A Quality Checklist for Technology-Centred Testing Studies

Barbara A. Kitchenham
School of Computing and Mathematics, Keele University, Keele, Staffordshire ST5 5BG, UK
B.A.Kitchenham@@cs.keele.ac.uk

Andrew J. Burn
Department of Computer Science, Durham University, Durham, Science Laboratories, South Road, Durham City, DH1 3LE, UK.
a.j.burn@durham.ac.uk

Zhi Li
School of Computing and Mathematics, Keele University, Keele, Staffordshire ST5 5BG, UK
z.li@epsam.keele.ac.uk

Abstract Background: **One aspect of undertaking a systematic literature review is to perform a quality evaluation of primary studies. Most quality checklists adopted from medicine, psychology and social studies assume that the experimental unit in an experiment is a human being. However, in empirical studies in software engineering the experimental unit may be a technology, an application or an algorithm. *Aim*: This paper presents a checklist we are developing to evaluate the quality of empirical technology-centred testing studies. *Discussion points*: The checklist was developed by considering entities used in technology-centred testing studies and the validation problems associated with them. The planned validation process includes face validation, usability and reliability assessment and external validation. As yet the external validation has not been performed. *Conclusions*: The checklists appear to be usable and after some experience applying them they appear to give consistent results. However, their validation is as yet incomplete. The method of developing and evaluating the checklist may be of use to other researchers requiring a means of assessing the quality of technology-centred software engineering studies.**

Keywords: systematic literature review, primary study quality, quality checklists, technology-centred studies; testing

1. INTRODUCTION

The Evidence-based Practices Informing Computing (EPIC) project aims to investigate the use of systematic literature reviews (SLR) in Software Engineering. This investigation involves undertaking systematic literature reviews that address a number of research questions. Two such research questions are:

> RQ3: To what extent is the adoption of an extended search space vital for answering detailed research questions?
> RQ8: What are the particular problems facing novices when they first undertake SLRs?

The research question numbers were initially identified in the EPIC Scoping document [1].

We are addressing these questions by replicating a previously reported SLR in the testing domain, which used a limited search, with a study addressing the same questions but using a much broader search strategy [4].

One aspect of undertaking a systematic literature review is to perform a quality evaluation of primary studies. Quality checklists adopted from medicine, psychology and social studies (as proposed by Kitchenham and Charters [7]) assume that the experimental unit in any experiment is a human being. However, in empirical studies in software engineering the experimental unit may be a technology, an application or an algorithm. Thus, although we specified a quality checklist for our SLR ([8], [9]), it was soon obvious that it was not appropriate for technology-centred studies. Note, there is something of a terminology problem with these studies. Although many researchers refer to these studies as “experiments”, they seldom involve concepts such as random allocation of experimental units to treatment that is an essential part of a controlled experiment, it may therefore be better to refer to them as quasi-experiments or benchmarking studies. The term benchmarking studies is most appropriate when the studies re-use existing experimental materials rather than search out new materials.

In order to address the problem of assessing the quality of technology-centred studies, we developed a checklist and are in the process of validating it. This paper describes the checklist, its construction and proposed validation, with the aim of providing some ideas for other researchers faced with constructing quality checklists for technology-centred studies (for example performance studies, or cost model evaluation studies). We also hope that the checklist might help researchers undertaking technology-centred empirical testing studies to improve the quality of their studies.

2. CHECKLIST CONSTRUCTION

One of us (Kitchenham) constructed a checklist based on a model of the objects used in a technology-centred empirical testing studies. The checklist was based on the extent to which the objects might cause bias in the studies and a review of three relevant papers that discussed the validation problems of technology-centred empirical studies ([3], [5], [13]).

The underlying model of a benchmarking study used to create the quality checklist is shown in Figure 1, the entities in the model include:

- Test Methods (algorithms or methods that are automated and that may relate to the construction of Test Cases, assessing the effectiveness of Test Cases, or selecting among a set of Test Cases)
- Test Object(s) (one or more software programs that are executed using the Test Method) sometimes called SUT (subject under test)
- Test Case (a set of input values used to execute a Test Object with the intention of detecting Faults). Note some studies are based on executing specific statements (such as modified statements) rather than detecting faults.
- Fault (one or more defects exist in one Test Object that may be revealed by executing one or more Test Cases).

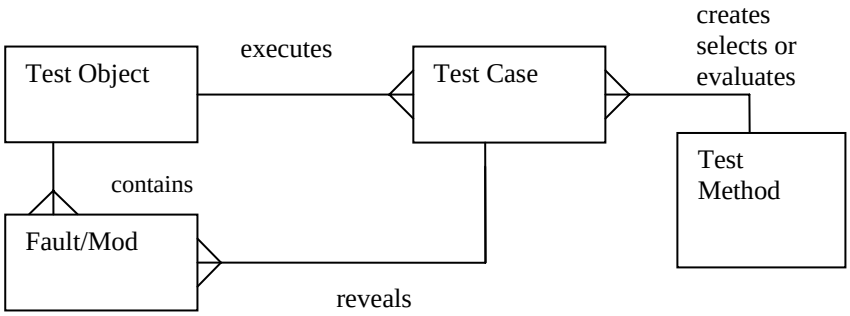

Figure 1 Model of the Enmities involved in a Testing Study

A review of validity issues identified in the three empirical testing papers and general data analysis and quality principles suggested that a quality checklist should be based on the following assumptions:

1. The most trustworthy experiments (in terms of external validity) use industrial Test Objects, Test Cases and Faults.
2. There must be some form of replications either in terms of multiple test cases, multiple test case sets, or multiple test case objects or there is no valid empirical study. Note replication for test objects may occur when a large program is made up of many components and each component is tested.
3. The outcome metrics should be valid measures of the concept they represent.
4. Statistical analyses should be appropriate.
5. Limitations should be reported.

Kitchenham used these criteria to develop a checklist of 8 questions (where one question was involved two sub-questions) and then used the checklist on two other papers ([10] and [12]). After the initial trial, the checklist was revised to give the checklist shown in Table 1.

Table 1Quality Checklist with results for two papers

Question No	Question	Available answers
1	Are the study measures valid?	None=0 / Some (0.33) / Most (0.cc) / All (1)
2	Was there any replication, i.e. multiple test objects, multiple test sets?	Yes (1) /No (0)
3	If test cases were required by the Test Treatment, how were the test cases generated?	Not applicable. By the experimenters (Yes=0) By an independent third party (Yes=0.5) Automatically (Yes=0.5) By industry practitioners when the test object was created (Yes=1)
4	How were Test Objects generated?	Small programs (Yes=0) Derived from industrial programs but simplified (Yes=0.5) Real industrial programs. (Yes=1)
5	How were the faults/modifications found?	Naturally occurring Yes=1, go to question 6

		If No go to questions 5a
5a	For seeded faults/modifications, how were the faults identified?	Faults introduced by the experimenters (Yes=0), Independent third party (Yes=0.25) Generated automatically (Yes=0.5) Go to 5b.
5b	For seeded faults/modifications, were the type and number of faults introduced justified?	Type & Number: Yes (0.5) Type or Number (Yes=0.25) No=0
6	Did the statistical analysis match the study design?	No=(0), somewhat (0.33) Mostly (0.66), Completely (1)
7	Was any sensitivity analysis done to assess whether results were due to a specific test object or a specific type of fault?	Yes=1 / Somewhat=0.5 / No=0
8	Were limitations of the study reported either during the explanation of the study design or during the discussion of the study results?	No=0, Somewhat=0.5, Extensively=1

3. QUALITY CHECKLIST EVALUATION

This section discusses how we plan to evaluate the checklist and our results to date.

3.1 Method

Any quality checklist, even a re-used one, should ideally be validated as part of a new SLR. We adopted a validation method that tried to balance rigour with available resources, based on a number of different validation approaches:

- Face validity where the checklist is reviewed by intended users.
- Usability evaluation where the checklist is trialled on several papers by its users.
- Reliability evaluation where the results of using the checklist on the same paper by different users are checked for consistency.
- External validity where the results of using the checklists are compared with the subjective opinion of paper quality made by one or more experts.

3.2 Results

Two of us (Burn and Li) reviewed the checklist for face validity and usability. They found that the checklist required some clarification. Two questions were difficult to answer:

1. Q6: Did the analysis match the design?
2. Q7: Was any sensitivity analysis done?

We agreed that for the design question, the users need to be alert to a design that has elements (i.e. multiple tests on the same piece of software, blocking based on complexity or size of software artefacts) that are not directly reflected in the analysis. If checklist users were unsure, the question should be referred to a statistical expert. For sensitivity analysis, the users need to look for any analysis that mentions:

- The impact of data outliers on results.
- Different analysis approaches and compared their consistency.
- Uses the terms stability analysis or sensitivity analysis.

If nothing of this sort was reported, paper should be given a No for sensitivity analysis.

Another problem was that some papers use both "real industrial" and “toy programs” as test objects. It was not clear how to answer the questions in these cases. We agreed that the users need to consider whether the paper should be regarded as reporting multiple studies and report results for each study separately, otherwise they should report the average across the different test objects.

Burn and Li then undertook a reliability assessment. They extracted the checklist answers from three papers. They initially used two papers ([1] and [6]) but felt the two papers were rather similar, and so included a third paper [11]. Initially they answered the quality questions for each paper independently and then met to answer the questions together:

- There were 5 initial disagreements (out of 9 questions – question 5 having two parts) for paper [1] due primarily to the problem of multiple types of test object, with one disagreement due to misunderstanding the question.
- There were two initial disagreements for paper [6], although they made little difference to the overall score.
- There were no initial disagreements for paper [11].

These results suggest that the quality criteria need some getting used to, but after some experience can be used to give consistent results.

External validity is yet to be assessed. We plan for one software expert and one statistical expert to undertake the evaluation, each reviewing 2-4 papers regarded as good quality and 2-4 papers regarded as relatively poor quality. This validation exercise depends on there being some major differences in the quality of papers being examined. If no major quality differences are detected by the quality checklist, the experts will be asked to confirm that their assessment of the quality confirms that there is not much to choose between the papers. Note the three papers evaluated by the Burn and Li all had very similar quality scores. The papers evaluated by Kitchenham also had similar scores to the papers evaluated by the Burn and Li. This suggests that we may not find extreme values for quality and will have to adopt the second form of assessment.

4. CONCLUSIONS

This paper has presented a proposed quality checklist for technology-centred empirical testing studies. We have explained how we constructed the checklist and how we planned to validate the checklist. The validation results to date have confirmed that the checklist is understood by its intended users and leads to reasonably consistent quality evaluations. We have yet to confirm that the quality checklist gives results that conform with an expert's view of paper quality.

However, we think the process by which we developed the checklist and our validation process may be of interest to other researchers undertaking systematic reviews of technology-centred empirical studies. The checklist itself may also be of interest to researchers undertaking testing studies. For example, to date we have reviewed 23 technology-centred empirical testing papers and 9 included little or no discussion of limitations and none of the studies discussed sensitivity analysis (i.e. whether the results could be due to some anomalous data point).

REFERENCES

[1] Bible, J., Rothermel, G., and Rosenblum, D.S. (2001) A Comparative Study of Coarse- and Fine-Grained Safe Regression Test Selection Techniques, ACM Transactions on Software Engineering and Methodology, 10(2), pp 149–183.

[2] Brereton, O. P., Kitchenham, B.A. (2007) The Scope of EPIC Case Studies. EPIC technical Report EPIC-2007-01.

[3] Briand, L.C. (2007) A critical Analysis of Empirical Research in Software Testing. Keynote Address, ESEM 2007, pp 1-8.

[4] Budgen. D. (2008) Supporting Novices undertaking Systematic Literature Reviews. EPIC Case Study Protocol No: CS001/07, May.

[5] Elbaum, S., Malishevsky, A.G. and Rothermel, G. (2002) Test case prioritization: A family of empirical studies. IEEE TSE, 28(2), pp 159-182.

[6] Graves, T.L., Harold, M.J., Kim, J.M., Porter, A., and Rothermel, G. (2001) An Empirical Study of Regression Test Selection Techniques. ACM Transactions on Software Engineering and Methodology, 10(2), pp 184-208.

[7] Kitchenham B.A. and Charters S.M. (2007). Guidelines for performing Systematic Literature Reviews in Software Engineering, Version 2.3. EBSE Technical Report EBSE-2007-01, Keele University and Durham University, July 2007.

[8] Kitchenham B.A., Brereton, O. P., Budgen, D. and Li, Z. (2008) Results from Case Study 1 – Quality Checklists, EPIC Technical Report, EBSE 2008-009.

[9] Kitchenham B.A., Brereton, O. P., Budgen, D. and Li, Z. (2009) An Evaluation of Quality Checklist Proposals – A participant-observer case study. EASE09, accepted for publication.

[10] Mansour, N., Bahsoon, R., and Baradhi, G. (2001) Empirical comparison of regression test selection algorithms. JSS, 57, pp 79-90.

[11] Offutt, A.J., Pan, J., Tewary, K., Zhang, T. (1995) An Experimental Evaluation of Data Flow and Mutation Testing, Software: Practice and Experience 26(2) pp 165-176.

[12] Porwal, R. and Gursaran. (2004) An experimental evaluation of weak-branch criterion for class testing. JSS 70, pp 209-224.

[13] Rothermel, G., Harrold, M.J. (1998) Empirical Studies of a Safe Regression Test Selection technique. IEEE TSE, 24(6), pp 401-419.

Acknowledgements
This study was funded by the UK Engineering and Physical Sciences Research Council project EP/E046983/1.