A Systematic Review of Web Engineering Research

M Shailaja,Y Laxmi Prasanna,G L N V Kartheek
Associate Professor[1,2], Assistant Professor[3]
Dept. of CSE,
mail-id: shailajacse@anurag.ac.in, prasannacse@anurag.ac.in, kartheek.cse@anurag.ac.in
Anurag Engineering College,Anatagiri(V&M),Suryapet(Dt),Telangana-508206

Abstract

This paper uses a systematic literature review as means of investigating the rigor of claims arising from Web engineering research. Rigor is measured using criteria combined from software engineering research. We reviewed 173 papers and results have shown that only 5% would be considered rigorous methodologically. In addition to presenting our results, we also provide suggestions for improvement of Web engineering research based on lessons learntby the software engineering community.

Introduction

The term "Web engineering" was first published in 1996 in a conference paper by Gallersen et al. [13]. Since then this term has been cited in numerous publications, and numerous activities devoted to discussing Web engineering have taken place (e.g. workshops, conference tracks, entire conferences). Web engineering is described as:
"the use of scientific, engineering, and management principles and systematic approaches with the aim of successfully developing, deploying and maintaining high quality Web-based systems andapplications"[32].

Engineering is widely taken as the disciplined application of scientific principles to solving practical problems [50], [29], [18], where scientific principles are the result of applying a scientific process [17]. A process in this context means that theories are neverset, i.e., theories may be modified or replaced as new evidence is discovered through the accumulation ofdata and knowledge.
The scientific process supports knowledge building, which in turn involves the use of empirical studies that test models previously created, in order to ensure
whether the current understanding of the discipline is correct. Therefore experimentation in Web engineeringis essential. Conventional wisdom, intuition, conjecture, and proofs of concepts are not reliable sources of credible knowledge [1],[3]. Disciplines suchas Physics, Medicine, and Manufacturing all employ the Scientific process, which undoubtedly contributes to their progress [3]. The software engineering literature has examples of empirical studies that can be as rigorous as those performed in other sciences (e.g. [31],[40],[41],[48]). Given the definition of Web engineering and its resulting implications, we wantedto examine the state of Web engineering research from the point of view of the following research question: How rigorous have any claims arisen from Web engineering research been?
Rigor was measured using criteria combined from [3],[12],[26], and described in Section 3.5.
We have selected 173 primary studies published in scholarly literature (e.g. Web engineering track of the WWW conference, IEEE Multimedia special issues on Web engineering, Web engineering conferences). Results have shown that only eight (5%) would be considered rigorous methodologically, suggesting that Web engineering research seems to differ significantly from more traditional forms of engineering. The lackof rigour might lead to Web engineering practitioners who are faced with an overwhelming number of choices on methods, technologies and tools to use without knowing which are suitable or not to their situation. A similar trend has occurred in the field of software engineering [12],[45],[8], however, many rigorous experiments, surveys and case studies have been conducted, with results that have contributed to the advancement of this discipline (see for example [11],[3]).
The remaining of this paper is organised as follows: Section 2 presents related work. Section 3 describes the systematic review, followed by the presentation of its results in Section 4. Section 5 discusses threats tothe validity of the results, and suggestions for improving the current state of Web engineering research and practice are given in Section 6. Finally, issues related to conducting systematic reviews are presented in Section7, and conclusions and comments on future work are described in Section 7.

1. Related Work

Three previous studies we are aware of, one in Computer Science [42] and two Software Engineering [16],[36], had similar aims to ours, however both were carried out using methods different from ours.

The first, by Perry et al. [36] aimed at summarising the strengths and weaknesses of empirical research in software engineering, and to provide suggestions ofimproving the research in this field. They discuss the state of empirical research by citing three studies in empirical software engineering, two of which chosenon the basis of “being influential”, and “widely quoted”.

The second study, by Glass et al. [16], examined thestate of research in Software Engineering from the point of view of the following research questions:

1. What topics do CS researchers address?
2. What research approaches do CS researchers use?
3. What research methods do CS researchers use?
4. On what reference disciplines does CS researchdepend?
5. At what levels of analysis do CS researchersconduct research?

They examined 369 articles from 6 journals in the software engineering literature, within the period of 1995 to 1999. These journals were chosen since they had been used several times for another study on Top scholars and Top institutions in the field of Systems/Software Engineering. All the articles from these 6 journals were examined.

Finally, the third study, by Ramesh et al. [42], examined the state of Computer Science research from the point of view of the same research questions investigated in [16]. In order to address these research questions, they examined 628 papers published between 1995 and 1999 in 13 research journals in the Computer Science field. The journals included were either published by ACM or IEEE. The aim was to examine a sample of approximately 500 articles, where articles from ACM and IEEE were as much as possibleequally represented in the sample set.

Our motivation for using a systematic review was that we wanted results that were unbiased and fair.

Systematic Review (SR)

Introduction

A systematic review is a method that enables the evaluation and interpretation of all accessible research relevant to a research question, subject matter, or event of interest [22], [23]. There are numerous motivations for carrying out a systematic literature review, amongstwhich the most common are [22]:

- To review the existing evidence regarding a treatment of technology, for example, to review existing empirical evidence of the benefits and limitations of a specific Web development method.
- To identify gaps in the existing research that will lead to topics for further investigation.
- To provide a context/framework so as to properly place new research activities.

A Systematic review generally comprises the following steps [1][35]:

- Formulation of a focused review question;
- Identification of the need for carrying out asystematic review;
- A comprehensive, exhaustive search and inclusionof primary studies;
- Quality assessment of included studies;
- Data extraction;
- Summary and synthesis of study results (meta-analysis);
- Interpretation of the results to determine theirapplicability;
- Report-writing.

Prior to the review, it is desirable to develop a protocol that specifies the plan that the systematic review will follow to identify, assess and collate evidence.

A well-formulated question generally has four parts [35], identified as PICO (Patient, Intervention, Comparison, Outcome):

- The population (e.g. the disease group, or aspectrum of the healthy population);
- The study factor (e.g. the intervention, diagnostictest, or exposure);
- The comparison intervention (if applicable);
- The outcome.

The question should be sufficiently broad to allow examination of variation in the study factor and acrosspopulations [1].

Research Question

Within the context of this paper we have carried outa systematic literature review using the basic approach identified in [23], in order to examine the state of Web engineering research from the point of view of the following research question: *How rigorous have any claims arisen from Web engineering research been?*

The criteria used to measure 'rigour' are presentedin Section 3.5 as part of Data Extraction.

This study's population and intervention did not strictly follow the guidelines suggested in the systematic reviews literature (see Table 1), since they were broader than what the guidelines recommend.

Table 1 – Differences between our approachand suggested by guidelines

	Defined according to our goal	Strictly defined according to guidelines
Population	Web engineering full research papers	Web applications, sites, projects, systems etc
Intervention	basis upon which claims were based	methods, technologies, frameworks etc
Outcome	no focus on the outcome itself	no focus on the outcome itself
Experimental design	any	any

Our review was more limited than a full systematic review because we did not follow up the references in papers nor did we extend our search to include grey literature sources such as PhD theses and technicalreports.

Search Strategy Used for Primary Studies

The resources we used to search for primary studies are listed below:

- Web engineering tracks at the World-Wide Web conference (2003, 2004) [51],[52].
- International Web engineering conferences (2003, 2004) [7],[28].
- IEEE Multimedia Special Issue on Web engineering (2 volumes published in 2001) [13],[15].
- IEEE Software Special issue on "Engineering Internet Software (1 volume published in 2002) [20].
- A book on Web engineering by Springer (LNCS) published in 2001 [32].
- IEEExplore electronic database.
- ACM Digital library.

In relation to IEEExplore and ACM Digital library electronic databases we ensured that our search was applied to journals, magazines and conference proceedings published over the last 9 years, i.e. since term Web engineering was coined.

Publications such as the Proceedings of the Ibero American Conferences on Web engineering (2001,2002), the International Journal of Web Engineering and Technology, and Journal of Web engineering were excluded from this Systematic review since we did not have access to these publications. The absence of these publications is afactor that has reduced the validity of our results. Many reasons may affect the availability of all relevant sources. For example, some conference proceedingsmay not be available on-line (e.g. Proceedings of the Ibero American Conferences on Web engineering (2001,2002)), and sometimes, even if they are, subscribing to databases that index such publications may be too expensive for an individual or Institution.

We experimented with several different search criteria using different combinations of strings obtained from the population and intervention. However the one that retrieved the highest number of useful literature was:

(Web engineering OR WWW engineering OR (s1)World-Wide Web engineering OR
Internet engineering)

The choice to concatenate 'engineering' to 'Web' and its synonyms was motivated by our goal, which was to retrieve papers specifically focused at 'Web engineering' research. Therefore the choice of search strings such as "(Web or WWW or World-Wide Web or Internet) and engineering" was removed. The search string was used in all instances, even when examining papers from special issues in Web engineering.

Study Selection Criteria and Procedures

We included any primary studies where there were claims, based on either advocacy research[1], proof of concept, common wisdom, experience report, empirical evaluation and data.

We excluded papers if no claims were given, and also if they were short papers or an introduction to special issues/workshops/tutorials/mini-tracks/book'sSection. In addition, we considered only once any

[1] authors describe a new concept/tool/method in detail, using arguments to persuade readers of its benefits, and recommend that it be transferred to practice. No empirical evaluation or even proof of concept is provided [12].

papers retrieved using IEEExplore and ACM Digital library that have already been included in our evaluation (e.g. IEEE Multimedia Special issues).

Data Extraction Strategy

The data extracted used the same criteria employed to measure rigor of claims, detailed using the followingquestions:

1. What were the claims based upon? (advocacy research, proof of concept, common wisdom, experience report, empirical evaluation and data)
2. If based on empirical evaluation and data:
   i. Was the survey/case study/formal experimentdesigned correctly? (Yes, No)
   ii. Is it based on a toy or real situation? (Toy,Real)
   iii. Were the measurements used appropriate tothe goals of the evaluation? (Yes, No)
   iv. Was the evaluation run for a long enoughtime? (Yes, No)

We adopted the classification for empirical evaluations as suggested in [10], where evaluations are classified as either surveys, case studies or formal experiments. A Survey is a retrospective study of a circumstance aiming at documenting relationships and outcomes. It generally targets at a large audience and is commonly named "research in the large". Case studies and experiments study a current activity as it happens. Their difference relies on the amount of control enforced. Formal experiments impose strict control over the variables under investigation, since this is the only way to obtain results of more general validity that can lead to building a theory. Control here is also important in order to allow for replications. Formal experiments are also known as "research in the small". In Software engineering most formal experiments use students as subjects and toy problems in toy situations as experimental objects [48],[6]. This happens because it is generally difficult to find companies that can afford to let a subset of their software development team to take part in an experiment, unless of course they get paid for it [47]. Case studies do not impose strict control over variables and therefore do not provide results that can be generalised outside their context of investigation [26]. In Software engineering these are commonly used in industrial settings on real projects [11],[40],[41],[47]. Case studies are commonly known as "research in the typical". The use of toy problems in toy situations, i.e., the use of artificial problems in artificial situations, is better than not conducting any evaluations at al. They can be of benefit to explore an initial idea or research design [10]. However, their drawback is in that artificial problems may be too simplistic compared to current industrial practice and difficult to scale up. Occasionally empirical evaluations are designed correctly, but measure and analyse insufficient or wrongdata.

For example, measuring Web applications' size with function points in order to construct effort estimation models is insufficient as function points measure only one aspect of a Web application's size (its functionality as perceived by users). Size should also take into account graphics and animations created from scratch or adapted from previous media as these also account for a project's development effort.

It is important that all measurements used are appropriate to the goals of the evaluation. In addition to measuring the correct variables, researchers must also take care to evaluate and manipulate the measurements in a way that is appropriate to the design and type of data collected [12],[25]. For example, calculating the average on data measured on an ordinal scale is incorrect [10],[4].

Finally, another issue of importance regarding empirical evaluations is the length of time an evaluation was carried out for. For example, within an industrial context where all Web distributed applications are deployed using the development environment J2EE, if we were to investigate the benefits of using .NET instead of J2EE we would need to let development teams have experience with a number of major .NET developments (in order to assume similar expertise to that using J2EE) otherwise conclusions could be misleading. The objective here is to make sure that any practices that promise a profound effect on development and maintenance have been investigated without bias, such as a learning effect [12].

Discussion

The searches using the IEEExplore and ACM Digital library were carried out on the 27th of October 2004. IEEExplore provides full-text search and allows for the use of complex Boolean expressions. Therefore we were able to search the database using the same search string presented in Section 3.2 (s1). Our search retrieved 207 papers of which 20 complied with our selection criteria. The ACM digital library also provides full-test search however it does not allow for the use of complex Boolean expressions. Moreover, even simpler search strings turned out to be problematic. We initially used the advanced search mechanism and the option "must have **any** of the words or phrases", with the following set of strings:

"Web engineering" "WWW engineering" "World- Wide Web engineering" "Internet engineering". No results were returned and after a few trials we realised that more than two phrases **always** resulted in no results. Therefore we decided to run four separate searches and to use in our systematic review the union of the results obtained. The number of papers retrieved for each search string is as follows:

Web engineering : 45 papers
WWW engineering: 1 paper
World-Wide Web engineering: 0 papers
Internet engineering : 15 papers

There were no common papers from these four searches therefore we retrieved 61 papers using ACM Digital library, of which 21 complied with ourselection criteria. In total we retrieved 343 primarystudies, of which 163 complied with our inclusion criteria.

Our search criteria missed 9 relevant papers we knew about. These papers, despite representing research in Web engineering, did not use any of our search strings. However, these papers were also added to our analysis. In addition, we also knew about one Web engineering paper published in 2004 in the Proceedings of the EASE conference. They were included in our review since this author had access to them. Therefore we had in total another 10 papers added to the analysis, leading to a total of 173 primary studies to review.

Our review did not include a blind analysis [26], where all papers are coded and author(s) name(s) omitted. Coding every paper and making the necessary adjustments was not feasible due to lack of resources.

4. Results of the Systematic Review

Our results are presented in Table 2. They havebeen organised by publication source and then addedup to provide a broader overview.

These results indicate that 68% of all reviewed papers are not classified as rigorous according to our criteria. Of these 61% are proof of concept papers,22% advocacy research and 17% experience reports.Of the 32% papers that can be classified as empirical, 27% were designed properly, 46% are based on real scenarios, 54% used measurement properly and 43% carried out their evaluations long enough. These percentages in isolation may seem reasonable, howeveronly 14% of the empirical papers were designed properly, were based on a real scenario, used measurement principles properly and lasted for a reasonable duration. This accounts for only 5% (eight) of all papers reviewed. Of these eight papers, seven were case studies: Web cost estimation (five), quality (one), and productivity (one); and one was a survey on multimedia and Web development techniques.

73% of the empirical papers have been designed incorrectly. Often the aim of their evaluations were not made clear, there were no hypotheses, no explanation on how the evaluation was going to be carried out in order to avoid bias, how the gathered data was going to be analysed, the highlighting of threats to its validity etc.

A sizeable amount of the research described in the 173 reviewed papers was carried out in Web engineering model building, i.e., the building of technologies, methods, tools, life cycle models, specification languages etc. Perhaps these authors think of themselves as the Web engineering theorists and leave it to the experimenters to validate their models. We could also speculate that theorists believe their models do not need to be tested since they are self-evident [1]. We identified 26 papers (15%) as being advocacy research, higher in number thanexperience reports. However, attention should really be drawn to the reviewing process for these 26 papers which were judged sound enough to be published. In addition, we reviewed papers where the authors were either unaware of previous work in the field or thought it unnecessary to be referenced.

Numerous papers used incorrect terminology, e.g. "experiment" is used instead of "experience report", and "case study" instead of "proof of concept". However, it is not all bad news for the Web engineering research. We came across three papers (2 full papers, 1 short paper) where concerns and waysfor improvement were presented.

The lack of a common terminology has been raised by Olsina et al. [34] as an impeding issue to the building of a robust ontology. They propose a measurement terminology as a starting point for discussion within the Web engineering community. Their proposal was motivated by standards such as the ISO 9126 [21] and previous work in software engineering by Kitchenham et al. [25] where astandard structure for defining software measures and for modelling complex data sets is suggested.

Calero et al. [5] propose a Web Quality Model to beused as a standard for quantifying Web qualityattributes. The motivation for their work is the lack of quality metrics that have been empirically and/or theoretically validated.

Finally, Oates et al. [33] review methods and empirical research strategies they assume as of potential usefulness to Web engineering research, and suggest that empirical studies should be carried outmore often in Web engineering. These are examples that indicate that there is some responsiveness within the Web engineering community to issues related tothe whole discipline, such as a common vocabulary/terminology, the need for empirical studies,and the need for common frameworks and guidelines that provide the basis for definition and comparison of empirical studies.

		IE3MM1	IE3MM2	IE3SW	W303	W304	ICWE03	ICWE04	IEEE	ACM	Book	10P	Total
Advocacy Research		0	1	0	0	0	8	2	4	5	4	2	26
Proof of Concept		2	3	0	2	8	11	17	4	11	14	0	72
Common wisdom		0	0	0	0	0	0	0	0	0	0	0	0
Experience report		0	0	6	0	1	1	0	5	4	2	0	19
Empirical Evaluation And data		3	2	1	1	5	7	7	7	6	9	8	56
Designed Correctly?	Y	1	1	0	0	1	0	0	0	4	2	6	15
	N	2	1	1	1	4	7	7	7	2	7	2	41
Toy/Real	R	2	2	1	0	1	2	2	2	2	6	6	26
	T	1	0	0	1	4	5	5	5	4	3	2	30
Measurements Appropriate?	Y	1	1	0	0	1	2	5	4	6	2	8	30
	N	2	1	1	1	4	5	2	3	0	7	0	26
Run for enough time?	Y	2	2	1	0	1	2	1	2	2	6	5	24
	N	1	0	0	1	4	5	6	5	4	3	3	32

IE3MM1 - IEEE Multimedia Special Issue on Web engineering Volume 1
IE3MM2 - IEEE Multimedia Special Issue on Web engineering Volume 2
IE3SW - IEEE Software Special issue on "Engineering Internet Software
W303 - Web engineering tracks at the World-Wide Web conference 2003
W304 - Web engineering tracks at the World-Wide Web conference 2004
ICWE03 - International Web engineering conference 2003
ICWE04 - International Web engineering conference 2004
IEEE - IEEExplore electronic database
ACM - ACM Digital library
Book - A book on Web engineering by Springer (LNCS)
10P - 10 papers not selected by our search criteria
Total - Total number of papers

Threats to the Validity of the SR

The systematic review presented here was conducted by a single reviewer, thus without any means to assess whether her tabulation and application of the selection criteria are correct. However this was the only viable option at the time.

Some relevant sources of data, such as the two Web engineering journals, were not employed becauseneither the reviewer nor her Institutions subscribed to those journals at the time.

5. Suggestions for the Web EngineeringCommunity

There are several lessons from Software engineering research and practice, accumulated over more than 30 years of existence, which Web engineering can learn from.

Common Terminology

Every discipline needs a common vocabulary [1],[8], which is necessary for several reasons, such as:
To allow researchers and practitioners tounderstand and cooperate with each other.
To provide the basis for gathering, validating, andanalysing trustworthy data.
To allow for the summary of findings fromseveral empirical studies.
To improve the research and reporting processes.

Fenton and Pfleger [10] proposed a conceptual framework to be used in developing a software measurement program to help with software measurement activities related to an organisation's software practices (e.g. software development and maintenance, case studies, formal experiments). This framework combines principles from the representational theory of measurement, the Goal- Question-Metric paradigm (method used to identify measures required in a particular situation) and processmaturity (identifies the level of maturity of an organisation in relation to its development processes). This framework has been extensively cited in the software engineering literature and has also been employed in practice.

Kitchenham et al. [25] suggested a method for providing a standard structure for defining softwaremeasures and for representing the metadata associated

to data sets, which enables contextual information related to measurement goals. A single method prevents problems associated with software data collection, which often arise from poor definitions of software measures. Their method has provided the basis for Olsina et al.'s work on ontologies however it has not been completely adopted as a standard by the Web engineering community.

Summarising findings from several empirical studies to discover common trends and results leads to the building of scientific theories. Two different ways of summarising results have been proposed in Software Engineering: meta-analysis [19],[30] and systematic literature reviews [23]. The objective of meta-analysis is to provide a quantitative and unbiased procedure for combining results from different studies. This combination allows researchers to measure the extent to which empirical results are consistent across different studies since it is impossible to generaliseresults from a single empirical study even if it is a formal experiment [37]. Systematic literature reviews have already been presented in Section 2.

Guidelines to improve the research and reporting process were proposed by Kitchenham et al. [26]. These guidelines were based on a review of research guidelines developed for medical researchers and the authors' own experience in doing and reviewing software engineering research. The intention of these guidelines is to support researchers, reviewers and meta-analysis in designing, conducting, and evaluating empirical studies. They may also be used by editorial boards as a basis for providing procedures for reviewers and for structuring policies for dealing with the design, data collection, analysis, and reporting of empirical studies. Since these guidelines have been published numerous empirical studies in Software engineering have used them. We have applied these guidelines to our research and therefore believe they can also assist empirical studies in Web engineering.

Basili et al. [3] provide a more specific framework aimed at helping define and combine formal experiments to overcome validity problems. They also suggest how results can be unified and used to generate laboratory manuals to be employed in further replications. Further details on practical application of this framework are provided in [6].

Carry out Empirical Studies

Empirical studies are the building blocks necessary for building evidence and to determine what situations are best for using particular technologies [38]. In Software engineering numerous empirical studies have provided the means for the building of theories (see [11],[45] for examples) and as a results of that Web engineering has the same theories to make use of ratherthan having to start from scratch. Small, controlled andwell-planned experiments and replications, even if making use of students and not conducted in industrial settings, can be beneficial. Carver et al. [6] discuss a series of benefits that researchers can gain from empirical studies using students, which are as follows:

Obtain preliminary evidence to confirm or refute Hypotheses;
Control factors that may affect the study;
Show software companies the relevance of the research;
Show software companies the usefulness of carrying out empirical studies in their own environments;
Show software companies the feasibility of carrying out full-fledged empirical studies in industrial environments;
Fine-tune the organization and details of an empirical study, before it is carried out in an industrial environment;
Produce an experimental "kit.";
Train junior researchers in the empirical research field.

Shaw [45] suggested Software engineering researchers select a suitable mix of short-term, pragmatic, potentially empirical contributions that helpalleviate commercial practice, and to invest in the long-term efforts to develop and make available basic scientific contributions. We believe this is equally applicable to Web engineering.

Other studies also provide guidelines on how tocollaborate with Companies in order to conduct formal experiments and case studies in industrial settings [43],[47].

Develop Professional Web Engineers

Several initiatives have occurred in software engineering in order to make software engineering a profession in its own right. Examples of such initiatives are the software engineering body of knowledge (SWEBOK)2, and several professional master-level degrees offered by educational institutions [9]. Whitehead [49] has recently also proposed a curriculum for a Masters in Web engineering, motivated by the masters in software engineering offered at Oregon [9] and Carnegie Mellon. The Masters degree offered at Oregon includes as part of its core material concepts such as metrics and measurement. At Carnegie Mellon these concepts are provided as an elective course. We can only hope that Masters degrees in Web engineering offered worldwide follow Oregon's example. Another way of identifying education needs in Web engineering is to obtain industry requirements for Web engineering professionals. Software engineering has accomplished this using surveys [24].

Research Networks and Special InterestGroups

The ACM has to date 35 Special Interest Groups (SIGs) that are used as a forum for addressing the needs of IT professionals3. Its participants are practitioners and researchers. There are also other research networks that facilitate the exchange of research results and the dissemination of these results to industry. An example in Software Engineering is the International Software Engineering Research Network (ISERN) where its members perform joint activities and have access to a network of laboratory environments. In Web engineering there is the Web engineering Community Portal4 aimed at providing the community with technical and strategic information on Web Engineering. It provides discussion forums and Interest Groups. As a community portal, however, the information provided should be unbiased. There are currently two Web engineering journals (JWE and IJWET). The community portal only provides information on JWE. We are also aware of an European expression of interest for a Web engineering research network – EWENE (European Web Engineering Network of Excellence)5 with a central goal to "push Web Engineering towards an autonomous discipline". Finally, the ACM SIGWEB used to have a working group on Web Engineering however its home page no longer exists. We believe such networks are important for strengthening research collaborations and industry involvement in Web engineering.

7. Issues Related to Conducting SRs

The issues related to using SRs we would like to raise are as follows:

The SR literature suggests that a SR should be carried out by a group of people with expertise in the subject area, knowledge of statistics etc [35]. Thus, to help PhD students carry out SRs supervisors should assist these students prepare their protocol, provide examples of good qualityprotocols, and ideally a database of technical terms/dictionary of synonyms that can "replace" expert knowledge and be used when preparing the search strings. For example, Web, World-Wide Web, Internet, WWW, Web application, Web site, Website, and Web software are all used as synonyms in the Web engineering literature.
Both software and Web engineering communities are faced with the lack of a common set of search functionality provided by those Institutions looking after electronic resources. IEEExplore provides much better search mechanisms than the ACM digital library or Springer science direct.
We need electronic SR tools to help manage the reviewing process when there are several reviewers (e.g. who has been allocated which primary source). Reference management systems (e.g. EndNote) can store references, abstracts, comments. However, if a publisher (e.g. IEEE, ACM) does not allow the references to be exported to a referencing system everything still needs to be done manually.
Not all Institutions have access to all the proceedings and journals that are relevant for the systematic review. This automatically compromisesthe validity of the SR's results (e.g. no access to theWeb engineering proceedings, journals).
A SR striving to be complete should also take into account publications written in languages other than English. In medical studies the translation of primary sources is also budgeted when planning fora review.
A community portal with forums discussing several issues related to systematic reviews in software and Web engineering, or even a special interest group in systematic reviews, similar to so many run by the ACM, may help disseminate the use of SRs and aid in making software and Web engineering evidence- based. A community portal could also store finished SRs, protocols, and provide information onongoing SRs.
Finally, a narrow focused research question may be difficult to obtain at an early stage of a research project, which may lead to the retrieval of hundreds of primary studies. One would argue why not reviewing only a sample of these primary studies? However, what would be the adequate sample size, and moreover, what would be the most reliable criteria for obtaining an unbiased sample?

8. Conclusions

We have presented the results of a systematic literature review [23] aimed at investigating the rigor of claims arising from Web engineering research. We measured rigor using criteria combined from [3],[12],[26].
We reviewed 173 Web engineering papers published in scholarly literature (e.g. Web engineering track of the WWW conference, IEEE Multimedia special issues on Web engineering, Web engineering conferences). Our results have shown that 68% of all reviewed papers are not classified as rigorous according to our criteria. Of these 61% are proof of concept papers, 22% advocacy research and 17% experience reports. Only 5% of all papers reviewed were designed properly, were based on a real scenario, used measurement principles correctly and lasted for a reasonable duration.
73% of the empirical papers have been incorrectly designed. Often the aim of their evaluations were not made clear, there were no hypotheses, no explanation on how the evaluation was going to be carried out in order to avoid bias, how the gathered data was going to be analysed, the highlighting of threats to its validity etc. A sizeable amount of the research described in the
173 reviewed papers was carried out in Web engineering model building, i.e., the building of technologies, methods, tools, life cycle models, specification languages etc. We identified 26 papers (15%) as being advocacy research, higher in number than experience reports. However, attention should really be drawn to the reviewing process for these 26 papers which were judged sound enough to be published. In addition, we reviewed papers where the authors were either unaware of previous work in the field or thought it unnecessary to be referenced.
Finally, we also found that numerous papers used incorrect terminology, e.g. "experiment" is used instead of "experience report", and "case study" instead of "proof of concept".
Our future work involves carrying out systematic reviews in other areas of Web engineering, such as to review existing empirical evidence of the benefits and limitations of different effort estimation techniques for Web cost estimation.

References

Australian National Health and Medical Research Council. How to review the evidence: systematic identification and review of the scientific literature, IBSN 186-4960329, 2000.

Basili, V.R. The role of experimentation in software engineering: past, current, and future, Proceedings of the 18th International Conference on Software Engineering, (25-30 March 1996), 442 – 449, 1996.

Basili, V.R. Shull, F., and Lanubile, F. Building knowledge through families of experiments, IEEE Transactions on Software Engineering, 25, 4, (July-Aug. 1999), 456 – 473,1999.

Briand, L.C.; Morasca, S.; Basili, V.R. An operational process for goal-driven definition of measures, IEEE Transactions on Software Engineering, (Dec. 2002), 28, 12, 1106 – 1125, 2002**.**

Calero, C. , Ruiz, J., and Piattini, M. A Web Metrics Survey Using WQM, Proceedings ICWE04, LNCS 3140, Springer- Verlag Heidelberg, pp. 147 - 160, July 2004, 2004.

Carver, J., Jaccheri, L., Morasca, S., and Shull, F. Issues in using students in empirical studies in software engineering education, Proceedings Ninth International Software Metrics Symposium, (3-5 Sept. 2003), 239 – 249.

Cueva Lovelle, J.M., González Rodríguez, B.M., Joyanes Aguilar, L., Labra Gayo, J.E., del Puerto Paule de Ruiz, M. (Eds.), Proceedings Third International Conference Web Engineering, Oviedo, Spain, (July 14-18), LNCS 2722, ISBN: 3-540-40522-4, 2003.

Ebert, C. The road to maturity: navigating between craft and science, IEEE Software, 14, 6, (Nov/Dec 1997), 77-82, 1997.

Faulk, S.R. Achieving industrial relevance with academic excellence: lessons from the Oregon master of software engineering, Proceedings of the 2000 International Conference on Software Engineering, (4-11 June 2000), 293 – 302, 2000.

Fenton, N., and Pfleeger, S. L. Software Metrics: A Rigorous and Practical Approach, Second Edition. International Thomson Computer Press, 1996.

Fenton, N., Marsh, W., Neil, M., Cates, P., Forey, S., Tailor, M. Making resource decisions for software projects, Proceedings 26th International Conference on Software Engineering, (23-28 May 2004), 397 – 406, 2004.

Fenton, N., Pfleeger, S.L., and Glass, R.L. Science and Substance: A Challenge to Software Engineers, IEEE Software, (July 1994), 86-95, 1994.

Gellersen, H., Wicke, R., and Gaedke, M. WebComposition: an object-oriented support system for the Web engineering lifecycle, Computer Networks and ISDN Systems, Volume 29, Issues 8-13, Pages 865-1553 (September 1997) Papers from the Sixth International World Wide Web Conference, Pages 1429- 1437,1996.

Ginige, A., and Murugesan, S. (Eds), IEEE Multimedia Special Issue on Web Engineering: Part 1, 8,1, (Jan-March 2001), 2001.

Ginige, A., and Murugesan, S. (Eds), IEEE Multimedia Special Issue on Web Engineering: Part 2, 8,2, (April-June 2001), 2001.

Glass, R. L., Vessey, I., and Ramesh, V. Research in software engineering: an analysis of the literature. Information and Software Technology, Volume 44, Issue 8, 1 June 2002, Pages 491-506.

Goldstein, M., and Goldstein, I. F. How we know : an exploration of the scientific process, New York : Plenum Press, 1978.

HarperCollins, Publishers, Collins English Dictionary, 2000

Hayes, W.; Research Synthesis in Software Engineering: A Case for Meta-Analysis, Proceedings of Sixth IEEE International Symposium on Software Metrics, (04 – 06 November, 1999), 143-153, 1999.

Hendrickson, E., and Fowler,M (Eds), IEEE Software Special Issue on Engineering Internet Software, (March-April, 2002), 2002.

ISO. International Standard ISO/IEC 9126. Information technology -- Software product evaluation -- Quality characteristics and guidelines for their use, International Organization for Standardization, International Electrotechnical Commission, Geneva, 1991.

Kitchenham, B.: Procedures for Performing Systematic Reviews. Joint Technical Report Software Engineering Group, Keele University, United Kingdom and Empirical Software Engineering, National ICT Australia Ltd, Australia (2004).

Kitchenham, B.A., Dyba, T., Jorgensen, M. Evidence-based software engineering, Proceedings 26th International Conference on Software Engineering, (23-28 May 2004), 273 –281, 2004.

Kitchenham, B.A., Budgen, D., Brereton, P., and Woodall, P. An investigation of software engineering curricula, The Journal of Systems and Software, 2004, article in press.

Kitchenham, B.A., Hughes, R.T., and Linkman, S.G. Modeling software measurement data, IEEE Transactions on Software Engineering, 27, 9, (Sept. 2001), 788 –804, 2001.

Kitchenham, B.A., Pfleeger, S.L., Pickard, L.M., Jones, P.W., Hoaglin, D.C., El Emam, K., and Rosenberg, J. Preliminary guidelines for empirical research in software engineering, IEEE Transactions on Software Engineering, 28, 8, (August 2002), 721 – 734, 2002.

Kitchenham, B. Pickard, L. Pfleeger, S.L. Case studies for method and tool evaluation, IEEE Software, 12, 4, (July 1995), 52 – 62, 1995.

Koch, Nora, Fraternali, Piero, and Wirsing, Martin (Eds.), Proceedings Fourth International Conference Web Engineering, Munich, Germany, (July 26-30), LNCS 3140, ISBN: 3-540- 22511-0, 2004.

Mifflin, Houghton, Company. The American Heritage Concise Dictionary, Third Edition, 1994.

Miller, J.; Can results from software engineering experiments be safely combined?, Proceedings Sixth International Software Metrics Symposium, (4-6 November, 1999), 152 – 158, 1999.

Mohagheghi, P.; Conradi, R.; Killi, O.M.; Schwarz, H. An empirical study of software reuse vs. defect-density and stability, Proceedings 26th International Conference on Software Engineering, (23-28 May 2004), 282 – 291, 2004.

Murugesan, S., and Deshpande, Y. (eds.), Web Engineering, Managing Diversity and Complexity of Web Application Development, Lecture Notes in Computer Science 2016, Springer Verlag, Heidelberg, Germany, 2001.

Oates, B., Griffiths, G., Lockyer, M., and Hebbron, B. Empirical Methodologies for Web Engineering, Proceedings ICWE04, LNCS 3140, Springer-Verlag Heidelberg, (July 2004), 311 – 315, 2004.

Olsina, L., Martin, M.A., Fons, J., Abrahao, S., and Pastor, O. Towards the Design of a Metrics Cataloguing System by Exploiting Conceptual and Semantic Web Approaches, Proceedings ICWE03, LNCS 2722, Springer-Verlag Heidelberg, (August 2003), 324 – 333, 2003.

Pai, M., McCulloch, M, Gorman, J.D., Pai, N., Enanoria, W., Kennedy, G., Tharyan, P., Colford, J.M. Jr. Systematic reviews and meta-analyses: An illustrated step-by-step guide. The National Medical Journal of India, 17(2), 86-95, 2004.

Perry, D. E., Porter, A.A., and Votta, L.G. Empirical studies of software engineering: a roadmap, ICSE 2000, 22nd International Conference on on Software Engineering, Future of Software Engineering Track, June 4-11, 2000, Limerick Ireland. ACM, 2000**.** 345-355.

Pickard, L.M., Kitchenham, B.A., and Jones, P.W. Combining empirical results in software engineering, Information and Software Technology, 40, 14, (1 December 1998), 811-821, 1998.

Pfleeger, S.L. Albert Einstein and Empirical Software Engineering, IEEE Computer, (October 1999), 32-38, 1999.

Pfleeger, S.L.; Jeffery, R.; Curtis, B.; Kitchenham, B. Status report on software measurement, IEEE Software, 14, 2, (Mar/Apr 1997), 33 – 43, 1997.

Porter, A.A., Votta, L.G.Jr., and Basili, V.R.; Comparing Detection Methods for Software Requirements Inspections: A Replicated Experiment, IEEE Transactions on Software Engineering, 21,6, (June 1995), 1995.

Prechelt, L., Unger, B., Tichy, W.F., Brossler, P., Votta, L.G. A controlled experiment in maintenance: comparing design patterns to simpler solutions, IEEE Transactions on Software Engineering, 27, 12 , (Dec. 2001), 1134 – 1144, 2001.

Ramesh, V., Glass, R.L., and Vessey, I. Research in computer science: an empirical study, the Journal of Systems and Software 70 (2004) 165–176.

Rombach, D. Fraunhofer: the German model for applied research and technology transfer, Proceedings of the 2000 International Conference on Software Engineering, (4-11 June),531 – 537, 2000.

Ruhe, G. Methodological contributions to professional education and training, Proceedings of 24th Annual Computer Software and Applications Conference, (25-27 Oct. 2000), 11 –16, 2000.

Shaw, M. Prospects for an Engineering Discipline of Software, IEEE Software, (Nov. 1990), 15-24, 1990.

Shull, F., Basili, V., Carver, J., Maldonado, J.C., Travassos, G.H., Mendonca, M., and Fabbri, S. Replicating software engineering experiments: addressing the tacit knowledge problem, Proceedings First International Symposium on Empirical Software Engineering, (3-4 Oct. 2002), 7 – 16, 2002.

Sjoberg, D.I.K., Anda, B., Arisholm, E., Dyba, T., Jorgensen, M., Karahasanovic, A., Koren, E.F., and Vokac, M. Conducting realistic experiments in software engineering, Proceedings First International Symposium on Empirical Software Engineering, (3-4 Oct. 2002), 17 – 26.

Thelin, T., Runeson, P., Wohlin, C., Olsson, T., Andersson, C. How much information is needed for usage-based reading? A series of experiments, Proceedings First International Symposium on Empirical Software Engineering, (3-4 Oct. 2002), 127 – 138, 2002.

Whitehead, E.J., Jr., A proposed curriculum for a Masters in Web engineering, Journal of Web Engineering, 1,1, 18-11, 2002.

Wikipedia, http://en.wikipedia.org/wiki/Main_Page, accessed on the 25th of October.

WWW03, Proceedings of the twelfth international conference on World Wide Web 2003, Budapest, Hungary, (May 20 – 24), 2003, http://www2003.org/cdrom/index.html

WWW04, Proceedings of the 13th international conference on World Wide Web 2004, New York, NY, USA, *(*May 17 – 20), 2004, http://www2004.org/proceedings/index.htm.