Available online at www.sciencedirect.com

ScienceDirect

The Journal of Systems and Software

ELSEVIER The Journal of Systems and Software 80 (2007) 571–583

www.elsevier.com/locate/jss

Lessons from applying the systematic literature review process within the software engineering domain

Pearl Brereton [a,*], Barbara A. Kitchenham [a], David Budgen [b], Mark Turner [a], Mohamed Khalil [c]

[a] *School of Computing and Mathematics, Keele University, Keele, Staffordshire ST5 5BG, UK* [b] *Department of Computer Science, Durham University, Durham,*

Science Laboratories, South Road, Durham City, DH1 3LE, UK

[c] *Department of Computer Sciences, University of Khartoum, P.O. Box 321, Khartoum, Sudan*

Received 7 March 2006; received in revised form 26 June 2006; accepted 10 July 2006

Available online 17 August 2006

Abstract

A consequence of the growing number of empirical studies in software engineering is the need to adopt systematic approaches to assessing and aggregating research outcomes in order to provide a balanced and objective summary of research evidence for a particular topic. The paper reports experiences with applying one such approach, the practice of systematic literature review, to the published studies relevant to topics within the software engineering domain. The systematic literature review process is summarised, a number of reviews being undertaken by the authors and others are described and some lessons about the applicability of this practice to software engineering are extracted.

The basic systematic literature review process seems appropriate to software engineering and the preparation and validation of a review protocol in advance of a review activity is especially valuable. The paper highlights areas where some adaptation of the process to accommodate the domain-specific characteristics of software engineering is needed as well as areas where improvements to current software engineering infrastructure and practices would enhance its applicability. In particular, infrastructure support provided by software engineering indexing databases is inadequate. Also, the quality of abstracts is poor; it is usually not possible to judge the relevance of a study from a review of the abstract alone.

© 2006 Elsevier Inc. Open access under CC BY-NC-ND license.

Keywords: Systematic literature review; Empirical software engineering

1. Introduction

Empirical studies are now being undertaken more frequently, as a means of investigating a widening range of phenomena in software engineering. However, as each study is inevitably limited in scope, researchers and decision-makers need to be able to rigorously and systematically locate, assess and aggregate the outcomes from all relevant empirical studies related to a particular topic of interest, in order to provide an objective summary of the relevant evidence. This need has been addressed within a

[*] Corresponding author. Tel.: +44 1782 583079; fax: +44 1782 584268.
E-mail address: o.p.brereton@cs.keele.ac.uk (P. Brereton).

0164-1212 © 2006 Elsevier Inc. Open access under CC BY-NC-ND license.

doi:10.1016/j.jss.2006.07.009

number of other disciplines, including clinical medicine, social policy, education and information systems, through the application of the evidence-based paradigm. The evidence-based paradigm advocates the objective evaluation and synthesis of empirical results of relevance to a particular research question through a process of *systematic literature review* and the integration of that evidence into professional practice (Sackett et al., 2000).

One of the criticisms that can be levelled at researchers in software engineering and computer science, particularly in contrast with those in information systems, is that they make little or no use of the methods and experiences available from other 'reference disciplines' (Glass et al., 2004). We therefore, in this paper, present and discuss our experiences of applying the systematic literature review process to the software engineering domain for the purposes of identifying those aspects of the process that transfer 'as is' to software engineering, those that need to be adapted to the particular characteristics of the domain and those areas where improvements to current software engineering infrastructure and practices are needed in order to gain greatest advantage from the process.

Section 2 provides an introduction to systematic literature review highlighting the main phases and stages of the process. This is followed, in Sections 3–5, respectively, by overviews of three systematic reviews relating to service based systems, the technology acceptance model and guidelines for conducting systematic literature reviews. Lessons learned from our experiences of undertaking these reviews are then described and discussed, and finally some conclusions are drawn.

2. Systematic literature review

The wider context for this study is that of investigating the use of the evidence-based paradigm in software engineering. The possibility of applying the evidence-based paradigm to the software engineering field was raised, discussed and enthusiastically supported at ICSE 2004

(Kitchenham et al., 2004). The goal of evidence-based software engineering (EBSE) is summarised by Kitchenham et al., as being: “to provide the means by which current best evidence from research can be integrated with practical experience and human values in the decision making process regarding the development and maintenance of software”.

Within the medical domain, the steps that need to be followed to practise evidence-based medicine have been identified and documented (Sackett et al., 2000), and these have subsequently been re-formulated to address evidence-based software engineering (Dybå et al., 2005; Kitchenham et al., 2004). The steps are:

1. convert the need for information (about a technique, procedure, etc.) into an answerable question;
2. find the best evidence with which to answer the question;
3. critically appraise the evidence for its validity (closeness to the truth), impact (size of the effect), and applicability (usefulness);
4. integrate the critical appraisal with software engineering expertise and with stakeholders’ values and circumstances;
5. evaluate the effectiveness and efficiency in executing steps 1–4 and seek ways to improve them.

The first three of these steps essentially constitute a systematic review of the literature, conducted in order to provide a balanced and objective summary that is relevant to meeting a particular need for information. A systematic (literature) review is “a means of evaluating and interpreting all available research relevant to a particular research question or topic area or phenomenon of interest” (Kitchenham, 2004). The research papers summarised in the review are referred to as *primary* studies, while the review itself is a *secondary* study. The accumulation of evidence through secondary studies can be very valuable in offering new insights or in identifying where an issue might be clarified by additional primary studies. For example, a study of software cost overruns showed that the results reported in a highly influential study carried out in the early 1990s (The CHAOS report) were significantly out of step with those reported in other studies (Jørgensen and Moløkken-Østvold, 2006). A critical evaluation of the CHAOS report by Jørgensen and Moløkken-Østvold identified several methodological problems. Another example, where new insights have emerged, is a systematic review of statistical power in software engineering experiments (Dybå et al., 2006). Here, the results show “that the statistical power of software engineering experiments falls substantially below accepted norms as well as the levels found in related discipline of information systems research”. The authors go on to make recommendations about how empirical software engineering researchers might address the reported shortcomings.

Performing a systematic review involves several discrete activities, which can be grouped into three main phases: planning; conducting the review; and reporting the review.

Fig. 1 illustrates the overall 10-stage review process.

Systematic literature reviews are primarily concerned with the problem of aggregating empirical evidence which may have been obtained using a variety of techniques, and in (potentially) widely differing contexts—which is commonly the case for software engineering. While they are used in information systems research (Webster and Watson, 2002), they are less common in software engineering (however, see (Glass et al., 2002) as an example of a secondary study that samples literature within the software engineering domain). Indeed, at the present time, outside of information systems research, reviews in any form, as well as review journals are really not part of the computing

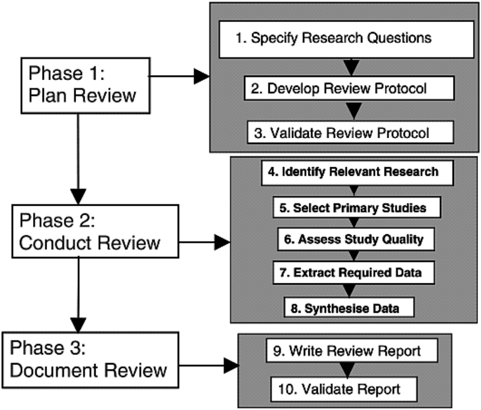

Fig. 1. Systematic literature review process.

research culture, which focuses almost entirely on publishing primary studies.

To understand the role of evidence, we need to recognise that, across a wide spectrum of disciplines of study, there is a common requirement to find objective practices that can be employed for aggregating the outcomes of different empirical studies in a consistent manner. The range of forms and issues is very wide: at the one end, aggregating (say) experimental studies measuring the mass of the electron is largely a matter of using mathematically based transformations to adjust for variations in experimental conditions; whereas drawing together the results from a set of surveys, that may have employed different sets of questions and been administered to rather different populations, presents a much less mathematically tractable problem. One of the key issues underlying this difference is the role of the human in the process of data collection: in the former the only involvement is as an external observer, while in the latter, the human is a participant in the treatment itself.

This process of aggregation is a fundamental one for any evidence-based approach that is seeking to provide objective summaries of empirical data. The area of medicine occupies an intermediate position between the two examples above. Clinical medicine (at least) is able to make extensive use of randomised controlled trials (RCTs) as its experimental paradigm, and in the RCTs used in clinical studies, the role of the human is as a subject, being a recipient of the experimental treatment. This makes it possible to carry out meta-analysis for aggregation of the outcomes, and this, along with the nature of some of the outcomes, has helped to cement the success of evidence-based medicine. In software engineering, however, RCTs and meta-analysis[1] are rarely possible and we need to establish whether the results of a systematic review are useful despite the absence of these procedures.

An essential element in conducting a secondary study such as a systematic literature review is to establish a *protocol* for the study (during the planning phase). The protocol aims to minimise bias in the study by defining in advance how the systematic review is to be conducted, and is itself a document that should be reviewed. The protocol embodies the detailed plan for the review, specifying the process to be followed, any conditions to apply when selecting primary studies, any boundary conditions, quality measures etc. It may also specify details of who will undertake the various tasks and sub-tasks. The protocol may be quite a substantial document (that for the second case study described in this paper is about twenty pages long) and should itself be subject to version control. In particular, the reasons for any changes that occur after the protocol has been agreed should be recorded. As a consequence, the protocol should also be usable as the basis for documenting a review.

In order to investigate the applicability of systematic literature reviews to the software engineering domain, the authors (with others) have undertaken, or are in the process of undertaking, a number of reviews which aim to address a range of software engineering questions. These reviews are summarised in the following sections using the structured abstract headings (context, objectives, methods, results, conclusions) which form part of the recommendations for reporting systematic reviews (Khan et al., 2001). In addition, for each of the reviews, there is a brief introduction and a description of its current state. The reviews have used Kitchenham's guidelines as described in (Kitchenham, 2004) or, in some cases, in earlier versions of the report. In Section 6, we discuss the lessons learned from these reviews against the process model shown in Fig. 1.

We would emphasise that the outcomes from these reviews are, or will be, reported elsewhere. For the purposes of this paper, we are concerned with reporting on the *process* of conducting systematic reviews in software engineering and with assessing its practicality. While not all of the reviews described here have been completed in full, all of them have progressed far enough to provide substantial experience of the first two phases of the review

[1] Although where meta-analysis is possible, there is a substantial body of methods to call upon (e.g. see (Egger et al., 1997)).

process.

3. Service-based systems: a systematic review of issues (R1)

This review was conducted with the aim of *identifying research issues*, and hence differs from the 'traditional' hypothesis-centred view of a research question—although it is probably quite typical of a whole class of research questions that may arise in a technology-based subject such as software engineering. The topic of software service models is itself one that has not yet led to any significant empirical studies and is one that is still being mapped out. So here the objective for the review was to identify both the issues being raised and also the extent to which the issues raised in different papers were consistent. The review is essentially complete although the final report is yet to be externally validated.

3.1. Context

In 2000, Brereton and Budgen published a classification of the issues that needed to be addressed for component-based systems (CBS) to achieve their full potential (Brereton and Budgen, 2000). Since that time there has been a considerable growth in research and practice in the related field of service-based systems (SBS), leading to the view (of the authors and of others) that a similar classification of issues for SBS would be useful.

A team of software engineering researchers from King's

College London and the Universities of Durham, Keele and Manchester carried out the review.

3.2. Objectives

The objectives of the review were:

- to identify the main issues that need to be addressed if SBS are to be successfully implemented and widely adopted;
- to identify the solutions that have been proposed to address the issues raised;
- to identify the research methods used to investigate proposed solutions;
- to provide a framework for positioning of new research activities;
- to identify the gaps in current research.

3.3. Methods

Six archival journals dating from 2000 were used as data sources and papers were selected for review if they had a key term (or synonym) in the title, abstract or keywords list. All of the selected papers were included in the review (i.e., no further quality criteria were applied, on the basis that these were journal papers that would already have been thoroughly reviewed). During the data extraction process (stage 7) reviewers were aided by the use of a standardised data recording form and guidance notes.

3.4. Results

The main issues that were identified as needing to be addressed if SBS are to be successfully implemented and widely adopted were *change*, *selection* and *co-ordination* and the solutions presented were focused mainly on technologies. The research methods used were primarily those of *concept implementation* and *conceptual analysis.* A framework based on the previously-developed CBS framework was proposed and the gaps identified included topics relating to business and people-oriented issues.

3.5. Conclusions

The objectives of the systematic review were successfully achieved and experiences of applying the methodology to a software engineering topic are also reported (Brereton et al., 2005).

4. Systematic literature review of the technology acceptance model and its predictive capabilities (R2)

This review is more ‘conventional’ in nature than R1, in that there is a clear set of research questions that can only be answered by appealing to empirical results. In addition, although at the time of writing the review is incomplete, a preliminary pilot review has already been undertaken by one of our students, raising enough issues to motivate this fuller study.

The review is at stage 7 (see Fig. 1), that is, the protocol has been developed and reviewed, primary studies have been selected and assessed and the required data is being extracted. As recommended in all medical guidelines, the review protocol has been intensively piloted. This means that although the study is still formally in the middle phase, most of the elements required to conduct a review have already been explored.

4.1. Context

The study is being conducted by the authors of this paper as part of the Evidence Based Software Engineering project and others (<http://evidence.cs.keele.ac.uk>) funded by EPSRC.[2] The technology acceptance model (TAM) was proposed by Davis (1989) and Davis et al. (1989) as an instrument to predict the likelihood of a new technology being adopted within a group or organisation, and there have been a number of subsequent variations of the original TAM model, including TAM2 (Venkatesh and Davis, 2000). Whenever the TAM is validated for internal consistency, it scores very highly against whatever measure is used (Davis, 1989; Szajna, 1994; van der Heijden, 2003). As a consequence, the results of applying the TAM are often accepted as predictors of usage and adoption well beyond the point of validation. However, the actual usage of the technology, rather than intention to use, is rarely monitored. This study aims to investigate whether the TAM is a reliable predictor of *actual* use, rather than *intention* to use, based upon employing both subjective (self-reported) and objective (computer logs) measures.

4.2. Objectives

The objectives of the review are:

- to identify the extent to which the TAM and its revisions are capable of providing reliable predictions of the *actual* usage of a technology as opposed to the *intention to use*;
- to determine if the type of usage measure (subjective or objective) affects the accuracy of TAM predictions;
- to identify if the version of the TAM, or the technology being evaluated, affects the accuracy of the predictions.

4.3. Methods

The search strategy for the review is primarily directed towards finding published papers (archival journals, conference proceedings, or technical reports) from the contents of five electronic databases, although each identified primary source has been checked for other relevant references. A number of search strings were constructed using relevant terms based on the research questions and the search was

[2] Engineering and Physical Sciences Research Council, UK.

restricted to papers published between 1989 and the present day. The year 1989 was chosen as the baseline as this was when the first paper to describe the TAM was published (Davis, 1989).

The selection of primary sources was initially based on a review of title, keywords, and abstracts although this was extended to include the conclusions section in the cases where the title, keywords and abstract provided insufficient information. All selected studies were then reviewed against a detailed set of inclusion criteria designed to identify whether or not a study can help to answer the specified research questions. In particular, we aimed to identify whether a study describes an empirical investigation in which the TAM was used, whether the actual usage variable is measured, whether the study includes particular TAM variables and whether it reports the relationship to actual usage. The data extraction process is being conducted using a standardised electronic data recording form.

4.4. Results and conclusions

Initially, 208 papers were identified as potentially relevant to the research questions; however, after applying the inclusion/exclusion criteria, 59 papers remained in the set of relevant papers. The number of false positives in the initial set (papers that may have been relevant but on detailed investigation turned out not to be so) was disappointingly high.

5. Systematic literature review of guidelines for conducting systematic literature reviews (R3)

This study is more of a 'meta-review', although still using the concepts of systematic review. It is at the planning phase and we are in the process of developing the protocol (stage 2) which is currently being piloted.

5.1. Context

The guidelines used for conducting systematic literature reviews vary across the different domains that commonly use such an approach. The purpose of this study, which is also part of the EBSE project, is to review existing guidelines for systematic literature reviews across a range of domains, identifying the similarities and differences between them, in an attempt to propose appropriate guidelines for performing systematic literature reviews in the domain of software engineering.

5.2. Objectives

The objectives of the review are:

- to search and explore other similar domains where guidelines may be found, and to determine whether any guidelines are more suitable for software engineering than medical guidelines;

- to find quality criteria that are appropriate for assessing the kinds of studies carried out in software engineering.

5.3. Methods

For this review, a preliminary investigation is being carried out to identify research domains similar to the domain of software engineering. A research domain will be regarded as similar to the domain of software engineering if similar forms of empirical studies are conducted in that domain. This preliminary investigation is based on meetings/interviews with a number of experts within a range of disciplines beginning with those located at Keele and Durham Universities and then following up pointers from these (Budgen et al., 2006).

According to the outcomes from this, publications specific to each domain considered to be sufficiently 'close' will be targeted. This search will include textbooks along with databases that specialise in research publications in the chosen domains. The gathered literature will then be qualified against predefined quality criteria. These criteria are designed to assess the guidelines in terms of who developed them and how they were validated, in order to decide if the proposed guidelines should be included in the review. A pilot study is being performed before the final review is conducted.

5.4. Results and conclusions

To date, the domains of education, criminology and nursing and midwifery have been identified as having empirical practices most closely aligned with software engineering.

6. Lessons learned

In this section, we review our experiences with the *process* of conducting systematic literature reviews. We have structured our discussion around the 10-stage model that was introduced in §2. Although two of our studies have still to be completed, we have undertaken extensive piloting of both research protocols. This has involved piloting most of the elements of the review except the final synthesis of, and documentation of, the systematic literature review results, and hence our lessons draw upon all three studies. For ease of reading we report our results against the most relevant stage, although the lessons learned from studies R2 and R3 are partly based on our experiences from piloting some of the stages during protocol development.

6.1. Stage 1: Specify research questions

Specifying the research question(s) is the most critical element of a systematic review. The research questions are used to construct search strings for automated searches, they determine the data that needs to be extracted from each primary study and constrain the aggregation process. The research questions are the part of the protocol that should not be changed after the protocol is accepted.

In all three reviews, the questions were refined as a result of piloting the protocol. This was particularly noticeable for R3 which started with two questions:

Question 1: What are regarded as the elements of a systematic literature review across a sample of domains?
Question 2: What are the guidelines that can be adopted in software engineering?

After reviewing the protocol and investigating some of the known information sources, we expanded these into a set of more detailed questions:

Question 1: What are regarded as the essential elements of a systematic literature review in the domains studied?
Question 2: What are regarded as the essential elements of systematic literature review in domains that are similar to software engineering?
Question 3: What are the criteria for assessing the quality

of primary studies?

Question 4: How are searches organised? This question leads to two separate sub-questions:

Q4.1. What strategies are used to identify primary studies that are related to the research topic?

Q4.2. What criteria are used to assess the completeness of the search strategy?

Question 5: What procedures are used for combining evidence from different primary studies other than formal meta-analysis (for example grading systems for study types)?

One way of addressing the difficulty of scoping research questions is through the introduction of a systematic pre-review mapping study of the research topic. This approach is taken at the EPPI-Centre (The Evidence for Policy and Practice Information and Co-ordinating Centre[3]) which undertakes and publishes reviews relating to social intervention. A systematic map is used to describe the kinds of research activity that have been undertaken relating to a research question. The map describes factors such as the distribution of studies, the range of ages covered by the studies, and the number of studies that evaluate specific policies and practices. Mapping is similar to data extraction in that it is done for each study by entering details into a form. The difference is that mapping is done as quickly as possible, for a large number of initial studies, and describes the studies rather than extracting specific details. It is different to synthesis as no interpretations of the descriptions are made in a map. However, it does provide a context for the later synthesis. The aim of such a map is to provide a context for the review, to help interpretation of the synthesis, to narrow down the synthesis question and inclusion/exclusion criteria and to reduce the number of potential studies.

6.1.1. Lesson learned

L1: Expect to revise your questions during protocol development, as your understanding of the problem increases.

L2: A pre-review mapping study may help in scoping research questions.

6.2. Stage 2: Develop review protocol

As outlined earlier, the protocol gives details of the plan for the review, including, for example, specifying the process to be followed and the conditions to apply when selecting primary studies, the quality metrics to be applied to primary studies and the allocation of reviewers to particular activities. In study R1, the distributed nature of the team led to some problems during the development of the review protocol. In the end, the protocol was developed by only two of the reviewers which meant that these reviewers became more familiar than the others with the systematic literature review process, and in particular, were

[3] Available from: <http://eppi.ioe.ac.uk/eppiweb/home.aspx>.

more familiar with the role of the protocol itself. This caused problems later on when one member of the team did not follow the search process specified in the protocol and other members did not fully understand the data extraction requirements.

In study R2, piloting the protocol proved essential in revealing data extraction and aggregation problems. In particular:

- The team members taking the role of data extractors were not as familiar with statistical terms as the team member who defined the data extraction form. Piloting the extraction process revealed several misunderstandings about the nature of correlation and regression constants.
- The pilot revealed that primary studies often reported multiple tests for a single study and the data extraction process needed to be refined to ensure that the results from each study were not double counted.
- The primary studies used different means of reporting their results and the data extraction and aggregation process needed to be amended to cater for these differences.
- The primary studies reported subsets of the required information which meant we needed to define a process for managing missing values.

In R3, the piloting exercise identified the need for information to be collected at the domain level as well as the primary study level. The domain level information was necessary to provide an assessment of how similar the domain is to software engineering. The piloting also revealed that the R3 review is unlike other software engineering reviews where researchers can search for primary studies in software engineering e-databases, journals, or conference proceedings. Instead, it is based on literature published in domains that are largely outside the authors' area of expertise, which means that performing a comprehensive search for guidelines has been problematical, due to a lack of domain knowledge. This led us to revise the scope of our study and to extend the methodology of the study beyond a systematic literature review to include interviews with domain experts.

6.2.1. Lessons learned

L3: All systematic review team members need to take an active part in developing the review protocol.

L4: Piloting the research protocol is essential. It will find mistakes in your data collection and aggregation procedures. It may also indicate that you need to change the methodology you intend to use to address the research questions.

6.3. Stage 3: Validate review protocol

The protocol is a critical element of a review and researchers need to specify and carry out procedures for its validation. For R1, validation of the review protocol included the execution of a pilot run of the data extraction process (and forms) on four papers using two of the reviewers. As a result of this, the forms were modified slightly and some notes for conducting data extraction were prepared. A particularly useful aspect of the notes was the inclusion of definitions for those research methods that were likely to be used in the selected papers, which greatly aided uniformity of coding. The protocol was also reviewed informally by Prof. Barbara Kitchenham but the results of this were not adequately (or formally) considered by the review team.

For R2, validation was performed through a formal review process using two external reviewers who were asked to complete a questionnaire addressing the completeness and quality of the review items. Their observations led to some revisions to the protocol.

6.3.1. Lessons learned

L5: Data extraction is assisted by having data definitions and data extraction guidelines from the protocol recorded in a separate short document.

L6: There needs to be an agreed validation process separate from the protocol piloting activity. Ideally, external reviewers should undertake this validation process.

6.4. Stage 4: Identify relevant research

Once the protocol is finalised and the systematic review enters the execution phase, the researchers must execute the search strategy defined in the protocol. Studies R1 and R2 adopted very different search strategies.

For R1, during the planning phase, it became evident that hundreds of papers had been published in recent conferences and workshops on topics relating to SBS. The options of sampling from these papers or of applying some quality criteria were debated but finally it was decided to extract primary sources solely from archival journals, largely on the pragmatic basis that we were unable to determine any clear quality criteria for deciding which conferences and workshops to include. This meant that the search strategy was restricted to specifying the archival journals to be used and the years which would be searched. The start year was identified from the first publication that referenced the term service-based software. In retrospect, the relative immaturity of the technology meant that there were only a few journal papers available, and, clearly, identifying relevant research during the period when a technology is developing rapidly can be somewhat problematical. (This is probably one of the aspects of software engineering that has no equivalent in the other domains we have examined.)

In contrast, study R2 needed to obtain as complete a set of papers as possible because it is looking for the results of a rarely-used evaluation process (i.e., validation of the TAM against actual use rather than validation against intention to use). For that reason a restricted search was not possible. Construction of the search strategy was based on the approach suggested by medical standards:

- The research question was decomposed into individual elements related to the technology (technology acceptance model), the study type (evaluation) and the response measure (correlation with actual effort) to obtain the main search terms.
- Key words obtained from known primary studies were assessed for other main terms.
- Synonyms for the main terms were identified.
- Search strings were constructed using Boolean "AND" to join the main terms and "OR" to include synonyms.

In addition, the R2 search protocol specified the starting point for searches as 1989 when the TAM was first introduced, identified a previous systematic review of TAM evaluation, and noted that the reference list of all primary studies would be searched to look for other candidate primary studies.

Medical guidelines recommend searching several electronic sources. We identified the following indexing services:

- IEEExplore;
- ACM Digital library;

- Google scholar (<scholar.google.com>);
- Citeseer library (<citeseer.ist.psu.edu>);
- Keele University's electronic library (<opac.keele.ac.uk>);

(('technology acceptance model' <and> (usage <or> 'actual usage') <and> **(assessment <or> evaluation) <and> empirical** <in> (metadata, pdfdata))) <and> (pyr >= 1989 <and> pyr <= 2005)

Fig. 2a. Boolean search expression a.

(('technology acceptance model' <and> (usage <or> 'actual usage') <and> **empirical <and> (assessment <or> evaluation)** <in> (metadata, pdfdata))) <and> (pyr >= 1989 <and> pyr <= 2005)

Fig. 2b. Boolean search expression b.

- Inspec (<www.iee.org/publish/inspec/>);
- ScienceDirect (<www.sciencedirect.com>);
- EI Compendex (<www.engineeringvillage2.org/controller/servlet/athensservice>).

However, piloting the searches immediately hit problems. The search terms proposed in the protocol were applied in two key electronic databases: IEEExplore[4] and the ACM Portal.[5] We found that the search engines of the two databases are organised around completely different models. It is therefore impossible to make direct use of the same set of search terms for both engines. For instance, the ACM Portal does not support complex logical combination, although it is supported by IEEExplore.

Further we found that, in some search engines, the evaluation of a Boolean search string is dependent upon the order of the terms, independent of brackets. For example, the two Boolean expressions illustrated in Figs. 2a and 2b give the same total number of results in IEEExplore but the order in which the results appear is significantly different.

This is particularly important as the results of the IEEExplore search engine are sorted in order of *relevance*. Other search engines also display similar inconsistencies. For example, Citeseer (http://citeseer.ist.psu.edu/) treats multiple words as a Boolean term and looks for instances of all the words together. The use of the Boolean expression 'and' in this engine looks for all of the words but not necessarily together. As an example, the Boolean search 'technology acceptance model' (without quotes) looks for instances of "technology acceptance model" (all three words together) and finds no documents. The amended search term using Boolean 'and' (i.e. 'technology and acceptance and model') returns 210 documents. However, as none of the returned results includes all of the terms together the relevance of the results is very low. The engine also offers the ability to search the database of the site using Google. A Google search of Citeseer using the search term "technology acceptance model" (with quotes and thus looking for the exact phrase) finds around forty documents. For this reason we decided to use different sets of search terms for each database, with each of the search terms being derived from the terms originally proposed in the protocol.

6.4.1. Lessons learned

L7: There are alternative search strategies that enable you to achieve different sort of search completion criteria. You must select and justify a search strategy that is appropriate for your research question.

L8: We need to search many different electronic sources; no single source finds all of the primary studies.

L9: Current software engineering search engines are not designed to support systematic literature reviews. Unlike medical researchers, software engineering researchers need to perform resource-dependent searches.

6.5. Stage 5: Select primary studies

The selection of primary studies is usually a two-stage process:

- The title and abstract of studies identified by the initial searches are reviewed (preferably by at least two

[4] <http://ieeexplore.ieee.org>.

[5] <http://portal.acm.org>.

researchers) and irrelevant papers are rejected. This review should err on the side of caution, if researchers cannot agree, the paper should be included.

- Full copies of the papers not previously rejected are obtained. These papers are reviewed by two or more researchers against the inclusion/exclusion criteria defined in the protocol to obtain a final list of primary studies. The two researchers should resolve any disagreements (if necessary with the help of an independent arbitrator).

This process may iterate if the search protocol specifies that the reference list of primary studies must be reviewed to look for other possible candidate primary studies.

For R1, as the search was performed manually, and the set of papers was relatively small, this step presented no real difficulties.

For R2 there was a problem deciding whether papers returned by the search engine were relevant to the study. The protocol specified that the initial selection of primary sources would be purely based upon a review of title, keywords, and abstract. However, in practice it was very difficult to determine whether a paper was a potential candidate for the TAM study by using only titles, abstracts, and keywords. In medicine, unlike in software engineering and computer science, abstracts are structured and contain information that is, along with titles and keywords, usually enough to determine the content of a paper. In contrast, in software engineering and computer science, keywords are not consistent between different major journals and between organisations such as ACM and IEEE. In the majority of cases it was also necessary to read the conclusion before making a decision to include or exclude a particular primary study. In fact, when decisions were made solely on the abstract of the papers, several studies were excluded that were known to be important prior to the pilot. The result was that the protocol was amended to state that conclusions will be included in the initial process of identifying primary studies.

6.5.1. Lesson learned

L10: The standard of IT and software engineering abstracts is too poor to rely on when selecting primary studies. You should also review the conclusions.

6.6. Stage 6: Assess study quality

It is important to assess the quality of primary studies in order to support the inclusion/exclusion process and the allocation of weighting to specific studies during the data synthesis stage. There are no universal definitions of study quality but it has been suggested that quality relates to the extent to which bias is minimised and external and internal validation are maximised (Khan et al., 2001).

In R1, the research team did not assess the quality of individual studies based on the rationale that by restricting their review to papers published in archival journals; they had already ensured that the papers should be of acceptable quality.

In R2, the review team established quality criteria based on the completeness of the data, with 'problem' papers being rated as questionable or incomplete. Questionable studies are studies that make claims about the relationship they found between the TAM and actual use but do not provide any data or statistical tests to support the claims. Incomplete studies report some, but not all, of the required information. For example a paper that notes a relationship between the major TAM elements (i.e., perceived usefulness and perceived ease of use) and actual use but does not say whether the relationship holds for each element separately would be graded as incomplete. If information is ambiguous or missing, it is important to contact the authors and ask them for the additional information. If the additional information is not forthcoming, the quality rating will be used in sensitivity analysis e.g. assessing whether the results of the incomplete or ambiguous studies affect the overall results.

For R3, the review team identified quality criteria to assess the appropriateness and authenticity of any guidelines identified. It defined a number of questions to quantify the source. Examples of these questions are: “Did a single individual or a group develop the guidelines?”, and “Did the group include individuals from different countries?” The quality rating developed for the guidelines covered a result range from 0, when the guidelines were prepared by a single person with no formal validation, to 5, for guidelines produced by a multinational group with a formal validation by independent reviewers. This rating should help in determining the reliability of the source of the guidelines.

6.6.1. Lessons learned

L11: All the medical standards emphasise that it is necessary to assess the quality of primary studies. However, it depends on the type of systematic literature review you are undertaking.

L12: It is important to be sure how the quality assessment will be used in the subsequent data aggregation and analysis.

6.7. Stage 7: Extract required data

The objective of this stage is to use data extraction forms to accurately record the information researchers obtain from the primary studies. To reduce the opportunity for bias, the data extraction forms should be defined and piloted when the study protocol is defined.

For R1, the research team found that individuals who also performed the pilot review produced mostly complete, good quality data. Others were sometimes unsure about what to do and sometimes did not follow the protocol, although they did not ask for clarification at any time. Possible reasons for this are:

- the instructions may have been unclear;
- some weeks had passed between the instructions being sent to reviewers and the reviews being carried out;
- the rationale for being systematic and the overall process were documented in the form of a draft paper rather than as a formal 'protocol'.

For R2, when the data extraction process was piloted, the two reviewers used two different approaches. The first approach required each reviewer to extract data from a paper individually and then compare the two data extraction forms and discuss any disagreements. This is the method recommended by the medical guidelines. The second approach was to have one reviewer extracting the data and the second reviewer acting as a checker. We found that there was no *major* difference between the two approaches in terms of effort and time. However, having one reviewer acting as a checker was slightly quicker and so may be worth considering if there are a large number of papers to review.

6.7.1. Lessons learned

L13: Having one reader act as data extractor and one act as data checker may be helpful when there are a large number of papers to review.

L14: Review team members must make sure they understand the protocol and the data extraction process.

6.8. Stage 8: Synthesise data

Once the data has been extracted, it must be synthesised in a manner suitable for answering the review questions. In medicine the majority of systematic literature reviews aim at formal meta-analysis of quantitative data. Two of the reviews, R1 and R3, are based on collection of qualitative data. Aggregation of qualitative data is based on simple tabular formats.

Therefore, only one of the three systematic reviews discussed in this paper (R2) considered the use of meta-analysis. However, attempting this for R2 immediately revealed the problem that regression coefficients cannot be aggregated in the same way that correlation coefficients can (Lipsey and Wilson, 2001). The pilot exercise revealed that some researchers reported correlations, while others reported the multiple correlation coefficient or the regression coefficients. The review team decided it could not base aggregation solely on meta-analysis but would also need to summarise data in simple tabular form.

When data is tabulated (as opposed to subjected to formal meta-analysis), it may not be clear whether the research questions of the review have been answered. The researchers may need to explain how the summarised data addresses the research questions.

6.8.1. Lessons learned

L15: Software engineering systematic reviews are likely to be qualitative in nature.

L16: Even when collecting quantitative information it may not be possible to perform meta-analysis of software engineering studies because the reporting protocols vary so much from study to study.

L17: Tabulating the data is a useful means of aggregation but it is necessary to explain how the aggregated data actually answers the research questions.

6.9. Stage 9: Write review report

Once the systematic review is completed and the questions answered, the systematic review must be documented. The medical guidelines suggest that the protocol can be used as the basis for the final report.

Only R1 has reached phase 3 and the review documentation is taking the form of a paper for submission to an academic software engineering journal. The team faced two challenges at this stage. One related to the limited quantity and quality of record keeping. Decisions were distributed across many email exchanges and the notes that were taken at a meeting of four of the five reviewers. Although we were able to construct the necessary process information it would have been better if we had kept a more detailed and formal project log. The other challenge that arose was the restriction in length imposed (or at least recommended) by some software engineering journals (and even more restricted by many software engineering conferences). Fully documenting a review within the maximum word count of many journals is a challenge and it would perhaps be advisable to make some of the details of a review available through a trusted website. This approach has been taken by the EPPI-Centre which publishes reviews at three levels of detail: brief introductions, summaries and full reviews.

6.9.1. Lessons learned

L18: Review teams need to keep a detailed record of decisions made throughout the review process.

L19: The software engineering community needs to establish mechanisms for publishing systematic literature reviews which may result in papers that are longer than those traditionally accepted by many software engineering outlets or that have appendices stored in electronic repositories.

6.10. Stage 10: Validate report

Once the systematic review is documented, the medical guidelines suggest that the document should be independently reviewed. We have published an internal technical report of R1 (Brereton et al., 2005) and are extending this for external validation (through submission to a peer-reviewed journal).

7. Discussion

In this section, we look back at the aims of the study and discuss how the lessons we have learned help in addressing them. The aims are:

- to identify aspects of the process that transfer 'as is' to software engineering;
- to identify aspects that need to be adapted to the particular characteristics of the software engineering domain;
- to identify areas where improvements to current software engineering infrastructure and practices are needed in order to gain greatest advantage from the process.

The discussion is organised around the stages of the review process and is summarised in Table 1.

Stage 1 (specify research questions) of the process essentially carries over 'as is'. Medical guidelines do not preclude the revision of the research questions during protocol development or the introduction of a pre-review scoping

Table 1
Adopting systematic review within the software engineering domain

Stage	Transfer ‘as is’	Adapt to SE domain	Adapt SE practices
1. Specify research questions	Activity can be applied within the SE domain		
2. Develop review protocol	Activity can be applied within the SE domain		
3. Validate review protocol	Activity can be applied within the SE domain		
4./5. Identify relevant research and Select primary studies			
6. Assess study quality			
7. Extract required data	Activity can be applied within the SE domain		

8. Synthesis data

9./10. Write review report
and Validate report

We recommend that all empirical work in SE should include the development of a study protocol

This activity could be carried out through the usual peer review of papers submitted for publication or a formal evaluation by experts could be adopted

Current SE digital libraries do not provide good support for these activities.Also, the standard of the abstracts of SE publications is poor

Quality measures appropriate to the types of empirical studies included in a review need to be developed

Methods of aggregation should be appropriate to the types of empirical studies included in a review

Mechanisms for publishing papers that are longer than those traditionally accepted by SE outlets are needed

exercise to help identify useful, answerable questions. However, we should note that the lack of published empirical studies in software engineering, compared to clinical medicine, and the nature of those studies, which use a wide range of empirical procedures (compared to the extensive use of RCTs in clinical studies) are likely to contribute to a greater degree of difficulty in establishing appropriate research questions.

All of the studies benefited from the production of a protocol. Studies R2 and R3 piloted the protocol extensively and identified a number of practical problems as a result. We found the creation and piloting of a study protocol of great benefit and would now recommend that all empirical research work should start with the development of a study protocol. However, it is critical that all members of a research team take an active part in the construction and piloting of the protocol or some of the benefits may be lost. Staples and Niazi also highlight the value of piloting the protocol but raise an additional concern about knowing when to terminate the piloting process and move on to the execution phase (Staples and Niazi, 2006). Reeves et al. stress the importance of initial tasks such as clarification of the question(s) the review is trying to answer, definition of terms and inclusion/exclusion criteria (Tip 2 in (Reeves et al., 2002)) and add word of caution that "it may be difficult to invite new members into your review group after this stage as any new members are likely to find it hard to 'catch up' and understand the norms and meanings jointly held by the group". This last point supports our view that all systematic review team members need to take an active part in developing the review protocol (L3).

We believe that a formal validation process for the review protocol is beneficial, and intend to investigate further the value of independent evaluation of both review protocols and final research reports in studies R2 and R3.

Our attempts to identify relevant research and select primary studies have highlighted a number of domain specific problems. In particular, the major on-line indexing databases do not provide adequate support for systematic searching because they use different underlying models and do not all support complex Boolean searches. This need to design different search strings for different databases has been raised by others (see for example Tip 6 in (Reeves et al., 2002) and also (Kitchenham et al., 2006) and (Staples and Niazi, 2006)) and clearly is a significant problem to be addressed. Furthermore, keywords are not standardised and abstracts are of too poor quality for researchers to determine whether papers are relevant to specific research questions. We therefore need to improve the quality of the papers we ourselves write. Abstracts are intended to be a complete standalone summary of a paper and it may be that we need to consider advocating the use of structured abstracts as used in some medical journals.

We have identified and applied a range of quality criteria to the selected primary studies for R2 and R3, however there is a need for further work in this area in particular to establish which criteria are appropriate for specific types of review, specific types of empirical study and specific types of data analysis.

The extraction of required data using standardised data recording forms appears to work well once reviewers have gained some experience of doing this and appreciate that it is a time consuming and skilled activity.

As discussed earlier, empirical approaches used in software engineering include a prevalence of qualitative methods, and the variation in reporting protocols even where quantitative data is available suggests that formal meta-analysis may not be possible. The reviews also highlight the need to explain how summarised data addresses the research questions.

For the final 2 stages (write review report and validate review report), there is a need for software engineering journals and conferences to accommodate longer papers or enable access to additional information through a trusted web site. Some journals have taken this on board. For example, the Journal of Information and Technology[6] provides information on its web site about procedures for undertaking reviews and "encourage[s] potential authors to identify areas for systematic reviews, write papers and submit them" (Dyer et al., 2005). The problems associated with publishing empirical studies, especially at prestigious conferences such as the International Conference of Software Engineering (ICSE) were also highlighted by Basili and Elbaum in an invited talk at ICSE 2006 (Basili and Elbaum, 2006).

8. Conclusions

Our experiences of attempting systematic literature reviews have confirmed that the basic steps in the systematic review process appear as relevant to software engineering as they do to medicine. However, some modifications to our normal practices could significantly improve its value as a research tool and as a source of evidence for practitioners. We particularly note that the reporting of empirical studies in software engineering has a number of shortcomings and the lack of conformity, especially in terms of searching facilities, across commonly used digital libraries is also a hindrance to systematic literature reviewers in the discipline.

Acknowledgements

We would like to acknowledge the support of EPSRC for funding the EBSE project (EP/C51839X/1) and thank Stuart Charters and Stephen Linkman for their contributions to general discussions. We also thank the Service-based Systems co-reviewers: Nicolas Gold, King's College London; Keith Bennett, University of Durham and Nikolay Mehandjiev, The University of Manchester. Michael Rusca performed an initial literature review of the TAM as a Keele University undergraduate project and we used his work as a starting point for our study on this topic.

[6] Available from: <http://www.elsevier.com/wps/find/journaldescription.cws_home/525444/description#description>.

Tore Dybå and Tracy Hall acted as reviewers for the protocol used in R2 (study of the TAM).

References

Basili, V., Elbaum, S., 2006. Empirically Driven SE Research: State of the Art and Required Maturity, Invited Talk, ICSE 2006, Shanghai.

Brereton, O.P., Budgen, D., 2000. Component based systems a classification of issues. IEEE Computer 33 (11), 54–62.

Brereton, O.P., Gold, N.E., Budgen, D., Bennett, K.H., Mehandjiev, N.D., 2005. Service-based systems: a systematic literature review of issues. Computer Science Technical Report, Keele University (TR/05-01).

Budgen, D., Charters, S., Turner, M., Brereton, P., Kitchenham, B., Linkman, S., 2006. Investigating the applicability of the evidence-based paradigm to software engineering. In: Proceedings of Workshop on Interdisciplinary Software Engineering Research, ICSE 2006. ACM Press, Shanghai, pp. 7–13.

Davis, F.D., 1989. Perceived usefulness, perceived ease of use, and user acceptance of information technologies. MIS Quarterly 13 (3), 319–340.

Davis, F.D., Bagozzi, R., Warshaw, PR., 1989. User acceptance of computer technology: a comparison of two theoretical models. Management Science 35 (8), 982–1003.

Dybå, T., Kitchenham, B.A., Jørgensen, M., 2005. Evidence-based Software Engineering for Practitioners. IEEE Software 22 (1), 58–65.

Dybå, T., Kampenes, V.B., Sjøberg, D., 2006. A systematic review of statistical power in software engineering experiments. Journal of Information and Software Technology 48 (8), 745–755.

Dyer, M., Shepperd, M., Wholin, C., 2005. Systematic Reviews in

Evidence-based Software Technology and Software Engineering, Editorial. Information and Software Technology 47 (1), 1.

Egger, M., Smith, G.D., Phillips, A.N., 1997. Meta-analysis: Principles and procedures. British Medical Journal 315, 1533–1537.

Glass, R.L., Vessey, I., Ramesh, V., 2002. Research in software engineering: an analysis of the literature. Information and Software Technology 44, 491–506.

Glass, R.L., Ramesh, V., Vessey, I., 2004. An Analysis of Research in Computing Disciplines. Commun. ACM 47 (6), 89–94.

Jørgensen, M., Moløkken-Østvold, K., 2006. How large are software cost overruns? A review of the 1994 CHAOS report. Information and Software Technology 48, 297–301.

Khan, K.S., ter Riet, G., Glanville, J., Sowden, A.J., Kleijnen, J. (Eds)., 2001. Undertaking Systematic Review of Research on Effectiveness. CRD Report Number 4 (Second Edition), NHS Centre for Reviews and Dissemination, University of York, UK.

Kitchenham, B., 2004. In: Procedures for Undertaking Systematic Reviews. Joint Technical Report, Computer Science Department, Keele University (TR/SE-0401) and National ICT Australia Ltd (0400011T.1).

Kitchenham, B.A., Dybå, T., Jørgenson, M., 2004. Evidence-Based Software Engineering. In: Proceedings of ICSE 2004. IEEE Computer Society Press, pp. 273–281.

Kitchenham, B.A., Mendes, E., Travassos, G., 2006. A systematic review of cross- vs. within-company cost estimation studies. In: Proceedings of EASE 2006. British Informatics Society Ltd.

Lipsey, Mark W., Wilson, David B., 2001. Practical Meta-analysisApplied Social Science Methods Series, 49. Sage Publications Inc.

Reeves, S., Koppel, I., Barr, H., Freeth, D., Hammick, M., 2002. Twelve tips for undertaking a systematic review. Medical Teacher 24 (4), 358–363.

Sackett, D.L., Straus, S.E., Richardson, W.S., Rosenberg, W., Haynes, R.B., 2000. Evidence-Based Medicine How to Practice and Teach EBM, second ed. Churchill Livingstone, Edinburgh.

Staples, M., Niazi, M., 2006. In: Experiences Using Systematic

Review GuidelinesProceedings of EASE 2006. British Informatics Society Ltd.

Szajna, B., 1994. Software evaluation and choice: Predictive validation of Venkatesh, V., Davis, F.D., 2000. A theoretical extension of the

the technology acceptance instrument. MIS Quarterly 1 8 (3), 319–324. technology acceptance model: Four longitudinal field studies. Manvan der Heijden, H., 2003. Factors influencing the usage of websites: The agement Science 46 (2), 186–204.

case of a generic portal in The Netherlands. Information and Webster, J., Watson, R.T., 2002. Analyzing the past to prepare for the

Management 40 (6), 541–550.

future: Writing a literature review. MIS Quarterly 26 (2), 13–23.