A preliminary empirical investigation of the use of evidence based software engineering by under-graduate students

Austen Rainer, Tracy Hall and Nathan Baddoo
School of Computer Science
University of Hertfordshire
College Lane Campus
Hatfield
Hertfordshire AL10 9AB
U.K.
{a.w.rainer; t.hall; n.baddoo}@herts.ac.uk

Recently, Dybå, Jørgensen and Kitchenham have proposed a methodology, Evidence-Based Software Engineering (EBSE), that is intended to help researchers and practitioners evaluate software technologies in practice. We report the conduct of a preliminary empirical investigation of the reported use of Evidence-Based Software Engineering by 15 final-year under-graduate students. The investigation produced inconsistent results: our quantitative data suggests that students are making good use of some of the EBSE guidelines, but our qualitative evidence suggests that the students are not using the guidelines properly. Possible explanations for these results include: the superficial use of EBSE by the students; the limited 'power' of our research instruments to assess the use of EBSE guidelines; the difficulties of combining a required coursework with a piece of research; and the different expectations of what makes good evaluations between professional researchers and novice software practitioners.

Keywords: evidence-based software engineering, evaluation, methodology,

1. INTRODUCTION

In a recent series of articles, Dybå, Jørgensen and Kitchenham (e.g. Dybå et al., 2005; Kitchenham et al., 2004; Jørgensen et al., 2005; Jørgensen, 2005) have proposed a methodology that uses empirical evidence, practical experience and human values to evaluate technologies in software engineering. The methodology is called *Evidence-Based Software Engineering* (EBSE) and has been developed partly in response to the success of Evidence-Based Medicine. Broadly speaking, Dybå et al. argue in their series of articles that EBSE can support and improve practitioners' technology adoption decisions given their particular circumstances.

Dybå et al. (2005) have explicitly recognised the *lack* of empirical evidence from software engineering to support their very proposal:

> "You might have noticed that we've offered no evidence of EBSE's benefits. Although we have no examples of other practitioners using EBSE, … [we]… present examples of our own use of EBSE. On the basis of this experience, and other ongoing industrial and educational initiatives… we believe that evidence-based practice is possible and potentially useful for software practitioners." (Dybå et al., 2005, p. 64)

Our paper offers independent empirical evidence, practical experience and personal opinion on Evidence-Based Software Engineering. The paper is a product of a new module, *Empirical Evaluation in Software Engineering*, taught in the final year of the BSc(Hons) Computer Science degree programme at the University of Hertfordshire in the UK. As part of the assessment for that module, students used EBSE to evaluate a technology of their choice. Students' reported use of EBSE has been used to provide empirical evidence of practical experience for this paper.

This paper should help academics and practitioners in the following ways:

- To design and conduct empirical investigations of EBSE and EBSE-like methodologies.
- To improve the teaching and learning of both the concepts and practice of EBSE in particular, and empirical evaluations in general. This, in turn, should help to promote the value of evidence-based software practice.
- To further the development of EBSE and EBSE-like methodologies, for example through the addition or refinement of guidelines.

The remainder of the paper is organised as follows: section 2 presents a brief overview of the EBSE methodology. Section 3 discusses our study design. Section 4 presents the results from our study. Section 5 considers some explanations for, and implications of, these results. Finally, section 6 provides conclusions.

2. A BRIEF OVERVIEW OF EVIDENCE-BASED SOFTWARE ENGINEERING

Since the emergence of software engineering in the late 1960's, researchers and practitioners have recognised that most of software engineering research has little practical justification (Glass, 1994). This is manifested by the fact that very few of the results from software engineering research lead to an uptake of tools or change of practice in industry (Glass, 2005; Glass et al., 2002). There has been a recurring debate over how to make software engineering research more accessible and, indeed, practical to software practitioners. Much of this debate has advocated empirical work (Harrison et al., 1999).

EBSE "… aims to improve decision making related to software development and maintenance by integrating current best evidence from research with practical experience and human values." (Dybå et al., 2005, p. 58/59). As a methodology, it is intended to direct research to the needs of industry, help practitioners to make more rational decisions about technology adoption, enable better choice of development technologies and increase the acceptability of software intensive systems that interface with users. The methodology involves five steps:

1. Convert a relevant problem or information need into an answerable question
2. Search the literature for the best available evidence to answer the question
3. Critically appraise the evidence for its validity, impact and applicability
4. Integrate the appraised evidence with practical experience and the customer's values and circumstances to make decisions about practice
5. Evaluate performance and seek ways to improve it

Dybå et al. make the point that EBSE complements software process improvement (SPI), and that EBSE is particularly beneficial in an area where SPI is traditionally weak i.e. finding and appraising an appropriate technology, prior to inclusion of that technology in an SPI programme. A related distinction between EBSE and SPI is that EBSE concentrates on improving practitioners use of existing findings (notably through advocating systematic reviews) in order to make decisions. In contrast, SPI concentrates on the introduction, assessment and management of a new intervention. It follows that EBSE is not intended as an approach for generating new evidence. Rather, the focus of EBSE is on gathering and appraising existing evidence. As suggested above, EBSE places a heavy emphasis on the use of systematic reviews, which are a means to provide a fair evaluation of a phenomenon of interest by using a trustworthy, rigorous, and auditable methodology (Kitchenham, 2004).

EBSE can be located within a broader context of software engineering research that has:

1. Recognised the current limitations of software engineering research (e.g. Fenton et al., 1994; Glass, 1994; Zelkowitz et al., 2003).
2. Advocated the use of empirical evidence to evaluate technologies, particularly the use of quantitative evidence for these evaluations e.g. the SEI's Capability Maturity Model (CMM) (Paulk et al., 1993) requires Level 4 companies to incorporate metrics into their decision-making; and Kitchenham and others (e.g. Kitchenham et al., 1995; Kitchenham et al., 2002) have proposed guidelines for improving software engineering research.

3. Proposed and used other methodologies e.g. the Goal Question Metric (Basili and Weiss, 1984) framework, and DESMET (Kitchenham et al., 1997).
4. Recognised the disparity between evidence-based approaches to technology evaluation and adoption as advocated by professional researchers in the field, against the actual approaches to technology evaluation and adoption commonly undertaken by software practitioners (e.g. Rainer et al., 2005).
5. Recognised the inherent difficulties of software engineering (e.g. Brooks, 1987), and the problems that these cause when adopting practices from other well-established disciplines (Sommerville, 2001).

3. STUDY DESIGN

3.1 Overview to the empirical investigation

One of the major difficulties when evaluating a methodology like EBSE is being able to demonstrate the effects of the methodology on the outcomes of a task e.g. that the technology adoption decision made by a practitioner using EBSE was improved *as a result of* using the EBSE methodology. Given this problem, we have not evaluated the efficacy of Evidence-Based Software Engineering i.e. we have not evaluated the outcomes of EBSE. Rather, we investigate the *use* of EBSE, attempt to discover reasons for why EBSE was or was not used (or not used properly) and through this discovery to identify limitations and opportunities for improvement of the methodology. We focus on the use of EBSE, rather the outcomes, because EBSE must be *used* as a pre-requisite for it to have an affect.

As all subjects (students) in our investigation were required to use EBSE, there is no ‘external comparison’ with the non-use of EBSE, or with the use of an alternative to the EBSE methodology. We do attempt to perform an ‘internal comparison’ of the extent to which the subjects exploited the guidelines advocated by the EBSE methodology. We rely on subjects’ practical experiences of using EBSE. This includes situations where a student may not use the methodology as intended, but these un-intended uses provide opportunities to better understand the limits of the methodology (or the users of the methodology).

Overall, our main research question is:

RQ1: What are the practical experiences of under-graduate students using EBSE, as reported by those students in their coursework?

3.2 Procedure

Students were issued with the coursework in week 5 of the Autumn term. The coursework required students to use EBSE to evaluate a technology of their choice that would potentially address a problem that they were experiencing in software engineering. The timing of the coursework meant that students had received five lectures and tutorials on the *Empirical Evaluation in Software Engineering* module prior to receiving the coursework. These five lectures and tutorials included one lecture specifically on EBSE, and one accompanying tutorial. Students were also provided with copies of the *IEEE Software* article on EBSE, and references to the other articles in the series i.e. (Dybå et al., 2005; Kitchenham et al., 2004; Jørgensen et al., 2005; Jørgensen, 2005). (Full details of the lecture and tutorial programme, and the coursework specification are available from the first author.) At the time that the coursework was designed, an initial mark scheme (for the coursework) had also been designed (but not revealed to students of course).

Students were provided with almost six weeks to individually complete and submit the coursework (late submissions were not allowed). The series of lectures and tutorials continued during that period. During these six weeks, opportunities were provided to students at lectures, tutorials and via the University’s student intra-net to ask questions to clarify the coursework specification. Most questions asked by students over the six weeks concerned the EBSE question that had to be formulated. When answering these questions, the first author emphasised the importance of developing a well-formulated question according to EBSE guidelines. Also, the EBSE appraisal checklist (for step 3 of the EBSE) was used in a small number of lectures and tutorials to help appraise evidence being discussed in those lectures and tutorials. Consequently, one might expect students to make better use of those

EBSE guidelines concerning the question to ask, and the checklist, compared to other guidelines provided by EBSE.

In total 39 completed courseworks were submitted. Of the 39 courseworks, seven courseworks were then randomly selected and used, in conjunction with the initial mark scheme and a further review of the main EBSE paper, to develop an assessment checklist for this investigation. The assessment checklist was then tested on four further courseworks, and subsequently slightly revised. A summary of the final version of the checklist is provided in Appendix A.

Of the remaining 28 courseworks, 15 were randomly selected for this investigation[1]. Of these 15 courseworks, five were assessed by two separate authors as an informal test of inter-rater reliability.

Students were set a word limit on the main coursework submission, but were permitted to submit additional material as appendices. The additional material could be used to provide supporting evidence (e.g. examples of search results, examples of forum discussions etc.) for the main submission. Appendices were reviewed when assessing the courseworks for this investigation.

3.3 Limitations of design

One of the potential limitations to this study design is that we are assessing the report submitted by the student, rather than directly assessing the use of EBSE during the conduct of the coursework. Consequently, a student may have used EBSE extensively, but may not have effectively reported their use of EBSE in their coursework. Strictly speaking, this investigation assesses students' reports of their use of EBSE.

Students were also provided with a guideline on how much time they should spend on this piece of coursework. A typical guideline for the degree programme, and one used here, was 30 hours. Students were encouraged not to treat this guideline as a target or a limit, but as an approximation of the appropriate time to spend on the coursework.

4. ASSESSMENT OF STUDENTS' USE OF EBSE

4.1 Characteristics of students

TABLE 1 NUMBERS OF YEARS OF STUDY ON THE DEGREE PROGRAMME

Number of years at University (rounded up)	Count of students
5	3
4	2
3	9
2	1
Total	15

Table 1 indicates that the 15 students had been studying at the University for a number of different years. Students can be accepted directly onto the final year of the degree programme, or directly into the first or second years of the degree programme. Similarly, students can study part-time. The table indicates that all students who submitted courseworks are at least into their second year of study on the degree programme (although this could be their second year of part-time study in the final year of the degree programme). The table indicates that most students have progressed through the first and second years of the degree programme. Consequently, they will have studied the following subjects as these are compulsory modules for the degree programme:

[1] Unfortunately, time constraints restricted our investigation to only 15 courseworks, however we intend to report on a full analysis of all courseworks in a future publication.

- Programming, specifically object-oriented programming using Java or Eiffel
- Software systems development e.g. database development, software development lifecycles
- Formal systems
- Computer Systems and Networks

Many students will also have chosen to study multi-media systems development in the second year (e.g. using VisualBasic). Students are not offered any modules on research methods or evaluation prior to the final year. Some students will have completed a one year professional placement in industry prior to commencing their final year of the degree programme.

Overall, the students' experiences in previous years of the degree programme are potentially relevant when it comes to step 4 of EBSE i.e. integrating the published evidence with one's personal, practical experiences.

4.2 Summary of results

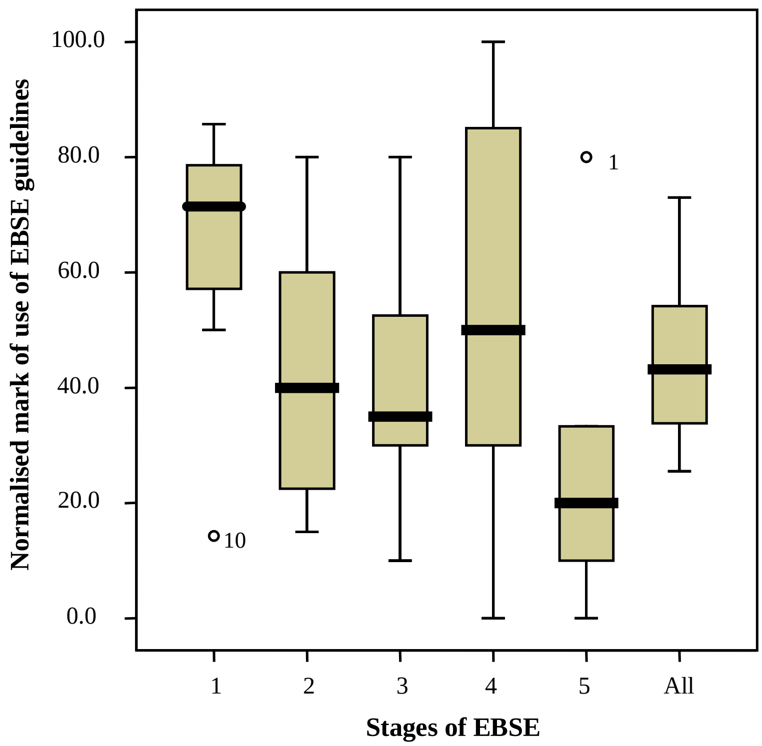

FIGURE 1 BOX-PLOTS OF NORMALISED MARKS OF THE USE OF EBSE GUIDELINES PER STAGE OF EBSE

Figure 1 presents box-plots of the normalised marks of the reported use that the students made of the EBSE guidelines. The normalised marks are derived from the application of the checklist given in the Appendix. The box-plots need to be interpreted carefully, particularly where comparisons are being made between stages of the EBSE. For example, in the first stage of EBSE, a practitioner is expected to develop a well-formulated question that might be characterised by five components: an intervention, a population of people who 'use' the intervention, an outcome, a situation in which the intervention is used, and a comparison. By contrast, the search for relevant evidence (step 2 of EBSE) is much harder to specify and hence assess. Therefore, students getting higher marks during step 1, compared to step 2, does not necessarily mean that their performance at step 1 is 'better' than their performance at step 2. We have tried to partially take account of this through normalising the marks for each stage.

The box-plots may also be useful for indicating the *range* of performance *within* each step, as well as the median performance relative to zero for that stage (but note that these box-plots are still subject to the limitations identified above). So, generally speaking, students seem to have:

- Provided well-formulated EBSE questions (step 1), as represented by the narrow range of values and also the high median value.
- Struggled to evaluate their use of EBSE (step 5), as indicated by the narrow range of values and also the *low* median value.

Interestingly, in both of these examples, there are outliers on the box-plots: a student who has performed particularly poorly at formulating an EBSE question, and a student who has performed particularly well at evaluating their use of EBSE.

4.3 EBSE Step 1: asking a question
In reviewing the students' stated EBSE questions, we observed the following:

- Students had problems constructing well-formulated EBSE questions. This seems to contradict the observation made in section 4.2 that generally students provide well-formulated questions. Closer inspection of the EBSE questions constructed by students suggests that while the components of a question may be present, the components are not well-defined and do not fit together in a coherent way. We will return to this point in section 5.
- Students have a preference for investigating web-related technologies, particularly databases and programming/scripting languages. A typical final-year project involves the development of a database with a web front-end.
- Most students have derived their EBSE question from a common personal experience i.e. their final-year project
- Some problems experienced by the students are actually about trying to first identify the alternatives prior to then evaluating them. As such, these kinds of problems are not easily expressed as an EBSE question. For example, Student 4 began with a poorly formulated question (i.e. "What is the best application/language for me to do my final year project in?") but recognised the limitations of their question in their evaluation, and considered that for EBSE they should have evaluated PHP in comparison to ASP.NET.

Student 15 had this to say about EBSE

> "I found step 1 [asking an EBSE question] particularly challenging. I had to change my question a couple of times in order to relate it to my software project. I had initially had (sic) a question comparing two scripting languages for the project. I chose to change it as I had no prior experience of ASP/PHP. As I had a reasonable understanding of databases, I felt I could produce a thorough and concise investigation in this subject area."

This suggests that some students are formulating their EBSE question partly on the basis of their existing experience.

4.4 EBSE Step 2: Searching for evidence
In reviewing the students' search for evidence, we observed the following:

- Students used a very limited number of search terms
- Students made very little use of material from scientific journals. Some students did search these databases but few (if any) found useful information. This may partly be due to the limited search terms they used.
- Students used limited criteria for identifying the best or better evidence e.g. it seems that more recent publications are considered to be more reliable.
- Many students used Google as a main search engine, but also AskJeeves, MSN etc.
- Many students used technology/practitioner web sites (e.g. www.developer.com)
- Students made little use of trade magazines.
- Students made little use of textbooks

- Students provided poor explanation in their reports of how their searches were conducted.
- Some students made use of official websites e.g. Microsoft.

4.5 EBSE Step 3: Critical appraisal of the evidence

In reviewing the students' appraisal of evidence, we observed the following:

- Students varied in their use of the EBSE checklist. A number of students simply did not use the checklist, a number of students used it poorly, whilst some students used it well. The varied use of the checklist is surprising as it had already been used in some tutorials and lectures.
- Students had little (reported) awareness of systematic reviews. Related to this, few students actually used systematic reviews.
- Separately from the EBSE checklist, students organised their appraisals by publication (i.e. evaluated each publication in sequence). Some students organised their appraisals in terms of technologies (e.g. presented papers on Oracle and then papers on Access) and some by issue (e.g. speed of database access, cost of database etc.)
- Some students critically appraise the technologies rather than the publications (evidence) on the technologies

For example, student 4 writes of an article that they reviewed "I found that the claims and facts he [Hull] expresses… to be valid", and the student adds "Through my own research into these claims and my own knowledge I have confirmed them". But nowhere in their coursework does the student present their own research, or state their own knowledge. Therefore, the student hasn't demonstrated their assessment of the validity of Hull's claims.

4.6 EBSE Step 4: Personal experience

In reviewing the students' integration of personal experience with the evidence, we observed the following:

- Some students talk about their opinions rather than their experiences.
- There are few specific examples of experience provided by students. This is surprising given the range of personal, practical experience that students will have gained through previous years of the degree programme (see section 4.1 above)
- Some students talk about their plans for their final year projects rather than their experiences

4.7 EBSE Step 5: Evaluating EBSE

In reviewing the students' evaluations of their use of EBSE, we observed the following:

- Students made varied use of EBSE guidelines for conducting evaluations i.e. some students simply did not use the Post-Mortem and After Action Reviews (specifically, the questions suggested in the main EBSE article).
- Students recognised that:
    - The time, effort and word limit for the coursework restricted the amount of time and effort that they could direct to their evaluations
    - They had problems asking a well-formulated question
    - EBSE was something that they benefited from (but, for some students, this isn't obvious from their report)

5. DISCUSSION

The assessment we report in section 4 of this paper presents a mixed impression. For example, the box-plots suggest that, generally speaking, students are performing consistently well when stating an EBSE question, however our qualitative evidence suggests that students' performance may be superficial. It seems that students can state an EBSE question comprising the appropriate components, but not all students *define* these components well, or organise them into a well-integrated question for subsequent EBSE-oriented evaluation. The inconsistencies between our quantitative evidence and our qualitative evidence suggests that:

1. The current checklist (see Appendix A) may primarily be capable of only supporting a more superficial assessment of the use of EBSE. This suggests that the checklist may require further development. In addition, the EBSE methodology may itself benefit from further development, in particular:
    - Additional information on how to implement the existing EBSE guidelines
    - Additional information on how a user of EBSE can self-assess their use of EBSE (which may be implemented as an extension of step 5 of the current EBSE methodology)

2. Using students to evaluate the use of EBSE raises the difficulty of assessing students according to the standards of a Bachelors degree, whilst also assessing students (as evaluators) according to the standards of professional research. Recall that the assessment checklist incorporated the mark scheme used to assess students as part of the degree module. This issue can be expressed as two questions:
    i. When conducting these kinds of empirical investigations, how do we take account of the fact that, generally speaking, under-graduate students will inevitably be less capable of research and evaluation, in comparison to qualified and experienced academics?
    ii. How do we design an empirical investigation that effectively incorporates both a degree assessment and a research assessment?

3. More generally, evaluations of the use of EBSE need to take account of the different expectations that professional researchers and software practitioners have with regards to the acceptable quality of evaluations. A number of the students reported, in their courseworks, that they believed they had performed a successful EBSE-based evaluation, an opinion that is not strongly supported by the evidence presented in this paper. Other papers (e.g. papers published at the ICSE 2005 *Realising Evidence Based Software Engineering* (REBSE) workshop) have recognised that software practitioners commonly adopt a very different set of standards for decision-making in software engineering compared to researchers. EBSE is presented as a methodology that both software practitioners and researchers can use. But each group has different expectations of quality evaluations and research. These issues can be expressed as a question (a variant of question i. above):
    i. When conducting these kinds of empirical investigations, how do we take account of the fact that, generally speaking, software practitioners will tend to have less time, effort, experience and expertise for conducting research and evaluation, in comparison to qualified and experienced academics?

One other issue warrants brief discussion. The evidence reported here suggests that students made little use of the scientific bibliographies recommended by the EBSE methodology. We think that this is partly because students lacked an understanding of how to properly search these bibliographies. But we also think that the kinds of problems students were tackling (e.g. “Is database engine X more suitable than database engine Y, for my circumstances?”) are not the kinds of problems researchers commonly investigate. We suspect that this discrepancy (between the question asked and the evidence available from the scientific bibliographies) would also often hold for software practitioners. For example, Microsoft has run an extended campaign (called *getthefacts*) that Microsoft claims provides evidence on the total cost of ownership of their Windows Server compared to Linux. This kind of issue is very important to software practitioners but unlikely to be addressed in scientific publications.

6. CONCLUSION

We have reported on an empirical investigation of students’ reported practical experiences of using the Evidence-Based Software Engineering (EBSE) methodology. Each student used EBSE to evaluate a technology of their choice that might help them with a software engineering problem that they were personally experiencing. We have provided some quantitative and qualitative evidence on students’ experiences of each of the five steps of the EBSE methodology. Our evidence provides mixed results, with inconsistencies between the qualitative and quantitative evidence. Students varied in their use of the EBSE guidelines, with some students making more use of the guidelines than others. Our qualitative evidence suggests that students tended to use EBSE superficially. This raised several issues, such as:

- The issue of assessing students according to the standards of a Bachelors degree, in contrast to assessing students (as evaluators) according to the standards of professional research.
- The different expectations and capabilities of professional researchers and software practitioners with regards to their software evaluations.
- The degree to which EBSE, as a methodology advocated for use by both practitioners and researchers, can take account of the different problems that professional researchers and software practitioners confront.

ACKNOWLEDGEMENTS

We are grateful to all of the 39 students who were involved in this investigation, and to Tore Dybå, Magne Jørgensen and Barbara Kitchenham for helpful comments during the initial planning of this investigation. We also thank the reviewers for their helpful comments, many of which we were unable to address due to space restrictions, but which we plan to include in future publications.

REFERENCES

Basili, V. R. and Weiss, D. M. (1984) A Methodology for Collecting Valid Software Engineering Data, *IEEE Transactions on Software Engineering,* **10**(6)**,** 728-737.

Brooks, F. P. (1987) No Silver Bullet: Essence and Accidents of Software Engineering, *IEEE Computer,* **20**(April)**,** 10-19.

Dybå, T., Kitchenham, B. A. and Jorgensen, M. (2005) Evidence-based software engineering for practitioners, *IEEE Software,* **22**(1)**,** 58-65.

Fenton, N., Pfleeger, S. L. and Glass, R. L. (1994) Science and substance: a challenge to software engineers, *IEEE Software,* **11**(4)**,** 86-95.

Glass, R. L. (1994) The software research crisis, *IEEE Software,* **11**(6)**,** 42-47.

Glass, R. L. (2005) A Sad SAC Story About The State Of The Practice, *IEEE Software,* **22**(4)**,** 119-120.

Glass, R. L., Vessey, I. and Ramesh, V. (2002) Research In Software Engineering: An analysis of the literatur, *Information and Software Technology,* **44**(8)**,** 491-506.

Harrison, W., Baddoo, N., Barry, E., Biffle, S., Parra, A., Winter, B. and Wuest, J. (1999) Directions and Methodologies for Empirical Software Engineering Research, *Empirical Software Engineering,* **4**(4)**,** 405-410.

Jørgensen, M. (2005) Teaching Evidence-Based Software Engineering to University Students, In *Evaluation and Assessment in Software Engineering (EASE)*, Keele University, 11 - 13 April, 2005.

Jørgensen, M., Dyba, T. and Kitchenham, B. (2005) Teaching Evidence-Based Software Engineering to University Students, In *11th IEEE International Software Metrics Symposium*, Como, Italy, 19-22 September 2005.

Kitchenham, B. (2004) Procedures for Performing Systematic Reviews, Keele University (UK) and National ICT Australia Ltd., July 2004, Keele University Technical Report TR/SE-0401 and NICTA Technical Report 0400011T.1.

Kitchenham, B., Linkman, S. and Law, D. (1997) DESMET: a methodology for evaluating software engineering methods and tools, *Computing & Control Engineering Journal,* **8**(3)**,** 120-126.

Kitchenham, B., Pfleeger, S. L., Pickard, L., Jones, P., Hoaglin, D. C. and Rosenberg, J. (2002) Preliminary guidelines for empirical research in software engineering, *IEEE Transactions on Software Engineering,* **28**(8)**,** 721-734.

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

Kitchenham, B. A., Dyba, T. and Jorgensen, M. (2004) Evidence-based software engineering, In *26th International Conference on Software Engineering (ICSE)*, Edinburgh, Scotland, UK.

Paulk, M. C., Curtis, B., Chrissis, M. B. and Weber, C. V. (1993) Capability Maturity Model, Version 1.1, *IEEE Software,* **10**(4)**,** 18-27.

Rainer, A., Jagielska, D. and Hall, T. (2005) Software engineering practice and evidence based software engineering research, In *International Conference on Software Engineering (ICSE) Workshop on Realising Evidence Based Software Engineering (REBSE)*, St Louis, Missouri, USA, 17 May 2005

Sommerville, I. (2001) *Software Engineering,* Addison-Wesley, Harlow, England.

Zelkowitz, M. V., Wallace, D. and Binkley, D. W. (2003) Experimental validation in software engineering, In *Lecture notes on empirical software engineering.* World Scientific, pp. 229-263.

APPENDIX A: MARK ALLOCATION FOR ASSESSING CONFORMANCE TO EBSE GUIDELINES

Section of EBSE		Marks
Step 1: Statement of the question to be investigated		
1.1 A question is clearly stated		1
1.2 Question clearly conforms to EBSE guidelines		
Intervention		1
Population		1
Outcome (effect)		1
Situation (context)		1
Comparison		1
1.3 Question is explicitly developed from personal experience		1
	Sub-total	7
Step 2: Identification of information resources etc.		
2.1 Search terms used are clearly stated		1
2.2 Search engines used are clearly stated		1
2.3 Clear list of identified resources are provided in coursework		1
2.4 Used the following search 'engines'		2
Online databases specified by EBSE e.g. IEEE Xplore		
Internet search engines		
Other databases e.g. Voyager		
2.5 Information resources exploited e.g. Scientific journals, books, trade magazines, web sites, internet forums, lecture materials, academic and practitioner personal comments		3
2.6 Clear description of how resources are selected or not selected		1
2.7 Clear description of how the best evidence was identified		1
	Sub-total	10
Step 3: Critical appraisal of the evidence		
3.1 EBSE checklist explicitly used		2
3.2 Explanations to support EBSE table are clearly provided		3
3.3 Awareness of systematic review		1
3.4 Explicit demonstration of attempt to conduct an unbiased appraisal		4
	Sub-total	10
Step 4: Critical comparison with personal experiences		
4.1 Discussion of personal experiences		1
4.2 Specific examples provided of previous experience (e.g. situations)		2
4.3 Critical comparison of personal experiences with evidence		2
	Sub-total	5
Step 5: Evaluation		
5.1 Use of either AAR or PA or both		10
Clear use of After Action Review (AAR)		
What was supposed to happen?		
What actually happened?		
Why were there differences?		
What did we learn?		
Clear use of Postmortem Analysis (PA)		
What went so well that we want to repeat it?		
What was useful but could have gone better?		
What were the mistakes that we want to avoid for the future?		
What were the reasons for the success or mistakes…?		
5.2 Other relevant evaluative comments in coursework report		5
	Sub-total	15
	Total	47