

Six years of systematic literature reviews in software engineering: An updated tertiary study

Fabio Q.B. da Silva*, André L.M. Santos, Sérgio Soares, A. César C. França,
Cleviton V.F. Monteiro, Felipe Farias Maciel

Centre for Informatics, UFPE, Cidade Universitária, 50.740-560 Recife, PE, Brazil

ARTICLE INFO

Article history:
Available online 23 April 2011

Keywords:
Systematic reviews
Mapping studies
Software engineering
Tertiary studies

ABSTRACT

Context: Since the introduction of evidence-based software engineering in 2004, systematic literature review (SLR) has been increasingly used as a method for conducting secondary studies in software engineering. Two tertiary studies, published in 2009 and 2010, identified and analysed 54 SLRs published in journals and conferences in the period between 1st January 2004 and 30th June 2008.

Objective: In this article, our goal was to extend and update the two previous tertiary studies to cover the period between 1st July 2008 and 31st December 2009. We analysed the quality, coverage of software engineering topics, and potential impact of published SLRs for education and practice.

Method: We performed automatic and manual searches for SLRs published in journals and conference proceedings, analysed the relevant studies, and compared and integrated our findings with the two previous tertiary studies.

Results: We found 67 new SLRs addressing 24 software engineering topics. Among these studies, 15 were considered relevant to the undergraduate educational curriculum, and 40 appeared of possible interest to practitioners. We found that the number of SLRs in software engineering is increasing, the overall quality of the studies is improving, and the number of researchers and research organisations worldwide that are conducting SLRs is also increasing and spreading.

Conclusion: Our findings suggest that the software engineering research community is starting to adopt SLRs consistently as a research method. However, the majority of the SLRs did not evaluate the quality of primary studies and fail to provide guidelines for practitioners, thus decreasing their potential impact on software engineering practice.

© 2011 Elsevier B.V. All rights reserved.

Contents

1. Introduction	900
2. Previous studies	900
3. Method	901
3.1. Research questions	901
3.2. Research team	901
3.3. Decision procedure	901
3.4. Search process	902
3.5. Study selection	902
3.6. Quality assessment	903
3.7. Data extraction process	903
4. Data extraction results	903
5. Discussion of research questions	904
5.1. RQ1: How many SLRs were published between 1st January 2004 and 31st December 2009?	904
5.2. RQ2: What research topics are being addressed?	904

* Corresponding author. Tel.: +55 81 99044085; fax: +55 81 21268430.

E-mail addresses: fabio@cin.ufpe.br (F.Q.B. da Silva), alms@cin.ufpe.br (A.L.M. Santos), scbs@cin.ufpe.br (S. Soares), accf2@cin.ufpe.br (A.C.C. França), cvmf@cin.ufpe.br (C.V.F. Monteiro).

5.3.	RQ3: Which individuals and organisations are most active in SLR-based research?	906
5.4.	RQ4: Are the limitations of SLRs, as observed in the two previous studies, FE and OS, still an issue?	906
5.4.1.	Review topics and extent of evidence.	906
5.4.2.	Orientation towards the practice.	907
5.4.3.	Quality evaluation of primary studies.	907
5.4.4.	Use of guidelines	909
5.5.	RQ5: Is the quality of the SLRs improving?	910
6.	Limitations of this study	911
7.	Conclusions.	911
	Acknowledgements	912
	References	912

1. Introduction

In 2004, Kitchenham et al. [14] introduced the concept of evidence-based software engineering (EBSE) as a promising approach to integrate academic research and industrial practice in software engineering. Following this paper, Dybå et al. [8] presented EBSE from the point of view of the software engineering practitioner, and Jørgensen et al. [20] complemented it with an account of the teaching aspects of EBSE to university students.

By analogy with evidence-based medicine [25], five steps are needed to practice EBSE:

1. Convert the need for information (about the practice of software engineering) into answerable questions.
2. Identify, with maximum efficiency, the best evidence with which to answer these questions.
3. Appraise the evidence critically: assess its validity (closeness to the truth) and usefulness (practical applicability).
4. Implement the results of this appraisal in software engineering practice.
5. Evaluate the performance of this implementation.

The preferred method for implementing Steps 2 and 3 is systematic literature review (SLR). Kitchenham [15] adapted guidelines for performing SLRs in medicine to software engineering. Later, using concepts from social science [23], Kitchenham and Charters updated the guidelines [16]. The literature differentiates several types of systematic reviews [23], including the following:

- Conventional SLRs [23], which aggregate results about the effectiveness of a treatment, intervention, or technology, and are related to specific research questions such as *Is intervention I on population P more effective for obtaining outcome O in context C than comparison treatment C?* (resulting in the PICOC structure [23]). When sufficient quantitative experiments are available to answer the research question, a meta-analysis (MA) can be used to integrate their results [10].
- Mapping (or scoping) studies (MS) [1] aim to identify all research related to a specific topic, i.e., to answer broader questions related to trends in research. Typical questions are exploratory, e.g., *What do we know about topic T?*

Greenhalgh [11] emphasises that evidence-based practice is not only about reading papers and summarising their results in a comprehensive and unbiased way. It involves reading the *right* papers (those both valid and useful) and then changing behaviour in the practice of the discipline (in our case, software engineering). Therefore, EBSE is not only about performing high-quality SLRs and making them publicly available (Steps 2 and 3). All five steps should be performed for a practice to be considered evidence-based. Nevertheless, SLRs can play an important role in supporting research and education, and informing practice on the impact or effect of

technology. Therefore, information about how many SLRs are available in software engineering, where they can be found, which topic areas have been addressed, and the overall quality of available studies can greatly benefit the academic community as well as practitioners.

In this article, we performed a mapping study of SLRs in software engineering published between 1st July 2008 and 31st December 2009. Our goal was to analyse the available secondary studies and integrate our findings with the results of the two previous studies discussed in Section 3. Our work is classified as a tertiary study because we performed a review of secondary studies. The study protocol is presented in Section 3. Sections 4 and 5 discuss the extracted data and their analysis. Finally, the conclusions are presented in Section 6.

2. Previous studies

Two previous tertiary studies have been performed aiming to assess the use of SLRs in software engineering research and, indirectly, to investigate the adoption of EBSE by software engineering researchers [18,19]. Additionally, our team recently performed a critical appraisal of the SLRs reported in these two studies with respect to the types of research questions asked in published reviews [7]. In this section, we briefly describe these three studies and their relationships.

The first study, developed by Kitchenham et al. [18] (hereinafter called the Original Study; OS), found 20 unique studies reporting literature reviews that were considered systematic according to the authors. The study utilised a manual search of specific conference proceedings and journal papers for peer-reviewed articles published between 1st January 2004 and 30th June 2008. The OS identified several problems or limitations of the existing SLRs, as follows:

- A relatively large number of studies (40%, 8/20) investigated research methods or trends rather than performing technique evaluation, which should be the focus of a (conventional) systematic review [23].
- The diversity of software engineering topics was limited: seven related to cost estimation, three related to test methods, and three related to software engineering experiments.
- The number of primary studies cited was much larger in the mapping studies than in the SLRs.
- Relatively few SLRs assessed the quality of the primary studies.
- Relatively few papers provided advice oriented towards practitioners.

The last two problems identified above are of the most concern, as the stated purpose of evidence-based practice is to inform and advise practitioners on (high-quality) empirical evidence that can be used to improve their practice. We investigated whether these problems persist.

One limitation of the OS was that the search was manual and performed on a relevant but restricted set of sources. Therefore, relevant studies may have been missed, as was, in fact, the case according to the findings of the second tertiary study performed by Kitchenham et al. [19]. This study (hereinafter called the First Extension Study; FE) used an automatic search on five search engines and indexing systems and found 33 additional unique studies published between 1st January 2004 and 30th June 2008. The FE identified some improvements on the issues found in the OS; the number of SLRs was increasing, as was the overall quality of the studies. However, the problem remained that only a few SLRs followed a specific methodology, included practitioner guidelines, or evaluated the quality of primary studies. The authors also emphasised that researchers from the USA, the leading country in software engineering research, authored a very small number of SLRs and this could be interpreted as a sign of limitation on the adoption of evidence-based software engineering or that this adoption is mainly concentrated in European research groups.

Our group [7] subsequently reported the results of a study performed on the 53 SLRs presented in the OS and FE with the goal of assessing the SLRs with respect to the types of research questions asked in the selected reviews, how the questions were presented in the reports, and how the questions were used to guide the search of the primary studies. We found that over 65% of the research questions asked in the 53 reviews were exploratory and that only 15% investigated causality questions. Additionally, we found that half of the studies did not state their research questions explicitly, and for those that did, 75% did not use the questions to explicitly guide the search for primary studies.

In our study, we also compared the types of research questions asked and the classification of the studies presented in the OS and FE studies as systematic reviews or mapping studies. To classify the research questions, we used the terminology defined by Easterbrook et al. [9]. We found that most studies classified as mapping studies in OS and FE asked exploratory questions, and the vast majority of the studies that asked causal or relational questions were classified as (conventional) systematic reviews. Using these results, we adopted a systematic approach to classify the reviews as mapping studies or systematic reviews based on their types of research questions; the same approach was used in the current study to classify the reviews and thus enable the comparison of our results with those of the OS and FE.

3. Method

The research group that developed the OS and FE reviews intended to repeat their study at the end of 2009 to “track the progress of SLRs and evidence-based software engineering” [18]. During the 14th International Conference on Evaluation and Assessment in Software Engineering (EASE’2010) in May 2010, at Keele University, we discussed this extension with two members of the group, Pearl Brereton and David Budgen, and they said that the extension had not yet been performed. We informed them of our intention of performing this extension and integrating the results with the OS and FE findings. At that meeting, two methodological decisions were made. First, our extension would be performed independently of their work and with as little exchange of information as possible. Second, we would use, as closely as possible, the same protocol used in the FE. These decisions would assure that one study would not influence or bias the other, and we would be able to compare the results, as the protocols would be the same. Therefore, the method used in our study follows closely the protocol defined by Kitchenham et al. [17] and the structure used by Kitchenham et al. [19].

Hereinafter, we use OS/FE to refer to the combination of the results related to the 53 SLRs from the OS (20) and FE (33). Finally,

we use OS/FE + SE to refer to the combination of results of OS/FE with our findings, based on 67 SLRs, representing a total of 120 secondary studies.

3.1. Research questions

The five research questions (RQ) investigated in the SE were equivalent to the research questions used in the FE [19]. We performed minor adjustments and added subquestions as follows.

RQ1: How many SLRs were published between 1st January 2004 and 31st December 2009?

RQ1.1: How many SLRs were published between 1st January 2004 and 30th June 2008?

RQ1.2: How many SLRs were published between 1st July 2008 and 31st December 2009?

The subquestions of RQ1 investigate the development of SLRs in two separate periods. To answer RQ1.1, we used the results of OS/FE [18,19], whereas for RQ1.2, we performed the processes of search, selection, quality assessment, and data extraction defined in Sections 3.3–3.5. Similarly, we addressed the next questions considering the two time periods as we explicitly did for RQ1, searching for new evidence, combining with the results of the previous studies, and integrating all findings.

RQ2: What research topics are being addressed?

RQ3: Which individuals and organisations are most active in SLR-based research?

The fourth question in the FE was, “Are the limitations of SLRs, as observed in the original study, still an issue?” We changed the fourth question to the following:

RQ4: Are the limitations of SLRs, as observed in the two previous studies, FE and OS, still an issue?

Finally, we kept the fifth question unaltered:

RQ5: Is the quality of SLRs improving?

3.2. Research team

A team of six researchers, who co-authored this article, developed this study. Three of them, Fabio Silva, André Santos, and Sérgio Soares (referred to as R_1 , R_2 , and R_3 , respectively) are full-time lecturers. Cleviton Monteiro, and César França (R_4 and R_5) are Ph.D. students, and Felipe Farias (R_6) is a Master's student. All researchers are affiliated with the Centre for Informatics, Federal University of Pernambuco, Brazil.

3.3. Decision procedure

Three important activities in a systematic review require decisions about possibly conflicting situations: study selection, quality evaluation, and data extraction. It is thus recommended that such activities be performed by at least two researchers. Therefore, a process to support decision-making and consensus is necessary. For this study, we defined a decision and consensus procedure (DCP), shown in Fig. 1.

The procedure starts with a list of unevaluated studies as the input for a decision process (study selection, quality assessment, or data extraction), which R_1 randomly allocates to two researchers (R_i and R_j). After individual evaluation, the results (r_i and r_j) are integrated by R_4 and R_5 into an Agreement/Disagreement Table (ADT). Next, R_1 randomly allocates the results from the ADT to

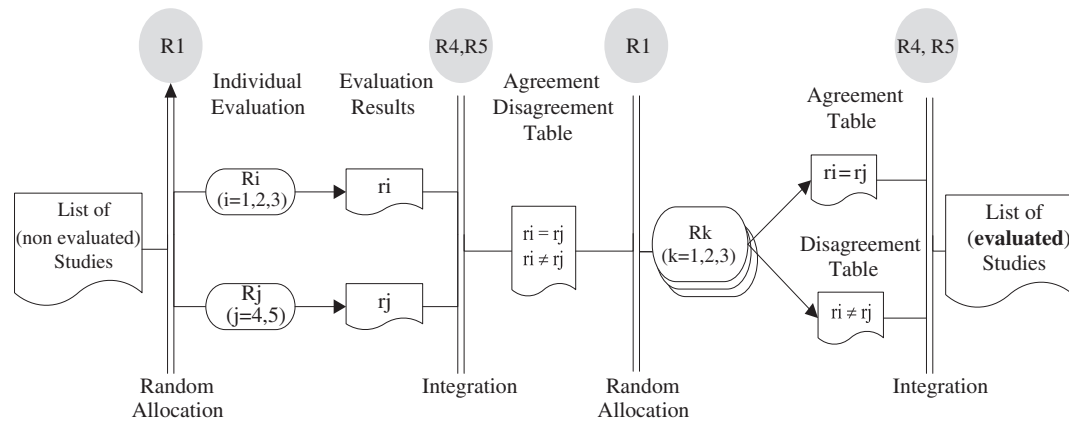


Fig. 1. The decision and consensus procedure (DCP).

the researchers, making sure that a different researcher (R_k) evaluates the results. The students do not participate in this stage. R_k judges the disagreements in the ADT and produces one of three results: an agreement for one of the previous decisions, a third result (r_k), or retention of the original disagreement. The remaining disagreements are resolved by a consensus of all six researchers. After consensus is reached, all results are integrated by R_4 and R_5 into the final list of evaluated studies.

3.4. Search process

We performed our search for peer-reviewed articles published between 1st July 2008 and 31st December 2009. Differing from OS and FE, we combined automatic and manual searches to increase coverage. The automatic search was performed by R_4 and R_5 on six search engines and indexing systems: ACM Digital Library, IEEEExplore Digital Library, Science Direct, CiteSeerX, ISI Web of Science, and Scopus. All searches were performed on the entire paper, including title and abstract, except for the ISI Web of Science, where the search was based only on title and topic due to limitations imposed by the search engine. This is the string used in the automatic search:

("software engineering") AND ("review of studies" OR "structured review" OR "systematic review" OR "literature review" OR "literature analysis" OR "in-depth survey" OR "literature survey" OR "meta analysis" OR "past studies" OR "subject matter expert" OR "analysis of research" OR "empirical body of knowledge" OR "overview of existing research" OR "body of published research" OR "evidence-based" OR "evidence based" OR "study synthesis" OR "study aggregation").

The syntax was the same for all engines, except for the ISI Web of Science, which required minor syntax changes due to the characteristics of the engine. The semantics of the strings remained unchanged. The search process was validated against the papers found in the OS and FE studies. Only three papers were not found by our search process. The study by Barcelos and Travassos [3] was obtained in the OS by directly consulting the authors. We searched the six engines looking for the article title directly and still did not find the study. The same happened with the study by Petersson et al. [22]. The study by Shaw and Clements [27] is indexed by the ACM digital library, but the authors use the term survey instead of review, and our search failed to find this article. Overall, we missed only one indexed article in a total of 53; we thus concluded that our search process was robust. The automatic search on the six engines returned 1389 documents. An initial fil-

tering was applied by reading their titles and abstracts and removing obviously irrelevant papers, resulting in a remaining 157 papers.

The lecturers (R_1 , R_2 , and R_3) performed a manual search of relevant journals and conference proceedings (Table 1). The source list was the same as in OS and FE except that since 2007, the International Symposium on Empirical Software Engineering and Metrics (ESEM) merged the International Symposium on Empirical Software Engineering (ISESE) and the International Symposium of Software Metrics (METRICS), which were used in the previous studies. The researchers searched the titles and abstracts of all published articles in each source. This search was done in parallel with the automatic search and produced 66 potentially relevant articles.

The lists from the automatic and manual search were merged and duplicates removed. The final list of potentially relevant studies contained 154 unique papers. This list was the input to the study selection activity.

3.5. Study selection

Study selection was performed by fully reading the 154 potentially relevant articles selected during the search process and excluding those articles that were not SLRs, i.e., literature reviews with defined research questions, search process, data extraction and data presentation, or were SLRs related to Information Systems, Human-computer Interaction or other Computer Science topics that were clearly not Software Engineering. When an SLR had been published in more than one journal or conference, both versions of the study were reviewed for purposes of data extraction, and the first publication was used in all time-based analyses used to track EBSE activity over time, which is consistent with the OS/FE as described in the original protocol [17]. Study selection was performed following the decision and consensus procedure described in Section 3.3.

After finishing the study selection, we performed a manual search of the reference list of each selected study and found two new studies [SE76, SE77]. The former was not found in the previous searches because the EASE Conference 2008 was on June 26th and 27th, outside of the time period of our study. However, we decided to include the study because it had also been missed in the OS/FE. The latter was not found by the initial automatic search of IEEEExplore. We attempted the search again, looking specifically for the paper's title, and could not find it. In this case, the manual search of the references proved to be an effective strategy, as otherwise, we would have missed one article. At the end of this stage, 77

Table 1
Manual search sources.

ACM Computer Surveys
ACM Transactions on Software Engineering Methodologies
Communications of the ACM
Empirical Software Engineering Journal
Evaluation and Assessment of Software Engineering
IEE Proceedings Software (now IET Software)
IEEE Software
IEEE Transactions on Software Engineering
Software Practice and Experience
Information and Software Technology
Int. Conference on Software Engineering
Int. Symposium on Empirical Software Engineering and Measurement
Journal of Systems and Software

articles were selected for quality assessment and data extraction (see Fig. 2).

3.6. Quality assessment

The OS and FE studies assessed the quality of the SLRs using the set of criteria defined by the Centre for Reviews and Dissemination (CDR) Database of Abstracts of Reviews of Effects (DARE) of York University [5]. This version of the DARE criteria was based on four questions:

- QA1: Are the review's inclusion and exclusion criteria described and appropriate?
- QA2: Is the literature search likely to have covered all relevant studies?
- QA3: Did the reviewers assess the quality/validity of the included studies?
- QA4: Were the basic data/studies adequately described?

The same scoring procedure used by Kitchenham et al. [19] was used in our study to assign scores to each question, which were then summed to yield the final quality score of the review. The answers for the quality assessment questions were obtained using the following criteria (the criteria for QA2 was modified from the original version used by Kitchenham et al. [19] to solve an ambiguity related to the use of automated search; the modifications are marked in bold face):

- QA1: Y (yes), the inclusion criteria are explicitly defined in the review; P (Partly), the inclusion criteria are implicit; N (no), the inclusion criteria are not defined and cannot be readily inferred.
- QA2: Y (yes), the authors have either searched four or more digital libraries and included additional search strategies or identified and referenced all journals addressing the topic of interest; P (Partly), the authors have searched **four digital libraries with no extra search strategies or three digital libraries (regardless of the use of extra search strategies)**, or searched a defined but restricted set of journals and conference proceedings; N, the authors have searched up to two digital libraries **(regardless of the use of extra search strategies)** or an extremely restricted set of journals.¹
- QA3: Y (yes), the authors have explicitly defined the quality criteria and extracted them from each primary study; P (Partly), the research question involves quality issues that are addressed by the study; N (No), no explicit quality assessment of individual

primary study has been attempted or quality data has been extracted but not used.

- QA4: Y (Yes), information is presented about each primary study so that the data summaries can clearly be traced to relevant studies; P (Partly), only summary information is presented about individual studies, e.g., studies are grouped into categories but it is not possible to link individual studies to each category; N (No), the results of the individual studies are not specified, i.e., the individual primary studies are not cited.

The scoring procedure was Y = 1, P = 0.5, and N = 0, as performed by Kitchenham et al. [19]. In the planning stage of our study, we noticed that the DARE criteria have changed and the current version has five questions [6]. Despite this change, we used the same quality criteria as OS and FE to allow for comparability of the results. The DCP was also used in the quality assessment producing the quality scores for all 77 papers.

To verify the consistency of our quality assessment with the one performed by Kitchenham et al. [19], we performed a blind assessment of 10 SLRs from the SE study and compared our scores with the results of Kitchenham et al. [19]. We agreed on all scores of eight studies, and disagreed on only one criterion in each of the two remaining studies, which was considered a very good agreement. Therefore, we are confident that our quality assessment can be compared to that presented by Kitchenham et al. [19].

3.7. Data extraction process

We extracted the following data from the 77 studies to answer the research questions:

- The **Year** of publication.
- The **Quality Score** of the study.
- The **Review Type**, related to whether the study is a conventional systematic literature review (SLR), a meta-analysis (MA) or a mapping study (MS).
- The **Review Scope**, related to the whether the study focused on a detailed technical question (RQ), on (research) trends in a particular software engineering topic area (SERT), or on research methods in software engineering (RT).
- The software engineering **Topic Area** addressed by the study.
- Whether the study explicitly **Cited EBSE papers** ([14,8,20]) or **Cited Guidelines** ([15,16]).
- The **Number of Primary** studies analysed in the SLR, as stated in the paper either explicitly or as part of tabulations.
- Whether the study **Included Practitioners Guidelines** explicitly as an identifiable part (section, table, etc.) of the paper.
- The **Source Type** in which the study was first reported (J = journal, C = Conference, WS = Workshop, BS = Book Series).

After analysing the results of the data extraction, we decided to exclude 10 studies: four were not on Software Engineering, three were reports of the results of two SLRs that appeared in the FE study, one was from 2010 (outside of the time period of this study), one was a shorter version of [SE01] published in another journal, and one received zero in the quality evaluation and did not have most of the required information. The DCP was used for data extraction, and at the end of this process, 67 articles were selected for further analysis and answering the research questions (see Appendix A).

4. Data extraction results

A summary of the data collected from the 67 SLRs in the above processes is shown in Table 2. Regarding the nature of the references to the EBSE papers and SLR Guidelines, similarly to the

¹ Note that scoring QA2 also requires the evaluator to consider whether the digital libraries were appropriate for the specific SLR.

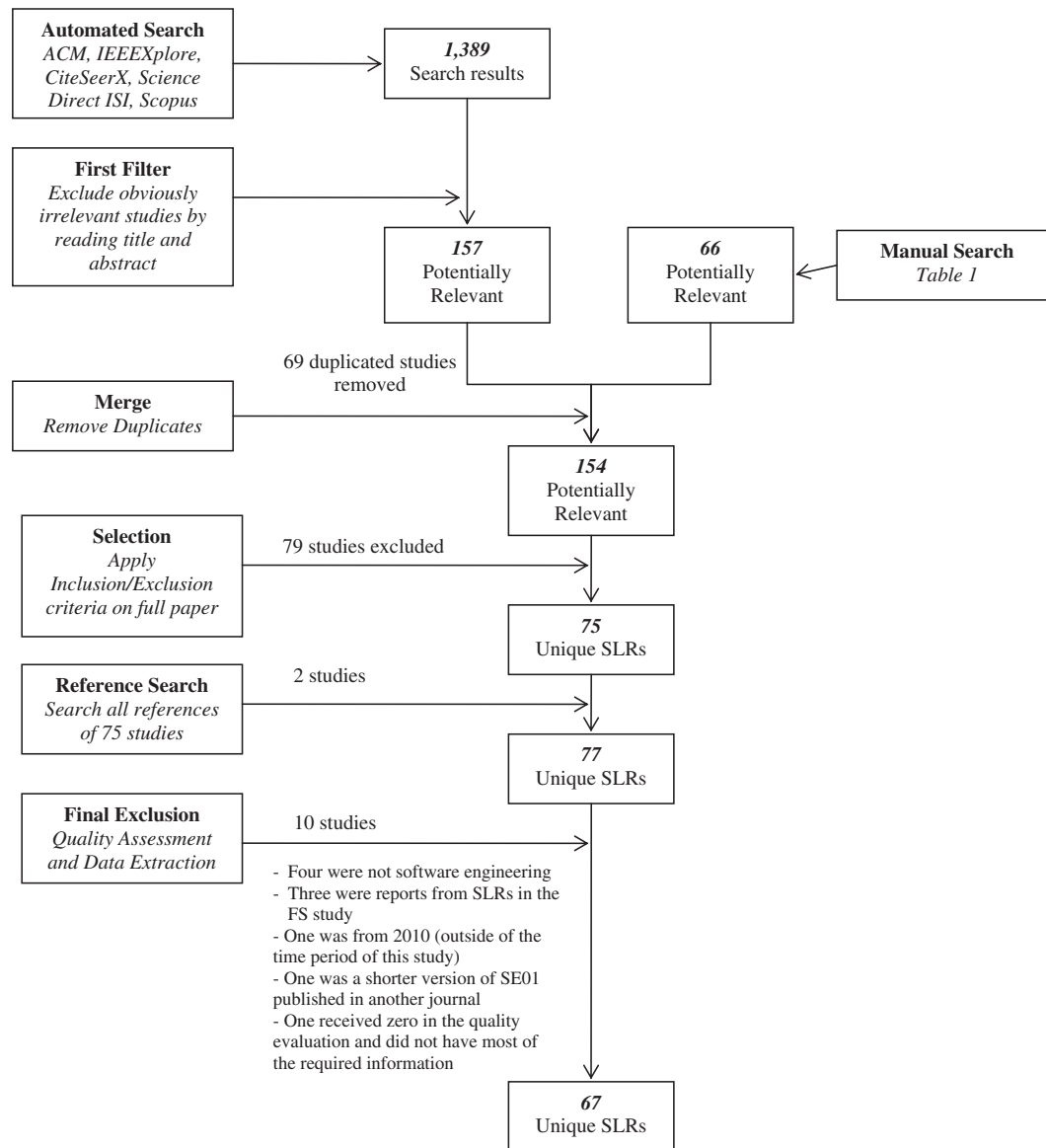


Fig. 2. Identification of included SLRs.

findings reported by Kitchenham et al. [18,19], all papers that cited the EBSE papers or the guidelines did so as a methodological justification for their study, so we considered all SLRs to be EBSE-positioned.

Table 3 shows the quality scores for each assessment question. We ordered the studies by the final score and divided the set into quartiles to allow for easier visualisation of how the entire set of studies performed in the assessment. The implications of the quality assessment results are discussed in Section 5.5.

5. Discussion of research questions

In this section, we address the research questions presented in Section 3.1. We show the results of our study (SE), compare them with the findings of OS/FE, and integrate the results (OS/FE + SE).

5.1. RQ1: How many SLRs were published between 1st January 2004 and 31st December 2009?

Table 4 shows the growth in published SLRs since 2004. The OS/FE studies found 53 studies between 2004 and June 2008

(4.5 years), and our extension (SE) found 67 studies between July 2008 and December 2009 (1.5 years). The studies published in 2009 account for 43% (51/120) of the total.

Table 4 also shows that the number of SLRs that cite either the EBSE papers or the SLR guidelines also increased in absolute number and also as a percentage of the studies in a given year. In fact, in the SE, 80% (53/67) of the SLRs cited the EBSE paper, the SLR Guidelines, or both.

5.2. RQ2: What research topics are being addressed?

As shown in Table 2, the 67 reviews in SE addressed 24 different software engineering topics, 14 of which were not addressed in the OS/FE. The most frequent topics in our SLRs were the following: Requirements Engineering (eight studies), Distributed Software Development (8), Software Product Line (7), Software Testing (6), Empirical Research Methods (5), Software Maintenance and Evaluation, and Agile Software Development (4).

To evaluate the coverage of software engineering topics, we used the same rationale used by Kitchenham et al. [19], and therefore, considered each SLR's relevance to education and practice by

Table 2

Systematic literature reviews in software engineering between July 2008 and December 2009.

Study Ref. (N = 67)	Year	Quality score	Review type	Review focus	Review topic	Cited EBSE paper	Cited guidelines	Number primary studies	Practitioners guidelines	Paper type
[SE01]	2008	4	MS	SERT	Human Aspects	N	Y ^d	92	N	J
[SE02]	2008	4	SLR	RT	Knowledge Management	Y ^{a,b}	Y ^d	68	Y	J
[SE03]	2008	1.5	MS	RT	Research Topics in Software Engineering	N	N	691	N	J
[SE04]	2008	1	MS	SERT	Software Project Management	N	N	48	N	C
[SE05]	2008	4	MS	SERT	Agile Software Development	N	Y ^f	36	Y	J
[SE08]	2008	2	MS	SERT	Software Testing	N	Y ^h	14	Y	C
[SE09]	2008	2	MS	SERT	Requirements Engineering	N	Y ^e	240	N	C
[SE10]	2008	1	MS	SERT	Usability	N	Y ^f	51	Y	C
[SE11]	2008	2.5	MS	SERT	Software Process Improvement	N	Y ^f	50	Y	C
[SE12]	2008	1.5	MS	SERT	UML	N	Y ^f	33	N	C ⁱ
[SE13]	2008	1	SLR	RT	Distributed Software Development	N	N	12	N	C
[SE14]	2008	3	SLR	RQ	Usability	N	Y ^{e,h}	63	Y	J
[SE18]	2009	4	MS	SERT	Software Testing	N	Y ^f	35	N	J
[SE19]	2009	3.5	SLR	RT	Software Testing	Y ^{a,b}	Y ^{f,h}	64	N	J
[SE20]	2009	2.5	MS	SERT	Software Maintenance and Evolution	N	Y ^{e,f}	34	N	J
[SE21]	2009	2.5	MS	SERT	Requirements Engineering	N	Y ^d	58	N	C
[SE22]	2009	2	SLR	RQ	Agile Software Development	Y ^a	Y ^{d,f}	9	N	C
[SE23]	2009	2	MS	RQ	Design Patterns	Y ^a	Y ^f	4	N	C
[SE24]	2009	2	MS	SERT	Software Maintenance and Evolution	N	Y ^d	12	Y	C
[SE25]	2009	2	MS	SERT	Risk Management	N	Y ^{e,g}	80	N	J
[SE26]	2009	2	MS	SERT	Software Fault Prediction	N	N	74	N	J
[SE27]	2009	3	MS	SERT	Software Product Line	N	Y ^f	34	N	C
[SE28]	2009	3	MS	SERT	Software Product Line	Y ^a	Y ^f	97	N	C
[SE29]	2009	1.5	MS	SERT	Requirements Engineering	N	N	46	N	C ⁱ
[SE30]	2009	3	MS	SERT	Software Maintenance and Evolution	N	Y ^d	176	N	J
[SE32]	2009	1.5	SLR	RT	Empirical Research Methods	Y ^a	Y ^d	16	N	C ⁱ
[SE33]	2009	1.5	SLR	RQ	Software Security	N	Y ^f	64	N	C
[SE34]	2009	2.5	MS	SERT	Empirical Research Methods	N	N	8	N	J
[SE35]	2008	3	SLR	RQ	Software Testing	N	Y ^{f,h}	28	N	C
[SE36]	2009	3.5	MS	SERT	Human Aspects	Y ^a	Y ^d	92	N	J
[SE37]	2009	3	MA	RQ	Agile Software Development	Y ^b	Y ^f	18	Y	J
[SE38]	2009	3	MS	SERT	Context Aware Systems	N	N	237	N	J
[SE39]	2009	3.5	SLR	RQ	Software Maintenance and Evolution	N	Y ^d	18	Y	C
[SE40]	2009	4	MS	SERT	Distributed Software Development	N	Y ^f	20	Y	C
[SE42]	2009	2.5	SLR	RT	Requirements Engineering	N	Y ^f	97	Y	J
[SE43]	2009	2.5	MS	SERT	Distributed Software Development	N	Y ^f	78	Y	J
[SE44]	2009	2	MS	SERT	Distributed Software Development	N	Y ^f	98	Y	C
[SE45]	2009	2	MS	SERT	Distributed Software Development	N	Y ^d	122	Y	C
[SE46]	2009	4	MS	SERT	Software Product Line	N	Y ^e	89	N	J
[SE47]	2009	3	MS	SERT	Software Product Line	N	Y ^d	23	N	C
[SE48]	2009	3.5	MS	SERT	Risk Management	Y ^a	N	27	N	C
[SE49]	2009	2	MS	SERT	Requirements Engineering	N	N	36	Y	C
[SE50]	2009	3	MS	SERT	UML	N	Y ^{e,g}	44	N	J
[SE51]	2009	3	SLR	RQ	Software Cost Estimation	N	N	12	N	C
[SE52]	2009	4	SLR	RQ	Software Development	N	Y ^d	5	Y	C ⁱ
[SE53]	2009	3	MS	SERT	Software Development	N	Y ⁺	40	Y	J
[SE54]	2009	2.5	MS	SERT	Software Product Line	N	Y ^d	39	N	C
[SE55]	2009	4	MS	SERT	Requirements Engineering	N	Y ^{e,g}	24	Y	J
[SE56]	2009	2	MS	SERT	Distributed Software Development	N	Y ^h	72	Y	J
[SE57]	2009	3	MS	SERT	Agile Software Development	Y ^b	Y ^d	50	N	C
[SE58]	2009	1	MS	SERT	Requirements Engineering	Y ^c	N	22	N	C
[SE59]	2009	4	SLR	RQ	Software Maintenance and Evolution	N	Y ^{d,e}	15	Y	C
[SE60]	2009	2	MS	SERT	Software Architecture	N	Y ^d	11	N	C ⁱ
[SE62]	2009	2.5	MS	SERT	Empirical Research Methods	N	Y ^e	63	N	WS
[SE63]	2009	4	MS	SERT	Requirements Engineering	Y ^b	Y ^{d,g}	149	Y	J
[SE64]	2009	2	MS	SERT	Software Testing	N	Y ^h	27	N	C
[SE65]	2008	2.5	MS	SERT	Software Product Line	N	Y ^d	17	Y	C
[SE66]	2009	1.5	MS	SERT	Software Testing	N	Y ^d	78	Y	J
[SE67]	2009	1.5	MS	SERT	Distributed Software	N	Y ^{d,g}	12	N	C ⁱ

(continued on next page)

Table 2 (continued)

Study Ref. (N = 67)	Year	Quality score	Review type	Review focus	Review topic	Cited EBSE paper	Cited guidelines	Number primary studies	Practitioners guidelines	Paper type
[SE68]	2009	2.5	MS	SERT	Development Service Oriented Systems Engineering	N	Y ^e	51	N	J
[SE70]	2009	3	MS	SERT	Software Evaluation and Selection	N	N	60	N	J
[SE71]	2009	2	SLR	RT	Empirical Research Methods	N	Y ^d	103	N	J
[SE72]	2009	2	SLR	RQ	Software Development	N	Y ^d	122	Y	BS
[SE74]	2009	0	MS	SERT	Empirical Research Methods	N	N	299	N	J
[SE75]	2009	3.5	MS	SERT	Human Aspects	N	Y ^d	92	N	J
[SE76]	2008	3	MS	SERT	Distributed Software Development	N	Y ^d	26	N	C
[SE77]	2008	3	MS	SERT	Software Product Line	N	Y ^f	19	N	C

^a [14].^b [8].^c [24].^d [15].^e [13].^f [16].^g [4].^h [12].ⁱ Short paper.

relating the topics covered with the Software Engineering 2004 Curriculum Guidelines for Undergraduate Degree Programs [28] and the Software Engineers' Book of Knowledge (SWEBOK) [2]. In this way, our findings can be directly compared to those found by Kitchenham et al. [19]. In Table 5, the relationships between the SLRs and the SE Curriculum and the SWEBOK are shown. To make the presentation clearer, we omitted from this table the SLRs that are only of interest to academics.

In Table 6, the distribution of SLR topics over the 2004 SE Curriculum and the SWEBOK are shown and compared to the findings of Kitchenham et al. [19]. This comparison shows that the coverage has increased but it is still very sparse for both the SE Curriculum and the SWEBOK; it also shows that the intersection between the previous OS/FE studies and the SE is very low, meaning that the more recent reviews addressed new topics. Finally, Software Configuration Management and Software Quality were not addressed by any of the 120 SLRs in the OS/FE + SE set of studies.

5.3. RQ3: Which individuals and organisations are most active in SLR-based research?

In the OS, a single researcher, Magne Jørgensen, from the Simula Lab, Norway, who was involved in eight studies, dominated the SLR publications. At the organisational level, Simula researchers contributed to 11 studies, just over half of the total. The FE showed a trend of reducing this concentration, as more researchers from different organisations and from other parts of the world began to adopt SLRs as a research method. In total, 103 researchers from 17 countries and 46 organisations were involved in the development of SLRs in the OS/FE.

In our study, the trend of the reduction in the concentrations of researchers, organisations, and countries was observed to continue. The number of researchers grew to 159, representing a 50% increase, which appears high when we consider that there was a 26% increase in the number of SLRs compared to the OS/FE. This indicates that the number of authors per study increased. In fact, we found that the percentage of studies co-authored by 3 or more researchers grew from 58% (31/53) to 67% (45/67), and the percentage of studies authored by a single researcher fell from 13% (7/53) to 1% (1/67). This indicates an evolution, as the SLR guidelines and literature emphasise the need for at least two researchers to perform an SLR to assure higher levels of quality, reduce bias, and increase the reliability of the results.

Another finding is that the number of researchers involved in more than one SLR also increased. In the OS, only five researchers were involved in more than three studies; in the FE, another seven researchers co-authored three or more studies, and in the SE, 10 new researchers entered this group. Furthermore, considering the OS/FE + SE dataset, there were 24 researchers who co-authored two studies, and 125 were involved in only one study. Table 7 lists the 21 researchers that have co-authored three or more SLRs since 2004. This is an indication that, at least for this group of researchers, the use of SLRs has gone from being a one-off activity to become part of the research methods regularly employed by these researchers.

In terms of affiliations, in the SE, we found 55 organisations with researchers involved in developing SLRs, which, combined with the 43 organisations from the OS/FE, total 90 distinct organisations involved since 2004. The countries in which these organisations are located also increased in number and become more widely distributed among the various regions of the world. The number of countries grew from 17 in the OS/FE to 25 in the OS/FE + SE, with eight new countries in the SE study. Asian countries, which did not appear in the OS/FE, contributed to 10 studies 15% (10/67) in the SE. Only 2 countries that appeared in the OS/FE were not found in the SE (Israel and Colombia). As shown in Table 8, researchers affiliated with European organisations still performed the vast majority of studies, a trend that remains since the OS. The participation of USA researchers can still be considered low, accounting for fewer than 12% (14/120) of the studies.

Altogether, this seems to indicate that SLRs are becoming more widespread in the scientific community. More researchers are using SLRs as a research method, and this use is spreading beyond Europe, where the majority of the promoters of EBSE and Systematic Reviews reside.

5.4. RQ4: Are the limitations of SLRs, as observed in the two previous studies, FE and OS, still an issue?

Some limitations of the SLRs identified in the OS and FE are discussed in this section.

5.4.1. Review topics and extent of evidence

As discussed in Section 5.2, the number of topics in software engineering covered by SLR and MS has increased since the OS and FE studies. There is no longer a concentration on a single topic

Table 3
Quality assessment scores.

Study Ref.	QA1	QA2	QA3	QA4	Final score	Quartiles
[SE01]	1	1	1	1	4	4th
[SE02]	1	1	1	1	4	
[SE05]	1	1	1	1	4	
[SE18]	1	1	1	1	4	
[SE35]	1	1	1	1	4	
[SE40]	1	1	1	1	4	
[SE46]	1	1	1	1	4	
[SE52]	1	1	1	1	4	
[SE55]	1	1	1	1	4	
[SE59]	1	1	1	1	4	
[SE63]	1	1	1	1	4	
[SE19]	1	1	0.5	1	3.5	
[SE36]	1	1	0.5	1	3.5	
[SE39]	1	1	0.5	1	3.5	
[SE48]	1	1	1	0.5	3.5	
[SE75]	1	1	1	0.5	3.5	
[SE14]	1	1	0	1	3	3rd
[SE27]	1	1	0	1	3	
[SE28]	1	1	1	0	3	
[SE30]	1	1	0	1	3	
[SE37]	1	1	0	1	3	
[SE38]	1	1	0	1	3	
[SE47]	1	1	0	1	3	
[SE50]	1	1	0	1	3	
[SE51]	1	1	0	1	3	
[SE53]	1	1	0	1	3	
[SE57]	1	1	0	1	3	
[SE70]	1	1	0.5	0.5	3	
[SE76]	1	1	0	1	3	
[SE77]	1	1	0	1	3	
[SE11]	1	1	0	0.5	2.5	2nd
[SE20]	1	0.5	0	1	2.5	
[SE21]	1	1	0	0.5	2.5	
[SE34]	0	0.5	1	1	2.5	
[SE42]	1	0	0.5	1	2.5	
[SE43]	1	0.5	0	1	2.5	
[SE54]	1	0.5	0	1	2.5	
[SE62]	1	0	1	0.5	2.5	
[SE65]	1	0.5	0	1	2.5	
[SE68]	1	1	0.5	0	2.5	
[SE08]	0.5	0.5	0	1	2	1st
[SE09]	1	1	0	0	2	
[SE22]	1	0.5	0	0.5	2	
[SE23]	1	1	0	0	2	
[SE24]	1	1	0	0	2	
[SE25]	1	1	0	0	2	
[SE26]	1	1	0	0	2	
[SE44]	1	1	0	0	2	
[SE45]	1	1	0	0	2	
[SE49]	1	1	0	0	2	
[SE56]	0	1	0	1	2	
[SE60]	1	1	0	0	2	
[SE64]	0.5	1	0	0.5	2	
[SE71]	1	1	0	0	2	
[SE72]	0	0	1	1	2	
[SE03]	0.5	1	0	0	1.5	
[SE12]	1	0.5	0	0	1.5	
[SE13]	1	0	0.5	0	1.5	
[SE29]	1	0.5	0	0	1.5	
[SE32]	0.5	1	0	0	1.5	
[SE33]	0.5	1	0	0	1.5	
[SE66]	0.5	1	0	0	1.5	
[SE67]	0	0.5	0	1	1.5	
[SE04]	0	0	0	1	1	
[SE10]	1	0	0	0	1	
[SE58]	0	0	1	0	1	
[SE74]	0	0	0	0	0	

(Software Cost Estimation), but there is still a concentration on six topics that were addressed by 55% of the reviews: Empirical Research Methods (16 studies), Software Cost (13), Requirements Engineering (10), Distributed Software Development (9), Software

Table 4
Number of SLRs per year.

Year	Number of SLR			Number of EBSE positioned SLR			
	OS/FE	SE	Total	OS/FE	SE	Total	% ^a
2004	6		6	1		1	17
2005	11		11	5		5	45
2006	9		9	6		6	67
2007	15		15	9		9	60
2008	12	16	28	10	12	22	79
2009		51	51		41	41	80
Total	53	67	120	31	53	84	70

^a Total EBSE positioned SLR/Total SLR in the same year.

Development (in general) (9), Software Testing (9), and Software Maintenance and Evolution (7).

The FE reported a reduction on the proportion of studies focusing on research methods between those in the OS (40%) and the new studies found in the FE (18%). In the SE, this trend was not observed, as we identified 27% of the studies (18/67) as being focused on research methods or primarily aimed at researchers. In the combined dataset of the OS/FE and our study, reviews of Empirical Research Methods are the most frequent topic of study, being addressed by over 13% (16/120) of the SLRs.

Consistent with the OS and FE studies, mapping studies (MS) analysed more primary studies than conventional systematic reviews (Table 9).

We found proportionally more MSs (82%, 55/67) than in the combined OS/FE data (32%, 17/53). Conversely, we found proportionally fewer conventional SLRs (18%, 12/67) than in the OS/FE (68%, 36/53). Two reasons may account for this difference. First, we classified the studies using the method presented by Da Silva et al. [7], and the researchers in the OS and FE used an unreported method. In fact, using the results of Da Silva et al. [7], the proportions in the OS/FE studies changed to 72% MS and 38% SLR, much closer to the proportions found in the SE. Moreover, the OS study did not distinguish between MSs and SLRs, classifying all studies as SLR, which increased the number of SLRs in the OS/FE studies.

Second, as described in Section 5.3, we found that an increasing number of newcomers (59), that is, researchers performing a systematic review for the first time, published reviews in new topic areas. Performing an MS of a topic area is a natural first step in research, especially if the area is more recently developed, for instance, agile development or distributed software development.

5.4.2. Orientation towards the practice

Twenty reviews in the SE study addressed research questions of possible interest to practitioners, including 11 that directly addressed technical evaluation questions (RQ). Furthermore, 36% (24/67) of the SE studies provided guidelines for practitioners, either explicitly or implicitly. These figures show an increase in the number of SLRs providing practitioner guidelines in comparison with the OS and FE studies, showing an increase in the orientation of the SLRs towards the practice of software engineering (Table 10).

However, 58% (39/67) of the reviews in the SE addressed trends in software engineering research that are only of indirect interest to practitioners, and eight studies investigated research methods of no interest in practice.

5.4.3. Quality evaluation of primary studies

The proportion of SLRs that undertook the evaluation of the quality of the primary studies increased when comparing the SE and OS/FE studies, as shown in Table 11. Although this indicates an improvement, the number of reviews performing a full and

Table 5
Relationships between SLRs and SE undergraduate Curriculum and SWEBOK.

Study Ref.	Review type	Quality score	Topic area	Useful for education	Useful for practitioner	Why?	SE Curriculum	SWEBOK
[SE01]	MS	4	Human Aspects	No	Possibly	Aimed at researchers, but solutions and limitations can inform practice		N/A
[SE02]	MS	4	Knowledge Management	No	Yes	Aimed at practitioners rather than undergraduates		Process Implementation and Change, Chapter 9, Section 1
[SE04]	MS	1	Software Project Management	Yes	Possibly	Aimed at researchers, but solutions and limitations can inform practice	MGT.pp.5 Resource allocation	Resource Allocation. Chapter 8, Section 2.4
[SE05]	MS	4	Agile Software Development	No	Yes	Aimed at practitioners rather than undergraduates		Software Life Cycle Models. Chapter 9, Section 2.1
[SE08]	MS	2	Software Testing	No	Yes	Aimed at practitioners rather than undergraduates		Acceptance/qualification testing. Chapter 5, Section 2.2.1
[SE10]	MS	1	Usability	No	Yes	Aimed at practitioners rather than undergraduates		Usability testing. Chapter 5, Section 2.2.12
[SE11]	MS	2.5	Software Process Improvement	Yes	Yes	Aimed at practitioners and provides examples for undergraduate education	PRO.con.6	Software Process Management Cycle. Chapter 9, Section 1.2
[SE14]	MS	3	Usability	No	Yes	Aimed at practitioners rather than undergraduates		Usability testing. Chapter 5, Section 2.2.12
[SE20]	MS	2.5	Software Maintenance and Evolution	Possibly	Yes	Rather specialised for undergraduates, but can be provide practical examples for education	EVO.pro.5	Software Maintenance Measurement, Chapter 6, Section 2.4
[SE24]	MS	2	Software Maintenance and Evolution	No	Yes	Aimed at practitioners rather than undergraduates		Software Maintenance Measurement, Chapter 6, Section 2.4
[SE27]	MS	3	Software Product Line	Possibly	Possibly	Aimed at researchers, but can be used for undergraduate education and solutions and limitations can inform practice	DES.ar.5 Domain-specific architectures and product-lines	Families of Programs and Frameworks. Chapter 3, Section 3.3
[SE33]	MS	1.5	Software Security	No	Possibly	Aimed at researchers, but solutions and limitations can inform practice		Functional and Non-functional Requirements. Chapter 2, Section 1.3
[SE35]	SLR	3	Software Testing	Possibly	Possibly	Aimed at researchers, but can be used for undergraduate education and solutions and limitations can inform practice	VAV.tst.10 Regression Testing	Regression testing. Chapter 5, Section 2.2.6
[SE37]	MA	3	Agile Software Development	Possibly	Yes	Rather specialised for undergraduates, but can be provide practical examples for education	PRF.psy.1 Dynamics of working in teams/groups	Coding. Chapter 4, Section 3.3
[SE38]	MS	3	Context Aware Systems	No	Possibly	Aimed at researchers, but solutions and limitations can inform practice		Software Structure and Architecture. Chapter 3, Section 3.1
[SE39]	SLR	3.5	Software Maintenance and Evolution	Yes	Yes	Aimed at practitioners and provides examples for undergraduate education	EVO.pro.1 Basic concepts of evolution and maintenance	Software Maintenance. Chapter 6
[SE40]	SLR	4	Distributed Software Development	Possibly	Yes	Rather specialised for undergraduates, but can be provide practical examples for education	MGT Software Management	Software Engineering Management. Chapter 8
[SE42]	MS	2.5	Requirements Engineering	No	Yes	Aimed at practitioners rather than undergraduates		Software Requirements. Chapter 2
[SE43]	MS	2.5	Distributed Software Development	No	Yes	Aimed at practitioners rather than undergraduates		Software Engineering Management. Chapter 8
[SE44]	SLR	2	Distributed Software Development	No	Yes	Aimed at practitioners rather than undergraduates		Software Engineering Management. Chapter 8
[SE45]	SLR	2	Distributed Software Development	No	Yes	Aimed at practitioners rather than undergraduates		Software Engineering Management. Chapter 8
[SE46]	MS	4	Software Product Line	Possibly	Possibly	Aimed at researchers, but can be used for undergraduate education and solutions and limitations can inform practice	DES.ar.5 Domain-specific architectures and product-lines	Families of Programs and Frameworks. Chapter 3, Section 3.3
[SE47]	MS	3	Software Product Line	Possibly	No	Aimed at researchers, but can be used for undergraduate education	DES.ar.5 Domain-specific architectures and	

Table 5 (continued)

Study Ref.	Review type	Quality score	Topic area	Useful for education	Useful for practitioner	Why?	SE Curriculum	SWEBOK
[SE48]	MS	3.5	Risk Management	Possibly	Possibly	Aimed at researchers, but can be used for undergraduate education and solutions and limitations can inform practice	product-lines; and VAV.tst Testing MGT.pp.6 Risk management	Risk Management. Chapter 8, Section 2.5
[SE49]	MS	2	Requirements Engineering	Possibly	Yes	Rather specialised for undergraduates, but can be provide practical examples for education	MAA.er Eliciting requirements	Requirements Elicitation. Chapter 2, Section 3
[SE52]	SLR	4	Software Development	Possibly	Yes	Rather specialised for undergraduates, but can be provide practical examples for education	PRO.imp.2 Life cycle models	Software Life Cycle Models. Chapter 9, Section 2.1
[SE53]	MS	3	Software Development	Possibly	Yes	Rather specialised for undergraduates, but can be provide practical examples for education	MAA.af.3 Analysing quality	Software Engineering Methods. Chapter 10, Section 2
[SE54]	MS	2.5	Software Product Line	No	Possibly	Aimed at researchers, but solutions and limitations can inform practice		Families of Programs and Frameworks. Chapter 3, Section 3.3
[SE55]	MS	4	Requirements Engineering	No	Yes	Aimed at practitioners rather than undergraduates		Software Requirements Specification. Chapter 2, Section 5.3
[SE56]	MS	2	Distributed Software Development	No	Yes	Aimed at practitioners rather than undergraduates		Risk Management, Chapter 8, Section 2.5
[SE57]	SLR	3	Agile Software Development	No	Possibly	Aimed at researchers, but solutions and limitations can inform practice		Process Planning. Chapter 8, Section 2.1
[SE58]	SLR	1	Requirements Engineering	No	Possibly	Aimed at researchers, but solutions and limitations can inform practice		Software Requirements Tools, Chapter 10, Section 1.1
[SE59]	SLR	4	Software Maintenance and Evolution	No	Yes	Aimed at practitioners rather than undergraduates		Maintenance Cost Estimation. Chapter 6, Section 2.3
[SE60]	MS	2	Software Architecture	No	Possibly	Aimed at researchers, but solutions and limitations can inform practice		Architectural Structures and Viewpoints. Chapter 3, Section 3.1
[SE63]	MS	4	Requirements Engineering	Possibly	Yes	Rather specialised for undergraduates, but can be provide practical examples for education	MAA.rv Requirements validation	Requirements Validation, Chapter 2, Section 6
[SE65]	MS	2.5	Software Product Line	No	Yes	Aimed at practitioners rather than undergraduates		Families of Programs and Frameworks, Chapter 3, Section 3.3
[SE66]	MS	1.5	Software Testing	No	Yes	Aimed at practitioners rather than undergraduates		Specification-based techniques. Chapter 5, Section 3.2
[SE70]	MS	3	Software Evaluation and Selection	No	Possibly	Aimed at researchers, but solutions and limitations can inform practice		N/A
[SE72]	MS	2	Software Development	No	Yes	Aimed at practitioners rather than undergraduates		Process Measurement. Chapter 9, Section 4.1
[SE75]	SLR	3.5	Human Aspects	No	Possibly	Aimed at researchers, but solutions and limitations can inform practice		Resource Allocation, Chapter 8, Section 2.4
[SE77]	MS	3	Software Product Line	No	Possibly	Aimed at researchers, but solutions and limitations can inform practice		Families of Programs and Frameworks, Chapter 3, Section 3.3

explicit quality evaluation was still very low, amounting to only 21% (14/67) of the reviews in the SE study.

Three situations that might explain the low percentage of quality assessment of the primary reviews were observed. First, some researchers appear to have confused quality assessment with explicitly stating the inclusion/exclusion criteria of the primary studies and may therefore have thought that no further quality assessment was necessary (e.g., [SE42]). Second, in some cases, quality assessment was thought to be unnecessary because the primary studies were retrieved from “trustworthy sources” (e.g., peer-reviewed journals), and this was considered to be sufficient to guarantee the quality of the primary studies (e.g., [SE43]). Third,

in other cases, the search process found so few relevant studies that the researchers may have feared that applying quality criteria would leave them with no studies to analyse (e.g., [SE39]). Researchers performing systematic reviews should attempt to address these situations, as none are acceptable explanations for not performing quality assessment.

5.4.4. Use of guidelines

The use of guidelines and citations to the EBSE papers in the reviews increased in the SE with respect to the OS/FE studies, as shown in Table 12. The increase in the use of guidelines significantly correlated with quality of the SLRs by Kitchenham et al.

Table 6

Distribution of SLRs over 2004 SE Curriculum and SWEBOK sections.

		OS/FE		SE		OS/FE + SE		Increase
		SLRs	Subs	SLRs	Subs	SLRs	Subs	OS/FE → SE
2004 SE Curriculum sections								
Computing essentials	4							
Mathematical & Engineering Fundamentals	3	2	2			2	2	0
Professional Practice	3			1	1	1	1	1
Software Modelling & Analysis	7	3	2	3	3	6	4	2
Software Design	7	4	0	3	1	7	2	2
Software V & V	5	4	5	1	1	5	5	0
Software Evolution	2			2	1	1	1	1
Software Process	2	2	1	2	2	4	2	1
Software Quality	5							
Software Management	5	11	2	3	2	14	2	0
Systems and Application Specialties	42	2	1			2	1	0
Total subsections	85	28	13	15	11	42	20	7
SWEBOK Chapter								
Software Requirements	7	2	2	5	4	7	5	3
Software Design	6	4	2	7	1	11	2	0
Software Construction	3	1	1	1	1	2	1	0
Software Testing	5	1	1	5	2	6	2	1
Software Maintenance	4	2	2	4	1	6	2	0
Software Configuration Management	6							
Software Engineering Management	6	5	2	9	1	14	2	0
Software Engineering Process	4	4	3	5	3	9	5	2
Software Engineering Tools and Methods	2			2	2	2	2	2
Software Quality	3							
Total Sections	46	19	13	38	15	57	21	8

Table 7

Authors with three or more studies.

Authors	Country	OS/FE	SE	OS/FE + SE
Jørgensen	Norway	9		9
Guilherme Travassos	Brazil	3	3	6
Shepperd	UK	6		6
Tore Dyba	Norway	3	3	6
Muhammad Ali Babar	Ireland		5	5
Hannay	Norway	4		4
Sarah Beecham	UK		4	4
Sjøberg	Spain	4		4
Tony Gorschek	Sweden		4	4
Ambrosio Toval	Spain		3	3
Helen Sharp	UK		3	3
Hugh Robinson	UK		3	3
Juristo	Spain	3		3
Kampenes	Norway	3		3
Kitchenham	UK	3		3
Maya Daneva	The Netherlands		3	3
Moløkken-Østfold	Norway	3		3
Moreno	Spain	3		3
Nathan Baddoo	UK		3	3
Thelin	Sweden	3		3
Tracy Hall	UK		3	3

Table 8

SLRs per country.

Region	OS/FE	% (N = 53)	SE	% (N = 67)	OS/FE + SE	% (N = 121)
N. America	9	17	7	10	16	13
S. America	5	9	8	12	13	11
Europe	45	85	56	84	101	83%
Asia	0	0	10	15	10	8
M. East	1	2	0	0	1	1
Oceania	5	9	2	3	7	6

Table 9

Median numbers of primary studies per SLR and MA and MS.

Statistic	2004	2005	2006	2007	2008	2009
Median	26.5	19.5	32	21	26.5	20
SLR and MA	6	8	7	8	8	11
Median	–	119	403.5	137	49	54.5
MS	–	3	2	7	20	40

Table 10

Practitioner guidelines in the SLRs.

Practitioners guidelines	OS/FE	SE	OS/FE + SE
N	44	43	87
Y	9	24	33
Y%	17%	36%	28%

[19]. However, a regression test using the Cited Guidelines as the factor and quality score as the dependent variable showed no statistical significance for the entire set of SLRs ($N = 120$).

5.5. RQ5: Is the quality of the SLRs improving?

Kitchenham et al. [19] observed that the quality of the SLRs increased from the OS to the FE study. This trend continued in our study, with a steady increase in the mean quality scores of the studies in every year except 2007 (Table 13). Considering the 6-year period between 2004 and 2009, the increase in quality was 12.5%.

Table 3 presents the SLRs with respect to their quality score; close inspection shows that almost all studies in all four quartiles

performed well on both QA1 and QA2, which are related to inclusion and exclusion criteria and coverage of the search process, respectively. This result was likely due to the increasing number of studies in which the SLR guidelines were used to plan the studies. Studies in the fourth quartile performed well on all questions. Additionally, most studies in the first quartile failed on QA3 or QA4, which are respectively related to quality assessment of primary studies and the synthesis and presentation of findings related to individual primary studies.

Table 11

Evolution of quality evaluation of primary studies.

Evaluate quality of primary studies?	OS/FE	SE	OS/FE + SE
N	37	22	59
Y ^a	16	45	61
Y%	30%	67%	51%

^a Includes full evaluation (score = 1) and implicit evaluation (score = 0.5).**Table 12**

Use of guidelines.

Citation	OS/FE	% (N = 53)	SE	% (n = 67)	OS/FE + SE	% (N = 120)
EBSE	7	13	12	18	19	16
Guidelines	27	51	51 ¹	76	81	68
EBSE and guidelines	3	6	10	15	13	11

^a Excluding three studies that cited non-EBSE-related review guidelines.**Table 13**

SLR quality.

Cited guidelines?	Statistics	2004	2005	2006	2007	2008	2009
No	# SLRS	6	6	4	7	6	10
	Mean	2.08	2.33	2.00	1.79	1.50	2.15
	σ	1.07	0.52	1.08	0.81	0.55	1.08
Yes	# SLRS	0	5	5	8	22	41
	Mean	–	2.20	3.10	3.00	2.80	2.72
	σ	–	0.27	0.65	0.60	0.78	0.81
All	# SLR	6	11	6	15	28	51
	Mean	2.08	2.27	2.61	2.43	2.50	2.61
	σ	1.07	0.41	0.99	0.92	0.92	0.89
	Increase		5%	7.5%	–5%	2.5%	2.5%

We then compared the mean of the quality scores of the SLR with respect to three other factors. First, the SLRs that explicitly provided guidelines for practitioners had higher mean quality scores (Mean = 2.85, σ = 0.91) than those that did not provide guidelines (Mean = 2.38, σ = 0.83). Second, SLRs published in Journals had higher quality scores (Mean = 2.69, σ = 0.94) than the studies published in Conferences (Mean = 2.44, σ = 0.81). Third, the SLRs with scope RQ performed better (Mean = 2.88, σ = 0.76), than those with SERT (Mean = 2.41, σ = 0.91) and RT (Mean = 2.28, σ = 0.79). We performed a regression analysis using these three factors, and the result was statistically significant for a 95% confidence level, as follows: Guidelines for Practitioners (B = 0.183, std. error = 0.038, p = 0.000), Journal (B = 0.117, std. error = 0.041, p = 0.005) and RQ (B = 0.081, std. error = 0.036, p = 0.025).

Finally, we correlated the number of primary studies in an SLR with the quality score using Pearson's coefficient and found that the inverse correlation was significant (r = –0.204, N = 120, p = 0.05). As reported by Kitchenham et al. [19], SLRs addressing larger numbers of primary studies had lower quality scores than those with fewer primary studies. A possible explanation is that when faced with too many studies to analyse, researchers may opt not to perform quality assessment, and they may also have more difficulties in presenting a good synthesis and summary of evidence for each paper, thus scoring lower on quality questions QA3 and QA4.

6. Limitations of this study

Two major problems in SLRs are finding all the relevant studies and assessing their quality. In our study, we employed a mixed process approach to find relevant studies that combined an automatic search in search engines, manual search on relevant journals

and conference proceedings, and backward search, that is, searching for relevant studies in the references of previously selected studies. We checked the coverage of our automatic search and only failed to recover one study in a set of 51, which can be considered good coverage if the automatic search is complemented by manual procedures.

Quality assessment of the SLRs was performed by at least two researchers, and conflicts were resolved by a third researcher or by consensus in cases in which the third point of view was also conflicting. This multi-evaluator procedure increased our confidence on the reliability of our quality assessment. However, we found the scoring procedure to be too subjective for question QA4 and inconsistent for QA2. We solved the inconsistency problem by consulting the researchers that performed the OS/FE studies. The problems with question QA4 caused many disagreements between evaluators that were only solved in the consensus meeting.

Two researchers, following the process described in Fig. 1 of Section 3.2, performed data extraction independently. Nevertheless, on many occasions, we found that quality assessment and data extraction may have been compromised by the way most of the SLRs were reported. Many reviews did not present sufficient information, or the organisation of the presentation made it very difficult to locate the needed information in the extraction process. More specifically, we often found that review protocols were not described in sufficient detail, in particular, regarding the search process and quality assessment of the primary studies. In many cases, information was not explicitly stated in the reports and had to be inferred from the text. Despite our efforts to reach consensus during data extraction and quality assessment, it is possible that the extraction and quality assessment processes may have resulted in inaccurate data.

7. Conclusions

This tertiary study analysed 1455 articles, of which 67 were considered to be systematic literature reviews in software engineering with acceptable quality and relevance. Among these studies, 15 appeared relevant to the undergraduate educational curriculum, 40 appeared of possible interest to practitioners, and 26 were directed mainly to researchers. Furthermore, the 67 studies addressed 24 different software engineering topics, covering 33% (15/46) of the SWEBOK sections.

Our study shows three important changes in the study set from the previous tertiary studies [18,19]. First, the coverage of topics in software engineering increased, and the concentration in a few topics decreased. Second, the number of researchers and, consequently, organisations undertaking systematic reviews increased and became more globally distributed. Finally, we found proportionally more mapping studies than conventional systematic reviews in our study. Together, these results appear to indicate that systematic reviews are increasingly being considered as an important tool in performing unbiased and comprehensive literature mappings of the research in specific topics in software engineering.

However, our findings also show three major limitations with the current use of SLRs in software engineering. First, a large number of SLRs still do not assess the quality of their primary studies, and this is consistent with the findings in the previous tertiary studies. Second, the integration of the results of the primary studies was poorly conducted by many SLRs. In the set of 67 reviews, only one ([SE37]) used a meta-analysis [10] to synthesise quantitative studies and two ([SE05,SE68]) used meta-ethnography [21] in the synthesis of qualitative studies. Although meta-analysis is mentioned in other studies [SE01,SE19,SE22,SE28,SE35,SE56,SE71], they did not employ the technique. Apart from meta-analysis and meta-ethnography, no other form of meta-synthesis [26] was used.

We believe that the underlying problem is that the studies we analysed, in particular the mapping studies, attempted to combine and synthesise results from too diverse a set of primary studies without the use of a proper methodology.

Third, as was identified in the OS/FE studies, the number of SLRs providing guidelines to practitioners remains small. Furthermore, we could not identify from the reported data in the SLRs whether the investigated question originated in the industrial practice of software engineering or as an academic problem. Because the practical origin of the problem and practitioner guidelines are essential for developing Steps 1, 4, and 5 of the EBSE approach discussed in Section 1, we concluded that EBSE is not being fully realised in practice. Nevertheless, the number of SLRs is increasing, along with the number of researchers and organisations performing them. This may indicate that SLRs are being adopted as a research method for the discovery of gaps and trends that could guide academic research in software engineering. This was corroborated by the increase in the proportion of mapping studies, as these are typically directed towards exploratory investigation of research trends.

A systematic review can be updated and extended in at least three complementary ways. A temporal update can be performed to expand the timeframe for the publication of the primary studies, without major changes in the original review protocol. This tertiary study is an example of a temporal update of a previous review [19]. A search extension can be performed to expand the number of sources and the search strategies (manual or automated) within the same timeframe as the original review to increase the coverage of the original study. The tertiary study performed by Kitchenham et al. [19] was a search extension of a previous study [18] (although it also increased the timeframe of the search). Finally, a temporal update and search extension can be combined. We found neither an update nor an extension of a previous SLR among the 120 studies. We believe that producing updates and extensions with proper integration of the findings with the original reviews is an important research activity, and more researchers should engage in such studies. In particular, we believe that external updates or extensions, in the sense of being performed by different researchers than the original authors, is important in detecting possible bias or imprecision in data extraction and analysis that may have been introduced by the original researchers.

As described in Section 6, we faced difficulties during quality assessment and data extraction due to the way various SLRs were reported. There was very little consistency in the way the articles are organised, including the use of section headings to indicate important parts or stages of the reviews. Additionally, many SLRs omitted essential data, including important parts of the review protocol, and often, the same information was presented inconsistently in different parts of the article. We believe that researchers performing and reporting systematic reviews would benefit from reading reviews that are organised to enhance readability and allow for better data extraction, assessment, and comparison among studies. We found a few reviews that provide good examples of organisation and content, including SE01, SE02, SE05, SE35, SE55].

In future work, we will investigate the extent to which the EBSE is being realised regarding the development of all steps defined in Section 1. One research approach will be to conduct a broad field survey with the researchers involved in the 120 SLRs to investigate the origins and motivations of their questions and the application of the results of their SLRs. We also plan to perform continual updates to this tertiary study on at least a yearly basis. Finally, we began to investigate the methods used by the researchers to integrate qualitative data. This is relevant due to the increasing incidence of qualitative research in software engineering. We expect to produce, from the best approaches of qualitative data analysis employed, guidelines for researchers performing SLRs of qualitative studies, as has been done in other fields [26].

Acknowledgements

Fabio Q.B. da Silva holds a research Grant from the Brazilian National Research Council (CNPq) #314523/2009-0. Sérgio Soares is partially supported by CNPq (305085/2010-7) and FACEPE (APQ-0093-1.03/08) André L.M. Santos and Sérgio Soares are partially supported by the National Institute of Science and Technology for Software Engineering (INES), Grants 573964/2008-4 (CNPq) and APQ-1037-1.03/08 (FACEPE). A. César C. França and Cleiton V.F. Monteiro are doctoral students at the Centre for Informatics of the Federal University of Pernambuco where they receive scholarship from CNPq and CAPES respectively. The authors would like to thank the Samsung Institute for Development of Informatics (Samsung/SIDI), for partially supporting the work developed in this research. Finally, the authors would like to thank the reviewers of the manuscript for important critiques and comments that lead to several improvements in the final article.

References

- [1] H. Arksey, L. ÓMalley, Scoping studies: towards a methodological framework, *International Journal of Social Research Methodology* 8 (1) (2005) 19–32.
- [2] A. Abran, J. Moore, P. Bourque, T. Dupuis (Eds.), *Guide to software engineering body of knowledge*, IEEE Computer Society (2004) 11–12.
- [3] R.F. Barcelos, G.H. Travassos, *Evaluation Approaches for Software Architectural Documents: A Systematic Review*, IDEAS, La Plata, Argentina, 2006. p. 14.
- [4] J. Biolchini et al., Systematic review in software engineering, *Technical Report RT-ES 679-05*, PESC, COPPE/UFRJ, 2005.
- [5] Centre for Reviews and Dissemination, What are the Criteria for the Inclusion of Reviews on DARE?, 2007 <<http://www.york.ac.uk/inst/crd/faq4.htm>> (accessed 24.07.07, no longer working, use [6]).
- [6] Centre for Reviews and Dissemination, About DARE, 2010. <<http://www.york.ac.uk/inst/crd/darefaq.htm>> (accessed 20.08.10).
- [7] F.Q.B. Da Silva et al., Critical Appraisal of Systematic Reviews in Software Engineering: a Tertiary Study, ESEM'2010, Bolzano-Bolzen, Italy, 2010.
- [8] T. Dybå et al., Evidence-based software engineering for practitioners, *IEEE Software* 22 (1) (2005) 58–65.
- [9] S.M. Easterbrook et al., Selecting empirical methods for software engineering research, in: F. Shull, J. Singer, D. Sjøberg (Eds.), *Guide to Advanced Empirical Software Engineering*, Springer, 2007.
- [10] G. Glass, Integrating findings: the meta-analysis of research, *Review of Research in Education* 5 (1977) 351–379.
- [11] T. Greenhalgh, *How to Read a Paper: The Basics of Evidence-based Medicine*, Blackwell Publishing, London, 2007.
- [12] R. Khan et al., *Reviews to Support Evidence-based Medicine: How to Review and Apply Findings of Healthcare Research*, Royal Society of Medicine Press Ltd., 2003.
- [13] B. Kitchenham et al., Preliminary guidelines for empirical research in software engineering, *IEEE TSE* 28 (8) (2002) 721–734.
- [14] B. Kitchenham et al., Evidence-based Software engineering, in: 26th International Conference on Software Engineering (ICSE), IEEE, Washington DC, 2004, pp. 273–281.
- [15] B.A. Kitchenham, Procedures for undertaking systematic reviews, Joint Technical Report, Computer Science Department, 2004, Keele University (TR/SE-0401) and National ICT Australia Ltd. (0400011T.1), 2004.
- [16] B. Kitchenham, S. Charters, Guidelines for performing systematic literature reviews in software engineering, Technical Report EBSE-2007-01, School of Computer Science and Mathematics, Keele University, 2007.
- [17] B.A. Kitchenham et al., Protocol for extending a tertiary study of systematic literature reviews in software engineering, EPIC Technical Report, EBSE-2008-006, 2008.
- [18] B. Kitchenham et al., Systematic literature reviews in software engineering – a systematic literature review, *Information and Software Technology* 51 (2009) 7–15.
- [19] B. Kitchenham et al., Literature reviews in software engineering – a tertiary study, *Information and Software Technology* 52 (8) (2010) 792–805.
- [20] M. Jørgensen et al., Teaching evidence-based software engineering to university students, *METRICS'05*, 2005.
- [21] G.W. Noblit, R.D. Hare, *Meta-Ethnography: Synthesizing Qualitative Studies*, Sage Publications, London, 1988.
- [22] H. Petersson et al., Capture–recapture in software inspections after 10 years research – theory, evaluation and application, *JSS* 72 (2) (2004) 249–264.
- [23] M. Petticrew, H. Roberts, *Systematic Reviews in the Social Sciences*, Blackwell Publishing, 2006.
- [24] A. Rainer, S. Beecham, 2008. Supplementary guidelines and assessment scheme for the use of evidence based software engineering, Technical Report CS-TR-469, University of Hertfordshire, Hatfield, UK.

- [25] D.L. Sackett et al., Evidence-based medicine: what it is and what it isn't, *BMJ* 312 (1996) 71–72.
- [26] M. Sandelowski, J. Barroso, *Handbook for Synthesizing Qualitative Research*, Springer Publishing Company, New York, 2007.
- [27] M. Shaw, P. Clements, The golden age of software architecture: a comprehensive survey, *IEEE Software* 23 (2) (2006) 31–39.
- [28] Software Engineering 2004 Curricula Guidelines for Undergraduate Degree Programs in Software Engineering, Joint Task Force on Computing Curricula IEEE Computer Society Association for Computing Machinery, 2004.

Appendix A – The systematic reviews

- [SE01] S. Beecham et al., Motivation in software engineering: a systematic literature review, *IST* 50 (9–10) (2008) 860–878.
- [SE02] F.O. Björnson, T. Dingsøyr, Knowledge management in software engineering: a systematic review of studied concepts, findings and research methods used, *Knowledge Management* 50 (11) (2008) 1055–1068.
- [SE03] K. Cai, D. Card, An analysis of research topics in software engineering – 2006, *JSS* 81 (6) (2008) 1051–1058.
- [SE04] F. Dong et al., Software multi-project resource scheduling, *ICSP* (2008) 63–75.
- [SE05] T. Dybå et al., Empirical studies of agile software development: a systematic review, *IST* 50 (9–10) (2008) 833–859.
- [SE08] B. Haugset, G.K. Hanssen, Automated Acceptance Testing: A Literature Review and an Industrial Case Study, in: *Agile*, IEEE, 2008, pp. 27–38.
- [SE09] A. Herrmann, M. Daneva, Requirements Prioritization Based on Benefit and Cost Prediction: An Agenda for Future Research, in: *16th RE*, IEEE, 2008, pp. 125–134.
- [SE10] E. Infran et al., A systematic review of usability evaluation in web development*, in: *Workshops on Web Information Systems Engineering*, 2008, pp. 81–91.
- [SE11] M. Kalinowski et al., Towards a defect prevention based process improvement approach, in: *34th Euromicro*, IEEE, 2008, pp. 199–206.
- [SE12] R. Pretorius, D. Budgen, A mapping study on empirical evidence related to the models and forms used in the uml, in: *IEEE-ACM ESEM*, IEEE, 2008.
- [SE13] D. Smite et al., Reporting empirical research in global software engineering: a classification scheme, in: *ICGSE*, IEEE, 2008, pp. 173–181.
- [SE14] L. Van Velsen et al., User-centered evaluation of adaptive and adaptable systems: a literature review, *The Knowledge Engineering Review* 23 (03) (2008) 261–281.
- [SE18] W. Afzal et al., A systematic review of search-based testing for non-functional system properties, *IST* 51 (6) (2009) 957–976. Elsevier.
- [SE19] S. Ali et al., A systematic review of the application and empirical investigation of search-based test case generation, *IEEE TSE* (99) (2009) 1–22.
- [SE20] H.C. Benestad, Empirical assessment of cost factors and productivity during software evolution through the analysis of software change effort, *Sciences – New York*, 2009.
- [SE21] D. Blanes et al., 2009. Requirements engineering in the development of multi-agent systems: a systematic review, in: *10th IDEAL*, pp. 510–517.
- [SE22] P. Brereton et al., Pair programming as a teaching tool: a student review of empirical studies, in: *22nd CSE&T*, 2009.
- [SE23] D. Budgen, C. Zhang, Preliminary reporting guidelines for experience papers, in: *EASE*, 2009, pp. 1–10.
- [SE24] R. Burrows et al., Coupling Metrics for Aspect-Oriented Programming: A Systematic Review of Maintainability Studies (Extended Version), *ENASE*, 2009.
- [SE25] J.a. Calvo-ManzanoVillalón et al., State of the art for risk management in software acquisition, *ACM SIGSOFT Software Engineering Notes* 34 (4) (2009) 1.
- [SE26] C. Catal, B. Diri, A systematic review of software fault prediction studies, *Expert Systems with Applications* 36 (4) (2009) 7346–7354.
- [SE27] L. Chen et al., Variability management in software product lines: a systematic review, in: *13th SPLC*, 2009, pp. 81–90.
- [SE28] L. Chen et al., A status report on the evaluation of variability management approaches, in: *13th EASE*, 2009.
- [SE29] N. Condori-Fernandez et al., A systematic mapping study on empirical evaluation of software requirements specifications techniques, in: *ESEM*, IEEE, 2009, pp. 502–505.
- [SE30] B. Cornelissen et al., A systematic survey of program comprehension through dynamic analysis, *IEEE TSE* 35 (5) (2009) 684–702.
- [SE32] P.S. dos Santos et al., Action research use in software engineering: an initial survey, in: *ESEM*, IEEE, 2009, pp. 414–417.
- [SE33] J. Du et al., An analysis for understanding software security requirement methodologies, in: *SSIRI*, IEEE, 2009, pp. 141–149.
- [SE34] H. Edwards et al., The repository grid technique: its place in empirical software engineering research, *IST* 51 (4) (2009) 785–798.
- [SE35] E. Engström et al., A systematic review on regression test selection techniques, *IST* 52 (1) (2010) 14–30.
- [SE36] T. Hall et al., A systematic review of theory use in studies investigating the motivations of software engineers, *TOSEM* 18 (3) (2009) (ACM).
- [SE37] J.E. Hannay et al., The effectiveness of pair programming: a meta-analysis, *IST* 51 (7) (2009) 1110–1122.
- [SE38] J. Hong et al., Context-aware systems: a literature review and classification, *Expert Systems with Applications* 36 (4) (2009) 8509–8522.
- [SE39] W. Hordijk et al., Harmfulness of code duplication – a structured review of the evidence, in: *13th EASE*, 2009.
- [SE40] E. Hossain, M.A. Babar, Using scrum in global software development: a systematic literature review, *ICGSE* (2009).
- [SE42] M. Ivarsson, T. Gorschek, Technology transfer decision support in requirements engineering research: a systematic review of REj, *RE* 2009 (2009) 155–175.
- [SE43] M. Jiménez et al., Challenges and improvements in distributed software development: a systematic review, *Advances in Software Engineering* (2009) 1–15.
- [SE44] S.U. Khan et al., Critical barriers for offshore software development outsourcing vendors: a systematic literature review, in: *16th APSEC*, IEEE, 2009, pp. 79–86.
- [SE45] S.U. Khan et al., Critical success factors for offshore software development outsourcing vendors: a systematic literature review, in: *Fourth ICGSE* (IEEE), 2009, pp. 207–216.
- [SE46] M. Khurum et al., A systematic review of domain analysis solutions for product lines, *JSS* 82 (12) (2009) 1982–2003.
- [SE47] B.P. Lamancha et al., Software product line testing – a systematic review, *ICSOFT* (2009) 23–30.
- [SE48] D. Liu et al., The role of software process simulation modeling in software risk management: a systematic review, in: *Third ESEM* (IEEE), 2009, pp. 302–311.
- [SE49] A. Lopez et al., Risks and safeguards for the requirements engineering process in global software development, in: *Fourth ICGSE* (IEEE), 2009, pp. 394–399.
- [SE50] F.J. Lucas et al., A systematic review of UML model consistency management, *IST* 51 (12) (2009) 1631–1645.
- [SE51] C. Mair, M. Shepperd, A literature review of expert problem solving using analogy, in: *13th EASE*, 2009.
- [SE52] S.M. Mitchell et al., A comparison of software cost, duration, and quality for waterfall vs. iterative and incremental development: A systematic review, *ESEM* (2009).
- [SE53] P. Mohagheghi et al., Definitions and approaches to model quality in model-based software development – a review of literature, *IST* 51(12), 1646–1669, 2009.
- [SE54] S. Montagud et al., Gathering current knowledge about quality evaluation in software product lines, *SPLC* (2009) 91–100.
- [SE55] J. Nicolás et al., On the generation of requirements specifications from software engineering models: a systematic literature review, *IST* 51 (9) (2009) 1291–1307.
- [SE56] J.S. Persson et al., Managing risks in distributed software projects: an integrative framework, *IEEE TOSEM* 56 (3) (2009) 508–532.
- [SE57] Z. Racheva et al., Value creation by agile projects: methodology or mystery?, *LNBP* 32 (4) (2009) 141–155.
- [SE58] A. Rainer et al., An assessment of published evaluations of requirements management tools, in: *13th EASE*, p. 209.
- [SE59] M. Riaz, A systematic review of software maintainability prediction and metrics, in: *3rd ESEM*, IEEE, 2009, pp. 367–377.
- [SE60] J. Savolainen, V. Myllärniemi, Layered architecture revisited—comparison of research and practice, in: *WICSA/ECSCA*, IEEE, 2009, pp. 317–320.
- [SE62] K. Stol et al., The use of empirical methods in Open Source Software research: Facts, trends and future directions, in: *Workshop on Emerging Trends in FLOSS Research and Development*, IEEE, 2009, pp. 19–24.
- [SE63] G.S. Walia, J.C. Carver, A systematic literature review to identify and classify software requirement errors, *IST* 51 (7) (2009) 1087–1109.
- [SE64] Z. Zakaria et al., Unit testing approaches for BPEL: a systematic review, in: *16th APSEC*, IEEE, 2009, pp. 316–322.
- [SE65] E.D. de Souza Filho et al., Evaluating domain design approaches using systematic review, *ECSCA* (2008) 50–65.
- [SE66] A.C. Dias-Neto, G.H. Travassos, Model-based testing approaches selection for software projects, *IST* 51 (11) (2009) 1487–1504.
- [SE67] T. Ebling et al., A systematic literature review of requirements engineering in distributed software development environments, in: *11th ICEIS*, 2009, pp. 363–366.
- [SE68] Q. Gu et al., Exploring service-oriented system engineering challenges: a systematic literature review, *Service Oriented Computing and Applications* 3 (3) (2009) 171–188.
- [SE70] A.S. Jadhav et al., Evaluating and selecting software packages: a review, *IST* 51 (3) (2009) 555–563.
- [SE71] V.B. Kampenes et al., A systematic review of quasi-experiments in software engineering, *IST* 51 (1) (2008) 71–82.
- [SE72] A. Trendowicz, Factors influencing software development productivity – state-of-the-art and industrial experiences, *Advances in Computers* 77 (09) (2009) 185–241.
- [SE74] M.V. Zelkowitz, An update to experimental models for validating computer technology, *JSS* 82 (3) (2009) 373–376.
- [SE75] H. Sharp et al., Models of motivation in software engineering, *IST* 51 (1) (2009) 219–233.
- [SE76] R. Prikladnicki et al., Patterns of distributed software development evolution: a systematic review of the literature, in: *12th EASE*, 2008.
- [SE77] M. Khurum et al., Systematic review of solutions proposed for product line economics, in: *2nd MESPUL*, 2008, pp. 386–393.