JOURNAL OF CRITICAL REVIEWS

ISSN- 2394-5125 VOL 06, ISSUE 07, 2019

Systematic literature reviews in software engineering – A tertiary study

N Mahesh Babu, Md Ayub Khan,P.Sandeep Reddy
Assistant Professor[1,2],Associate Professor[2]
Dept. of CSE,
mail-id:mahesh.nallagatla@gmail.com, ayubkhanmd32@gmail.com, sandeepreddycse@anurag.ac.in
Anurag Engineering College,Anatagiri(V&M),Suryapet(Dt),Telangana-508206

A B S T R A C T

Context: In a previous study, we reported on a systematic literature review (SLR), based on a manual search of 13 journals and conferences undertaken in the period 1st January 2004 to 30th June 2007.
Objective: The aim of this on-going research is to provide an annotated catalogue of SLRs available to soft- ware engineering researchers and practitioners. This study updates our previous study using a broad automated search.
Method: We performed a broad automated search to find SLRs published in the time period 1st January 2004 to 30th June 2008. We contrast the number, quality and source of these SLRs with SLRs found in the original study.
Results: Our broad search found an additional 35 SLRs corresponding to 33 unique studies. Of these papers, 17 appeared relevant to the undergraduate educational curriculum and 12 appeared of possible interest to practitioners. The number of SLRs being published is increasing. The quality of papers in con- ferences and workshops has improved as more researchers use SLR guidelines.
Conclusion: SLRs appear to have gone past the stage of being used solely by innovators but cannot yet be considered a main stream software engineering research methodology. They are addressing a wide range of topics but still have limitations, such as often failing to assess primary study quality.

Keywords: Systematic literature review Mapping study Software engineering Tertiary study

Introduction

In a series of three papers Kitchenham, Dybå and Jørgensen sug- gested that software engineers in general, and empirical software engineering researchers in particular, should adopt evidence-based practice as pioneered in the fields of medicine and sociology [1–3].
They proposed a framework for
ing experiments [9] which led to a series of follow-on SLRs including [10,11]. In addition, some mapping studies are con- cerned about how academics undertake research in software engineering (e.g. [13]) rather than what we know about a spe- cific software engineering topic. The study reported in this paper is a mapping study.

ISSN- 2394-5125 VOL 06, ISSUE 07, 2019

Some mapping studies (like this one) provide a more detailed review of the topics covered in each primary study including issues such as major outcomes and qualityevaluations of primary studies.
We believe secondary studies can play a vital role both in sup- porting further research efforts and also in providing information about the impact of methods and tools to assist software engineer- ing practitioners and managers [2,1]. However, these studies need to be readily available to those who would benefit from them. For example, researchers entering a new field would benefit from map- ping studies in the area, whereas standards writers would benefit
able. SLRs are referred to asanalyse are referred to as types of SLRs:
rom conventional SLRs evaluating the benefits of specific tech- niques.

For this reason we believe it is important to because it was a SLR of secondary studies.The
mates of the difference between methods, *meta-analysis* can be used to undertake a formal statistically-based aggregation. However, we have found that meta-analysis is seldom possi- ble for SLRs in software engineering because there are often insufficient primary studies.

Mapping studies. These studies aim to find and classify the pri-
goal of the study was to identify how many SLRs had been pub- lished, what research topics were being addressed, and the limita- tions of current SLRs. For that study we used a manual search of a targeted set of 13 conferences and journals during the period Jan- uary 1st 2004 to 30th June 2007. The sources were selected be- cause they were known to include empirical studies and
mary studies in
They have coarser-
literature surveys, and had been used as sources for other mapping
grained research questions such as ‘‘What do we know about topic x?” They may be used to identify available literature prior to undertaking conventional SLRs. They use the same methods for searching and data extraction as conventional SLRs but rely more on tabulating the primary studies in spe- cific categories. An example is the study of software engineer-studies (e.g. [9,13]). This search identified 20 SLRs of which eight were mapping studies and one a meta-analysis.

In this paper, we report the results of a broad automated search covering the period 1st January 2004 to 30th June 2008, and con- trast them with our previous results. In effect we compare three sets of SLRs:

- Those reported in the original study, covering the time period January 2004 to June 30th 2007 [12].
- Those found in the time period January 2004 to June 30th 2007 that were found by the broad automated search and were not included in the original study. We discuss the differ- ences between the results of the manual search and the broadautomated search in [14].
- Those found in the time period July 1st 2007 to June 30th 2008.

These are illustrated in Fig. 1. For convenience and to simplify referencing, these sets papers are respectively referred to as T1, T2-1 and T2-2 respectively in the rest of the paper (T for 'tertiary'). The original study [12] is referred to as T1, this one as T2.

Section 2 reports our methodology. Section 3 reports data we extracted from each SLR. Section 4 answers our research questions. We report the limitations of our study in Section 5 and our conclusions in Section 6.

1. Method

We applied the basic SLR method as described by Kitchenham and Charters [8]. The main differences between the methods used in this study compared with the method used for the original studywere that:

- We used a broad automated search rather than a restricted manual search process.
- Three researchers collected quality and classification data. For the papers found in the same time period as the original search, they took the median or mode value (as appropriate) as the consensus value. For the set of papers found after the time period of the original search, they used a ''consensus" and ''minority report" process for data extraction as described in Section 2.3.

The research questions

The first three research questions investigated in this study were equivalent to the research questions used in the original study [12]:

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

RQ2 What research topics are being addressed?

The third question in our original study was ''Who is leading the research effort?". However, since we actually measured activitynot leadership we have revised the research question to be:

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

In our original study, our fourth research question was ''What are the limitations of current research?" The original study identi- fied several problems with existing SLRs:

- A relatively large number of studies were investigating research methods rather than software engineering topics(8 of 20).
- The spread of software engineering topics was limited.
- The number of primary studies was much greater for map- ping studies than for SLRs.

Therefore in this study we have changed the fourth question to be:

RQ4 Are the limitations of SLRs, as observed in the originalstudy, still an issue?

ISSN- 2394-5125 VOL 06, ISSUE 07, 2019

We also address an additional research question: RQ5 Is the quality of SLRs improving?

The search process

Pretorius was responsible for most of the search process. She searched four digital libraries and one broad indexing service: IEEE Computer Society Digital Library; ACM; Citeseer; SpringerLink; Web of Science. In addition, Kitchenham searched the SCOPUS indexing system. All searches were based on title, keywords and abstract. The searches took place between July and August 2008. For all the sources except SCOPUS the researcher used a set of sim- ple search strings and aggregated the outcome from each of the searches for each source:

Since SCOPUS allowed easy construction of complex searches, and reducing the number of searches reduces the problem of inte- grating of search results, the SCOPUS search was based on only twocomplex searches over two time periods:

Search 1, conducted separately for 2004–2007, and then for 2008 only
TITLE-ABS-KEY(‘‘software”) AND TITLE-ABS-KEY(‘‘evidence-based software engineering” OR ‘‘review of studies” OR ‘‘struc- tured review” OR ‘‘systematic review” OR ‘‘literature review” OR ‘‘literature analysis” OR ‘‘in-depth survey” OR ‘‘literature survey” OR ‘‘meta-analysis” OR ‘‘Past studies”) AND SUBJAREA(comp).
Search 2, conducted separately for 2004–2007, and then for 2008 only
TITLE-ABS-KEY(‘‘software engineering research”[1] OR ‘‘subject matter expert” OR roadmap) AND SUBJAREA(comp).

The search process was validated against the papers found by the original study (set T1). The automated search found 15 of the
18 studies found in the original study through manual search(and excluding the two papers that were found by other means).

- Relatively few SLRs assessed the quality of primary studies.
- Relatively few papers provided advice that was oriented to the needs of practitioners.

[1] Note the two researchers used slightly different strings which may not be strictly equivalent.

Two of the missed papers were ‘‘border line” for inclusion (one was a small-scale review that was not the main topic of the paper and the other a computer science rather than software engineering study), and the remaining missed study used the term ‘‘review” but not ‘‘literature review”. Therefore we concluded that the auto- mated search was almost as good as the manual search for the most important software engineering sources.

However, the search missed three relevant papers of which we were aware:

- Jefferies et al. [15].
- Bailey et al. [16]
- MacDonell and Shepperd [17].

We re-checked the output of the SCOPUS search performed in July 2008 and confirmed that the papers had not been detected by the search (i.e. we had not missed the papers when looking at the search results). Using the same search string on 6/07/2009 for the time period 2007–2008 (we refer to this as the ‘‘2009 search”), we found all three papers, thus it seems that the three pa- pers were not in the indexing system when the original search was performed. We also checked all the papers found by the search andno other relevant papers were identified.

We reviewed the results from the 2009 search in more detail to assess whether it was likely that our initial search (conducted dur- ing July–August 2008) had missed any other relevant papers. The 2009 search found 9 of the other 16 SLRs published in the time per- iod July 1st 2007 to June 30th 2008. Seven of the SLRs missed by the 2009 search used non-standard terminology:

- Did not use the term ‘‘literature review” (e.g. used terms such as ‘‘literature survey” [18] or ‘‘assembly of studies” [19], or just the term ‘‘review” without any qualification [20,21]).
- Did not use any terms related to review (e.g. explained that they ‘‘searched publication channels” [22] or ‘‘analyzed soft- ware engineering experiments” [10]).
- Did not appear to be indexed by the SCOPUS system [23].

The remaining two papers missed by the 2009 search used the term ‘‘literature review” and were also missed by the 2008 search [24,25]. However, when the term ‘‘AND TITLE-ABS-KEY(‘‘soft- ware”)” was removed from the July 2009 search, the two papers were included in the search output (although the number of papers returned increased from 134 to 578). Thus, the search did not iden- tify the papers as mainstream software engineering literature re- views which appears reasonable since one paper was about web design for accessibility [24] and the other was about collaborative conceptual modeling [25].

Thus, the July 2009 SCOPUS-based search found all the papers that used standard terminology and were mainstream software engineering studies, found the three relevant papers missed in the original search, and did not find any other relevant studies. Therefore, we concluded that we had probably not missed any other relevant mainstream papers, and did not need to undertake any more detailed searches for missing studies.

Study selection

Pretorius integrated the results for the different searches and undertook an initial screening of

the 1757 papers found, based on title, abstract and keywords. This screening was based on excluding studies that were obviously irrelevant, or duplicates, or SLRs that we had already found [12].

he remaining 161 papers were then subject to a more detailed assessment:

Step 1: Three researchers screened each paper for inclusion independently. Two researchers from a pool of five researchers, excluding Kitchenham, were assigned at random to each paper. Kitchenham reviewed every paper. Each paper was screened to identify papers that could be *rejected* based on abstract and title on the basis that they *did not include literature reviews* or *were not software engineering topics*. Any disagreements were discussed but the emphasis was on not rejecting any disputed papers. This led to the exclusion of 42 papers.

Step 2: We obtained full copies of the remaining 119 papers and undertook a more detailed second screening using the following inclusion and exclusion criteria:

That there was a full paper (not a PowerPoint presen- tation or extended abstract)

That the paper *included a literature review* where papers were included *based on a defined search process*. The paper should be related to software engineering rather than IS or computer science.

As previously the process was:

- To assign two of five researchers at random to review each paper and for Kitchenham to review each paper.
- Disagreements were discussed and resolved.
- The emphasis was on not rejecting any possibly relevant papers.

This selection process rejected 54 papers that performed a liter- ature survey but did not have any defined search process. Another 25 papers were rejected because either they included only a re- lated research section, or were duplicate papers, or were not soft- ware engineering papers. The remaining 40 papers were split into two sets. The first set of 14 papers (T2-1) comprised papers pub- lished in the time period 1st January 2004 to 30th June 2007; the second set (T2-2) comprised 26 papers published after 1st July 2007. We completed the quality assessment and data extraction for the 14 papers in T2-1 before we began work on the quality assessment and data extraction for the 26 papers in T2-2.

Of the 26 papers in T2-2, four papers were excluded since they were published after 20th June 2008, leaving 22 papers, of which three referred to the same SLR, leaving 20 individual SLRs. Including the three studies missed by the search process, we found a total of 23 SLRs that were included in the data extraction process (see Fig. 2).

Quality assessment

Each SLR was evaluated using the York University, Centre for Reviews and Dissemination (CDR) Database of Abstracts of Re- views of Effects (DARE) criteria [26]. The criteria are based on fourquestions:

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

The questions were scored as follows:

- Question 1: Y (yes), the inclusion criteria are explicitly defined in the paper, P (Partly), the inclusion criteria are implicit; N (no), the inclusion criteria are not defined and can- not be readily inferred.
- Question 2: Y, 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, the authors have searched 3 or 4 digital libraries with no extra search strategies, or searched a defined but restricted set of journals and conference proceedings; N, the authors have searched up to 2 digital libraries or an extremelyrestricted set of journals. Note that scoring question 2 also requires the evaluator to consider whether the digital libraries were appropriate for the specific SLR.
- Question 3: Y, the authors have explicitly defined quality cri- teria and extracted them from each primary study; P, the research question involves quality issues that are addressed by the study; N no explicit quality assessment of individual papers has been attempted or quality data has been extracted but not used. (Note the decision to penalize papers that col- lected quality data but did not use it was applied only to papers published after 30th June 2007.)
- Question 4: Y, Information is presented about each paper so that the data summaries can clearly be traced to relevant papers; P, only summary information is presented about indi- vidual papers e.g. papers are grouped into categories but it is not possible to link individual studies to each category; N, the results of the individual studies are not specified i.e. the indi- vidual primary studies are not cited.

The scoring procedure was Y = 1, P = 0.5, N = 0.

These are the same criteria that were used to evaluate quality in the original tertiary study 1 [12], except for scoring N for Q3 if pa- pers collected quality data but did not use it. It should be noted that the information provided to help determine the answer for each question is intended to provide support for the assessment; it is not a strict mutually exclusive classification process.

We used two different methods for quality data extraction. For the additional papers in set T2-1, three researchers extracted infor- mation from each paper. Two researchers were assigned at random from the pool of four researchers while Kitchenham reviewed allthe papers. The median value was taken to represent the consensusview.

We used a more rigorous process to answer the quality ques- tions for papers published in the time period 1st July 2007 to 30th June 2008 (set T2-2). We refer to the process as a ''consensus

and minority report" process whereby:

1. Two from a pool of five researchers, excluding Pretorius and Kitchenham, were randomly allocated to each study.
2. The two researchers independently answered the quality questions, and provided a justification for each answer.
3. The two researchers compared their results and came to a consensus.
4. Kitchenham answered the quality questions for all SLRs pro- viding a justification for each answer.
5. The consensus result was then compared with a third inde- pendent extraction (performed by Kitchenham) and the two original data extractors discussed any disagreements until they reached final consensus. Note Kitchenham did not take part in the final discussion in order for one person not to havetoo much influence on the results.

This process is illustrated in Fig. 3.

Data extraction process

The data extraction for quality data and classification data was undertaken at the same time using the procedures described above. In addition to the quality assessment the following data, which were collected in the original study, were also extracted:

The type of study (SLR or mapping study).
The review focus i.e. whether the SLR was software engineer- ing oriented or research methods oriented.
The number of primary studies included in the SLR (to address the issue of whether there are sufficient primary studies in software engineering for SLRs to be useful).

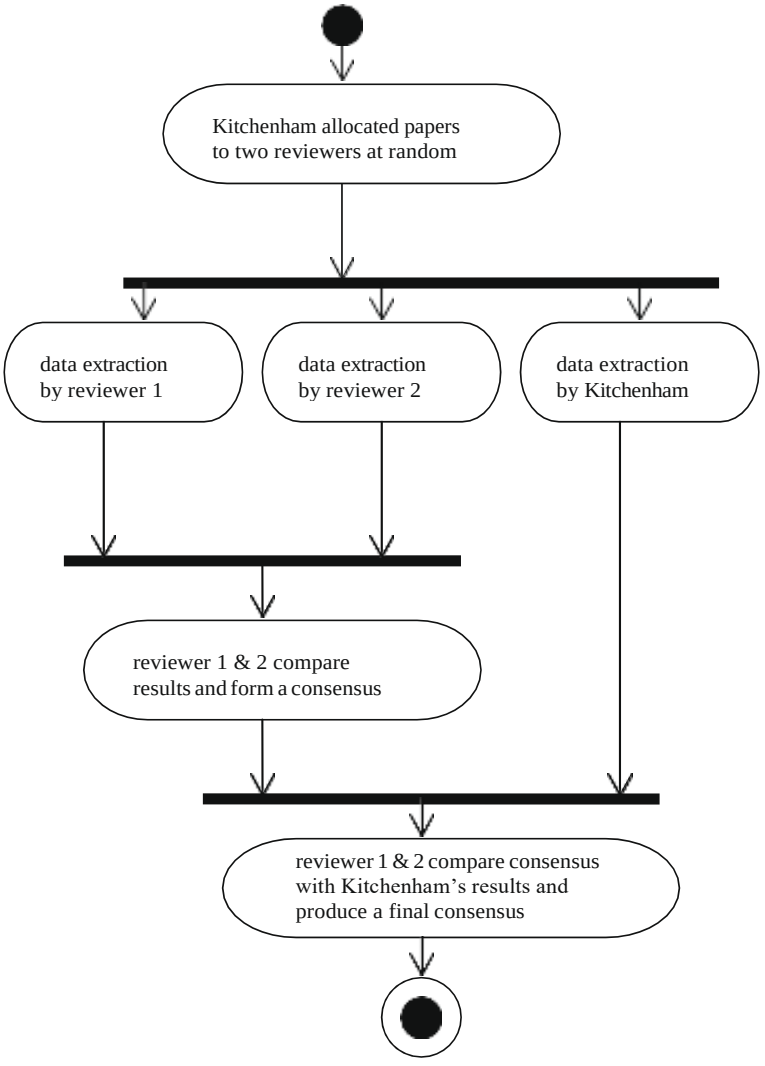

Fig. 3. ‘‘Consensus and minority report” data extraction process.

For papers published before 1st July 2007 (set T2-1), the median of nominal scale data items was used, and any disagreement about the number of primary studies was discussed and a final value agreed. Data for all other aspects was extracted by all three researchers, but these were non-subjective factors i.e. publication sources, publication type, authors, and author's affiliations and there was no disagreement. For papers published after 30th June 2007 (set T2-2), data extraction for subjective values was done by three researchers using the ‘‘consensus and minority report process”; data for non-subjective factors were extracted by one per- son, i.e. Kitchenham.

During the data extraction for papers found after June 30th 2007, four further papers were excluded from the set of relevant studies after agreement among three reviewers for the following reasons:

- Two papers were excluded because apart from the search there was nothing systematic about their literature review [27,28].

ISSN- 2394-5125 VOL 06, ISSUE 07, 2019

- Two papers were excluded because of incompleteness. One specified that it was presenting preliminary results based on the papers found to date [29]. The other was a short paper which did not report any aggregation of the identified papers [30].

Thus, we found 19 relevant studies in the time period 1st July 2007 to 30th June 2008, so overall, the broad search covering January 1st 2004 to June 30th 2008 (set T2) found 33 additional studies which reported an SLR (even if the authors themselves did not use the term ‘‘systematic” to describe their review). The process by which we arrived at 19 relevant SLRs is shown in Fig. 2. Data obtainedfrom the SLRs are reported in detail in Sections 3 and 4, and com- pared with the 20 SLRs discussed in the original study (set T1) [12].

3. Data extraction results

The 33 SLRs that were published in the time period 1st January 2004 to 30th June 2008 (excluding those reported in the original tertiary study) are shown in Table 1. For each review we identify:

Whether it posed detailed technical questions (RQ) or was interested primarily in trends in a particular software engi- neering topic area (SERT) or the way in which software engi- neers undertake research (RT).
The quality score assigned to the study. The year of publication.
Whether the study positioned itself explicitly as an EBSE study by citing any of the EBSE papers [2,1] or the SLR guide- lines [7,8].

Table 1
Additional software engineering SLRs published from 1st January 2004 to 30th June 2008 (studies above the double lines were published before July 1st 2007, studies below the double lines were published after June 30th 2007).

Study	Review focus	Quality total score	Year	Cited EBSE paper	Cited guidelines	Paper type	Number primary studies	Practitioner guidelines	Review topic
[32]	RQ	2.5	2005	No	Yes	Conference	8	N	Cost estimation – impact of clients on estimate accuracy
[33]	RQ	2	2005	No	No	Journal	70	Y	Cost estimation – Guidelines for estimating uncertainty
[36]	RT	2.5	2005	No	Yes[a]	Workshop	50	N	Cost estimation – data sets used to evaluate models
[39]	RT	1.5	2005	No	No	Workshop	119	N	Evidence produced by empirical software engineers
[40]	RT	2	2005	No	Yes	Conference	13	N	Classifying context in SE experiments
[41]	SERT	2.5	2005	No	No	Conference	105	N	Mobile systems development
[34]	RQ	3.5	2006	Yes	Yes	Conference	26	Y	Requirements elicitation techniques
[37]	SERT	1.5	2006	No	No	Workshop	57	N	Conceptual model of outsourcing
[42]	SERT	1	2006	No	No	Technical report	750	N	Software architecture
[35]	SERT	1.5	2007	Yes	No	Workshop	653	N	Cost estimation challenges
[38]		1.5	20	No	No	Journal	80	N	Approaches for

	SERT		07						mining software repositories in the context of evolution
[43]	SERT	1.5	2007	No	No	Book chapter (working conference)	4089	N	Requirements Engineering publications
[44]	RT	1.5	2007	No	No	Book chapter (workshop)	133	N	Evidence produced by empirical software engineers
[45]	SERT	2	2007	No	No	Book chapter (working conference)	155	N	Developing open source software – analysis of research
[11]	RT	2.5	2007	No	Yes	Journal	103	N	Empirical software engineering – effect size
[16]	SERT	2.5	2007	No	Yes	Conference	138	No	Software design – Object-oriented
[17]	RQ	4	2007	No	Yes	Conference	10	No	Cost estimation-local vs. global estimation models
[19]	RQ	3.5	2007	No	Yes	Book chapter (conference)	5	N	Software development process – tailoring and introduction of rational unified process
[20]	RQ	2.5	2007	No	Yes	Journal	11	N	Reuse – economic benefits
[21]	SERT	1	2007	No	No	Journal	137	N	Tool integration – a research agenda
[24]	SERT	3.5	2007	No	Yes	Conference	53	N	Web application development – design for accessibility

[10]	RQ	2.5	2008	No	Yes	Journal	103	N	Empirical software engineering – the value of laboratory experiments
[15]	SERT	2.5	2008	No	Yes	Conference	28	No	Re-engineering – multi-channel access
[18]	RQ	1.5	2008	No	No	Book chapter (conference)	21	N	Metrics – measurement programme evolution
[22]	RQ	2.5	2008	Yes	No	Book chapter (conference)	25	N	Model-driven engineering
[23]	RQ	3	2008	No	Yes	Workshop	14	N	Architecture – definition of architectural knowledge
[25]	SERT	1.5	2008	No	No	Book chapter (workshop)	46	Y	Collaborative conceptual modeling
[31]	SERT	3.5	2008	No	Yes	Journal	85	N	Model based testing
[46]	RT	3	2008	No	Yes	Workshop	23	N	Empirical software engineering – data set quality
[47]	RQ	3	2008	No	Yes	Journal	45	Y	Software process improvement – in SMEs
[48]	SERT	2	2008	No	Yes	Journal	100[b]	N	Metrics – overview
[49]	RQ	3.5	2008	No	Yes	Journal	43	N	Software process improvement – motivations for adoption
[50]	RQ	3	2008	Yes	Yes	Book chapter (workshop)	96	N	Software process simulation modeling

[a] This paper did not reference the guidelines in the context of using the method, so we do not count this paper as EBSE-related.

[b] Not explicitly reported – estimated from number of references.

ISSN- 2394-5125 VOL 06, ISSUE 07, 2019

- The source in which the SLR was reported (i.e. journal, work- shop, conference, book chapter).
- The number of primary studies used as stated by the authorexplicitly or in tabulations.
- Whether the study included guidelines for practitioners.
- The topic of the SLR.

With respect to the nature of references to the EBSE papers and SLR Guidelines:

All the papers that cited one of the EBSE papers used the EBSE concept as a justification for their study.

- All papers that quoted one of the Guidelines papers, except [36], did so in the context of the methodology they selected. In addition one paper, quoted the Guidelines as an input to the Biolochini et al. template [51] which they adopted for their review [23]. Another paper referenced both the guide- lines and the Biolchini et al. paper to describe their review method [47]. Several other papers said they were similar to,

or inspired by, or informed by the Guidelines or used criteria for inclusion exclusion as suggested by the Guidelines [10,19,20,11].

Thus we consider all papers that referenced either the Guide- lines or the EBSE papers, to be EBSE-positioned papers, except [36]. With respect to the number of primary studies, some values seem unusually high, but were based on the reports of the authors:

- Davis et al. [43] based their study on a database of 4089 requirements engineering papers collected over a period of 18 years and describe how the database was obtained.
- Shaw and Clements [42] searched CiteSeer using the term ‘‘software architecture”. They consolidated variant citations and ignored self-citations which they say yielded ‘‘a sample of about 5500 citations to about 750 papers”. They compared the top 24 papers in that group with a similar set of refer- ences found in 2001 giving a set of 34 papers. The paper itself references a total of 97 papers including the 34 papers.
- Shepperd states that he searched ISI Scientific Citation Index Web of Knowledge with terms related to software cost esti- mation and found 320 journal papers and 333 conference papers [35].

Table 2 shows that we categorized all three of these studies as mapping studies and that all three scored less that 2 on the quality scale. Thus, a large number of papers can be obtained but the resulting studies may lack quality, in particular traceability from primary studies to conclusions (Q4) and repeatability (Q1) are likely to be compromised and individual papers will probably not be assessed for quality (Q3).

Table 2
Scores for each quality question (studies above the double lines were published before July 1st 2007, studies below the double lines were published after June 30th 2007).

The 14 additional papers found in the time period 1st January 2004 to 30th June 2007 (set T2-1) were discussed previously in the context of the implications of broad and restricted search strat- egies [14,52]. Table 2 shows the quality assessments and type for each paper. The impact of the ‘‘consensus and minority report pro- cess” for assessing quality is discussed in [52].

4. Discussion of research questions

This section addresses our specific research questions and iden- tifies any changes between SLRs discussed in our original study and SLRs found in this study.

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

In the year from July 1st 2007 to June 30th 2008 (set T2-2), we found 19 SLRs of which eight were mapping studies. All but three were EBSE-positioned SLRs (i.e. referenced either one of the EBSE papers or the SLR guidelines). This compares with 34 studies of which 18 were mapping studies in the three and one half year per- iod 1 January 2004 to 30th June 2007 (T1 + T2-1). The number of studies per year is shown in Table 3. Table 3 suggests an increase in the number of SLRs with the number of studies per year between 2004 and 2006 being comparable with the number of studies per half year since 2007. Furthermore between July 2007 and June 2008, there has been an increase in the proportion of reviews posi- tioning themselves as EBSE SLRs.

What research topics are being addressed?

Table 1 suggests that many different topics are being addressed. In order to have some baseline to evaluate the extent to which soft- ware engineering topics are being addressed, we considered how well the SLRs of relevance both to education and to practice related to the Software Engineering 2004 Curriculum Guidelines for Undergraduate Degree Program [53] and the Software Engineers' Book of Knowledge (SWEBOK) [54] (see Table 4). We believe that SLRs could be used by academics to help prepare course material and text books, so we were interested in assessing coverage with respect to the undergraduate curriculum. In addition, we used the SWEBOK because from an EBSE viewpoint SLRs are supposed to be of relevance to practitioners, and the SWEBOK is intendedto identify knowledge needed by practitioners with up to 5 years experience.

Table 5 shows the distribution of SLR topics against the curric-
ulum and confirms that coverage of core SE topics is extremely sparse. (A similar result is found if papers are mapped to the SWEBOK [55].)

Restricting SLRs to those deemed to be good quality SLRs (i.e. those that scored 2 or more on the quality scale) would remove four of the SLRs relevant to practitioners and educators. However, even if an SLR is of relatively low quality, it might still provide a

Table 4
Relationship between SLRs and the undergraduate curriculum and the SWEBOK (good quality SLRs in bold).

Study ref.	Review type	Quality total score	Review topic	Useful for education	Useful for practitioners	Why?	Mapping to topic area in the SE curriculum (using the curriculum section codes)	Mapping to SWEBOK (using the SWEBOK section references)
[32]	SLR	2.5	Cost estimation – impact of clients on estimate accuracy	Yes	Yes	Overview of existing research plus another large survey	MGT.pp.4 software management; project planning, Section 4	Effort, schedule, and cost estimation. chapter 8, Section 2.3
[33]	SLR	2	Cost estimation – guidelines for estimating uncertainty	Yes	Yes	Practical evidence-based guidelines for managers	MGT.pp4 software management; project planning, Section 4	Effort, schedule, and cost estimation. chapter 8, Section 2.3
[36]	SLR	2.5	Cost estimation – data sets used to evaluate models	No	No	More appropriate for researchers.		
[39]	MS	1.5	Evidence produced by empirical software	No	No	Aimed at researchers		
	SL	2		No	No	More appropriate for		

[40]	R		engineers Classifying context in SE experiments			researchers		
[41]	MS	2.5	Mobile systems development	Yes	Yes	Good overview of topic area	SAS.mob system and application specialties; systems for small and mobile platforms	N/A
[34]	SLR	3.5	Requirements elicitation techniques	Yes	Yes	Important results confirming the importance of using structured interviews	MAA.er.2 software modeling and analysis; eliciting requirements, Section 2	Requirements elicitation techniques chapter 2, Section 3.2
[37]	MS	1.5	Conceptual model of outsourcing	Yes	Yes	An example of a process model in an important topic area	PRO.imp.2 software process; implementation. Section 2	Maintenance issues outsourcing chapter 6, Section 2.2.5
[42]	MS	1	Software architecture	Yes	Possibly	High level overview of the topic with a list of popular references	DES.ar software design; architectural design	Software design software architecture chapter 3, Section 3
[35]	MS	1.5	Cost estimation challenges	No	No	More appropriate for researchers		
[38]	MS	1.5	Approaches for mining software repositories in the context of evolution	No	No	More appropriate for researchers		
[43]	MS	1.5	Requirements engineering	No	No	High level statistical analysis of numbers of		

			publications			papers in different categories, so more		
[44]	MS	1.5	Evidence produced by empirical software engineers	No	No	appropriate for researchers. Aimed at researchers		
[45]	MS	2	Developing open source software	No	No	High level statistical analysis of numbers of papers in different categories, so more		
[11]	SLR	2.5	Empirical software engineering – effect size	No	No	appropriate for researchers Aimed at researchers		
[16]	MS	2.5	Software design – object-oriented	No	No	Statistical analysis of papers in various categories, so more appropriate for researchers		
[17]	SLR	4	Cost estimation-local vs. global estimation models	Possibly	Possibly	Rather specialized topic area (also an intentional replication of a previous SLR)	MGT.pp4 Software Management; Project planning, Section 4	Software project planning. effort, schedule, and cost estimation. chapter 8, Section 2.3
[19]	SLR	3.5	Software	Possibly	Yes	Rather specialized for undergraduates but	PRO.imp.6 software process; implementation,	chapter 9, Section

	development			2.4
	process – tailoring and introduction of rational unified process	illustrates the practical problems of using even well-defined processes	Section 6	

[21]	MS	1	Tool integration –	No	No	Aimed at researchers		
[20]	SLR	2.5	a research agenda Reuse – economic benefits	Possibly	Yes	Good overview of arguments concerning the value of reuse	FND.ec.3 mathematical and engineering fundamentals; engineering economics for	Software construction. practical considerations chapter 4, Section 3.5
[24]	MS	3.5	Web application developme nt – design	Possibly	Yes	Has some general discussion of issues but doesn't specify all the papers used in the review	software, Section 3 Des.con.6 software design; design concepts, Section 6	Software design chapter 3
[10]	MS	2.5	for accessibilit y	No	No	Aimed at researchers		
[15]	MS	2.5	Empirical software engineering – the value of laboratory experiments Re- engineering – multi- channel access	Possibly	Yes	This is a specialized topic but relevant to mobile computing	SAS.mob system and application specialties; systems for small and mobile platforms	Maintenance techniques. re- engineering chapter 6, Section 4.2
[18]	SLR	1.5	Metrics – measurement	Possibly	Yes	Rather specialized but highlights important	MGT.ctl.3 software management; project control,	Software engineering

			programme evolution			practical considerations about management process evolution	Section 3	measurement. Establish and sustain measurement commitment, chapter 8, Section 6.1
[22]	SLR	2.5	Model-driven engineering	Yes	Yes	An important topic highlighting its current limitations	MAA.md software modeling and analysis; modeling foundations	Software design strategies. Other methods chapter 3, Section 6.6
[23]	SLR	3	Architecture – definition of architectural knowledge	Possibly	Yes	Highlights the difficulty of defining architectural knowledge	Des.ar software design; architectural design	Software design software architecture chapter 3, Section 3
[25]	MS	1.5	Collaborative conceptual modeling	Possibly	Yes	Rather specialized but highlights important practical considerations	MAA.er.2 software modeling and analysis; eliciting requirements, Section 2	Requirements analysis. conceptual modeling chapter 2, Section 4.2
[31]	MS	3.5	Model based testing	Possibly	Yes	The information is quite high-level but the topic is of importance and the additional on-line material provides	VAV.tst Software verification and validation; testing	Testing from formal specifications chapter 5, Section 3.2.5
[46]	SLR	3	Empirical software	No	No	information about all relevant studies More appropriate for researchers		

			engineering – data set					
[47]	SLR	3	quality	No	Yes	Statistical analysis of papers but with some		Software engineering process,
			Software process improvement – in SMEs			guidelines for practitioners		process assessment methods
[48]	MS	2	Metrics – overview	Possibly	Possibly	Narrative discussion giving an overview of the metrics area.	FND.ef.3 Mathematical and engineering fundamentals; engineering economics for software, Section 3	chapter 9, Section 3.2 Software engineering measurement chapter 8, Section 6. Process and product measurement chapter 9, Section 4
[49]	SLR	3.5	Software process	No	Yes	Aimed at practitioners rather than undergraduates.	MGT.ctl.3 Software Management; Project control, Section 3	Software engineering process. Process assessment.
			improvement – motivations					process assessment
[50]	SLR	3	for adoption	No	No	Aimed at researchers not practitioners.		models chapter 9, Section 3.1
			Software process simulation modeling					

Table 5
Distribution of SLRs over the SE sections of the University curriculum guidelines.

Section of Total	Number of sub-sections	January 1st 2004 to June 30th additional sub-topics		January 1st 2004 to June 2007 manual search [12]		July 1st 2007 to June 30th 2007 papers (broad search)		Total Number sub-topics	
		SLRs	Num subs addressed	SLRs	Num subs addressed	SLRs	Num subs addressed		
Software modeling and analysis	7	41			[34]	1	[22,25]	2	3	3
Software design	7	37	[56]	0	[42]	0	[23,24]	0	4	0[a]
Software validation and verification	40	5	[61,65,66][b],[c]	5			[31]	0	5	5
Software evolution	2	13								
Software process	2	14			[37]	1	[19]	1	2	2
Software quality	5	28	[57]	1				–	1	1
Software management	5	32	[58–60,62–64]	1	[32,33]	1	[17,18,48]	2	11	2
Computing essentials	4	41	[67]	1					1	1
Mathematical and engineering fundamentals	3	22					[22,48]	2	2	2
System and application specialties	42				[41]	1	[15]	1	2	1
Total		233	12	8	6	4	11[d]	8	29	17

[a] The papers addressed a general topic, not a specific technique. [b] This paper addressed four sub-topics of the subsection testing. [c] This paper addressed testing methods and inspection methods. [d] Paper [48] addressed two topics.

useful starting point for academics planning course material or writing text books – as long as all the relevant primary studiesare fully cited.

Excluding general overview papers, we found four studies from those identified in set T2-1 in the time period 1st January 2004 to 30th June 2007 (additional to those in set T1 discussed in [12]), and eight studies published between July 1st 2007 and June 30th 2008 (i.e. set T2-2), that are of possible interest to practitioners (omit- ting the replication SLR [17]). These 12 papers are:

- Jørgensen's paper [33] which presents seven evidence-based guidelines to help estimators in industry assess the uncer- tainty in software cost estimates.
- Davis et al.'s paper [34] on requirements elicitation methods which unexpectedly found that structured interviews appeared to be the most effective elicitation method, out-per- forming other more sophisticated methods.
- Mohagheghi and Conradi paper [20] which reported evidence of the benefits of reuse found in industry studies.
- Four papers related to model-driven engineering [22,31,25,19]. All except [31] based their reviews on indus- try-based studies.
- Five other papers including two papers related to process improvement [47,49], one paper considering outsourcing [37], one paper considering the impact of client ion estima- tion accuracy [32] and a paper considering the evolution of metrics programmes [18].

A problem common to these 12 SLRs was a lack of reliable quan- titative results concerning the benefits of the various techniques which makes it difficult to offer clear-cut recommendations to practitioners. However, it is encouraging to see that seven of the SLRs concentrated on industrial case studies or industrial surveys that might be of more relevance to practitioners, rather than on small-scale studies and laboratory experiments.

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

In the period January 2004 to June 2007, the studies were dom- inated by a specific researcher, Magne Jørgensen, who co-authored eight studies. Martin Shepperd contributed to four studies and three other researchers contributed to three studies (Sjöberg,

Moløkken-Østvold, and Juristo). In contrast between July 2007 and June 2008, 51 researchers in total co-authored studies and no researcher contributed to more than two studies. Six research- ers co-authored more than one SLR (Brereton, Kitchenham, Han- nay, Mohagheghi, Shepperd, and Turner).

In terms of affiliations, between January 2004 and 30th June 2007, Simula Laboratory researchers contributed to 11 studies; Brunel University and Universidade Politécnica de Madrid researchers contributed to three studies. In the following year, SIN- TEF employees co-authored four different reviews, Simula Labora- tory and Keele University researchers co-authored three different reviews.

In terms of adopter types [68], these results suggest that groups and individuals undertaking systematic literature reviews are no longer just the *innovators*, but can increasingly be regarded as *earlyadopters*.

With respect to nationalities, as we observed previously, few re- views are co-authored by researchers living in the USA. Only 7 of the 34 studies published before 30th June 2007, and only one of the 19 studies published after 30th June 2007 were co-authored by researchers with US affiliations. Most of the studies were co- authored by Europeans (42 of the 53).

RQ4: Are the limitations of SLRs, as observed in the original study, still an issue?

Review topics and extent of evidence

Compared with our previous study [12], the 33 reviews dis- cussed in this paper addressed a broader range of software engi- neering topics. There is no longer a preponderance of cost estimation studies and more general software engineering topics have been addressed. In the previous study, 8 of the 20 SLRs were concerned with research methods. In the 33 studies reported in this paper, 15 were primarily aimed at researchers; of these, six were concerned with research methods, while the other nine pa- pers were mapping studies related to a specific software engineer- ing topic. Thus, the proportion of papers directed at research methods has reduced from 40% to 18%.

As we found previously, mapping studies analyze more primary studies than conventional SLRs (see Table 6). However, there ap- pears to be sufficient primary studies to undertake SLRs in some topic areas.

Table 6

Median number of primary studies including in mapping studies and SLRs.

Statistic	Time period 1st January 2004 to 30th June 2007 targeted search [12] T1	Time period 1st January 2004 to 30th June 2007 broad search (extra papers) [14] T2-2	Time Period 1st July to 30th June 2007 broad search T2-2
Median number of primary studies in SLRs	20	26	23
Number of SLRs (Including meta-analyses)	11	5	11
Median number of primary studies in mapping studies	103	133	92.5
Number of mapping studies	9	9	8

Practitioner orientation

Twelve of the reviews (addressing 10 different topic areas) seemed targeted at issues of interest

to practitioners with seven of the reviews explicitly concentrating on industrial studies. However, only four papers explicitly provided practitioner- oriented advice.

Evaluating primary study quality

The number of SLRs that undertake quality evaluations of the primary studies is still very low. Only six SLRs (including one map- ping study) performed a full quality evaluation and two more per- formed a partial quality evaluation.

RQ5: Is the quality of SLRs improving?

Table 7 compares the mean quality score for SLRs in set T1. It illustrates the difference in quality scores for papers that cited the SLR guidelines and those that did not. Table 8 shows the aver- age quality for studies published in different sources. It appears that the lack of quality in ‘‘grey literature” studies observed in
[14] is not discernable in the most recent time period.

We collected information about several factors that might potentially impact quality: publication year, review type (i.e. whether it was mapping study or not), whether it referenced the guidelines (where paper [36] was excluded from the count of pa- pers that cited the guidelines), whether it referenced the EBSE pa- pers, and publication type (journal, conference or workshop). We undertook a regression analysis using all these factors with the to- tal quality score as the dependent variable and found only two fac- tors that were statistically significant:

1. Guidelines with a parameter estimate = 0.55 and 95% confi- dence interval (0.257 to 1.123).
2. Mapping Study with a parameter estimate = —0.48 and 95%confidence interval (—0.876 to —0.090).

The results of the regression analysis changes slightly if studies published in Springer book chapters were treated as a separate publication category. In that case the impact of guidelines is no longer significant at the 0.05 level (with $p = 0.06$), but the differ- ence between mapping studies remains significant ($p < 0.01$).

A threat to the validity of these results is that our assessment of SLR quality might have been influenced by knowledge that an SLR did or did not reference the guidelines. For the 14 SLRs published between 1st January 2004 and 30th June 2007, this threat was re- duced by organizing the data collection form such that quality data was extracted before other data. For the 19 SLRs published after 30th June 2007, the citation information was collected by Kitchen- ham after she extracted the quality data, and other researchers were not asked to collect citation information. However, there was no attempt to formally blind the reviewers to the SLR refer- ences during the quality extraction process.

Overall the quality of studies appears to have improved, per- haps as a result of more studies utilizing the SLR guidelines. In par- ticular, the quality scores of book chapters and workshop papers are higher in the period July 1st 2007 to June 30th 2008 than in the earlier time period.

However, mapping studies, on average, have a lower quality score than conventional SLRs. This is because mapping studies seldom assess the quality of primary studies and often do not have clear traceability between individual studies and their individual characteristics.

ISSN- 2394-5125 VOL 06, ISSUE 07, 2019

Table 7
SLR quality.

Cited guidelines	Statistic	Time period 1st January 2004 to 30th June 2007 targeted search [12] T1	Time period 1st January 2004 to 30th June 2007 broad search (extra papers) [14] T2-1	Time period 1st July 2007 to 30th June 2008 broad search T2-2
No	Number of SLRs	12	12	5
	Mean	2.42	1.75	2.0
Yes	Number of SLRs	8	2	14
	Mean	2.70	3.00	2.93

Table 8
SLR quality for different sources.

Source	Time period 1st January 2004 to 30th June 2007 T1 and T2-1		Time period 1st July 2007 to 30th June 2008 T2-2 quality	
	Number of SLRs	Average quality	Number of SLRs	Average
Journal	17	2.42	8	2.56
Conference	8	2.69	4	3.12
Workshop	5	1.9	2	3
Book chapter	3	1.7	5	2.4
Conference	0	n/a	3	2.5
Working conference	2	1.75	0	n/a

Workshop	1	1.5	2	2.25
Technical report	1	1	0	n/a

5. Study limitations

One of the major problems with SLRs is finding all the relevant studies. In this case, we used an automated search of six sources which found most of the papers we found in a previous manual search. However, the search missed three papers that should have been found, since it appears that they were not indexed when the original searches took place. The additional search performed in July 2009 found all papers that used conventional terminology and were mainstream software engineering papers. However, there is a probability that we have missed some studies that are on the borderline between software engineering, information tech- nology and computer science. In addition, our search strings were designed to find the maximum number of known SLRs, so it is possible that they missed some studies that used different terminol- ogy to describe their literature review (e.g. ‘‘study aggregation” or ‘‘study synthesis”) without using terms such as ‘‘literature review” or ‘‘literature survey”.

We have also omitted a search for technical reports or graduate theses. We make the assumption that good quality grey literature studies will appear as journal or conference papers – particularlynow that interest in systematic reviews is increasing. Furthermore, the main reason for grey literature not being formally published is publication bias, which occurs when negative results are not published. However, this does not appear to be a problem for system- atic reviews in software engineering. For example, two recent meta-analyses reported fairly negative results but were still pub- lished [69,70].

Another limitation is that the quality assessment in the study was performed in two different ways: by using a median in studies published before June 30th 2007 and a ‘‘consensus and minority report” process for papers published after June 30th 2007. Further- more a different process was used in the original study i.e. an extractor and checker process. However, quality comparisons within each group of studies are comparable, so our conclusion that recent SLRs score better with respect to quality is reliable.

A final issue with respect to SLR selection and extracting subjec- tive data, is that one person (Kitchenham) reviewed and extracted data from all the papers. Although this might potentially have introduced bias, we felt it was important for one person to have an overview of all the papers and Kitchenham took this role since she was the most experienced researcher in this field.

6. Conclusions

The results of this study show two main changes comparedwith our previous study:

- The number of SLRs being published appears to be increasing. However, it is still the case that many literature reviews are not performed in accordance with any methodology. Over the time

period January 1st 2004 to 30th June 2008, we found 53 SLRs (of varying degrees of quality) but we also found 54 literature reviews that did not use any defined search strategy (see Section 3.2). This set of 54 studies does not include either the 14 candidate SLRs we rejected in the original tertiarystudy [12] or the four studies we excluded during data extrac- tion (see Section 3.4).

- The quality of SLRs being published appears to have increased among all types of publication, particularly workshop publi- cations, with the exception of studies where the authors appear unaware of the SLR guidelines.

These results make it clear that the task of cataloguing high quality SLRs must be based on a broad search of all sources and

not a targeted manual search of a limited number of software engi- neering conferences and journals. However, ensuring complete- ness of an automated search remains problematic. We strongly recommend authors to use the terms ‘‘systematic review” or ‘‘sys- tematic literature review” in their keywords or title if they want their studies to be easily found. It seems also that there may be considerable delay between a conference paper being published and information about the paper appearing in any indexing sys- tem. Furthermore, this problem is likely to affect any SLR not just our tertiary study and suggests that, for completeness, automated searches need to be backed up with manual searches of the most recent relevant conference proceedings. It may also be wise to undertake another search using an indexing system such as SCO- PUS prior to publishing the results of an SLR.

The results of this study are consistent with previous results in

that:

Few SLRs include practitioner guidelines.

Few SLRs evaluate the quality of primary studies. We suggest authors should provide a rationale if they do not evaluate pri- mary study quality, for example, this is reasonable for a large scale mapping study where follow-on SLRs would be expected to consider the quality of the primary studies.

- Few SLRs are authored by researchers from the USA.

This study has also considered the relationships between SLR topics and the software engineering undergraduate curriculum. Currently coverage of the undergraduate curriculum topics is lim- ited. Whether this is because of a lack of primary studies we do not know. However coverage is increasing. Furthermore, we believe it would be a worthy long-term goal for empirical software engineer- ing research to provide empirical support for topics identified in the undergraduate curriculum and assist academics preparing edu- cational material or text books.

Acknowledgement

This study was funded by the UK Engineering and Physical Sci- ences Research Council project EPIC/E046983/1.

References

B.A. Kitchenham, T. Dybå, M. Jørgensen, Evidence-based software engineering, in: Proceedings of the 26th International Conference on Software Engineering, (ICSE '04), IEEE Computer Society, Washington DC, USA, 2004, pp. 273–281.
T. Dybå, B.A. Kitchenham, M. Jørgensen, Evidence-based software engineering for practitioners, IEEE Software 22 (1) (2005) 58–65.
M. Jørgensen, T. Dybå, B.A. Kitchenham, Teaching evidence-based software engineering to university students, in: 11th IEEE International Software Metrics Symposium (METRICS'05), 2005, p. 24.
Khan S. Khalid, Regina Kunz, Jos Kleijnen, Gerd Antes, Systematic Reviews to Support Evidence-based Medicine, Springer, 2003.
A. Fink, Conducting Research Literature Reviews. From the Internet to Paper, Sage Publication, Inc., 2005.
Mark Petticrew, Helen Roberts, Systematic Reviews in the Social Sciences: A Practical Guide, Blackwell Publishing, 2005.
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).
B.A. Kitchenham, S. Charters, Guidelines for Performing Systematic Literature Reviews in Software Engineering Technical Report EBSE-2007-01, 2007.
D.I.K. Sjøberg, J.E. Hannay, O. Hansen, V.B. Kampenes, A. Karahasanovic, N.K. Liborg, A.C. Rekdal, A survey of controlled experiments in software engineering, IEEE Transactions on Software Engineering 31 (9) (2005) 733– 753.
J. Hannay, M. Jørgensen, The role of deliberate artificial design elements in software engineering experiments, IEEE Transactions on Software Engineering 34 (2) (2008) 242–259.
V.B. Kampenes, T. Dybå, J.E. Hannay, I. Dag, K. Sjøberg, A systematic review of effect size in software engineering experiments, Information and Software Technology 49 (11–12) (2007) 1073–1086.
B. 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.
R.L. Glass, I. Vessey, V. Ramesh, Research in software engineering: an analysis of the literature, Information and Software Technology 44 (8) (2002) 491–506.
B. Kitchenham, P. Brereton, M. Turner, M. Niazi, P. Pretorius, D. Budgen, The impact of search procedures for systematic literature reviews – a participant- observer case study, in: Proceedings of Symposium on Empirical Software Engineering & Metrics, ESEM, 2009, pp. 336–345.
C. Jefferies, P. Brereton, M. Turner, A systematic literature review to investigate reengineering existing systems for multi-channel access, in: Conference on Software Maintenance and Reengineering (CSMR), Athens, April 2008, pp. 258–262.
J. Bailey, D. Budgen, M. Turner, B. Kitchenham, P. Brereton, S. Linkman, Evidence relating to object-oriented software design: a survey, in: Proceedings of Empirical Software Engineering and Measurement, IEEE Computer Society Press, 2007, pp. 482–484.
S. MacDonell, M. Shepperd, Comparing local and global effort estimation models reflections on a systematic reviews, in: Proceedings of Empirical Software Engineering and Measurement,

IEEE Computer Society Press, 2007.

L. Harjumaa, J. Markkula, M. Oivo, How does a Measurement Programme Evolve in Software Organizations? PROFES 2008, LNCS 5089, 2008, pp. 230– 243.

G.K. Hanssen, F.O. Bjørnson, and H. Westerheim, Tailoring and Introduction of the Rational Unified Process, EuroSPI 2007, LNCS 4764, 2007, pp. 7–18.

P. Mohagheghi, R. Conradi, Quality, productivity and economic benefits of software reuse: a review of industrial studies, Empirical Software Engineering 12 (2007) 471–516.

M.N. Wicks, R.G. Dewar, A new research agenda for tool integration, Journal of Systems and Software 80 (2007) 1567–1585.

P. Mohagheghi, V. Dehlen, Where Is the Proof? – A Review of Experiences from Applying MDE in Industry, ECMDA-FA, LNCS 5095, 2008, pp. 432–443.

R.C. de Boer, R. Farenhorst, In search of 'architectural knowledge', in: SHARK '08: Proceedings of the 3rd International Workshop on Sharing and Reusing Architectural Knowledge, May 2008, pp. 71–78.

A.P. Freire, R. Goularte, R.P.M. Fortes, Techniques for developing more accessible web applications: a survey towards a process classification, in: SIGDOC '07: Proceedings of the 25th Annual ACM International Conference on Design of Communication, October 2007, pp. 162–169.

M. Renger, G.L. Kolfschoten, G.-J. de Vreede, Challenges in Collaborative Modeling: A Literature Review, CIAO! 2008 and EOMAS 2008, LNBIP 10, 2008, pp. 61–77.

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).

F. Dong, M. Li, Y. Zhao, J. Li, Y. Yang, Software Multi-Project Resource Scheduling: A Comparative Analysis, ICSP 2008, LNCS 5007, pp. 63–75.

M.I. Kamata, A.Y. Yoshida, H. Yoshida, N. Aoji, Figure out the current software requirements engineering – what practitioners expect to requirements engineering? in: 14th Asia-Pacific Software Engineering Conference, 2007, pp. 89–96.

A. Trifonova, S.U. Ahmed, L. Jaccheri, SArt: towards innovation at the intersection of software engineering and art, in: Proceedings of the 16th International Conference on Information Systems Development, 2007.

E. Hasnain, T. Hall, Investigating the Role of Trust in Agile Methods Using a Light Weight Systematic Literature Review, XP 2008, LNBIP 9, 2008, pp. 204– 207.

A.D. Neto, R. Subramanyan, M. Viera, G.H. Travassos, F. Shull, Improving evidence about software technologies, a look at model-based testing, IEEE Software 25 (6) (2008) 242–249.

S. Grimstad, M. Jorgensen, K. Møløkken-Østvold, 2005 The clients' impact on effort estimation accuracy in software development projects, in: 11th IEEE International Software Metrics Symposium (METRICS'05), p. 3.

M. Jørgensen, Evidence-based guidelines for assessment of software development cost uncertainty, IEEE Transactions on Software Engineering (2005) 942–954.

A. Davis, O. Dieste, A. Hickey, N. Juristo, A.M. Moreno, Effectiveness of requirements elicitation techniques: empirical results derived from a systematic review, in: 14th IEEE International Requirements Engineering Conference (RE'06), 2006, pp. 179–188.

M. Shepperd, Software Project Economics: A Roadmap, FOSE'07, 2007.

M. Mair, M. Shepperd, M. Jørgensen, An Analysis of Data Sets Used to Train and Validate Cost

Prediction Systems, PROMISE'05 Workshop, 2005.
A. Yalaho, A conceptual model of ICT-supported unified process of international outsourcing of software production, in: 10th International Enterprise Distributed Object Computing Conference Workshops (EDOCW'06), 2006.
H. Kagdi, M.L. Collard, J.I. Maletic, A survey and taxonomy of approaches for mining software repositories in the context of software evolution, Journal of Software Maintenance and Evolution: Research and Practice 19 (2) (2006) 77– 131.
J. Segal, A. Grinyer, H. Sharp, The Type of Evidence Produced by Empirical Software Engineers, REBSE'05, 2005.
M. Höst, C. Wohlin, T. Thelin, Experimental context classification: incentives and experience of subjects, in: ICSE'05, Proceedings of the 27th International Conference on Software Engineering, ACM, 2005.
J.H. Hosbond, P.A. Nielsen, Mobile Systems development – a literature review, in: Proceedings of IFIP 8.2 Annual Conference, 2005.
M. Shaw, P. Clements, The Golden Age of Software Architecture: A Comprehensive Survey. Technical Report CMU-ISRI-06-101, Software Engineering Institute, Carnegie Mellon University, 2006.
A. Davis, A. Hickey, O. Dieste, N. Juristo, A.M. Moreno, A Quantitative Assessment of Requirements Engineering Publications – 1963–2006, LNCS 4542/2007, Requirements Engineering: Foundation for Software Quality, 2007, pp. 129–143.
A. Höfer, W.F. Tichy, Status of empirical research in software engineering, in:
V. Basili et al. (Eds.), Empirical Software Engineering Issues, Springer-Verlag, 2007, pp. 10–19 (LNCS 4336).
J. Feller, P. Finnegan, D. Kelly, M. MacNamara, Developing Open Source Software: A Community-Based Analysis of Research, IFIP International Federation for Information Processing, 208/2006, Social Inclusion: Societal and Organizational Implications for Information Systems, 2006, pp. 261– 278.
G.A. Liebchen, M. Shepperd, Data sets and data quality in software engineering, in: PROMISE '08: Proceedings of the 4th International Workshop on Predictor Models in Software Engineering, May 2008, pp. 39–44.
F.J. Pino, F. García, M. Piattini, Software process improvement in small and medium enterprises: a review, Software Quality Journal 16 (2008) 237–261.
C.G. Bellini, R.D.C.D.F. Pereira, J.L. Becker, Measurement in software engineering from the roadmap to the crossroads, International Journal of Software Engineering and Knowledge 18 (1) (2008) 37–64.
M. Staples, M. Niazi, Systematic review of organizational motivation for adopting CMM-based SPI, Information and Software Technology 50 (2008) 605–620.
H. Zhang, B. Kitchenham, D. Pfahl, Reflections on 10 years of software process simulation modeling: a systematic review. International Workshop on Software Process Simulation Modeling, LNCS 5007, 2008, pp. 345–356.
J. Biolchini, P.G. Mian, A.C.C. Natali, G.H. Travassos, Systematic Review in Software Engineering. Technical Report ES 679/05, 2005.
B. Kitchenham, P. Brereton, M. Turner, M. Niazi, S. Linkman, R. Pretorius, D. Budgen, Refining the systematic literature review process – two participant- observer case studies, Empirical

Software Engineering, submitted for publication.
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.
A. Abran, J. Moore, P. Bourque, T. Dupuis (Eds.), Guide to Software Engineering Body of Knowledge, IEEE Computer Society, 2004.
B. Kitchenham, D. Budgen, P. Brerton, Evidence-based software engineering and systematic literature reviews. Upgrade, The Journal of CEPIS (Council of European Professional Informatics Societies), 2009.
R.F. Barcelos, G.H. Travassos, Evaluation approaches for Software Architectural Documents: A systematic Review, Ibero-American Workshop on Requirements Engineering and Software Environments (IDEAS), La Plata, Argentina, 2006.
D. Galin, M. Avrahami, Are CMM program investments beneficial? Analyzing past studies, IEEE Software 23 (6) (2006) 81–87.
S. Grimstad, M. Jorgensen, K. Molokken-Ostvold, Software effort estimation terminology: the tower of Babel, Information and Software Technology 48 (4) (2006) 302–310.
M. Jørgensen, A review of studies on expert estimation of software development effort, Journal of Systems and Software 70 (1–2) (2004) 37–60.
M. Jørgensen, Estimation of software development work effort: evidence on expert judgement and formal models, International Journal of Forecasting 3 (3) (2007) 449–462.
N. Juristo, A.M. Moreno, S. Vegas, M. Solari, In search of what we experimentally know about unit testing, IEEE Software 23 (6) (2006) 72–80.
B. Kitchenham, E. Mendes, G.H. Travassos, A systematic review of cross- vs. within-company cost estimation studies, IEEE Trans on SE 33 (5) (2007) 316– 329.
C. Mair, M. Shepperd, The consistency of empirical comparisons of regression and analogy-based software project cost prediction, in: International Symposium on Empirical Software Engineering, 2005.
K.J. Moløkken-Østvold, M. Jørgensen, S.S. Tanilkan, H. Gallis, A.C. Lien, S.E. Hove, A survey on software estimation in the norwegian industry, in: Proceedings Software Metrics Symposium, 2004.
H. Petersson, T. Thelin, P. Runeson, C. Wohlin, Capture–recapture in software inspections after 10 years research – theory, evaluation and application, Journal of Systems and Software 72 (2004) 249–264.
P. Runeson, C. Andersson, T. Thelin, A. Andrews, T. Berling, What do we know about defect detection methods?, IEEE Software 23 (3) (2006) 82–86
M. Torchiano, M. Morisio, Overlooked aspects of COTS-based development, IEEE Software 21 (2) (2004) 88–93.
E.M. Rogers, Diffusion of Innovations, forth ed., FreePress, New York, 1995.
J.E. Hannay, T. Dybå, E. Arisholm, D.I.K. Sjøberg, The effectiveness of pair- programming: a meta-analysis, IST 51 (7) (2009) 1110–1122.
M. Ciolkowski, What do we know about perspective-based reading? An approach for quantitative aggregation in software engineering, in: Third International Symposium in Empirical Software Engineering and Measurement, 2009, pp. 133–144.