and conference proceedings in phase 2.

Source	Acronym
Information and Software Technology	IST
Journal of Systems and Software	JSS
IEEE Transactions on Software Engineering	TSE
IEEE Software	IEEE SW
Communications of the ACM	CACM
ACM Computing Surveys	CSUR
ACM Transactions on Software Engineering and Methodology	TOSEM
Software Practice and Experience	SPE
Empirical Software Engineering Journal	EMSE
Requirements Engineering Journal	REJ
IEE Proceedings Software (now IET Software)	IET SW
International Conference on Software Engineering	ICSE
International Conference on Automated Software Engineering	ASE
International Requirements Engineering Conference	RE
International Software Product Line Conference	SPLC
International Conference on Software Reuse	ICSR
International Conference on Aspect-Oriented Software Development	AOSD

The number of studies retrieved from ACM, IEEE, Elsevier, and Wiley was 21, 34, 13, and 9 respectively.[1] All these retrieved lists were reviewed by each of the four researchers, i.e., Alves, Niu, Alves, and Valença. Then, in a group discussion, a list of relevant studies (L_1) were identified. Meanwhile, a couple of problems with respect to automatic search results were highlighted: (1) The results contained much irrelevant information, e.g., tutorial or workshop summaries and (2) Some known relevant work, e.g., [15,16], was missing from the results. These findings were in accordance with the experience that infrastructure support provided by software engineering indexing databases is inadequate to support complex Boolean searches [17]. Therefore, instead of refining the search strings and constraints, we decided to perform a manual search to complement the results.

In phase 2, a manual search was conducted of specific conference proceedings and journal papers from 1990 to 2009. The selected journals and conferences are given in Table 2. These sources were selected because they covered most of the studies identified in phase 1 (L_1) and presented a comprehensive collection of flagship venues on SPL RE. Each journal and conference proceedings was reviewed by one of the four researchers. The reviewer applied the inclusion and exclusion criteria (cf. Section 2.3), and the validated manual search results were recorded in list L_2.

The two lists, L_1 and L_2, were then merged by removing duplicated reports. One researcher performed this task; another researcher checked and confirmed the results. The merged list, L_3, contains 45 papers (S1–S45 in Table 4–Appendix).

It is worth mentioning that we have been collecting related work in systematic reviews of SPL RE.[2] Although the focus and research questions of each related work are different from ours, we are keen to examine the completeness of our chosen studies and are willing to correct any omissions. To that end, we performed a phase 3 search, as shown in Fig. 1b. We identified Khurum and Gorschek's effort [7] as one of the most recent and closest study to ours, and used it as a baseline for comparison. In [7], 89 primary studies were identified, among which eight appeared in L_3 and five had very close variants (i.e., almost the same authors and same title) in L_3. While it turned out that even with the list of 89 papers, Khurum and Gorschek missed some relevant studies (e.g., S24, S36, and S41 in L_3), we found only 4 papers in [7] (S46–S49 in Table 4) that should have been included in our search. Accordingly, the final list contains 49 papers (cf. Table 4), which serve as the primary studies of our SLR.

[1] The list of retrieved studies is available at http://twiki.cin.ufpe.br/twiki/pub/SPG/GenteAreaPublications/List.pdf.

[2] Section 5. discusses related work in more detail.

2.3. Inclusion and exclusion criteria

Our review included papers published between January 1st 1990 and August 31st 2009 following the inclusion criteria:

- Requirements engineering for software product lines.
- Domain analysis for product line engineering.
- Systematic requirements reuse.

The following criteria excluded papers from our review:

- Studies focused on product line architectures, domain design and implementation, and opportunistic reuse.
- Short papers, editorials, and summaries of keynote, workshop, or tutorial.
- Duplicate reports of the same study (when several reports of a study exist in different sources, the most complete version of the study was included in our SLR).[3]

The inclusion and the exclusion criteria were evaluated in the following way: each venue in Table 2 was reviewed by one of the four researchers, who then read title, keywords, and abstract to determine a paper's relevance according to each criterion. When necessary, the content of the paper was also examined. For each reviewer's result, two other researchers independently performed sanity checks. The differences were reconciled collaboratively.

2.4. Data extraction and assessment

We created a data extraction form to extract relevant information from studies in order to answer the research questions. The data extracted from each study and the corresponding assessment strategies were the following:

- The source (journal or conference), the title, and the author(s).
- The relationship to other studies; we recorded two types of paper relationship: (1) earlier version, in which the earlier version was excluded and (2) relevant work, in which all related papers were included.
- The name and short description of the proposed method, technique, or framework (addressing RQ1).
- The level of tool support, ranging from automatic, semi-automatic, to manual (addressing RQ1). This level was assessed based on the description of the underlying tool support within the study or from a referring web site: if partial human intervention was necessary to deploy the proposed method, the level was rated as semi-automatic; if human intervention was not necessary or minimal, the level was rated as automatic. Otherwise, it was rated as manual.
- The types of requirements artifacts involved, e.g., as inputs and outputs of the proposed framework (addressing RQ1.1).
- The RE activities (plan and elicit, model and analyze, communicate and agree, realize and evolve) supported (addressing RQ1.2).
- The SPL adoption strategies (proactive, extractive, reactive) followed (addressing RQ1.3). This assessment was based on whether the study described a method to build reusable requirements from scratch, from existing requirements documents of isolated applications, or from existing SPL requirements and a new application's requirements, respectively.
- The qualitative and quantitative evidence of the method's applicability, usefulness, and effectiveness (addressing RQ2).
- The evidence level(s); we defined six levels in Section 2.1 (addressing RQ2).
- The quality assessment of the study; we used the binary grading, "Yes" or "No", to assess the 11 quality criteria defined in Table 1 (addressing RQ3).

A pilot data extraction was designed in order to achieve a consistent rating among the researchers. Each researcher individually extracted the data on a couple of randomly selected studies. Then, a meeting was held to communicate the experiences from the individual dry runs in order to reduce the ambiguities, clarify the uncertainties, discuss the subtleties, and reconcile the differences. Once a shared understanding was established, one researcher coordinated the data extraction and checking tasks, which involved all the authors of this paper. Allocation was not randomized, but was based on the expertise of the individual researchers. Meanwhile, conflict of interest (e.g., self-assessing) was avoided.

2.5. Deviations from protocol

As explained in Section 2.2, we extended the original two-phase search process by incorporating the studies identified in related work (Fig. 1b). Although continuously examining the literature requires special attention and extra work, we felt that the effort was well spent. We not only produced a more complete list of primary studies, but also better justified our inclusion and exclusion criteria. Additionally, we added S45 to our list of primary studies, an omission which was pinpointed by two reviewers of the paper.

3. Results and analysis

Following the data extraction and assessment method described in Section 2.4, we now assess the selected papers with respect to the research questions. Tables 5–7 (Appendix) show the results, which we next summarize and analyze.

3.1. RQ1: What SPL RE methods and/or tools are available to the practitioners?

All papers are the representative publication of the corresponding method. Table 5 displays the results for RQ1. In terms of tool

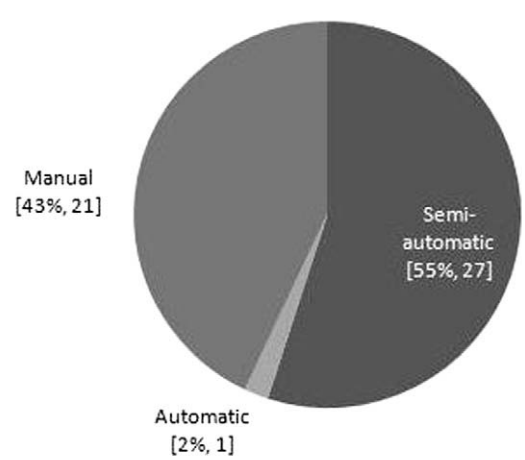

Fig. 2. RQ1 (tool support in the studied approaches).

[3] We encountered two instances of duplication: 1. Moon et al. TSE'05 (S1 in Table 4) was considered a more complete version of the conference paper: M. Moon and K. Yeom, "An Approach to Develop Requirement as a Core Asset in Product Line", *8th International Conference on Software Reuse*, July 2004, pp. 23–34; 2. Lauenroth and Pohl RE'08 (S28 in Table 4) was considered a more complete version of the conference short paper: K. Lauenroth and K. Pohl, "Towards Automated Consistency Checks of Product Line Requirements", *22nd IEEE/ACM International Conference on Automated Software Engineering*, November 2007, pp. 373–376.

support, 27 (55%) studies provide semi-automatic tools, 1 (2%) has automatic support, and 21 (43%) have no support, as shown in Fig. 2. Some tools are described as extensions of integrated development environments, whereas others are stand alone. It is surprising to realize that almost half of the studies do not explicitly discuss tool support, which raises usability, utility, and scalability shortcomings in these approaches. This might reflect that many studies were in preliminary stages and were not ready for cross-site validation or adoption.

The only claimed automatic tool support in S22 is provided by Vector Space Model [18] in the process of determining the similarity between requirements, where each requirement represents a vector and their similarity amounts to finding the angle between these vectors. However, as the authors point out, domain expertise and human intervention are needed to improve the matching accuracy and define configurations during the merge [15].

3.1.1. RQ1.1: What types of requirements artifacts are dealt with?

The papers describe approaches/tools manipulating requirements in various forms: plain textual, structured text, use cases, features, goals, agents, viewpoints, orthogonal variability models, state-machines, temporal logic, objects and events (Fig. 3). Accordingly, textual and features are the leading formats to express requirements. Although this shows the important role of feature models in describing SPL variability, it is consistent with the fact that requirements are mostly textual [19]. It is encouraging that a wide spectrum of requirements artifacts have been tackled. This allows the practitioner to choose the most appropriate approach for different problems.

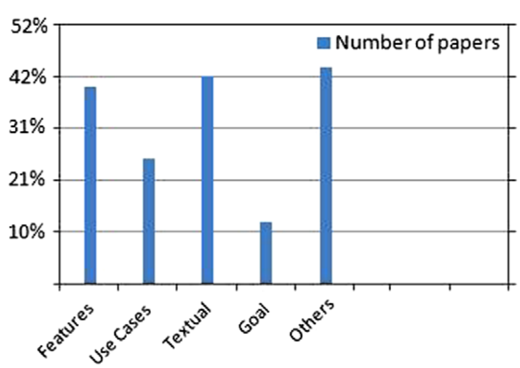

Fig. 3. RQ1.1 (requirements formats).

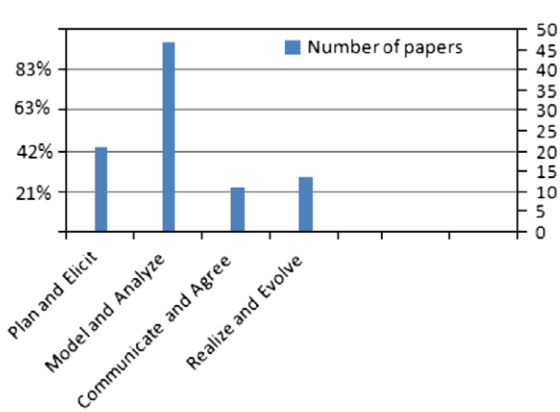

Fig. 4. RQ1.2 (RE activities addressed by the methods).

Twenty one (43%) papers have further specified input requirement models and output requirement models. Input requirement models (e.g., textual requirements) are at a lower level of abstraction than output requirement models (e.g., features, goals, agents, viewpoints). Seven papers (14%) relate features to use cases, whereas six papers (12%) relate features to textual requirements. In the first case, S42 is the pioneer work in relating feature to use cases. In particular, it presents FeatRSEB, which integrates FODA [20] and RSEB [21]. Regarding the studies relating features to textual requirements, they rely on the assumption that features are a higher level of abstraction over textual requirements, e.g., a cluster of requirements. In this context, the seminal study S30 initially defines a manual similarity relationship between requirements based on the notion of resources, then executes a clustering algorithm on this relationship to extract features.

3.1.2. RQ1.2: Which RE activities do the methods address?

In terms of RE activities, Plan and Elicit are addressed in 22 (43%) papers, Model and Analyze in 48 (98%) papers, Communicate and Agree in 11 (23%) papers, and Realize and Evolve in 14 (29%) papers. These results are represented in Fig. 4. The emphasis on Model and Analyze is important to addressing variability, which is at the core of SPLE. However, the comparatively lower support

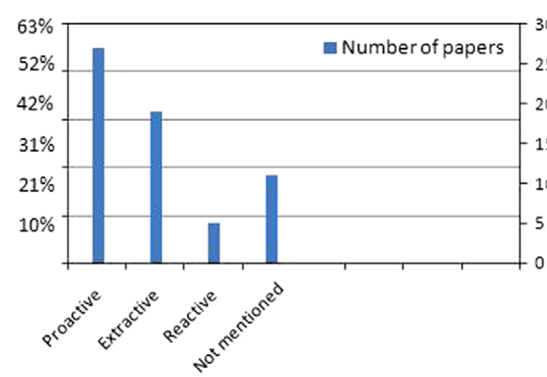

Fig. 5. RQ1.3 (SPL adoption strategies).

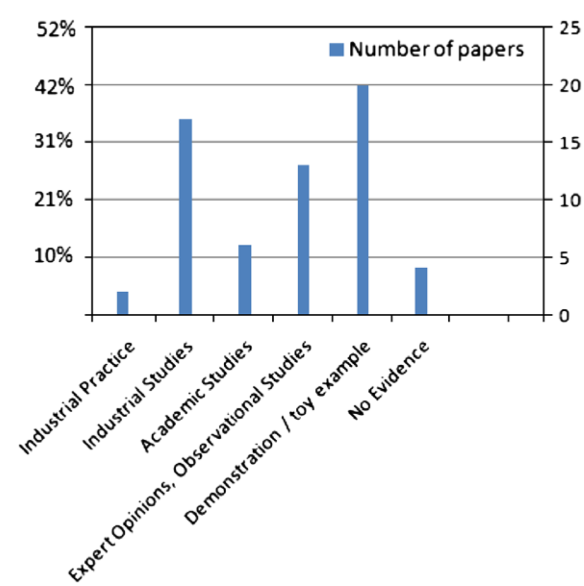

Fig. 6. RQ2 (evidence available to adopt the proposed methods).

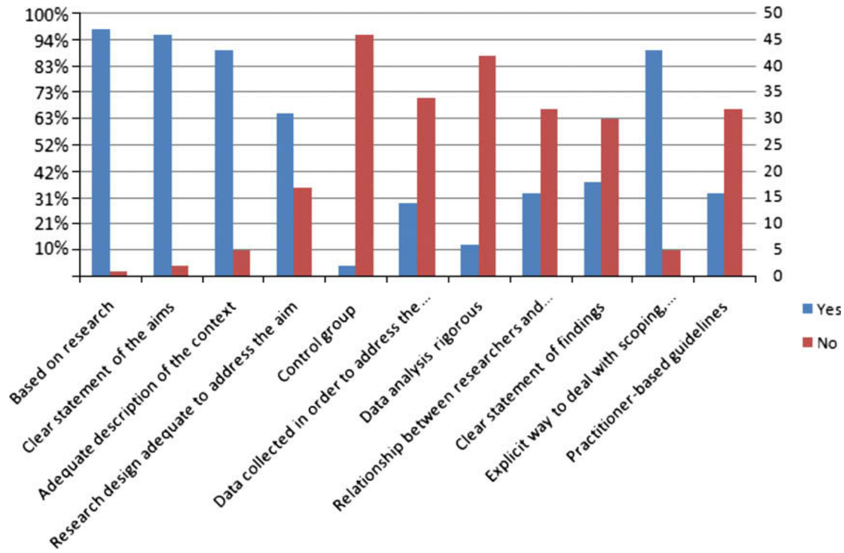

Fig. 7. RQ3 (limitations of the current research).

for the other activities raises a concern regarding the suitability of the approaches: the myriad of RE documents and formats makes planning and eliciting harder, in particular in terms of scalability. The larger number of stakeholders with competing or conflicting goals in SPLs complicate Communicate and Agree activity. Finally, only two studies (S11 and S23) cover all requirements engineering activities.

3.1.3. RQ1.3: What SPL adoption strategies do the methods follow?

Regarding adoption strategies, 28 (57%) approaches are proactive, 19 (39%) approaches are extractive, and only 5 (10%) are reactive. Only one approach (S33) embeds all three strategies. 4 (8%) studies combine extractive and reactive models, whereas 8 (16%) combine extractive and proactive strategies. The others do not fit into this classification scheme. Fig. 5 shows the results.

The data is consistent with the result for RQ1.2, since the emphasis on proactive means primary effort on Model and Analyze, and less concern for reactive is reflected in lower emphasis on Realize and Evolve. The moderate emphasis on extractive corresponds to the moderate effort in Plan and Elicit.

According to Krueger [5], the proactive approach is more prone to risks than the other two, since it usually requires a higher upfront investment. Further, the combination of extractive and reactive is considered to be a viable approach for most organizations to transition from one-of-a-kind system development to product line engineering. Therefore, the RQ1.3 results indicate that the approaches in the studies assessed need to perform better with respect to this criterion in order to achieve a higher adoption rate of SPLE in practice.

3.2. RQ2: How much evidence is available to adopt the proposed methods?

RQ2 detailed results are represented in Table 6 and summarized in Fig. 6. The data is presented in a bar graph instead of a pie chart, since it adds more than 100%. Accordingly, the main strategies for empirical evaluation are Demonstration/Toy examples (41%), industrial studies (35%), and observational studies/expert opinions (27%). According to our evidence evaluation scheme (Section 2.1), the emphasis on the first and on the third show the low level of evidence of the proposed methods. At the other end of the spectrum, the least frequently used strategies were industrial practice (6%) and academic studies (12%). Therefore, according to the same evaluation scheme, few approaches have high-level of evidence. No evidence occurred in 4 (8%) approaches.

Additionally, 31% of the approaches employ exactly two types of evaluation, the most frequent combination of types being observational studies/expert opinions and industrial studies (40% of the combinations). In particular, only three (6%) of the studies combine academic studies with another type of evaluation (twice with toy examples and once with expert opinions). Therefore, there were no studies with both academic and industrial evidence. Although from one point of view it could be arguable that such combined evidence would hardly fit a single study for space reasons, the Software Product Line Conference community is represented by comparable numbers of members from industry and academia.[4] Further implications of this result are discussed in Section 4. No study employed three or more types of evaluation.

Given these data and according to our classification scheme for evaluating RQ2 (Section 2.1), we observe that the majority of studies have some form of preliminary evaluation. However, this apparent benefit is diminished when these data are cross-referenced with the limitations of these studies (Section 3.3): their rigor and credibility is low. From the 45 (96%) approaches with evaluation, only 12 (24%) have suitable credibility, whereas none of them has suitable rigor. In particular, the only two studies using control group (S12 and S35) are evaluated by academic studies and expert opinions, respectively. Overall, this cross-reference compromises the soundness of their evidence. Therefore, given the increasing widespread adoption of SPL and their potential impact and required investment, the results show that further development is mandatory in the field to provide better evidence regarding the quality and suitability of proposed methods, as discussed in Section 4.1.

3.3. RQ3: What are the limitations of the current research in the field? (quality assessment)

RQ3 detailed results are represented in Table 7 and summarized in Fig. 7, according to the quality assessment criteria presented in

[4] www.splc.net.

Table 1. In terms of reporting (RQ3.1–RQ3.3 in Table 1), most approaches perform fairly well: at least 90% of the studies are based on research, have a clear statement of the aims of the research and an adequate description of its underlying context. We justify this result based on the quality standards required by the venues/journals, where the studies were published (Table 2).

Nevertheless, regarding rigor and credibility (RQ3.4–RQ3.9 in Table 1), most approaches perform poorly, in particular, with worse results for control group (RQ3.5), which is considered only in two (4%) approaches and data analysis rigor (RQ3.7), which is satisfactory in six (12%) methods. This compromises the validity and usefulness of the studies, since noncompliance in these items implies no sound established advantage over other approaches. This is further complicated by the fact that only 30% of the studies established relationships between the researcher and participants and collected data in a way to address the research issue, thus leading to high potential for bias and inconclusive results. Overall, 76% of the approaches fail in rigor and credibility. In a way, these results are expected given the informal manner that methods are empirically evaluated. Nonetheless, this indicates that the field needs further maturation with respect to validity and credibility of the results, a key issue in Evidence-Based Software Engineering.

Finally, in terms of relevance, 90% of the approaches have an explicit way of dealing with scoping, commonality, and variability (RQ3.10), whereas only 35% propose practitioner-based guidelines (RQ3.11). The former result is justified by the nature of the research goals (reflected in the Inclusion/Exclusion criteria, Section 2.3). The latter result is consistent with 35% of the approaches in the industrial studies level of evidence (Section 3.2) and indicates that, despite their research contribution, insufficient or non-existent guidance is offered for applying most of the proposed approaches. Therefore, the field needs to evolve towards more practical guidance to intensify industry adoption.

4. Discussion

This section summarizes the main findings of this study (Section 4.1) and discusses the relevance of gathered evidence to the software engineering community (Section 4.2). Limitations of this study are presented in Section 4.3.

4.1. Main findings

One of the goals of this study was to investigate the state of research and practice in the field of RE for SPLs. The major implications for research include the following issues:

- *Requirements heterogeneity*. Results from Section 3.1.1 shows that a myriad of requirements artifacts have been tackled. This result is positive for SPL RE, since it shows that the studies handle the inherent heterogeneity of requirement formats, thus allowing the practitioner to choose the most appropriate approach for different problems. Although features are a prominent format, the leading is textual. Further, despite FODA-based approaches, we remark that other early approaches contributed to SPLE as a field to emerge, for example, the work in S45, addressing assumptions that can or cannot vary across SPL instances instead of features.
- *Scalability issues*. As reported in Section 3.1.1, some approaches have input and output models, where the former are requirements in various formats and the latter are some abstraction over requirements, such as features in S22, S23, and S30. This abstraction is essential to provide a useful model of the SPL, for instance, its configurability view. Nonetheless, this abstraction also poses some challenges given the heterogeneous and mostly textual nature of the underlying requirements. Therefore, in the SPL context, scalability is an issue, since not only each application requirements document may be large but also there exists the significant number of documents referring to many applications in the domain.
- *Need for practitioners involvement*. The field of SPL RE has attracted several research initiatives over the last two decades. Meanwhile, studies have been frequently conducted with industrial partners and claim their methods address industrial needs. However, what we observed from this review is that the majority of the methods resulted from research proposals rather than emerged from established industrial practice. Consequently, we believe that researchers should develop their methods in close collaboration with practitioners, instead of only involving practitioners to evaluate their research proposals.
- *Incompleteness of RE activities*. According to results presented in Section 3.1.2, very few papers covered all RE activities. This is an acceptable result, as we may expect that work in progress cover specific topics rather than address large areas of research. One possible reason is that conference papers report work in progress that needs to be further detailed and improved. Another reason is that supporting the whole RE activities for SPLs is a hard problem that requires cooperation from several researchers working jointly. Therefore, the finding suggests the need to integrate fragmented approaches in order to define a comprehensive solution addressing the complete RE process.
- *Low level of evidence and validation*. The data reported in Section 3.2 show that several methods have been empirically validated by means of case studies in different domains including automotive, house automation, and spacecraft control. However, our analysis reveals that the majority of papers rely on toy examples or superficial "case studies" as a way to explain how the proposals work. With respect to the quality of research validation (Section 3.3), studies do not present well designed quantitative or qualitative evaluations. None of the methods were evaluated following an appropriate study design with rigorous recruitment strategy and critical analysis of results. Therefore, claims regarding the suitability of methods are supported by weak evidence based mainly on the authors' own experience using the methods by means of informal case studies. To increase the adoption of research proposals in industrial settings, researchers should improve the rigor of empirical validation and present reliable arguments to demonstrate the methods adequacy to a particular situation.

Regarding the state of practice in the field, we obtained the following findings:

- *Lack of comparative studies*. None of the reviewed papers reported a comparative study of SPL RE methods. This type of study is highly important to inform decision makers interested in adopting a new method. Assessing alternative methods using the same comparison baseline facilitates the selection of the best approach for a particular situation. With this respect, the studies we assessed do not highlight the suitability of their proposals for a particular situation nor whether the method is not adequate for specific cases. For instance, it is not clear whether the approach scales properly, to which domains and organizational environments the method can be applied, how practitioners can conduct a stepwise adoption, what the return on investment will be, and so on. Understanding the characteristics of situations where a method is appropriate helps practitioners identifying similar situations to apply a method.
- *Limited tool support*. Another common problem is the limited availability of tools to support the use of methods. Although most studies suggest they have tool support, we could not find these tools on the Web. The result of a Web search identified only two tools available for download (an open source and a proprietary tool). In order to increase the chances of a method being adopted in practice, suitable tools must be available. These tools have to be tailorable enough to be useful for the domain of interest.

• *Limited guidance and adoption strategies*. Based on the results reported in Section 3.3, we observed that a major limitation of studies is that they do not provide sufficient guidance for practitioners interested in adopting proposed methods. The absence of objective guidelines may prevent the use of a new method by one who has little or no previous knowledge in the SPL field. Furthermore, practitioners may have problems judging the benefits, limitations and risks of immature methods. In complex domains, which are frequently the ones that could receive high gains from adopting a SPL solution, practitioners must have strong confidence that a new method will work as promised by their proponents. Further, according to Section 3.1.3, there is still a substantially higher focus on the risk-prone proactive adoption strategy rather than on the combination of the extractive and reactive strategies, these latter having more potential for SPL adoption in practice.

• *Limited maturity*. Finally, from a practitioner's point of view, it is hard to determine the maturity of SPL RE methods. The body of evidence provided by the studies is fairly poor and thus it is hard to objectively assess the quality and overall usefulness of a particular method. Fostering the maturity of research proposals is a fundamental step to promote the popularization of technologies.

Based on the main findings obtained from our study, we make some suggestions to researchers and practitioners in RE for SPLs:

- Enhance the use of Natural Language Processing and Information Retrieval techniques in addressing variability within the mostly textual nature of requirements.
- Focus not only on the proactive product line adoption strategy, but also on extractive and reactive strategies and their combinations.
- Conduct more comparative studies, e.g., by empirically assessing the cost-effective degrees of different methods and techniques.
- Build an empirical base for sharing the cross-checking data, including requirements documents, requirements models, (prototype) tools, validation results, etc. The effort is in line with the recent advancements in empirical software engineering, such as PROMISE (Predictive Models in Software Engineering)[5] and MSR (Mining Software Repositories).[6]
- Conduct and report empirical studies more rigorously, e.g., by following the tutorials [22,23].

4.2. Strength of evidence

As recommended in [14], we adopted the GRADE (Grading of Recommendations Assessment Development and Evaluation) [24] approach to assess the evidence level of our selected studies. The GRADE system rates the strength of evidence provided by empirical studies using the scale: ⟨high, moderate, low, very low⟩. According to GRADE, the strength of evidence is determined by four criteria: study design, study quality, consistency, and directness.

To assess the strength of evidence in our SLR, we consider the following aspects. With respect to study design and according to the results from Section 3.2, studies were mainly toy examples. Additionally, only 6% of the papers presented case studies reporting industrial practice. Therefore, we consider the evidence provided by the study design very low. Papers faced serious problems regarding the quality of empirical studies. Methods were not clearly described, data collection and analysis were not rigorous, and issues of validity were not treated by the authors of primary studies. The consistency of studies refers [24] to the "similarity of estimates of effect across studies". Due to the fact that there were not comparable studies in our review, we cannot assess the consistency criterion. Finally, as defined in [24] directness is "the extent to which the subjects, settings, treatments, and outcome measures are similar to those of interest". We found little effort has been made to present relevant industrial studies and practitioner-oriented guidelines. Therefore, we argue that studies may have insignificant effect on industrial practice. However, papers are more research-oriented and address relevant issues for the research community. Our judgement is that directness can be rated as moderate. Consequently, due to the limitations found in the primary studies, we rate the strength of evidence of the papers assessed in our systematic review as low. This result suggests that the area of SPL RE needs further efforts in order to be considered a mature discipline.

4.3. Limitations of this study

The main limitations of our review are eventual omission of papers in the search process and bias in the extraction of data. Regarding the search process, we have started with an automatic search strategy using the main software engineering electronic databases. Confirming lessons reported in [17], due to limitations of the search engines, we realized that the automatic search missed important papers in the field. To alleviate these problems, we decided to conduct a manual search to improve the quality of search results. As presented in the exclusion criteria, we excluded short papers published in workshops, tutorials, keynotes. We believe that excluding these papers has not an impact on the overall results obtained, as such literature is unlikely to present significant information of established SPL RE methods. After having conducted the search phase, we became aware of paper [7], which presents a systematic review on domain analysis solutions. As we have discussed in Section 2.5, this SLR reviewed 89 papers, where four additional papers were also relevant to our study. Therefore, we decided to change our initial research protocol to include these papers in our review.

Our aim was to do our best to ensure the completeness of our selection. Another potential risk that we may have missed relevant papers is due to lack of agreed terminology in the SPL field and to the possible existence of relevant papers that do not mention the keywords we specified explicitly. This means that our choice of keywords may not have encompassed the complete set of papers published in our field of interest (i.e., SPL RE methods). Indeed, as mentioned in [25], "it is important to recognize that software engineering keywords are not standardized and that they can be both discipline – and language-specific". Hence, even carefully defining consistent keywords and related terms, there is a risk that pertinent studies are omitted. In order to minimize this risk, we conducted a manual search (cf. Phase 2 in Fig. 1a), applying inclusion and exclusion criteria (cf. Section 2.3), and also using title and abstract–and not only keywords–to determine a paper's relevance.

With respect to bias in the data extraction, we had some difficulties to extract relevant information from the papers. The majority of papers do not present objective details regarding several issues we wanted to address in the research questions. For instance, several papers do not explicitly mention what SPL adoption strategy they support nor what requirements phase the method can be employed. In several occasions we had to interpret the subjective information provided by the papers. To minimize interpretation bias, we conducted discussion meetings among the co-authors of this paper during the data extraction phase. Another frequent problem was the lack of information regarding the empirical methods employed by these studies to conduct evaluation. We found that, in some primary studies, data collection and analysis were poorly described. Overall, empirical studies were not conducted in a rigorous manner. This situation may have

[5] promisedata.org.
[6] msr.uwaterloo.ca.

compromised the accuracy of data extracted regarding the quality assessment criteria. Therefore, we acknowledge that there is a possibility of having misunderstandings in the way we have extracted data from the primary studies.

With respect to data assessment, according to Section 2.4, tool support has been evaluated based on the description of the underlying tool support within the study or from a referring web site. Indeed, it can be hard to make such assessment by just reading descriptions and thus ideally the tools would be installed. However, requiring tool installation could have significantly reduced the number of studies considered, given the restricted availability of their underlying tools, which has been confirmed in Section 4.1 (7th bullet) and also noted elsewhere [8]. Therefore, we envision as future work such effort of assessing tool support based on its installation, which is in line with the suggestions at the end of Section 4.1 (before last bullet).

The threats to validity of our study can be addressed as follows:

- *Construct validity* concerns establishing correct operational measures for the concepts being studied. The main constructs in our study are the concepts of "RE for SPLs" and "systematic literature review". Regarding the first, we identified some roots of the field and compared related work. We also teased out the key characteristics, such as SCV (scoping, commonality, variability) analysis and adoption strategies (proactive, extractive, reactive), of RE for SPLs. As for the second construct, we followed the guidelines to design our research questions, search criteria, and assessment protocol. We also documented the deviations made to address possible threats to construct validity.
- *Internal validity* concerns establishing a causal relationship, whereby certain conditions are shown to lead to other conditions. In particular, the main threats are selection of primary studies and individual bias in assessment. Our major source of data was peer-reviewed journal and conference papers on RE for SPLs. As discussed earlier, we extended our literature review to several rounds in order to incorporate the most complete primary studies possible in order to increase reliability of the conclusions. The other threat arises from individual researchers bias when assessing his or her assigned primary studies. We ameliorated the threat by following a pre-defined protocol, carrying out several dry runs individually, and consolidating the differences collaboratively.
- *External validity* concerns establishing the domain to which a study's findings can be generalized. The scope of our systematic literature review was on RE for SPLs that spanned from 1990 to 2009. Our results might not generalize to broader time periods, or broader primary study selections, e.g., books and technical reports. The results of our current study were drawn from qualitative analysis. Quantitative analysis and inferences may be further considered to enable analytical and statistical generalizations.
- *Reliability* concerns demonstrating that the operations of a study can be repeated with the same results. We expect that replications of our study should offer results similar to ours. Of course, the characteristics of research questions, search strings, and selected primary studies may differ from our current study, but the underlying trends should remain unchanged.

5. Related work

Researchers have recently conducted systematic reviews in order to gather and evaluate the available evidence in the area of SPL RE. We identified four SLRs: domain analysis (DA) solutions for SPLs [7], DA tools [8], RE for SPLs presented in Chapter 3 of [9], and variability management in SPL [10]. Table 3 summarizes the comparison.

In [7], the authors aimed to assess the applicability, usefulness, and utility of the proposed DA solutions, especially the workability in industrial settings. In contrast, we aimed to evaluate quality assessment (reporting, rigor, credibility, relevance), evidence level, and to characterize the approaches in terms of tool support, RE activities, and adoption strategies. The authors of [7] could not provide conclusive answers to 3 out of 4 research questions, due to the lack of rigorous study design and data analysis. We found similar results in our study. Their results also supported the lack of industrial practitioner involvement (applications and empirical validation) in developing domain analysis tools. As for the selection of primary studies, we covered 1990–2009, a strict superset of the 1998–2007 time frame considered in [7]. However, 89 papers were selected in [7], which shows a significantly broader scope and

Table 3
Comparison of related systematic reviews.

SLR	Research question	Answer
[7] (89 primary studies identified between 1998 and 2007)	Are DA solutions based on industrial needs?	A majority of proposals are
	Are DA solutions applied and/or validated?	Inconclusive due to the absence of detailed results or replicated studies
	Are DA solutions usable?	Inconclusive due to the lack of qualitative and quantitative data analysis
	Are DA solutions useful?	(Same as above)
[8] (19 primary studies identified, no time period specified)	Do DA tools support a specific or a generic process?	Most tools (>78%) are developed to support a specific process
	What are the main functionalities of the tools?	The tools are mainly focused on the modeling phase
	Where the tools were developed and used?	The majority were developed and used in academia
[9] (16 primary studies identified, no time period specified)	What RE activities are supported?	Common: elicitation, analysis, and specification; Less common: verification, management
	What RE techniques are proposed?	Classified nine categories: feature, goal, use case, viewpoint, etc.
	How is variability dealt with?	All surveyed approaches support variability management
[10] (33 primary studies identified, between 1990 and 2007)	What approaches have been proposed for managing variability in software product lines?	33 approaches are identified and their short descriptions are given
	How has the research on developing variability management approaches been evolved?	A chronological view is provided. Notable branches include: feature orientation, Koala, and OVM (orthogonal variability modeling)
	What are the key issues that have drove [driven] the evolution of different variability management approaches?	10 issue groups are identified: variability modeling, identifying commonality and variability, process support, etc.

more relaxing inclusion criteria. When comparing their selection with ours, we found 12 overlapping papers and five almost identical papers. The distinctions reflect different focuses between SPL DA and SPL RE. For example, SPL RE discusses viewpoints and goals, whereas DA investigates releasing and configurations.

Lisboa et al. reviewed the tool support aspect of DA [8]. They found that there is hardly a generic process for all DA tools, most tools focused on supporting the modeling phase, and the majority of the tools are developed and used only in academic settings. Our study also revealed that few approaches cover all RE phases, in particular, most have an emphasis on modeling. It is interesting to note that because of available executables and other reasons, the authors selected 19 studies from the 31 potentially relevant tools [8]. Our study confirmed that many research prototype tools do not make themselves accessible for cross-site validation and adoption.

In Chapter 3 of [9], the author provided a literature review so as to define a unified RE process for SPLs. The review was based on 16 primary studies, whose representative papers were all included in our selected list. Note that the search strings defined in [9] were very similar to ours. The relatively small number of studies resulted from the automatic search was in accordance with our experience. The research questions and findings in [9] were also a subset of ours. Furthermore, we refined the high-level questions into a set of more concrete sub-questions, and clarified the criteria used to answer them.

The authors of [10] have recently performed a systematic review on variability management (VM) in SPL. Their focus was on variability modeling in general and not particularly within RE as in our work. Their identified primary studies were from 1990 to 2007, whose range has much overlap with our covered time period (1990–2009). Their SLR scope had much emphasis on how different VM approaches evolved over the years. The results from their SLR supported our findings in terms of the lack of tool support, the lack of studies on product line evolution and product derivation, the strength on SCV (scoping, commonality, variability) analysis, and the potential scalability problems. In addition, their review identified VM-specific issues, such as variability binding time (build-time, delivery time, runtime, etc.) and mechanisms (object-oriented framework, load table, plug-ins, etc.).

SLRs have gained much importance in software engineering because they help to assess and aggregate outcomes and evidence from the ever increasing number of empirical studies on a particular topic. They are superior to traditional literature reviews in terms of rigor, completeness, and repeatability. Guidelines and experiences about SLRs have been reported, e.g., [11,17,14]. From our comparison to related work, we felt that no single SLR has collected *all* relevant primary studies. The combination of automatic and manual search would offer more complete results. Moreover, SLRs are empirical studies themselves. Tertiary studies (e.g., [26]) or mapping studies (e.g. [27]) can shed light on assessing and aggregating the outcomes from related SLRs.

Table 4
Selected primary studies.

ID	Title	Author(s)	Venue
S1	An approach to developing domain requirements as a core asset based on commonality and variability analysis in a product line	M. Moon, K. Yeom, and H.S. Chae	TSE 31(7): 551–569, 2005
S2	Ten steps towards systematic requirements reuse	W. Lam, J.A. McDermid, and A.J. Vickers	REJ 2(2): 63–113, 1997
S3	Structuring product family requirements for *n*-dimensional and hierarchical product lines	J.M. Thompson and M.P.E. Heimdahl	REJ 8(1): 42–54, 2003
S4	Analogy-based domain analysis approach to software reuse	C.-H. Lung, J.E. Urban, and G.T. Mackulak	REJ 12(1): 1–22, 2007
S5	Multi-level feature trees	M.-O. Reiser and M. Weber	REJ 12(2): 57–75, 2007
S6	Managing requirements inter-dependency for software product line derivation	D. Sellier, M. Mannion and J.X. Mansell	REJ 13(4): 299–313, 2008
S7	An integrated domain analysis approach for teleoperated systems	J. Nicolas, J. Lasheras, A. Toval, F.J. Ortiz, and B. Alvarez	REJ 14(1): 27–46, 2009
S8	Reusable software requirements and architectures for families of systems	H. Gomaa	JSS 28(3): 189–202, 1995
S9	Achieving requirements reuse: a domain-specific approach from avionics	W. Lam	JSS 38(3): 197–209, 1997
S10	Domain analysis for software reuse	A. Sutcliffe	JSS 50(3): 175–199, 2000
S11	Goal and scenario based domain requirements analysis environment	J. Kim, M. Kim, and S. Park	JSS 79(7): 926–938, 2006
S12	DRAMA: A framework for domain requirements analysis and modeling architectures in software product lines	J. Kim, S. Park, and V. Sugumaran	JSS 81(1): 37–55, 2008
S13	Managing requirements specifications for product lines – an approach and industry case study	M. Eriksson, J. Borstler, and K. Borg	JSS 82(3): 435–447, 2009
S14	Capturing quality requirements of product family architecture	E. Niemela and A. Immonen	IST 49(11–12): 1107–1120, 2007
S15	Rigorous engineering of product line requirements: a case study in failure management	C. Snook, M. Poppleton, and I. Johnson	IST 50(1–2): 112–129, 2008
S16	Approach to modelling feature variability and dependencies in software product lines	H. Ye and H. Liu	IET SW 152(3): 101–109, 2005
S17	Addressing quality attributes in domain analysis for product lines	S. Jarzabek, B. Yang, and S. Yoeun	IET SW 153(2): 61–73, 2006
S18	Multiple-view modelling and meta-modelling of software product lines	H. Gomaa and M.E. Shin	IET SW 2(2): 94–122, 2008
S19	Using viewpoints to define domain requirements	M. Mannion, B. Keepence, and D. Harper	IEEE SW 15(1): 95–102, 1998
S20	Feature-oriented product line engineering	K.C. Kang, J. Lee, and P. Donohoe	IEEE SW 19(4): 58–65, 2002
S21	A framework for constructing semantically composable feature models from natural language requirements	N. Weston, R. Chitchyan, A. Rashid	SPLC'09: 211–220
S22	An exploratory study of information retrieval techniques in domain analysis	V. Alves, C. Schwanninger, L. Barbosa, A. Rashid, P. Sawyer, P. Rayson, C. Pohl, and A. Rummler	SPLC'08: 67–76
S23	On-demand cluster analysis for product line functional requirements	N. Niu and S. Easterbrook	SPLC'08: 87–96
S24	Automating mappings between use case diagrams and feature models for software product lines	A. Bragana and R. Machado	SPLC'07: 3–12
S25	A Methodology for the derivation and verification of use cases for product lines	A. Fantechi, S. Gnesi, G. Lam and E. Nesti	SPLC'04: 255–265
S26	Requirements management for product lines: extending professional tools	K. Schmid, K. Krennrich, and M. Eisenbarth	SPLC'06: 122–131
S27	Model-based requirements engineering for product lines	G. Buckle	SPLC'00:193–204

Table 4 (*continued*)

ID	Title	Author(s)	Venue
S28	Dynamic consistency checking of domain requirements in product line engineering	K. Lauenroth and K. Pohl	RE'08: 193–202
S29	Extracting and modeling product line functional requirements	N. Niu and S. Easterbrook	RE'08: 155–164
S30	An approach to constructing feature models based on requirements clustering	K. Chen, W. Zhang, H. Zhao, H. Mei	RE'05:31–40
S31	Modelling requirements variability across product lines	S. Buhne, K. Lauenroth, and K. Pohl	RE'05:31–40
S32	On goal-based variability acquisition and analysis	S. Liaskos, A. Lapouchnian, Y. Yu, E. Yu, and J. Mylopoulos	RE'06:76–85
S33	Consistency management of product line requirements	J. Savolainen and J. Kuusela	RE'01:40–47
S34	Requirements engineering for product families	J. Kuusela and J. Savolainen	ICSE'00:61–69
S35	Reusing single-system requirements from application family requirements	M. Mannion, H. Kaindl, J. Wheadon and B. Keepence	ICSE'99:453–462
S36	The domain analysis concept revisited: a practical approach	E. Almeida, et al.	ICSR06: 43–57
S37	Performing domain analysis for model-driven software reuse	D. Lucredio et al.	ICSR'08: 200–211
S38	Feature-oriented analysis and specification of dynamic product reconfiguration	J. Lee and D. Muthig	ICSR'08: 154–165
S39	Supporting software variability by reusing generic incomplete models at the requirements specification stage	R.P. Redondo et al.	ICSR'04: 1–10
S40	Requirements-reuse using GOPCSD: component-based development of process control systems	I.A.M. El-Maddah and T.S.E. Maibaum	ICSR'04: 318–328
S41	FODAcom: an experience with domain analysis in the Italian telecom industry	A.D. Vici	ICSR'98: 166–175
S42	Integrating feature modeling with the RSEB	M.L. Griss, J. Favaro, and M. d Alessandro	ICSR'98:76–85
S43	Object-oriented technology and domain analysis	S. Cohen and L.M. Northrop	ICSR'98: 86–93
S44	Concept analysis for product line requirements	N. Niu and S. Easterbrook	AOSD'09: 137–148
S45	Commonality and Variability in Software Engineering	J. Coplien, D. Hoffman and D. Weiss	IEEE SW 15(6): 37–45, 1998
S46	Product-line requirements specification (PRS): an approach and case study	S.R. Faulk	RE'01: 48–55
S47	Safety analysis of requirements for a product family	R.R. Lutz et al.	RE'98: 24–31
S48	COVAMOF: a framework for modeling variability in software product families	M. Sinnema, S. Deelstra, J. Nijhuis, and J. Bosch	SPLC'04: 197–213
S49	Tool-supported verification of product line requirements	P. Padmanabhan, R.R. Lutz	ASE'05: 447–465

Table 5
RQ1 (research question 1) results.

Study	Tool support	RQ1.1 requirements format	RQ1.2: RE activities supported				RQ1.3 SPL adoption		
			P/E	M/A	C/A	R/E	Pr	Ex	Re
S1	**S**	**O**: domain usecase model	Yes	Yes	No	No	No	Yes	No
S2	**S**	N/A	No	Yes	No	Yes	No	Yes	No
S3	**M**	**O**: Venn diagrams, configuration diagram	Yes	Yes	No	No	No	Yes	Yes
S4	**M**	Features, use cases	Yes	Yes	No	No	N/A	N/A	N/A
S5	**S**	Features	No	Yes	No	Yes	N/A	N/A	N/A
S6	**S**	**I**: textual, **O**: decision model	Yes	Yes	No	No	Yes	No	No
S7	**M**	Features, use cases	No	Yes	No	No	Yes	No	No
S8	**S**	**O**: graphical domain models	No	Yes	No	Yes	Yes	No	No
S9	**S**	**I**: textual, **O**: structured textual	Yes	No	No	No	No	Yes	No
S10	**M**	Objects, agents, goals, events	No	Yes	No	No	Yes	No	No
S11	**S**	**I**: textual, **O**: goal, use cases	Yes	Yes	Yes	Yes	Yes	No	No
S12	**S**	Goals, scenarios	Yes	Yes	No	No	Yes	No	No
S13	**S**	**I**: textual, use cases, **O**: feature models	No	Yes	Yes	Yes	N/A	Yes	No
S14	**M**	**I**: textual, **O**: UML models	Yes	Yes	No	No	Yes	Yes	No
S15	**S**	**I**: textual, **O**: UML-B models	No	Yes	No	No	Yes	No	No
S16	**M**	Features	No	Yes	No	Yes	Yes	No	Yes
S17	**M**	Features, goals	No	Yes	No	No	Yes	No	No
S18	**S**	**I**: use cases, **O**: multiple-view meta-model	No	Yes	Yes	Yes	Yes	No	No
S19	**S**	**I**: textual, **O**: domain dictionary, viewpoints	Yes	Yes	Yes	No	Yes	Yes	No
S20	**M**	Features	Yes	Yes	No	No	Yes	No	No
S21	**S**	**I**: textual, **O**: feature models	No	Yes	Yes	Yes	Yes	Yes	No
S22	**A**	**I**: textual, **O**: feature models	Yes	Yes	No	No	No	Yes	No
S23	**S**	**I**: textual, **O**: feature models	Yes	Yes	Yes	Yes	Yes	Yes	No
S24	**M**	**I**: use cases, **O**: feature models	No	Yes	No	No	N/A	N/A	N/A
S25	**M**	**I**: use cases, **O**: use cases with variability tags	No	Yes	No	No	Yes	No	No
S26	**S**	**I**: textual, **O**: structured text	No	Yes	No	No	N/A	N/A	N/A
S27	**S**	**I**: structured text, **O**: use cases	No	Yes	No	No	Yes	Yes	No
S28	**M**	OVM, state-machines	No	Yes	No	No	N/A	N/A	N/A
S29	**S**	**I**: textual, **O**: OVM	No	Yes	No	No	Yes	Yes	No
S30	**S**	**I**: textual, **O**: feature models	No	Yes	No	No	No	Yes	No
S31	**S**	Graphical, structured text	No	Yes	No	No	N/A	N/A	N/A
S32	**S**	Goal tree, variability labels	Yes	Yes	Yes	No	No	Yes	No
S33	**M**	Definition hierarchy	No	Yes	No	Yes	Yes	Yes	Yes
S34	**S**	Definition hierarchy	Yes	Yes	Yes	No	Yes	No	No

(*continued on next page*)

Table 5 (*continued*)

Study	Tool support	RQ1.1 requirements format	RQ1.2: RE activities supported				RQ1.3 SPL adoption		
			P/E	M/A	C/A	R/E	Pr	Ex	Re
S35	**S**	**I**: textual, **O**: configurations	Yes	Yes	No	No	Yes	No	No
S36	**M**	Features	Yes	Yes	Yes	No	Yes	No	No
S37	**M**	Features	Yes	Yes	Yes	No	Yes	No	No
S38	**M**	Features	No	Yes	No	No	N/A	N/A	N/A
S39	**M**	Graphical models, temporal logic formulas	No	Yes	No	No	N/A	N/A	N/A
S40	**S**	State-machines, KAOS goals	No	Yes	No	No	N/A	N/A	N/A
S41	**M**	Features, use cases, textual	Yes	Yes	No	Yes	Yes	No	No
S42	**S**	Features, use cases	Yes	Yes	No	No	Yes	Yes	No
S43	**M**	Features, use cases	Yes	Yes	No	No	Yes	No	No
S44	**S**	**I**: textual, **O**: concept lattices	Yes	Yes	Yes	No	No	Yes	Yes
S45	**S**	Sets	Yes	Yes	No	No	Yes	No	No
S46	**M**	**I**: textual, **O**: hierarchy of objects	No	Yes	No	No	Yes	No	No
S47	**M**	**I**: textual, **O**: textual	No	Yes	No	Yes	No	Yes	Yes
S48	**M**	**O**: graphical models	No	Yes	No	Yes	No	Yes	No
S49	**S**	Features	No	Yes	No	Yes	N/A	N/A	N/A

A: Automatic, **S**: Semi-automatic, **M**: Manual, **I**: Input, **O**: Output, **P/E**: Plan and elicit, **M/A**: Model and analyze, **C/A**: Communicate and agree, **R/E**: Realize and evolve, **Pr**: Proactive, **Ex**: Extractive, **Re**: Reactive.

6. Conclusion

This paper presented a SLR of RE within SPLE. Accordingly, the overall quality of research in the field needs improvement in terms of empirical validation. Results reveal that, although there is some level of evidence to adopt the studied methods, this level is mostly given by unproved claims and statements. In addition, these approaches still have serious limitations in terms of validity and credibility of their findings, e.g., very few apply control group with which to compare the treatments, or perform rigorous data analysis, or address threats to validity. Consequently, we expect that further empirical studies are performed with sufficient rigor to enhance the body of evidence in the SPL RE field. In this context, there is a clear need for conducting studies comparing alternative methods.

Additionally, a major drawback of studies is that they do not provide sufficient guidance for practitioners interested in adopting proposed methods, thereby limiting their use in industrial settings. In particular, there are very few commercial or open source tools accessible. In order to address these issues, we suggest that a promising focus for future research should be invested in developing reliable tool support and objective guidance to adopt SPL RE methods.

As a research trend, in order to effectively perform scalable domain analysis, we expect the growth of approaches that handle the textual nature of SPL requirements in more automatic form through the increased use of Natural Language Processing and Information Retrieval techniques, e.g., S22, S23, S29, and S30. Another remarkable result is that the proactive strategy is the most common adoption strategy suggested by the methods. However, according to [5], the proactive approach is the most expensive and risk-prone strategy. In this context, we believe that further work combining extractive and reactive strategies has the potential to increase the level of SPL adoption in practice. Another expected trend is performing comparative studies of SPL RE methods, since this type of study is highly important to inform decision makers interested in adopting a new method.

Given the current state of research of RE in SPLE, researchers should consider opportunities to extend and improve present research in an integrative fashion instead of conducting disconnected studies. We believe that joint academic and industrial efforts are crucial to foster RE in SPLE. This will bring additional responsibility for researchers to show that new ideas are not simply promising, but also that they are effective, which is a necessary base to further disseminate research results into practice.

Table 6
RQ2 (research question 2) results.

Study	Lev1	Lev2	Lev3	Lev4	Lev5	Lev6
S1			Yes		Yes	
S2			Yes			Yes
S3		Yes		Yes		
S4				Yes		
S5					Yes	
S6					Yes	
S7					Yes	
S8		Yes				
S9			Yes		Yes	
S10				Yes		
S11		Yes				
S12				Yes		
S13			Yes		Yes	
S14		Yes	Yes			
S15	Yes					
S16		Yes				
S17		Yes				
S18		Yes				
S19			Yes		Yes	
S20		Yes				
S21		Yes				
S22			Yes		Yes	
S23		Yes	Yes			
S24						
S25	Yes					
S26		Yes				
S27	Yes					
S28		Yes				
S29			Yes	Yes		
S30		Yes				
S31		Yes	Yes			
S32			Yes			
S33		Yes				
S34					Yes	
S35			Yes		Yes	
S36					Yes	
S37	Yes					
S38					Yes	
S39		Yes				
S40		Yes				
S41					Yes	
S42						Yes
S43		Yes	Yes			
S44					Yes	
S45						Yes
S46					Yes	
S47					Yes	
S48		Yes			Yes	
S49		Yes		Yes		

Lev1: No evidence, Lev2: Toy examples, Lev3: Expert opinions, Lev4: Academic studies, Lev5: Industrial studies, Lev6: Industrial practice.

Table 7
RQ3 (research question 3) results.

Study	RQ3.1	RQ3.2	RQ3.3	RQ3.4	RQ3.5	RQ3.6	RQ3.7	RQ3.8	RQ3.9	RQ3.10	RQ3.11
S1	Yes	Yes	Yes	Yes	No	Yes	No	Yes	Yes	Yes	No
S2	No	No	Yes	Yes	No	No	No	No	Yes	Yes	Yes
S3	Yes	Yes	Yes	No	No	No	No	No	No	Yes	No
S4	Yes	Yes	Yes	Yes	No	Yes	No	No	Yes	No	No
S5	Yes	Yes	Yes	Yes	No	No	No	No	No	Yes	No
S6	Yes	Yes	Yes	Yes	No	No	No	Yes	Yes	Yes	Yes
S7	Yes	Yes	Yes	Yes	No	Yes	No	No	Yes	Yes	No
S8	Yes	Yes	Yes	No	No	No	No	No	No	Yes	No
S9	Yes	Yes	Yes	Yes	No	Yes	No	Yes	Yes	Yes	Yes
S10	Yes	Yes	Yes	No	No	Yes	No	No	Yes	No	No
S11	Yes	Yes	No	N/A	No	No	No	No	Yes	Yes	Yes
S12	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	No
S13	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	No
S14	Yes	Yes	Yes	No	No	No	No	Yes	No	Yes	No
S15	Yes	Yes	Yes	No	No	No	No	Yes	No	Yes	No
S16	Yes	Yes	Yes	Yes	No	No	No	No	Yes	No	No
S17	Yes	Yes	Yes	Yes	No	No	No	No	Yes	Yes	No
S18	Yes	Yes	N/A	N/A	No	No	No	No	Yes	Yes	Yes
S19	Yes	Yes	Yes	Yes	No	Yes	No	Yes	Yes	Yes	No
S20	Yes	Yes	Yes	Yes	No	No	No	No	No	Yes	No
S21	Yes	Yes	Yes	Yes	No	No	No	No	Yes	Yes	Yes
S22	Yes	Yes	Yes	Yes	No	Yes	No	Yes	Yes	Yes	No
S23	Yes	Yes	No	N/A	No	No	Yes	No	Yes	Yes	Yes
S24	Yes	Yes	No	No	No	No	No	No	No	Yes	No
S25	Yes	No	Yes	No	No	No	No	No	No	Yes	No
S26	Yes	Yes	Yes	No	No	No	No	Yes	No	Yes	No
S27	Yes	Yes	Yes	No	No	No	No	No	No	Yes	No
S28	Yes	Yes	Yes	Yes	No	Yes	Yes	No	Yes	Yes	No
S29	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes
S30	Yes	Yes	Yes	Yes	No	Yes	No	No	No	Yes	Yes
S31	Yes	Yes	Yes	No	No	No	No	No	Yes	Yes	No
S32	Yes	Yes	No	N/A	No	No	No	No	No	Yes	Yes
S33	Yes	Yes	Yes	No	No	No	No	No	No	Yes	No
S34	Yes	Yes	Yes	Yes	No	No	No	No	Yes	No	No
S35	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes
S36	Yes	Yes	Yes	Yes	No	No	No	Yes	Yes	Yes	Yes
S37	Yes	Yes	Yes	Yes	No	No	No	No	No	No	No
S38	Yes	Yes	Yes	Yes	No	No	No	No	Yes	Yes	No
S39	Yes	Yes	Yes	Yes	No	No	No	No	No	Yes	No
S40	Yes	Yes	Yes	No	No	No	No	Yes	No	Yes	No
S41	Yes	Yes	Yes	Yes	No	No	No	No	Yes	Yes	No
S42	Yes	Yes	Yes	Yes	No	No	No	No	Yes	Yes	Yes
S43	Yes	Yes	Yes	No	No	No	No	No	No	Yes	No
S44	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes
S45	Yes	Yes	Yes	Yes	No	No	No	No	Yes	Yes	Yes
S46	Yes	Yes	Yes	Yes	No	No	No	No	Yes	Yes	Yes
S47	Yes	Yes	Yes	Yes	No	No	No	No	Yes	Yes	Yes
S48	Yes	Yes	Yes	Yes	No	No	No	Yes	Yes	Yes	No
S49	Yes	Yes	Yes	No	No	No	No	No	No	Yes	No

Appendix A. Selected primary studies and raw assessments

See Tables 4–7.

References

[1] P. Clements, L. Northrop, Software Product Lines: Practices and Patterns, Addison-Wesley, 2001.

[2] P. Zave, Classification of research efforts in requirements engineering, ACM Computing Surveys 29 (4) (1997) 315–321.

[3] J. Bosch, Design and Use Software Architectures: Adopting and Evolving a Product-Line Approach, Addison-Wesley, 2000.

[4] J. Coplien, D. Hoffman, D.M. Weiss, Commonality and variability in software engineering, IEEE Software 15 (6) (1998) 37–45.

[5] C.W. Krueger, Easing the transition to software mass customization, in: Proceedings of the 4th International Workshop on Software Product-Family Engineering (PFE 2001), Bilbao, Spain, October 3–5, 2001, pp. 282–293.

[6] B.A. Kitchenham, T. Dybå, M. Jorgensen, Evidence-based software engineering, in: Proceedings of the 26th International Conference on Software Engineering (ICSE 2004), 2004, pp. 273–281.

[7] M. Khurum, T. Gorschek, A systematic review of domain analysis solutions for product lines, Journal of Systems and Software 82 (12) (2009) 1982–2003.

[8] L.B. Lisboa, V.C. Garcia, D. Lucrédio, E.S. de Almeida, S.R. de Lemos Meira, R.P. de Mattos Fortes, A systematic review of domain analysis tools, Information and Software Technology 52 (1) (2010) 1–13.

[9] D.F.S. Neiva, RiPLE-RE: A Requirements Engineering Process for Software Product Lines, M.Sc. Dissertation, Universidade Federal de Pernambuco, Brazil, 2009.

[10] L. Chen, M.A. Babar, N. Ali, Variability management in software product lines: a systematic review, in: Proceedings of the 13th Software Product Line International Conference (SPLC 2009), San Francisco, CA, USA, 2009, pp. 81–90.

[11] B. Kitchenham, Procedure for Undertaking Systematic Reviews, Technical report TR/SE-0401, Computer Science Department, Keele University, 2004.

[12] B. Nuseibeh, S. Easterbrook, Requirements engineering: a roadmap, in: The Future of Software Engineering, 2000, pp. 35–46.

[13] J. Nicolás, A. Toval, On the generation of requirements specifications from software engineering models: a systematic literature review, Information and Software Technology 51 (2009) 1291–1307.

[14] T. Dybå, T. Dingsøyr, Strength of evidence in systematic reviews in software engineering, in: Proceedings of the Second International Symposium on Empirical Software Engineering and Measurement (ESEM 2008), 2008a, pp. 178–187.

[15] V. Alves, C. Schwanninger, L. Barbosa, A. Rashid, P. Sawyer, P. Rayson, C. Pohl, A. Rummler, An exploratory study of information retrieval techniques in domain analysis, in: Proceedings of the 12th Software Product Line Conference (SPLC 2008), Limerick, Ireland, 2008, pp. 67–76.

[16] M. Mannion, B. Keepence, D. Harper, Using viewpoints to define domain requirements, IEEE Software 15 (1) (1998) 95–102.

[17] P. Brereton, B.A. Kitchenham, D. Budgen, M. Turner, M. Khalil, Lessons from applying the systematic literature review process within the software engineering domain, Journal of Systems and Software 80 (2007) 571–583.
[18] G. Salton, A. Wong, C.S. Yang, A vector space model for automatic indexing, Communications of the ACM 18 (11) (1976) 613–620.
[19] M. Luisa, F. Mariangela, N.I. Pierluigi, Market research for requirements analysis using linguistic tools, Requirements Engineering Journal 9 (1) (2004) 40–56.
[20] K.C. Kang, S.G. Cohen, J.A. Hess, W.E. Novak, A.S. Peterson, Feature-Oriented Domain Analysis (FODA) Feasibility Study, Technical report CMU/SEI-90-TR-21, Software Engineering Institute, Carnegie Mellon University, 1990.
[21] I. Jacobson, M. Griss, P. Jonsson, Software Reuse: Architecture, Process, and Organization for Business Success, Addison-Wesley, 1997.
[22] D.E. Perry, S.E. Sim, S.M. Easterbrook, Case studies for software engineers, in: Proceedings of the 28th International Conference on Software Engineering (ICSE 2006), Shanghai, China, 2006, pp. 1045–1046.
[23] W.F. Tichy, F. Padberg, Empirical Methods in Software Engineering Research, in: Proceedings of the 29th International Conference on Software Engineering (ICSE 2007) Companion Volume, Minneapolis, MN, USA, 163–164, 2007.
[24] D. Atkins, D. Best, P.A. Briss, M. Eccles, Y. Falck-Ytter, S. Flottorp, G.H. Guyatt, R.T. Harbour, M.C. Haugh, D. Henry, S. Hill, R. Jaeschke, G. Leng, A. Liberati, N. Magrini, J. Mason, P. Middleton, J. Mrukowicz, D. O'Connell, A.D. Oxman, B. Phillips, H.J. Schünemann, T.T. Edejer, H. Varonen, G.E. Vist, J.W. Williams, S. Zaza, Grading quality of evidence and strength of recommendations, BMJ 328 (7454) (2004).
[25] T. Dybå, T. Dingsøyr, Empirical studies of agile software development: a systematic review, Information and Software Technology 50 (9-10) (2008) 833–859.
[26] B.A. Kitchenham, O.P. Brereton, D. Budgen, M. Turner, J. Bailey, S. Linkman, Systematic literature reviews in software engineering – a systematic literature review, Information and Software Technology 51 (2009) 7–15.
[27] M. Jørgensen, M. Shepperd, A systematic review of software development cost estimation studies, IEEE Transactions on Software Engineering 33 (1) (2007) 33–53.