A follow-up empirical evaluation of evidence based software engineering by undergraduate students

Austen Rainer and Sarah Beecham
School of Computer Science
University of Hertfordshire
College Lane Campus
Hertfordshire AL10 9AB, U.K.

{a.w.rainer; s.beecham}@herts.ac.uk

Context: Evidence Based Software Engineering (EBSE) has recently been proposed as a methodology to help practitioners improve their technology adoption decisions given their particular circumstances. Formally, Systematic Literature Reviews (SLRs) are a part of EBSE. There has been a noticeable take up of SLRs by researchers, but little has been published on whether, and then how, the EBSE methodology has been applied in full.

Objectives: To empirically evaluate the use of EBSE by undergraduate students. To consider how insights into the students' performance with EBSE can be applied to software practitioners' expected performance with EBSE. To gain insights into the use of Supplementary EBSE Guidelines and associated Assessment Schemes.

Method: 37 final-year undergraduate students completed a coursework assessment that required them to use EBSE to evaluate one or more of four Requirements Management Tools (RMTs): Telelogic DOORs®, Borland's Caliber® Analyst, Compuware Optimal Trace™, and GODA ARTS. Students were provided with a range of EBSE resources, including a set of Supplementary EBSE Guidelines, to assist them with their EBSE evaluation. An Assessment Scheme was developed to assess the degree to which students followed the Supplementary Guidelines. A feedback questionnaire, completed by 12 of the 37 students, complemented the Assessment Scheme.

Results: 78% of students chose to evaluate a RMT that is currently a market leader. 62% of students subsequently recommended the adoption of their chosen RMT. Some students made a recommendation because the Guidelines indicated they should, rather than making a recommendation because the evidence 'entailed' that recommendation. Only 8% of students intentionally made no recommendation, and this seems to be on the basis of the poor quality of evidence available on the chosen RMT(s). All 12 students who completed the feedback questionnaire reported that this was the hardest or nearly the hardest assessment that the student had ever done! 67% of these 12 students reported that they had not given themselves sufficient time to complete the evaluation, and 83% reported that they had to balance their time on this evaluation with other commitments. The 12 students found EBSE steps 1 and 4 to be the easiest and EBSE steps 2 and 3 to be the hardest, and they generally reported that they had received sufficient support from the EBSE resources made available to them.

Conclusion: This study presents independent, empirical evidence of the full use of EBSE, to complement the growing body of published SLRs. We believe that the findings on students' use of EBSE are relevant to understanding professional practitioners' prospective use of EBSE. Both students and professionals would: find EBSE very challenging, in particular the SLRs; be subject to constraints and trade-offs; may not be able to find relevant and rigorous evidence; and may make errors in their critical thinking whilst conducting the evaluation. Our findings should benefit researchers, practitioners and educators.

Keywords: EBSE, evidence based software engineering, empirical methods, Systematic Literature Review, SLR, evaluation of software methods and tools, decision making, teaching, Requirements Management Tools, RMTs, Telelogic DOORs®, Borland Caliber® Analyst, Compuware Optimal Trace™, and GODA ARTS.

1. INTRODUCTION

Evidence Based Software Engineering (EBSE) has recently been proposed as a methodology to help practitioners improve their technology adoption decisions given their particular circumstances (Dybå et al., 2005). In simple terms, EBSE first recommends the conduct of a Systematic Literature Review (SLR) (Kitchenham, 2004) to identify and appraise evidence relevant to the problem or technology under consideration. EBSE then recommends that the evaluators integrate the results of the SLR with (their) practical experience, circumstances and (professional) values. There has been a noticeable take up of SLRs by researchers, as demonstrated by the number of SLRs published in the academic literature in the last two to three years (see (Kitchenham, 2007) for a review). Conversely, there has been very little attention directed at the second part of EBSE; indeed, in a previous paper (Rainer et al., 2006) published at the 2006 *Evaluation and Assessment in Software Engineering* (EASE'06) conference, we reported what we believe to be the first and, to date, the only empirical investigation of the full use of the EBSE methodology. (Dybå et al., (2005) report *anecdotal* evidence on the use of EBSE by students.).

In this paper, we report on a further investigation of EBSE, again by students but this time using EBSE to evaluate one of four industry-recognised requirements management tools (RMTs). The paper offers independent empirical evidence of the EBSE methodology, and extends our preliminary investigation (Rainer et al., 2006). Our research into EBSE complements the higher education teaching we undertake in the *Empirical Evaluation in Software Engineering* module, taught in the final year of the BSc(Hons) Computer Science degree programme at the University of Hertfordshire in the UK. (See (Rainer et al., 2007b) for a discussion of related modules offered in other universities.) As part of the assessment for that module, students use EBSE to evaluate technologies.

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.
- To promote the evaluation of the full EBSE methodology, and not only the use of Systematic Literature Reviews.

The remainder of the paper is organised as follows: section 2 discusses the importance of empirical evaluations for researchers and practitioners and briefly summarises the current status of EBSE; section 3 describes the design of the investigation reported here; section 4 presents quantitative results of the students' performance of EBSE; finally, section 5 discusses the results, draws some conclusions, and provides some directions for further research.

2 EVIDENCE BASED SOFTWARE ENGINEERING BACKGROUND

2.1 The importance of empirically-based evaluations

The software engineering research community has long-argued (e.g. (Fenton, 1994)) that the software industry frequently adopts technologies without first undertaking a structured evaluation (ideally an *empirically*-based evaluation) of those technologies. At the same time, the research community has also recognised that researchers themselves are prone to propose technologies without undertaking a structured evaluation of those technologies e.g. (Fenton, 1994, Glass, 1994, Glass, 1995, Tichy et al., 1995, Höfer and Tichy, 2007, Zelkowitz and Wallace, 1997, Zelkowitz et al., 2003).

2.2 A brief survey of evaluation methodologies in software engineering

To date, the empirical software engineering community appears to have proposed *two* methodologies specifically intended to improve practitioners' technology adoption decisions i.e. DESMET and Evidence Based Software Engineering (EBSE). DESMET (e.g. (Kitchenham et al., 1997)) was a collaborative project involving academia and industry, and partly funded by the UK government, which sought to develop a method to evaluate software engineering methods and tools. Much more recently, and as already noted, EBSE has been proposed as a methodology to support technology adoption decisions in software engineering. A number of other related initiatives have been proposed in the software engineering community. These include software process improvement models and frameworks, such as the Goal Question Metric (GQM) method and the Experience Factory. On a much larger scale, technologies such as the Capability Maturity Model (CMM) (Paulk et al., 1993) and the more recent Capability Maturity Model Integration (CMMI) focus on improving the organisation and, by implication, an organisation's ability to make effective decisions, for example in the Decision Analysis and Resolution (DAR) key process area. Underpinning many of these initiatives are measurement programmes designed to establish an empirical and ideally quantitative foundation on which to make better decisions. There has also been research on technology transfer, diffusion and assimilation and more generally, research on models of decision making in general. There are also methodologies for evaluating COTS products (e.g. (Donzelli et al., 2005)).

2.3 A brief overview of the Evidence Based Software Engineering methodology

As already stated, EBSE has been proposed as a methodology to help practitioners' improve their technology adoption decisions given their particular circumstances. EBSE comprises five steps, as summarised here in Table 1. The first three steps are closely related to Systematic Literature Reviews (SLRs), which are a means to provide a fair evaluation of a phenomenon of interest by using a trustworthy, rigorous, and auditable methodology (Kitchenham, 2004).

Table 1 The five steps of EBSE

Step	Description
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

During the first step of EBSE, the evaluator constructs an EBSE question comprising five components: a statement of the intervention (i.e. the instrument of change, such as the introduction of a new tool) of interest, a statement of a baseline against which the performance of the intervention is compared, a statement of the users of the intervention (e.g. the users' experience and skills), a statement of the situation within which the users apply the intervention (e.g. the type of project, type of application) and a statement of the particular outcome that intervention is expected to change and ideally improve. A simple example of an EBSE question is: Does the Telelogic DOORs Requirements Management Tool [intervention] more effectively trace requirements [outcome], when used by experienced requirements analysts [users] in the development of a large software system [situation], when compared to the use of no formal requirements management tool [baseline]? In step 4, the evaluator attempts to make a recommendation on whether or not to adopt the intervention, on the basis of the evaluator's conclusions from steps 3 and 4. It may be that the evidence is contradictory or inconclusive and a recommendation cannot be made.

The EBSE methodology was initially developed with reference to the widely-adopted Evidence Based Medicine (EBM) methodology, but more recently EBSE has drawn on other evidence based disciplines (Budgen et al., 2006). Kitchenham et al. (2004) in particular recognised the problems with 'transplanting' an EBM-like methodology into software engineering.

Dybå et al. (2005) argue 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 for practitioners, between EBSE and SPI, is that EBSE concentrates on improving practitioners' use of existing findings (notably through conducting or reviewing systematic literature reviews) in order to make decisions. In contrast, SPI concentrates on the introduction, assessment and management of a new technology intervention by those practitioners. 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.

2.4 A summary of the status of Evidence Based Software Engineering

EBSE is the subject of two research projects, both funded by the UK's EPSRC research council. The first project, now completed, developed a protocol for undertaking Systematic Literature Reviews (SLRs), examined the value of structured abstracts for supporting systematic literature reviews (Budgen et al., 2007), commenced a tertiary review of published systematic literature reviews in software engineering (Kitchenham, 2007), and sought to identify other evidence based (EB) disciplines that may be more like software engineering than EBM. The second project has recently commenced.

Since the initial publications on EBSE, a rapidly growing body of publications have reported the conduct of SLRs in software engineering research and, as already indicated, Kitchenham has reported preliminary results of a tertiary review of these SLRs at EASE'07 (Kitchenham, 2007). Broadly speaking, Kitchenham found that nine out of the 23 published SLRs in software engineering research that she reviewed have focussed on reporting research *trends* rather than reporting the efficacy of (technology) interventions. Kitchenham expresses disappointment with this ratio of reviews, as reviews of research trends primarily interest researchers rather than practitioners. As noted earlier, work has been undertaken on evaluating Structured Abstracts, which are considered to be particularly helpful to researchers undertaking SLRs.

Dybå et al. (2005) and Jørgensen et al. (Jørgensen, 2005, Jørgensen et al., 2005) report anecdotal evidence on the success of teaching EBSE to undergraduate students at a Norwegian university, and we have reported on a preliminary study of the use EBSE by students at a UK university (Rainer et al., 2007b, 2007a, Rainer et al., 2006). EBSE has been the subject of two workshops, *Realising Evidence Based Software Engineering* (REBSE), both co-located with the *International Conference on Software Engineering* (ICSE). Overall, the indications are that EBSE has been used almost exclusively by researchers, with some application of EBSE by undergraduate students. We are not aware of any published, empirically-based evidence of professional software practitioners directly using EBSE to make technology adoption decisions in their projects or organisations.

3 STUDY DESIGN

3.1 Rationale for the study design

In (Rainer et al., 2006) we focused on students' *use* of EBSE rather than the efficacy of EBSE. We did this because with the sample sizes available to use it is extremely difficult to separate out the effects of the methodology itself from the effects of a user's abilities, including training effects on that user. In the current study, we begin to consider the *outputs* from an EBSE evaluation as a step towards the future investigation of the efficacy of EBSE. We are also aware that one criticism of the EASE'06 paper is the use of students to investigate EBSE. We argue that students are a suitable population of users for investigating the use and efficacy of EBSE. In particular:

- Students can test a new methodology prior to releasing it to the professional community,
- Students can help to highlight where a methodology works, and works well, and where a methodology does not work.
- Students may have similar problems to professionals, such as: deadline pressures, time and resource constraints, the need to balance different objectives, and the need to produce an output conforming to a particular client's requirements.
- It may be easier to replicate a study if a similar population of users (i.e. students) is used in each replication.
- It may be easier to control aspects of the study.
- The research methods can be refined prior to the conduct of a more public and commercially sensitive study e.g. where a company is reluctant to discuss its problems.
- Students are the next generation of professionals, and studies of this kind provide an opportunity for students to experience the conduct of EBSE in particular but also empirically-based evaluations more generally.

3.2 Research model and research questions

Figure 1 presents the research model for this investigation, using the notation of Structured Analysis and Design Technique (SADT). As indicated in the figure, the model distinguishes between: the EBSE evaluation itself, the candidate input RMTs, controls on the evaluation, resources to be used in the evaluation, and the results and recommendation (if any) from the evaluation. In section 3.1, we recognised the effect that a user's ability could have on the output of the evaluation. At this stage, we do not explicitly represent a user in our model. The most obvious point within the model to include users is as a resource. Future research may decide to distinguish between the effects of controls, inputs, the activity and the resources on the output.

Figure 1 Research model

On the basis of the research model, we investigate the very general research question: What can we learn about EBSE, and its application, from students' use of EBSE?

3.3 Procedure

Students were issued with a coursework which required them to use EBSE to evaluate one or more of four available Requirements Management Tools (RMTs): Telelogic DOORs®, Borland Caliber® Analyst, Compuware Optimal Trace™, and GODA ARTS. The evaluation task was closely based on a DESMET evaluation being undertaken by a commercial organisation in collaboration with the University of Hertfordshire. On this basis, the *target* of the EBSE evaluation, Requirements Management Tools, is a realistic target for technology evaluations. (For confidentiality reasons, we cannot report on the DESMET work here.)

The timing of the coursework meant that students received five lectures and tutorials on the *Empirical Evaluation in Software Engineering* module *prior* to receiving the coursework, these lectures covered general topics relating to decision making, claims, arguments and evidence, and definitions. During the coursework, the students received two lectures specifically on EBSE (including a tutorial that presented results reported in (Rainer et al., 2006)) and

lectures on related topics e.g. a discussion of the claims made in the Standish Group's 1994 CHAOS Report and the responses from Jørgensen and Moløkken (2005; see also (Jørgensen and Moløkken, 2006) and (Glass, 2005)). Students were also provided with copies of the *IEEE Software* article on EBSE (Dybå et al., 2005), and references to the other articles in the series (Kitchenham et al., 2004; Jørgensen et al., 2005; Jørgensen, 2005). In addition, we developed and provided students with Supplementary EBSE Guidelines (Rainer and Beecham, 2008) to assist them with their evaluation, and we also developed and applied a complementary Assessment Scheme to assess the degree to which students followed the Supplementary EBSE guidelines. Both the Supplementary Guidelines and the Assessment Scheme have been developed from our previous work (Rainer et al., 2007b, 2007a, Rainer et al., 2006). During the coursework period, opportunities were provided to students at lectures, tutorials and via the University's student intra-net to ask questions to clarify the coursework specification.

Students were given six weeks to individually complete and submit the coursework. Students are allowed to submit coursework up to one week late, but their assessment mark is capped at a minimum pass. The assessment capping was not applied to the measures reported here. Two students submitted late, one with extenuating circumstances and one without.

Students were set a word limit of 2500 words 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. A typical student is expected to direct 30 hours at assessments of this kind.

Overall, 37 students participated in the assessment. Of those 37 students, 12 also completed the feedback questionnaire on their experiences of using EBSE to evaluate one or more RMTs.

3.4 Comparison with previous EBSE investigations

Table 2 summarises the three investigations that we have been undertaken to date in conjunction with the *Empirical Evaluation in Software Engineering* module. The first evaluation, in 2005, was reported in EASE'06 and the third evaluation, in 2007, is the focus of the current paper. The second evaluation has not been published in the public domain. All three evaluations received ethics approval from the Faculty.

TABLE 2: Summary of three investigations of EBSE using students

Investigation	Year	Number of case for quantitative analysis	Number of cases for qualitative analysis
1	2005	38	7
2	2006	41	18
3	2007	37	19

3.5 Feedback from students

The final lecture and tutorial for the term occurred on the same day as the submission of the coursework. During the tutorial, the first author sought feedback from the attending students on the coursework. Students were invited, but not required, to complete a two-page feedback questionnaire and most but not all students did so. In addition to the 11 students who completed the questionnaire in the tutorial, a twelfth student later submitted the feedback form.

4. RESULTS

4.1 Students' performance using EBSE

Figure 2 presents box plots for the students' performance on each step of EBSE, together with the students' overall performance. The percentages are derived from the application of the Assessment Scheme (see (Rainer et al., 2007a)). As with the results reported in the EASE'06 paper, the box-plots presented here in Figure 2 need to be interpreted carefully. The different steps of EBSE are not equally difficult and although we have attempted to take account of this variance in difficulty by reporting percentages, these percentages may still not be accurate. Related to this, it is not clear what percentage 'threshold' would constitute sufficient performance by the student in the respective step of EBSE. On the one hand, a score of 100% in, for example, EBSE step 3 may indicate that the respective student has sufficiently followed the EBSE guidelines for that step; on the other hand, a percentage of 50% may indicate that the respective student has sufficiently followed the EBSE guidelines with a percentage of 100% indicating that the respective student has excelled at following the EBSE guidelines¹.

¹ During the presentation of the EASE'06 results at an informal research workshop, one audience member responded that they would consider a percentage of less than 80% as indicating a failure, on the part of the respective student, to sufficiently undertake that step of EBSE.

The box-plots may be useful for indicating the *range* of performance within each step, as well as the median performance for that step. Generally speaking:

- Each of the box-plots are evenly distributed
- The median values for each box-plot lie in the region of 45% - 60%.

Together, these two observations suggest that the Assessment Scheme is sufficiently discriminating in presenting the range of students' use of the EBSE Supplementary Guidelines.

FIGURE 2: Box plots of students' performance in each step of EBSE, and overall

4.2 Recommendations from the EBSE evaluations

TABLE 4: Breakdown of RMTs as intervention, or baseline

RMT	Intervention		Baseline		Total	
	Count	%	Count	%	Count	%
Telelogic DOORs	19		7		26	
CaliberAnalyst	10		15		25	
Compuware Optimal Trace	7		7		14	
GODA ARTS	2		6		8	
Manual	0		9		9	
Total	37		44			

Table 4 presents a breakdown of how frequently each RMT was used as the intervention or baseline in the EBSE evaluations. (See section 2.3 for an explanation of the terminology used in the table.) Some students proposed multiple baselines (e.g. more than one RMT as the baseline/comparison) or alternatively did not explicitly indicate whether an RMT was intended as the intervention or the baseline. One student proposed multiple interventions.

TABLE 5: Recommendations from EBSE evaluations

Recommendation	Count	Percentage
Intervention	23	62%
Baseline	5	14%
Ambiguous	6	16%
No recommendation	3	8%

Telelogic DOORs and Borland Caliber Analyst are the most frequently evaluated RMTs, with DOORs being the most frequently evaluated *intervention* and Borland Caliber Analyst the most frequently evaluated *baseline*. GODA ARTS was rarely used as the intervention or as the baseline. Unsurprisingly, no EBSE evaluation considered a manual requirements management approach as the intervention although this was in fact permissible in the coursework scenario given to students.

Table 5 indicates that most students, in Step 4 of their EBSE evaluation, recommend the intervention. Qualitative impressions suggest that students tended to seek evidence to *support* the proposed intervention in their EBSE question, and tended not to seek evidence that contradicted the adoption of their proposed intervention or, alternatively, that supported the adoption of the baseline.

4.3 Feedback from students

4.3.1 The making of a recommendation in the EBSE evaluation

All 12 students who completed the feedback form responded that they made a recommendation in their coursework. (See section 2.3 for a brief discussion of the guidance in step 4 to make a recommendation.) Verbal feedback provided during the tutorial, but not recorded on the questionnaire, suggests that at least some students made a recommendation because the Supplementary Guidelines required it rather than the student being confident about the making of a recommendation.

4.3.2 The degree of challenge from the coursework

We asked students to indicate, on a scale of 1 – 7, how challenging they found the coursework. A value of 1 represented the statement “The easiest coursework I have ever done” and a value of 7 represented the statement “The hardest coursework I have ever done”. For the sample of 12 students, the median and mode was 6 and the mean was 6.25. The lowest value was 5 and one student responded by extending the scale to 8! Re-setting that particular value to 7 actually changes the modal value from 6 to 7! and very slightly reduces the mean value to 6.16. Overall, this sample of students is clearly indicating that this was the hardest coursework that they have had to undertake on their degree programme. This is the final year of the degree programme and this coursework was the first coursework that these students had been set in their final year. It may be that students would revise their opinion of the coursework having completed other courseworks, including their final year dissertations.

4.3.3 The easiest and hardest steps in EBSE

We asked students what they considered to be the easiest and hardest steps in EBSE. Table 6 presents the results. The table indicates that students could more clearly identify *one* easiest step but could not clearly identify only one hardest step. Because the students had difficulty identifying only one hardest step, Table 6 reports two columns for the hardest step together with a total. The steps considered to be easiest were Step 4 and Step 1 which clearly contrast to those steps identified by students as the hardest steps. Although students found it hard to identify a single hardest step, there is some indication that Step 2 is considered harder than Step 3 (because no student selected Step 2 as their second choice). Again, verbal feedback provided during the feedback tutorial suggests that students found Step 2 very frustrating because while the guidelines were clear, there was an on-going iterative process of searching on the Internet for articles with rigorous and relevant evidence.

TABLE 6: The easiest and hardest steps in EBSE

EBSE step	Easiest choice 1	Hardest		Total
		choice 1	choice #2	
Step 1	4	0	0	0
Step 2		7	0	7
Step 3		4	4	8
Step 4	6	0	1	1
Step 5	2	1	0	1

4.3.4 The adequacy of resources

We also asked students whether they received sufficient resources to help them with their coursework. We used a scale of 1 – 3, with a value of 1 indicating ‘Not enough support’, a value of 2 indicating ‘Enough support’, and a value of 3 indicating ‘More than enough support’. Table 7 indicates that, generally, students thought the resources were sufficient, with the Supplementary Guidelines receiving the highest ‘score’.

TABLE 7 The adequacy of resources

Resource	Median	Mode
The lectures and tutorials on EBSE	2.5	3
The Supplementary Guidelines on EBSE	2.6	3
The EBSE articles posted on StudyNet	2.3	3
Other articles posted on StudyNet	2.3	2
StudyNet Classroom Discussion	2.1	2
Personal emails with tutor	2.1	2

4.4 The impact of the coursework specification as a control

The University's student intra-net allows lecturers to monitor students' use of the online teaching resources available to a module. We can in principle use data from the monitoring facility to gain insights into, for example, when students first started to use the resources related to this coursework. For ethical reasons we are not able to report here the actual data from this monitoring facility. Data from the monitoring facility does indicate however that there was a substantial increase in module accesses in the three days prior to the coursework submission date, which would be consistent with the stereotype of many students not (seriously) starting their coursework assessment until close to the submission date. This is a kind of deadline effect, and can be treated as evidence of the influence of the coursework specification as a *control* on the evaluation. As indicated in section 3.1, we believe that we can generalise this point beyond students and courseworks: external controls on an evaluation will influence the conduct of that evaluation. In a professional situation, conflicting priorities, limited resources and deadline pressure may mean that evaluators are not able to devote the time, effort and attention that they ideally want to direct to their evaluation.

5. DISCUSSION AND CONCLUSION

5.1 Summary and brief discussion of results

In section 3, we stated a general research question *viz.* What can we learn about EBSE, and its application, from students' use of EBSE? In seeking to answer that general question, we report here the findings of a third investigation of the use of the EBSE methodology by undergraduate students, and in so doing we complement the findings from our first investigation (Rainer et al., 2006). We also compare the overall performance of students for each of the five steps of EBSE for each of the three studies we have conducted. Our three studies suggest that:

- The behaviour of students undertaking an EBSE evaluation in a coursework can provide insights that are not only relevant to the teaching and evaluation of EBSE for students, but insights that are also relevant to the teaching, use and evaluation of EBSE by professional practitioners. For example: students and professional practitioners deal with conflicting objectives, the need to allocate limited resources to multiple activities, deadline pressures, requirements to complete evaluations to a certain specification, and the difficulties of finding evidence that is relevant and rigorous. Students and professionals (and indeed researchers) may also be subject to similar human errors in critical thinking, such as confirmation bias.
- Undergraduate students are able to undertake an EBSE evaluation however they find the activity to be very challenging. We suspect that professional practitioners would also find such evaluations to be challenging. Related to this, we cannot reasonably expect undergraduate students or professionals to perform an evaluation of the quality of doctoral students, or indeed of highly-experienced, post-doctoral researchers and academics. Undergraduate students and professionals lack the time, resources and skills to undertake highly-technical evaluations. By contrast, PhD students and Research Fellows may be allocated an extended period of time and resource that is dedicated to undertaking an EBSE evaluation.
- Students find EBSE Step 1 and Step 4 to be the easier steps to perform. We believe that for Step 1 this is due to the well-defined, discrete nature of the task to be performed (i.e. to define an EBSE question). In our EASE'06 paper, we speculated that students successfully define a *superficial* EBSE question, and we found further evidence for this speculation in analyses reported in (Rainer et al., 2007b) and (Rainer et al., 2007a). For Step 4, we speculate that students perceive this task to be easier because they are drawing on their own experience but again our further analyses suggest that students are not very effective at reflecting on their own practice and experience. Our findings suggest that professionals would also find EBSE Step 1 and 4 the easiest steps to perform.
- Students find EBSE Step 2 and Step 3 to be the hardest steps to perform. There is some indication that Step 2 is considered to be the hardest step and this may be due to the frustration that arises when searching the Internet for relevant and rigorous information, and not knowing whether and when one has found the best information or whether one has exhausted the search space. Our findings suggest that professionals would also find EBSE Step 2 and 3 the hardest steps to perform.
- Many students seemed to select the RMT(s) to evaluate on the basis of the tool vendors market position and, related to this, on the quantity of information that was available on the Internet about the vendor and the

tool. This is an understandable strategy for a coursework: more information on the Internet suggests that it will be easier to find relevant, high-quality information.

- Many students ultimately recommended the RMT that they had selected as the intervention, but this needs to be qualified by the fact that some students said that they made a recommendation because Guidelines stated that they should, rather than because students thought the evidence justified that recommendation. Accepting this qualification, a confirmation bias affect may be present here, where students seek evidence to confirm their choice of intervention and, ultimately, make a recommendation that confirms their choice of intervention.
- Students found the resources available for the coursework to be at least sufficient, with the Supplementary Guidelines receiving the highest ranking.

5.3 Further research

We are already undertaking the following research activities to further investigate EBSE as a technology evaluation methodology:

1. Further analysis of the current dataset. This includes: the analysis of information resources selected during EBSE Step 2; the analysis of the subsequent acceptance and rejection in EBSE step 3 of the articles identified in step 2; the analysis of the relationship between the search terms selected in EBSE step 2 and the EBSE question constructed.
2. A comparison of students' performance with a researcher's performance on the coursework.
3. A comparison of EBSE evaluations of RMTs by students with a DESMET evaluation of RMTs by a practitioner.

5.4 Conclusion

Evidence Based Software Engineering (EBSE) has been proposed as a methodology to help practitioners' improve their technology adoption decisions given their particular circumstances. EBSE first recommends the conduct of a Systematic Literature Review (SLR) to identify and appraise evidence relevant to the problem or technology under consideration. EBSE then recommends that the evaluators integrate the results of the SLR with (their) practical experience, circumstances and (professional) values. There has been a noticeable take up of SLRs by researchers but, conversely, there has been very little attention directed at the integration of the results of the SLRs with practical experience, circumstances and values. In this paper we report on a further investigation where students have used EBSE to evaluate one of four industry-recognised requirements management tools (RMTs). This study offers independent empirical evidence of the full EBSE methodology, and extends our preliminary investigation in (Rainer et al., 2006). Whilst we have used students in our investigation, we argue that insights from students' use of EBSE can be applied to professional practitioners' use of EBSE. In particular: EBSE is very challenging, with the SLR component of EBSE being the most challenging. The conduct of EBSE is subject to resource and time constraints, trade-offs and deadline pressures. Recommendations on particular tool(s) to adopt may be influenced by the requirement for a recommendation rather than the evidence supporting that recommendation. When selecting and evaluating information, evaluators need to be aware of the possible human error in their critical thinking, such as confirmation bias.

ACKNOWLEDGEMENTS

We thank the 37 students for participating in the coursework, and the 20 students who agreed for us to use their courseworks in our research. We also thank students from the two previous studies. All three studies have received ethics approval under Faculty Ethics Protocols 06/04, 07/07 and 0708/78. Results from the students' evaluation should not be interpreted as a recommendation for the use or non-use of any of the four Requirements Management Tools. Telelogic DOORS is a trademark of Telelogic.AB. Caliber is a trademark of Borland Software Corporation. Optimal Trace is a trademark of Compuware Corporation. Finally, we thank the reviewers for their detailed, constructive comments.

REFERENCES

- Budgen, D., Charters, S., Turner, M., Brereton, P., Kitchenham, B. & Linkman, S. (2006) Investigating The Applicability Of The Evidence Based Paradigm To Software Engineering. *2nd International Workshop On Interdisciplinary Software Engineering Research (WISER'06) May 20, 2006*. Shanghai, China.
- Budgen, D., Kitchenham, B., Charters, S., Turner, M., Brereton, P., Linkman, S. (2007) Preliminary Results Of A Study Of The Completeness And Clarity Of Structured Abstracts, *11th International Conference On Evaluation And Assessment Of Software Engineering (Ease)*, Keele University, U.K., 2 - 3 April 2007
- Donzelli, P., Zelkowitz, M., Basili, V., Allard, D. & Meyer, K. (2005) Evaluating COTS Component Dependability In Context. *Software, IEEE*, 22(4), 46-53.
- Dybå, T., Kitchenham, B. A. & Jorgensen, M. (2005) Evidence-Based Software Engineering For Practitioners. *IEEE Software*, 22(1), 58-65.
- Fenton, N. (1994) Software Measurement: A Necessary Scientific Basis. *IEEE Transactions On Software Engineering*, 20(3), 199-206.
- Glass, R. L. (1994) The Software Research Crisis. *IEEE Software*, 11(6), 42-47.

- Glass, R. L. (1995) A Structure-Based Critique Of Contemporary Computing Research. *Journal Of Software Systems*, 28(1), 3-7.
- Glass, R. L. (2005) It Failure Rates--70% Or 10-15%? *IEEE Software*, 22 (3), 112-111.
- Höfer, A. & Tichy, W. F. (2007) Status Of Empirical Research In Software Engineering. In Basili, V. R., Rombach, D., Schneider, K., Kitchenham, B., Pfahl, D. & Selby, R. W. (Eds.) *Empirical Software Engineering Issues: Critical Assessment And Future Directions*. Springer.
- Jørgensen, M. (2005) Teaching Evidence-Based Software Engineering To University Students. *Evaluation And Assessment In Software Engineering (Ease)*. Keele University, 11 - 13 April, 2005.
- Jørgensen, M., Dyba, T. & Kitchenham, B. (2005) Teaching Evidence-Based Software Engineering To University Students. *11th IEEE International Software Metrics Symposium*. Como, Italy, 19-22 September 2005.
- Jørgensen, M. & Moløkken, K. (2005) How Large Are Software Cost Overruns? A Review Of The 1994 Chaos Report. Simula Research Laboratory.
- Jørgensen, M. & Moløkken, K. (2006) How Large Are Software Cost Overruns? A Review Of The 1994 Chaos Report. *Information And Software Technology*, 48(4), 297-301.
- Kitchenham, B. (2004) Procedures For Performing Systematic Reviews. Keele University Technical Report TR/SE-0401 and NICTA Technical Report 0400011T.1
- Kitchenham, B. (2007) Current State Of Evidence-Based Software Engineering (Keynote Address). *11th International Conference On Evaluation And Assessment In Software Engineering*. 2-3 April 2007.
- Kitchenham, B., Linkman, S. & Law, D. (1997) DESMET: A Methodology For Evaluating Software Engineering Methods And Tools. *Computing & Control Engineering Journal*, 8(3), 120-126.
- Kitchenham, B. A., Dyba, T. & Jorgensen, M. (2004) Evidence-Based Software Engineering. *26th International Conference On Software Engineering (ICSE'04)*. Edinburgh, Scotland, U.K.
- Paulk, M. C., Curtis, B., Chrissis, M. B. & Weber, C. V. (1993) Capability Maturity Model, Version 1.1. *IEEE Software*, 10(4), 18-27.
- Rainer, A. & Beecham, S. (2008) Supplementary Guidelines And Assessment Scheme For The Use Of Evidence Based Software Engineering. Hatfield, U.K., University Of Hertfordshire Technical Report CS-TR-469.
- Rainer, A., Beecham, S. & Hall, T. (2007a) Assessing Undergraduate Students' Use Of Evidence Based Software Engineering. Hatfield, U.K., University Of Hertfordshire. Technical Report CS-TR-462
- Rainer, A., Beecham, S. & Hall, T. (2007b) Assessing Undergraduate Students' Use Of Evidence Based Software Engineering: A Preliminary Evaluation Of A Research Instrument. *Workshop Series On Empirical Software Engineering (WSESE'07)*. Riga, Latvia, 2nd - 4th July 2007.
- Rainer, A. W., Hall, T. & Baddoo, N. (2006) A Preliminary Empirical Investigation Of The Use Of Evidence Based Software Engineering By Under-Graduate Students. *2006 Evaluation And Assessment In Software Engineering (EASE'06) Conference*. Keele University, U.K., 10th – 11th April, 2006.
- Tichy, W. F., Lukowicz, P., Prechelt, L. & Heniz, E. A. (1995) Experimental Evaluation In Computer Science: A Quantitative Study. *Journal Of Systems Software*, 28(1), 9-18.
- Zelkowitz, M. V. & Wallace, D. (1997) Experimental Validation In Software Engineering. *Empirical Assessment In Software Engineering (EASE'97)*. University of Keele.
- Zelkowitz, M. V., Wallace, D. & Binkley, D. W. (2003) Experimental Validation In Software Engineering. *Lecture Notes on Empirical Software Engineering*. World Scientific.