Accepted Manuscript
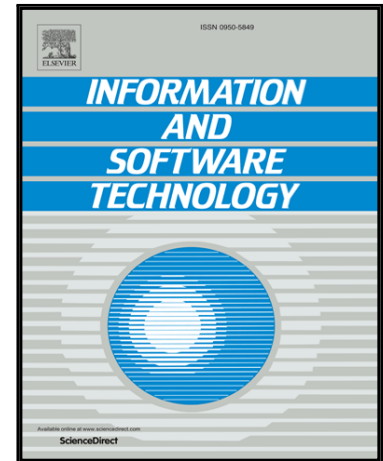
Reporting Systematic Reviews: Some Lessons from a Tertiary Study

David Budgen, Pearl Brereton, Sarah Drummond, Nikki Williams

Please cite this article as: David Budgen, Pearl Brereton, Sarah Drummond, Nikki Williams, Reporting Systematic Reviews: Some Lessons from a Tertiary Study, *Information and Software Technology* (2017), doi: 10.1016/j.infsof.2017.10.017

Reporting Systematic Reviews: Some Lessons from a Tertiary Study

David Budgen[a,∗], Pearl Brereton[b], Sarah Drummond[a], Nikki Williams[c,1]

[a]Durham University, School of Engineering & Computing Sciences, Durham DH1 4LA
[b]Keele University, School of Computing & Maths, Staffordshire ST5 5BG
[c]Cranfield University, Centre for Electronic Warfare, Information & Cyber, Defence Academy of the United Kingdom, Shrivenham SN6 8LA

Abstract

Context: Many of the systematic reviews published in software engineering are related to research or methodological issues and hence are unlikely to be of direct benefit to practitioners or teachers. Those that are relevant to practice and teaching need to be presented in a form that makes their findings usable with minimum interpretation.

Objective: We have examined a sample of the many systematic reviews that have been published over a period of six years, in order to assess how well these are reported and identify useful lessons about how this might be done.

Method: We undertook a tertiary study, performing a systematic review of systematic reviews. Our study found 178 systematic reviews published in a set of major software engineering journals over the period 2010-2015. Of these, 37 provided recommendations or conclusions of relevance to education and/or practice and we used the DARE criteria as well as other attributes related to the systematic review process to analyse how well they were reported.

Results: We have derived a set of 12 'lessons' that could help authors with reporting the outcomes of a systematic review in software engineering. We also provide an associated checklist for use by journal and conference referees.

Conclusions: There are several areas where better reporting is needed, including quality assessment, synthesis, and the procedures followed by the reviewers. Researchers, practitioners, teachers and journal referees would all benefit from better reporting of systematic reviews, both for clarity and also for establishing the provenance of any findings.

Keywords:
Systematic review, reporting quality, provenance of findings

1. Introduction

The idea of adapting the use of secondary studies (systematic reviews) to form a tool of empirical software engineering was first proposed in 2004 [1]. Since then, they have become a well established tool for empirical research.

However, what may easily be overlooked is that the motivation for using a systematic review in software engineering usually differs from those that occur in other disciplines, such as health, education and the social sciences. For those disciplines, both systematic reviews and the primary studies that form their inputs are commonly sponsored and commissioned by government and research agencies to support practice and policy-making [2]. This influences both the topics that are studied as well as the way that the outcomes are reported.

In software engineering the funding for such studies (when available) is more likely to be from research grants and the choice of topic is apt to be driven by the interests of the researchers involved. Hence systematic reviews in software engineering are more likely to be concerned with identifying research practices, often taking the form of mapping studies [3, 4]. Many also appear to be undertaken to underpin study for a PhD [5], with the focus of the research questions being upon research trends or research practice.

In 2011 we undertook a tertiary study (a systematic review of systematic reviews) to identify how well the systematic reviews then available could be used as a source of material to help inform introductory teaching about software engineering (and hence by implication, could provide useful knowledge to underpin software engineering practice) [6]. For convenience we will refer

---

∗Corresponding Author

Email addresses: `david.budgen@durham.ac.uk` (David Budgen), `o.p.brereton@keele.ac.uk` (Pearl Brereton), `sarah.drummond@durham.ac.uk` (Sarah Drummond), `nikki.williams@cranfield.ac.uk` (Nikki Williams)

[1]The work reported in this paper was undertaken when Nikki Williams was employed by Keele University.