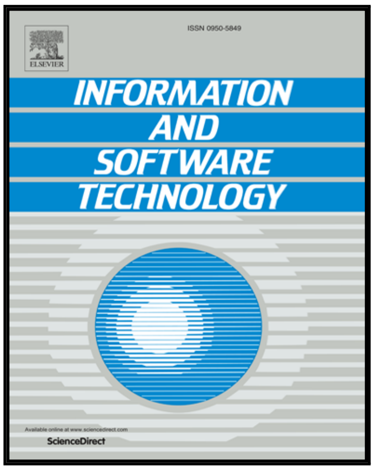
Accepted Manuscript



The contribution that empirical studies performed in industry make to the findings of systematic reviews: A tertiary study

David Budgen, Pearl Brereton, Nikki Williams, Sarah Drummond

PII: S0950-5849(17)30379-8
DOI: 10.1016/j.infsof.2017.10.012
Reference: INFSOF 5899

To appear in: *Information and Software Technology*

Received date: 27 April 2017
Revised date: 11 September 2017
Accepted date: 16 October 2017

Please cite this article as: David Budgen, Pearl Brereton, Nikki Williams, Sarah Drummond, The contribution that empirical studies performed in industry make to the findings of systematic reviews: A tertiary study, *Information and Software Technology* (2017), doi: 10.1016/j.infsof.2017.10.012

The contribution that empirical studies performed in industry make to the findings of systematic reviews: A tertiary study

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

[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: Systematic reviews can provide useful knowledge for software engineering practice, by aggregating and synthesising empirical studies related to a specific topic.

Objective: We sought to assess how far the findings of systematic reviews addressing practice-oriented topics have been derived from empirical studies that were performed in industry or that used industry data.

Method: We drew upon and augmented the data obtained from a tertiary study that performed a systematic review of systematic reviews published in the period up to the end of 2015, seeking to identify those with findings that are relevant for teaching and practice. For the supplementary analysis reported here, we then examined the profiles of the primary studies as reported in each systematic review.

Results: We identified 48 systematic reviews as candidates for further analysis. The many differences that arise between systematic reviews, together with the incompleteness of reporting for these, mean that our counts should be treated as indicative rather than definitive. However, even when allowing for problems of classification, the findings from the majority of these systematic reviews were predominantly derived from using primary studies conducted in industry. There was also an emphasis upon the use of case studies, and a number of the systematic reviews also made some use of weaker 'experience' or even 'opinion' papers.

Conclusions: Primary studies from industry play an important role as inputs to systematic reviews. Using more rigorous industry-based primary studies can give greater authority to the findings of the systematic reviews, and should help with the creation of a corpus of sound empirical data to support evidence-informed decisions.

Keywords:
Systematic review, primary study, industry study, case study

1. Introduction

Knowledge about the effectiveness of established and emerging practices in software engineering can be derived in a number of ways, ranging from using 'expert opinion' through to conducting rigorous empirical studies. Although all have value, it has been argued that the emphasis has too often been on use of the former [1].

In the period since the idea of using secondary studies (systematic reviews) as a source of software engineering knowledge was proposed in 2004 [2], these have become a well established tool for consolidating different sources and forms of study. Terms such as 'evidence-based' or 'evidence-informed' are usually associated with their use. Because a systematic review aggregates and synthesises the findings from many 'primary' studies in an unbiased manner it can be considered as a form of *value multiplier*, in the sense that its findings should carry much greater authority than the outcomes of a single empirical study. Since empirical studies conducted in industry should themselves already carry a certain degree of authority, their use in systematic reviews is particularly important for generating findings that should carry much greater weight than expert opinion. The study described in this paper examines how far primary

*Corresponding Author
Email addresses: `david.budgen@durham.ac.uk` (David Budgen), `o.p.brereton@keele.ac.uk` (Pearl Brereton), `nikki.williams@cranfield.ac.uk` (Nikki Williams), `sarah.drummond@durham.ac.uk` (Sarah Drummond)
[1] The work reported in this paper was undertaken when Nikki Williams was employed by Keele University.

studies conducted in industry do actually contribute to the findings of systematic reviews.

In 2011 we undertook a tertiary study (a systematic review of systematic reviews) to identify how well the information available from published systematic reviews could be used to help inform introductory teaching about software engineering and hence, by implication, should also be suited to informing software engineering practice [3]. In this paper we refer to this as ETS1 (Education Tertiary Study 1). More recently, we have extended and refined this study, and have identified a set of 48 systematic reviews published up to the end of 2015 [4]. We refer to this study as ETS2.

One way in which ETS2 differs from ETS1, apart from the period covered, is that for each systematic review included, we have required that its findings should not only provide knowledge about software engineering, but also that the findings should be supported by some form of *provenance* showing how they were derived, so making it possible to make some assessment of the confidence that can be placed in them. As a result, ETS2 is based upon a core set of 48 systematic reviews that address a range of software engineering practices, and provide conclusions and/or recommendations about practice that are explicitly derived from and supported by ‘primary’ empirical studies.

Since these systematic reviews address topics relevant to practice, rather than research, an obvious question to ask is how far their findings are based upon using primary studies that have been conducted in industry, or have used industry data? In this paper we describe a supplementary analysis of these studies, aimed at addressing the following research question:

> *“For those systematic reviews that address topics relevant to practice and teaching, to what degree are the findings derived from the use of primary studies that have been conducted in an industry context?”*

To answer this, we have interpreted ‘derived’ as being the proportion of primary studies that have been conducted in an industry context. Ideally, what we would really like to know is in what way these primary studies contribute to the individual findings of a systematic review. However, as systematic reviews rarely report upon their analysis or synthesis processes in sufficient detail to determine this, we have had to use proportion as a surrogate measure.

We also need to explain what is meant by ‘industry context’. For this study, we consider this to be where an empirical study (such as a case study) is either performed in an industry setting and/or with participants who are employed in industry; or where the study makes use of industry artifacts in some way.

Inevitably, since the systematic reviews rarely report the characteristics of the primary studies in detail, there are some limitations upon the confidence that we can place upon the counts of primary studies obtained from our analysis.

Despite these limitations, what does emerge very clearly is that, taken as a whole, the findings of this set of 48 systematic reviews are substantially derived from primary studies that have been conducted in an industrial setting, to an extent that we were not really expecting. This highlights the important role that such studies can play in providing well-founded software engineering knowledge, and hence the importance of finding ways to improve their quality. We are also able to make some observations about the forms of empirical studies that have been used as the primary studies.

The rest of this paper is structured as follows. The next section provides a brief background about the roles and use of systematic reviews in software engineering, as well as the role performed by the primary studies. We then describe our research method—and since much of the detail of this is reported elsewhere, we confine our detailed description to the elements specific to this study. Similarly we provide only an outline of the way that the study was *conducted*, placing our main emphasis upon the findings. We then discuss the findings and make observations about how far these appear to have been influenced by empirical studies in industry.

2. Background

The systematic review is now a well-established tool of empirical software engineering, and the book by Kitchenham, Budgen and Brereton describes their use in software engineering, as well as providing an updated set of guidelines for conducting and reporting them [5]. However, although systematic reviewers often comment on the poor quality of reporting provided by the authors of the primary studies, the processes and findings of systematic reviews are not always reported particularly well either [6].

This section provides a brief summary of the forms that systematic reviews can take; followed by a discussion about the sort of knowledge they can provide; and finally outlines some relevant characteristics of the context for primary studies used in software engineering.

2.1. Forms of Systematic Review

A systematic review is classified as a *secondary study*, since it aims to identify all empirical studies relevant to the chosen topic (referred to as the *primary studies*) and to synthesise their results in order to produce its findings. As such therefore, a systematic review does not involve making any direct measurements related to the topic, its role is entirely concerned with aggregation and synthesis of the findings from other studies.

The degree and form of synthesis can vary. Many systematic reviews are less concerned with synthesising the findings of the primary studies and more with categorising their characteristics (such as the type of research question they address), usually using some model or framework. Such studies are referred to as *mapping studies*, and while they can perform a useful role in terms of identifying what aspects of a topic have or have not been studied, the lack of findings means that they do not contribute to the analysis described in this paper. *Tertiary studies* are usually a form of mapping study performed to categorise secondary studies. The underlying study for this paper (ETS2) is a tertiary study, identifying and categorising the secondary studies that address software engineering topics of relevance to teaching and practice.

An obvious question is why systematic reviews are viewed as an important form of empirical study. And in the context of this paper, we might also ask what contribution can they make to improving the practice of conducting studies performed in industry?

To answer the first question, one reason why they are viewed as important is that they are *systematic*, conducted according to a pre-defined plan (the *research protocol*) that is designed to minimise possible bias arising from different factors, including any pre-conceived ideas of the researchers or 'cherry-picking' among primary studies [5]. Another reason is that the process of synthesis should help avoid an over-reliance upon specific studies. All human-centric studies (and most software engineering studies are of this form) can be expected to demonstrate a degree of *variation* in their outcomes, especially (as in software engineering) where the participants may need to be selected on the basis of their skills and experience [7].

For studies performed in industry there are additional sources of possible bias, such as the culture of any organisations concerned. So, synthesising the outcomes from a set of such studies can help with distinguishing those effects that arise from the 'intervention' being studied (such as the use of a test-first strategy) from the effects that are produced by the practices and culture of the host organisation.

The second question is essentially one of motivation, and partly relates to the role of a tertiary study as a mapping study. Identifying how extensively industry-based studies are used in systematic reviews, and the types of study commonly used, can help determine where improvements in the conduct of such primary studies could make a particularly valuable contribution.

2.2. Knowledge provided by systematic reviews

The findings of a systematic review can take a range of forms. In the case of mapping studies, the findings are usually concerned with *categorisation* of the primary studies, and so concentrate upon the research issues addressed by the primary studies, although they may report on other characteristics of these such as the date and venue of publication (to identify trends).

Systematic reviews may also report on other aspects of the primary studies that they have identified, some of which may be related to the *provenance* of the findings. Many perform a quality analysis of the primary studies, usually by employing some form of checklist, seeking to assess how rigorously the primary study was performed.

Where a systematic review seeks to *synthesise* the outcomes of the primary studies, it generally provides a set of findings related to the research topic itself. Ideally it also identifies in what way these are supported by the individual primary studies. Stronger forms of synthesis are also likely to take into account the quality of the findings from individual primary studies, giving greater weight to those possessing higher degrees of rigour [8].

In software engineering, the primary studies can take a range of forms, with case studies and observational forms of study being used quite widely. So the secondary study may well provide information about the form of each primary study, together with additional information such as the number of participants in an experiment or the number of cases used in a case study. However, relatively few reports describing systematic reviews provide clear summaries of such information, and many provide little detail about the primary studies.

In this study we are particularly interested in one of these 'other' aspects of the primary studies, namely in what context, and by whom, the core tasks of the primary studies were performed.

2.3. The primary studies

The types of primary study included in a systematic review will constrain the choice of forms of synthesis that can be employed. Systematic reviews have become a major influence in clinical medicine, where the primary studies usually take the form of randomised controlled trials (RCTs), facilitating the use of statistical meta-analysis for their synthesis. While controlled experiments and quasi-experiments provide the nearest equivalent to an RCT in software engineering, the involvement of human skill complicates the use of meta-analysis.

Case studies, usually based on the positivist approach advocated by Robert K Yin have become much more widely used in recent years, particularly for studies that are based in an industrial setting where experimentation would be inappropriate [9, 10]. A consequence is that many systematic reviews use less rigorous and non-statistical forms of synthesis. Sometimes they also use a form of synthesis that is weaker than others that might possibly have been employed [8].

A relevant factor here is the *context* in which such such studies are performed. Although the affiliation of the researchers is one element of this, other significant contributions to the 'context' can include the following.

- The nature of any *source material* used, which can include such things as specifications, design material, test cases and code. These can be related to 'toy' problems, widely accepted 'standard' datasets, and large-scale systems.
- The choice of the *participants*, particularly for experiments or surveys. A simple categorisation often used for describing these is as either 'student' or 'practitioner'. However the category of 'student' can cover a wide range from inexperienced undergraduates to (say) part-time postgraduate students who have at least five years experience in industry. And the extent to which students can act as surrogates for practitioners will also be partly dependent upon the topic [11].
- The *setting* in which the study is performed, which may be an academic 'laboratory' environment through to forming an ancillary activity within an industrial organisation.

As a very simple generalisation, experiments and quasi-experiments are often used to study technical issues, and are performed with both students and practitioners as participants; while case studies are largely undertaken to study practice.

3. Research Method

Since the analysis presented in this paper draws upon the data collected for a tertiary study (ETS2) for much of its material, we have not attempted to discuss the complete study design in this section. Instead, we have focused upon providing a description of the searching and inclusion/exclusion steps, as they explain how we selected our source material. We have omitted much of the detail about issues such as quality assessment and data extraction, which are described in [4], although we have described the additional data extraction performed to support our analysis.

3.1. Searching for systematic reviews

Our analysis is based upon the set of 48 systematic reviews used in ETS2. These have been identified using two different procedures, depending on the period when the review was published. For ease of reference in the rest of this paper, we have labelled these as *Source-set1* and *Source-set2*. (However, we should emphasise that all of the review procedures, such as inclusion/exclusion were performed in the same way for all of the systematic reviews in the two sets.)

- *Source-set1*. This consisted of the 120 reviews found in the three *broad* tertiary studies [12, 13, 14] that covered the period up to the end of 2009. These were performed in the early period for conducting systematic reviews, and used a mix of manual and electronic searching to achieve a comprehensive degree of coverage of known reviews for that period.
- *Source-set2*. With the rapidly-growing use of systematic reviews, performing broad tertiary studies that identified and included all published systematic reviews was recognised as becoming both too large a task, as well as one likely to be of diminishing value. So for the period January 2010 to December 2015, we confined our searching to five major software engineering journals that published systematic reviews. Our rationale for doing so was that these provided good sources of systematic reviews in software engineering, while we had also observed that many systematic reviews published in conferences were mapping studies. (And those that were not were likely to be published in an extended form in a journal.) We were also concerned that any material found should be readily accessible to teachers and practitioners, which was a further reason for confining our searching to a set of well-known journals.

Our sources are summarised in Table 1.

3.2. The inclusion/exclusion criteria

The selection of candidate systematic reviews was based upon a two-stage process. In the first stage, randomly assigned pairs of authors performed an initial

Table 1: Details of the sources used

Period	Sources
2004-2009 (*Source-set1*)	Tertiary Study 1 [12] Tertiary Study 2 [13] Tertiary Study 3 [14]
2010-2015 (*Source-set2*)	IEEE Transactions on S/W Eng. Empirical Software Engineering Information & Software Technology Journal of Systems & Software Software Practice & Experience

selection, based mainly on the topic of the systematic review and its suitability for teaching. In the second stage, again working in pairs, we performed more detailed data extraction and quality assessments. During this, we excluded a candidate systematic review if closer inspection showed that it did not adequately meet the inclusion/exclusion criteria as discussed below.

The inclusion/exclusion criteria used for ETS2 are summarised in Table 2. A major difference between ETS1 and ETS2 is represented by criteria I1 and I3.

Table 2: Inclusion and Exclusion Criteria

Inclusion Criteria
I1. The paper is published in a journal, and either included in the three broad tertiary studies, or one of the five journals in the appropriate periods. I2. The topic of the paper is appropriate for introductory teaching of SE I3. The paper contains conclusions or recommendations relevant to teaching and explicitly supported by the outcomes.
Exclusion Criteria
E1. Systematic reviews addressing research trends. E2. Systematic reviews addressing research methodological issues. E3. Mapping studies with no synthesis of data. E4. Systematic reviews that address topics not considered relevant to introductory teaching of SE.

For criterion I1, we decided to restrict our study to use only journal papers for both periods (in ETS1, we also included conference papers from *Source-set1*). This was on the basis that these were not artificially constrained in length when presenting their results and would therefore be more comprehensive and useful (and more readily accessible).

For criterion I2, we determined suitability on the basis of the fit of a topic to those covered in the SEEK (Software Engineering Education Knowledge) included in the 2014 revision to the ACM/IEEE curriculum guidelines for software engineering programmes [15].

The use of criterion I3 formed a more significant constraint than was used in ETS1. We now required that a systematic review not only addressed a topic relevant to teaching and practice, but that it also had useful findings that were relevant to that topic. We also required that there be some form of *provenance* for these in terms of links between the study data and the outcomes.

We differentiated between *conclusions* and *recommendations* chiefly on the basis of the degree of provenance provided in the report of the systematic review.

- A *conclusion* presents knowledge about the review topic that a teacher or student (or practitioner) could use to aid their understanding.

- A *recommendation* is essentially a conclusion that has a degree of confidence associated with it that means that it could help when making decisions about practice.

Whenever possible, we also consulted the original authors to confirm that we had extracted these correctly.

3.3. Extracting industry-related profiles

The data extracted from each systematic review when conducting ETS2 is summarised in Table 3. Wherever possible, the details were accompanied by notes about where the information was to be found. (Note: the DARE criteria[2]—Database of Attributes or Reviews of Effects—are a widely used five-point scheme for assessing how well a systematic review was performed)

Table 3: Data extracted for the main tertiary study (ETS2)

	Form of data extracted
1.	Bibliographical information
2.	Quality scores (based on the DARE criteria)
3.	Details of any quality assessment performed on the primary studies
4.	Details of the 'body of evidence' (number and types of primary study)
5.	Material associated with the body of evidence (search period, search engines etc.)
6.	Any conclusions that are reported or can be derived
7.	Any recommendations that are reported or could be derived

[2]http://www.crd.york.ac.uk/CRDWeb/AboutPage.asp

For the purpose of this analysis, we performed some additional data extraction based on assessing the degree to which industry-related studies were employed in each systematic review.

Based on a small pilot exercise with five of the systematic reviews, we sought details of the number and type of each primary study used in a review, categorised by two factors: the *setting* where the study was performed (academic or industry); and the *participants* who were involved (academic or industry). Since not all systematic reviews are human-centric, where a review had no participants we sought details of the *research material* used in the study (academic or industry).

Again, we recorded the details of where information about these characteristics could be found in the report of the review.

4. Conduct of the study

We first discuss how we selected systematic reviews from each of our two source sets. Figure 1 provides a summary of how this was organised. We then discuss the details of the supplementary data extracted for this analysis

4.1. The three tertiary studies: Source-set1

In conducting ETS1, we performed an initial selection process on the studies included in the three broad tertiary reviews to determine which ones could potentially provide information for teaching and practice. So for this study, we began with the set of studies selected for ETS1 and re-assessed them using the more comprehensive inclusion/exclusion criteria for ETS2. We performed this task using random pairings of the authors for each study, resolving any differences by discussion.

The total number of secondary studies included in the three tertiary studies was 120 (20 + 33 + 67 respectively). We had selected 43 of these for ETS1 and so for ETS2 we used these as our 'baseline' set. We first excluded the ones published in conference proceedings, leaving 18 journal papers, after which our inclusion/exclusion procedures left us with 12 studies. Since two of these papers (those with index values #54 and #118) used the same data, we actually had 11 systematic reviews that required additional data extraction.

Table 4 provides a summary of these systematic reviews. We refer to this as *Dataset1*. For each review, we provide the index value assigned as part of this study, the citation, the period covered by the review (where known), and a brief summary of the topic of the review (usually condensed from the title). We then give the counts for the four categories of primary study we used for our analysis (these are discussed in Section 4.3) as far as we were able to extract these.

4.2. The journal searches: Source-set2

For the five journals, we undertook a manual search of the contents pages for issues published over the period 2010-2015, examining titles and abstracts. This was performed by one of the authors (DB). Since not all systematic reviews necessarily have indicative terms in their titles, this was supported by an electronic search that was performed by an independent reviewer. The latter was performed in April 2016, and the details of this are provided in [6]. Together these resulted in 156 systematic reviews, to which we added two further ones from other journals that had been recommended by researchers, giving a total of 158 systematic reviews in *Source-set2*.

Although some of these had been used in ETS1, we decided that it would be better to treat the whole period covered by Source-set2 in a consistent manner. So all of the systematic reviews included in *Source-set2* were assessed for relevance using the same procedures.

Once again, we employed a two-phase process for inclusion/exclusion in which pairs of reviewers first performed an initial filter to determine potential relevance using the inclusion/exclusion criteria, followed by an in-depth data extraction. The pairings were organised on a random basis, apart from where one of us (DB or PB) was one of the authors of a systematic review, which then had to be assessed by other members of the team. If the reviewers disagreed in the first phase, the paper was included in the second phase, while in the second phase any differences were resolved by discussion between the team members.

The first phase of this resulted in 74 candidate papers, following which we performed the process of data extraction, which also involved determining whether the paper contained suitable conclusions or recommendations. This resulted in our excluding a further 37 studies on the basis either that we could not identify usable conclusions or recommendations, or that we were unable to identify explicit links between the data presented in the review and any conclusions provided. This left a total of 37 systematic reviews that we refer to as *Dataset2*.

Table 5 provides a summary of the 37 systematic reviews making up *Dataset2*. This uses the same format as Table 4.

As a further consistency check we contacted the lead authors of all of the 48 systematic reviews included in the two datasets and asked them to check our interpretation of the outcomes in terms of the conclusions and

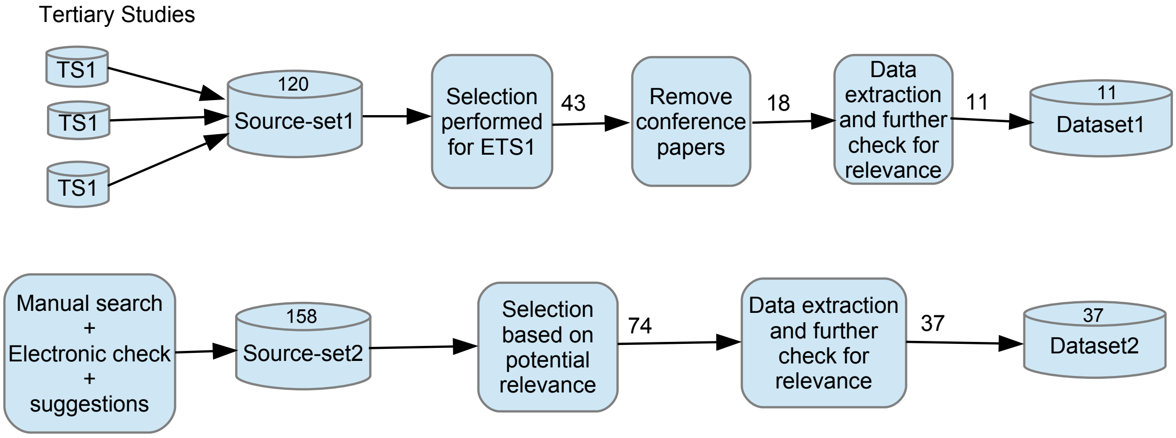


Figure 1: Schematic description of the study selection process

Table 4: Details of the systematic reviews included in this study: Dataset1 (2004-2009)

Index	Ref.	Period covered	Topic	Primary Study Counts			
				expl. ind.	impl. ind.	acad.	not stated
8	[16]	to 2006	Estimation of s/w development work effort	14		2	
15	[17]	1992-2002	Capture-recapture in s/w inspections	1			24
22	[18]	unclear	Assessment of development cost uncertainty				40
39	[19]	1994-2005	Benefits of software reuse	11			
50	[20]	1996-3/2006	SPI in small & medium s/w enterprises	45			
52	[21]	unclear	Motivations for adopting CMM-based SPI	49			
54	[22]	1980-6/2006	Motivation in software engineering				79
66	[23]	1996-2007	Search-based non-functional testing	17		18	
82	[24]	1969-2006	Regression test selection techniques	4			32
84	[25]	to 2007	Effectiveness of pair programming	5		14	
102	[26]	1995-2005	Managing risks in distributed s/w projects		72		

recommendations. We heard from 25 of these authors, none of whom suggested other than minor changes to wording.

4.3. *The data extraction process*

For the rest of this paper, unless otherwise specified, our analysis applies to the combined set of 48 systematic reviews from the two datasets.

Here we confine our description to the processes involved in the additional data extraction performed for this analysis, fuller details of the main data extraction are provided in [4]. As indicated in the previous section, we based this additional extraction around a model that used the concepts of *setting* and *participant* to categorise the primary studies included in each systematic review. While data extraction for ETS2 was performed using random pairings of team members, to ensure greater consistency of interpretation, the additional data was extracted by two of us (PB and DB), resolving any differences by discussion.

Few reports of systematic reviews provided clear and explicit details about these characteristics of the primary studies. Indeed, some provided little more than a list of references to the primary studies. Even where quite extensive details were provided, these were not necessarily 'joined up'. So for example, we might be able to identify how many primary studies were case studies, and how many primary studies took place in industry, but not be able to determine how many of the case studies were performed in industry. In many ways this is quite understandable—since the authors of the reports had no reason to anticipate that this question might be asked

Table 5: Details of the systematic reviews included in this study: Dataset2 (2010-2015)

Index	Ref.	Period covered	Topic	Primary Study Counts			
				expl. ind.	impl. ind.	acad.	not stated
121	[27]	2000-2007	Evidence in global software engineering	37		16	3
123	[28]	unclear	Domain analysis tools		7	12	
124	[29]	1970-2007	Characterising s/w architecture changes				130
126	[30]	1989-2006	Does the TAM predict actual use?				79
130	[31]	1997-2008	Evidence for aspect-oriented programming		6	16	
134	[32]	to 3/2005	Empirical studies on elicitation techniques	7		7	18
135	[33]	1980-2008	Antecedents to personnel's intention to leave		72		
138	[34]	to 2009	Measuring & predicting software productivity	25		13	
146	[35]	2000-2010	Dependency analysis solutions	38			27
150	[36]	to 6/2010	Agile product line engineering	14			25
154	[37]	1995-2009	Effectiveness of s/w design patterns	11		7	
155	[38]	2000-2010	Fault prediction performance	35		1	
157	[39]	to 2/2011	Effects of Test-Driven Development	10		23	4
160	[40]	to 4/2009	Reconciling s/w development methods		42		
161	[41]	1993-2011	Identifying stakeholders for req. elicitation		42		
167	[42]	2006-2011	Evaluating commercial cloud services		82		
174	[43]	unclear	Industrial use of s/w process simulation		87		
175	[44]	to mid-2008	Barriers to selecting outsourcing vendors	77			
193	[45]	to 7/2010	Using social software for global s/w dev.	61		23	
197	[46]	to 10/2011	Software fault prediction metrics	81		25	
205	[47]	2000-2011	Test-Driven Development	22		19	
215	[48]	to 12/2013	S/W development in start-up companies		30		13
217	[49]	1997-2011	Influence of user participation on success		82		
219	[50]	to 2012	Linking OO measures and quality attributes		33	5	61
222	[51]	1990-2012	The Kanban approach		37		
228	[52]	1997-1/2008	Lightweight software process assessment	22			
236	[53]	2001-2013	Impact of global team dispersion	40		3	
239	[54]	to 2011	Using CMMI with agile development	59		1	
241	[55]	1980-2012	User-involvement and system success		87		
244	[56]	1990-2012	Analogy-based development effort estimation	61			
246	[57]	2003-4/2013	Barriers to newcomers on OSS projects		20		
249	[58]	2002-10/2012	User-centred agile development		26		57
252	[59]	2002-2013	Metrics in Agile/Lean development	30			
259	[60]	1992-2/2014	Use case specifications research	27		11	81
260	[61]	to 5/2015	Use of SE practices in science			43	
268	[62]	1996-2/2008	Requirements for product derivation support				118
276	[63]	1996-10/2013	Decision support model for adopting SPL		31		

(an issue that we return to later). Even so, as we have observed in our study of reporting quality, the reporting of systematic reviews in software engineering is apt to be of rather mixed quality and completeness [6].

In some cases we were able to infer that primary studies were very likely to have been performed in industry, usually because of the topic of the systematic review. Overall though, we were often unable to identify any information that would allow us to categorise the studies using our model. This was not always a matter of reporting—some systematic reviews are not human-centric, so there is no concept of a human participant. Where this occurred, we tried to use substitutes such as the source material that could be considered to provide industrial or academic participation for the study. In Tables 4 and 5, we have therefore reported our findings using the following four categories.

- Studies that *explicitly* involve industrial participation (or material) in some form, clearly reported as such in the report of the systematic review.
- Studies that *implicitly* involve industrial participation (or material), where this could be determined with a reasonable degree of confidence, either because of the topic of the review, or on the basis of comments made by the authors.
- Those studies that were clearly identified as having been performed in an *academic* setting, usually using student subjects or 'toy' problems (or both).
- The studies that we were unable to classify, either because no details were given, or because we had no means of determining how the primary studies were distributed across each category.

The variety of topics, reporting style, and levels of detail provided, meant that the additional data extraction we performed required some discussion for nearly every paper, usually to resolve differences of interpretation when assigning them to categories. However, for the purposes of this paper, although the categorisation described above is a less detailed one than we originally hoped would be possible, it does allow us to draw some useful conclusions.

5. The contribution from industry studies

In this section we provide some analysis and further interpretations related to the contributions that industry studies make to the 48 systematic reviews. In particular we look at the proportion of primary studies that have been performed in industry or used industry data; the types of industry studies they included; and how far these systematic reviews might have included the use of weaker and less rigorous forms of primary study such as 'experience' or 'opinion' papers.

Before doing so we should make some observations about the available data and possible ways that it might have introduced error and bias our analysis.

- It is possible that some of the primary studies might be used in more than one systematic review, particularly for topics such as agile methods, estimation, and testing where several systematic reviews cover different aspects. This overlap is likely to be a relatively small effect as no topics have many related systematic reviews.
- We have had to make many interpretations of the data reported as being used in the systematic reviews, and in doing so, have had to assume that different teams of systematic reviewers are using terms such as 'case study' to mean the same thing. We have tried to do this in as consistent manner as possible.
- We have tried to provide the counts for *empirical* studies wherever possible, as some systematic reviews do include quite a wide range of less rigorous study types, which might include 'opinion', 'observational' and 'theory' papers. Unfortunately, these are not always clearly distinguished from the more rigorous forms.
- We have included *experience reports* with the empirical studies where there was an indication that these were derived from experiences incurred in an industry setting.
- Where the participants in a study are students and nothing is said about the researchers, we have assumed that these are academics.

What these do mean is that there is inevitably some degree of uncertainty about the 'true' value of many of the counts. So these should be viewed as being *indicative* rather than *definitive*.

5.1. The overall profile for industry studies

As a first element in the answer to our research question, we consider the overall proportion of studies that are associated with some form of industry context.

If we look at Tables 4 and 5 we can see that there is a clear preponderance of primary studies that we were able to identify as having been performed in an industry context or using industry data. The variation between secondary studies and their use of different inclusion/exclusion criteria, suggests that totalling the primary studies in each category is unlikely to be a very reliable measure, and so as a better way of gauging impact, we have looked at frequency.

In Table 4, using the definitions of our categories provided in Section 4.3, there are eight studies from 11 (82%) that explicitly or implicitly make use of industry studies, (ignoring the one study categorised for systematic review #15). For five of these, industry studies are the predominant form used. We were unable to categorise the primary studies used in three reviews, with the exception of the one study from #15. The proportion using academic studies is quite low (three from 11, or 27%). If we assume that similar proportions occur for the 'not stated' studies then we can reasonably conclude that most of the findings from these reviews are likely to be largely based upon primary studies performed in industry.

For Table 5 the proportion of systematic reviews that are clearly using industry studies is even higher, 33 from 37 (89%), so that taken together we have 41 from 48 (88%) of our reviews for which we can say that the provenance of the findings is likely to be at least in part based upon primary studies conducted in industry. For 18 of these, all of the findings are likely to be based upon industry data. Only one study (#260) is completely based upon academic primary studies, and here the topic of the review is such that this can be considered as being appropriate.

There are also five reviews where we lack enough information to categorise any of the primary studies with any confidence. It is noticeable that all were performed during a relatively early period for the use of systematic reviews. This suggests that systematic reviewers increasingly consider that providing at least some degree of categorisation for the primary studies used can create a useful element of provenance for their findings.

The basis on which the set of 48 systematic reviews was selected is obviously favourable to the use of industry-based primary studies. Even so, looking at the individual ratios of industry/academic primary studies, there is clearly a marked emphasis upon the use of industry studies.

5.2. *The types of empirical study used in industry*

We now examine the types of primary study used in an industry setting, concentrating upon those that employed experiments and case studies.

Table 6: Distribution of study types where known

Review #	**Industry Studies**			**Academic Studies**		
	Case Study	Expt.	Other	Case Study	Expt.	Other
8		4	10		1	1
39	7		4			
50	45					
52		1	48			
66		17			18	
84		5			14	
130				9	7	6
134	1	5	1		7	
138	10		15	2		11
150	14	>1	>2			
154		5	6		7	
157		10			23	
175	26		51			
205	13	6	3		19	
219	11	1	21		5	
228	22					
239	15		44			1
241	20	11	56			
246	20	2				
249	17		>9			
252	21		9			
259	27			11		

We were able to extract figures for the types of primary study from 22 (46%) of the 48 systematic reviews, although we were not always able to categorise all of the primary studies used in a review.

Table 6 shows the data we were able to obtain. A review is only included if we were able to categorise at least one of its primary studies as a case study or an experiment. For the primary studies that were based in industry, the case study was clearly (and perhaps not unexpectedly) the form most frequently used. When both case studies and experiments were used in a review, the proportion of case studies was usually much higher. And for this group, the 'other' category did include a number of surveys (see next subsection), again as might be expected when eliciting expertise from practitioners.

For studies based in academia, case studies were used relatively infrequently and the most common form used for these was some form of experiment (again, the term is often used rather loosely). In an academic context this proportion is perhaps not very surprising.

5.3. *The 'other' studies*

Table 6 shows a predominance of case studies being used as the study type for the industry studies, but there are two groups of studies that should be considered a little further. The first is those listed as 'other' in the table, the second is those that are not included at all because we know little about them.

For the first group the most notable thing about the 'industry other' column is how often there are more studies listed there than in the other two columns (7 studies from 21). So to clarify this further, we drilled down into this group. Table 7 provides a fuller picture for these.

Table 7: The 'other' studies where known

Review #	Total other	Survey	Exper. Report	Remainder
8	10			5 field studies + 5 mixed forms
39	4		3	1 example application
52	48	2	45	1 interview
134	1			1 'non-standard design'
138	15	6		A range of modelling forms
150	>2		>2	
154	6		6	
175	51	15	15	11 interviews + 10 'other'
205	3	3		
219	21			21 'historical data'
239	44	6	37	1 action research
241	56	46	1	7 field studies, 1 action research, 1 grounded theory
249	>9		unspec.	4 ethnographic studies, 3 interviews, 2 action research
252	9	2	7	

As this shows, we can explicitly identify the use of experience reports in 9 of the 22 systematic reviews listed in Table 6, although they were only used extensively in 3 systematic reviews (#52, #175 and #239) The rest of the primary studies include something of a medley of forms.

For the second group, there is relatively little that we can report. For most of the other systematic reviews, we could not determine the types of study used in any detail, or could not match study types to setting. For two of them (#215 and #217), although there was no explicit use of case studies or experiments, so that they were not included in Table 6, there were large numbers of surveys (#217) and 'evaluation research' studies (#215). Also, we should note that in the case of the 22 systematic reviews analysed above, several had a total number of studies that was greater than those we were able to classify.

So what we can say is that there is evidence of quite explicit use of forms such as 'experience reports' and 'opinion papers' within these systematic reviews, although these were only predominant in two systematic reviews (#52 and #239). Obviously, we don't know the details of how these were used—experience reports can provide a useful form of triangulation on occasion—but their inclusion suggests that systematic reviewers may have found themselves short of good empirical material.

6. Discussion

We first explain how this study (that we have labelled as STS2, for Supplementary Tertiary Study 2) relates to the other analyses we have undertaken. We then consider the limitations upon our findings that are implicit from the organisation and conduct of this study, since these have implications for any further discussion. After that, we consider what our findings about the empirical studies conducted in industry tell us about their contribution to any outcomes from the systematic reviews, what this might indicate about the maturity of the use of the systematic review as a research tool, and how these might co-evolve in the future.

6.1. Relationship to other analyses

Figure 2 shows an abstract summary of the relationships between our educational and supplementary tertiary studies.

Stemming from our original tertiary study (ETS1), we have performed three related analyses.

- **ETS2** has extended the original tertiary study, both in terms of the period covered, and also by the use of stricter inclusion criteria (as described in Section 3.2). The motivation for this study was to identify sound empirical findings that might be used to inform practice and teaching.

- **STS1** used part of the dataset from ETS2 (37 systematic reviews published in the period 2010–2015) and analysed the rigour and 'completeness' of reporting for these. The motivation was to identify guidelines and lessons about how to report the procedures and findings of a systematic review, as in conducting ETS2, we had often found that key information about reviews was missing or unclear.

- **STS2** (as reported in this paper) has analysed the 48 systematic reviews used in ETS2 to determine how extensively industry-based primary studies

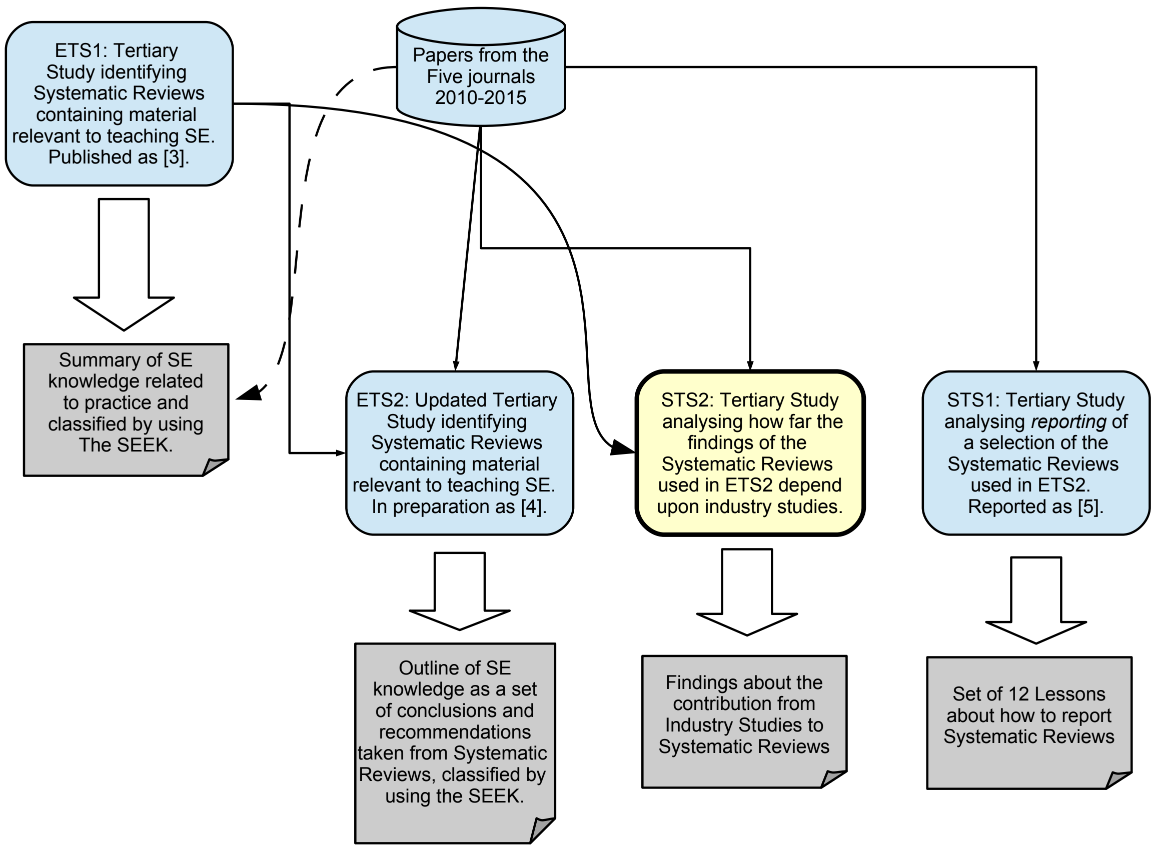


Figure 2: Relationships between the set of tertiary studies

contributed to their findings, and what types of study were used. The motivation for this study was to determine how far these systematic reviews addressing ‘practice’ topics had findings that stemmed from industry studies, and hence how authoritative they could be considered to be.

Together, these form a comprehensive analysis of a carefully selected set of systematic reviews and provide a ‘state of the art’ picture of how evidence-based studies have progressed in software engineering. Their findings should demonstrate how far sound empirical evidence is available in software engineering, and also help to motivate researchers to improve their practices where appropriate.

6.2. Limitations

We can identify a number of potential limitations that stem from the way that we conducted both our tertiary study and also the analysis of industry-related primary studies that we present in this paper.

- The way that we selected our secondary studies. We did not attempt to find all of the systematic reviews that were published during the period covered by this tertiary review. However, we might usefully note that of the 11 studies included in Dataset1 (drawn from the three broad tertiary studies) eight were from the set of journals we used for Dataset2, and the other three were from more specialist journals.
- In common with many of the systematic reviews analysed here, we have used a very broad interpretation of ‘empirical’ in our study, both when considering the primary studies (especially the non-human-centric forms), and also in selecting the systematic reviews. The latter was mainly driven by our interest in finding material for teaching, and so we did include systematic reviews that were more concerned with evaluation than with synthesis.
- The process of data extraction. Our concern here is

with the additional data extracted as part of STS2, since where we have used data from ETS2, this has mostly been relatively objective contextual material. The main issue is, of course, one of determining whether specific primary studies were conducted in an industry context. A problem for any tertiary study is that it is necessary to interpret such knowledge by using the information provided in the secondary studies, and in this case, the information is often provided indirectly. Even where secondary studies do provide quite detailed information about the characteristics of the primary studies, few provide much about this particular aspect. As a result, most studies have required fairly detailed analysis supported by extensive discussion to reach agreement on an appropriate interpretation. Where there was any doubt about classification, we made use of the 'other' category.

- Analysis of the profile of industry studies. As discussed earlier, many of our counts are indicative rather than definitive, simply because each systematic review uses its own set of inclusion criteria as well as slightly different interpretations of terms such as 'empirical' or 'case study'.

Overall we do not consider that any of these factors are likely to invalidate our analysis in any way. Their main influence is providing a degree of measurement uncertainty for the values obtained from the analysis.

6.3. *The contribution from industry studies*

The predominance of randomised controlled trials (RCTs) in clinical medicine, where this form is considered as a 'gold standard' for empirical studies, has probably influenced expectations in the software engineering community, including our own. Where RCTs (as well as experiments and quasi-experiments as used in software engineering) are available as primary studies, this makes it possible to perform a meta-analysis for their synthesis. As a result, discussion of evidence-based procedures tends to place emphasis upon such forms of primary study, since synthesis through quantitative procedures such as meta-analysis or vote-counting can potentially result in more definitive findings.

However, this does not seem to be the actual situation for software engineering, at least, as regards the systematic reviews that we analysed. Our analysis of reporting [6] showed that only two of the 37 systematic reviews in Dataset2 (#157 and #217) used meta-analysis for synthesis (although an element of vote-counting was used in another nine). For this analysis, our findings show that the use of case studies is widespread, indeed, that most systematic reviews are something of a 'mixed economy', and hence tend to use qualitative synthesis forms to aggregate the results from a range of primary study types.

What is clear from Tables 4 and 5 is that, regardless of study type used, there is a strong predominance of primary studies that have been conducted in some form of industry context. While the use of some of the associated study types might present challenges for synthesis, this does mean that the findings from these systematic reviews have more authority than purely academic studies, and hence should be particularly relevant to industry as well as teaching.

This variation does mean that these systematic reviews rarely have very strong findings (even the findings from #84—a review that uses a meta-analysis to analyse a set of experiments—are constrained by the wide variation in the research questions used by the primary studies). Equally, we should note that the effect sizes resulting from the use of software engineering techniques tend to be relatively small. The context in which a technique is employed may also be an important factor in determining its effectiveness—so another benefit of a predominance of industry studies is that they are likely to help identify contexts that are relevant to practitioners. Hence the value of a systematic review lies mainly in identifying where, and under what conditions, the use of a technique or tool may be particularly effective. However, that is something that Brooks long ago pointed out, silver bullets in the form of techniques that 'always' confer a benefit for the user simply don't exist in software engineering [64].

6.4. *How can empirical evidence influence practice & teaching?*

The use of systematic reviews can be considered as an 'innovation' in terms of research practice. As we observed at the start of the paper, the innovatory aspect for systematic reviews can be viewed as being their role as a 'value multiplier'—strengthening the provenance and enhancing the impact of the findings from individual empirical studies. The process and mechanisms by which innovations are *diffused* successfully within a community (or fail to diffuse) have been widely studied [65]. Indeed, the vocabulary describing the major categories of "diffusion of innovation" (innovators; early adopters; early majority; late majority; laggards) is in relatively common use.

In software engineering we suggest that it is possible to identify two quite distinct but related innovation cycles with regard to evidence-based concepts, for which the major tool is the systematic review. The first of these is in the research community where we can identify a number of 'innovators', most notably Kitchenham, Dybå and Jørgensen who wrote the foundational paper [2]. Over the following decade, the empirical software engineering community have formed the category of 'early adopters', producing hundreds of publications, some of which were included in this review. As a number of researchers in other areas of software engineering begin to incorporate systematic reviews into their toolset, we are now starting to see the emergence of the 'early majority'.

The second cycle is centred upon *users* of evidence-based findings. Here we are probably still barely in the 'innovator' phase—which of course can only begin when a suitable mass of useful evidence-based findings are available. Our tertiary study suggests that such a mass is becoming available, although the findings are often not presented in a manner that is relevant for practitioners[3].

It is worth observing that when Barends and Briner examined the experiences of evidence-based medicine to help them understand the challenges faced by evidence-based management, they identified a number of factors that may also be relevant to software engineering [66]. These included the following.

- For clinical medicine, evidence-based practice originated as a *teaching method* in the early 1990's.
- There was an available base of material to support evidence-based teaching derived from existing systematic reviews that had been performed over the previous decade.
- Medicine had already accepted the value of using empirical studies (largely in the form of randomised controlled trials), and also, evidence-based practice "came along at a time when medicine was getting challenged in a way and losing its authority to some extent". Evidence-based practice therefore offered a means to retain that authority, based strongly upon the provenance of the findings from systematic reviews.
- Medical practice has a strong professional ethos and regulatory system, so once evidence-based medicine was included in the professional exams it gained much greater influence.

There is some resonance between the challenges faced by both management and software engineering, while some aspects are clearly different to those occurring for medicine. However, the need to find ways to challenge and overcome the inertia created by 'practitioner belief' is clearly a common one [67].

So, if we look at the four success factors identified above and consider how well these are met by the current state of software engineering research and practice, we can conclude the following.

- Teaching of software engineering is currently far from being evidence-informed, both in terms of readily-available material, or of its use. How to achieve this is an open research question and an important one, since many tools and techniques now used on an everyday basis in industry permeated there from the young staff who had learned about them as students.
- A base of useful material is emerging, and (good) industry studies are an important element in underpinning this. Our analysis demonstrates the important role that such studies have in systematic reviews, while at the same time, their use of weaker study types suggests that more and more rigorous studies are needed.
- Software engineering as a discipline is beginning to acknowledge the role of empirical studies in helping assess what works and when, and this acceptance is likely to increase if more studies are seen as being based in industry. However, education of students is probably going to be the important motivator here.
- The software engineering discipline lacks a professional regulatory context, but the professional bodies do often provide accreditation of university degrees, and can play a useful role in encouraging a more evidence-informed approach to teaching and practice.

A fuller discussion of these issues belongs elsewhere. For this paper the above arguments help to reinforce the view that software engineering needs more (and better) systematic reviews; better presentation of outcomes; and provenance that may help practitioners to accept the findings. To achieve the last of these requires the underpinning of sound and relevant primary studies—that is, conducted in a realistic industry context.

[3] There is a useful suggestion in [68] to present findings in a "1-3-25 format: one page of take-home messages; a three-page executive summary, and a 25-page report [69]". Where journals publish systematic reviews, they could well require that the researchers provided all three as a condition of acceptance, to assist with dissemination.

7. Conclusions

In terms of the purpose of our study, and its research question, we suggest that there are several conclusions that we can draw about the use of empirical studies in industry in systematic reviews.

1. With regard to our original research question, then from our set of 48 systematic reviews, selected on the basis of having findings that are relevant to teaching and practice, 41 of the reviews demonstrated a clear predominance of the use of industry-based primary studies, insofar as we could categorise these. For 18 of the studies, *all* of the findings stem from the use of industry-based primary studies. We can therefore conclude that empirical studies from industry make a large contribution to the findings of such systematic reviews.
2. For the primary studies we can identify as having been conducted in an industry setting, the positivist case study plays an important role. This would therefore argue that finding ways to facilitate and conduct such studies as rigorously as possible is important to software engineering as a discipline.
3. Some systematic reviews are including weaker forms of industry-based primary study such as experience reports. The full scale of this has to remain a matter for conjecture, but this is clearly undesirable, and reinforces the previous conclusion about needing more rigorous studies. We should also observe that in classifying some studies as case studies, we may well be doing so on the basis of rather imprecise descriptions, and so our figures for case studies may well include some that should be more correctly classified as experience reports.

From these we can conclude that primary studies conducted in industry play an important role in evidence-based software engineering. Also, greater rigour in their conduct is required in order to provide systematic reviews with the provenance needed to support evidence-informed decision making.

There are lessons for the ways that systematic reviews are conducted and reported too, particularly regarding demonstrating provenance for findings. And the associated issue of dissemination is an important one. We have noted that for clinical medicine, the development of an evidence-based approach to teaching was an important precursor for practice, but if we are to do the same for software engineering we will need a corpus of material that is sound and also well reported. We would particularly recommend that journals should require that all systematic reviews they publish are accompanied by a short summary of the findings (and their provenance) written for practitioners and researchers.

Acknowledgements

Our thanks to Professor Barbara Kitchenham for advice and observations, as well as for conducting independent electronic searching for publications that we might have missed. Our thanks also to the authors of the systematic reviews we studied, many of whom were good enough to check the accuracy of our extracted conclusions and to pass comment on these. Finally, our thanks to the anonymous reviewers who made some very helpful suggestions for improving this paper.

References

[1] B. Kitchenham, D. Budgen, P. Brereton, M. Turner, S. Charters, S. Linkman, Large-Scale Software Engineering Questions–Expert Opinion or Empirical Evidence?, IET Software 1 (2007) 161–171.

[2] B. Kitchenham, T. Dybå, M. Jørgensen, Evidence-based software engineering, in: Proceedings of ICSE 2004, IEEE Computer Society Press, 2004, pp. 273–281.

[3] D. Budgen, S. Drummond, P. Brereton, N. Holland, What scope is there for adopting evidence-informed teaching in software engineering?, in: Proceedings of 34th International Conference on Software Engineering (ICSE 2012), IEEE Computer Society Press, 2012, pp. 1205–1214.

[4] D. Budgen, P. Brereton, N. Williams, S. Drummond, A tertiary review of evidence about software engineering practice, 2017. Paper in preparation.

[5] B. A. Kitchenham, D. Budgen, P. Brereton, Evidence-Based Software Engineering and Systematic Reviews, Innovations in Software Engineering and Software Development, CRC Press, 2015.

[6] D. Budgen, P. Brereton, S. Drummond, N. Williams, Reporting systematic reviews: Some lessons from a tertiary study, Submitted for publication, 2017.

[7] D. Budgen, Aggregating empirical evidence for more trustworthy decisions, in: T. Menzies, L. Williams, T. Zimmerman (Eds.), Perspectives on Data Science for Software Engineering, Morgan Kaufman, 2016, pp. 181–186.

[8] D. S. Cruzes, T. Dybå, Research synthesis in software engineering: A tertiary study, Information and Software Technology 53 (2011) 440 – 455.

[9] R. K. Yin, Case Study Research: Design & Methods, Sage Publications Ltd, 5th edition, 2014.

[10] P. Runeson, M. Höst, A. Rainer, B. Regnell, Case Study Research in Software Engineering: Guidelines and Examples, Wiley, 2012.

[11] D. Sjøberg, J. Hannay, O. Hansen, V. Kampenes, A. Karahasanović, N.-K. Liborg, A. Rekdal, A survey of controlled experiments in software engineering, IEEE Transactions on Software Engineering 31 (2005) 733–753.

[12] B. Kitchenham, P. Brereton, D. Budgen, M. Turner, J. Bailey, S. Linkman, Systematic literature reviews in software engineering — a systematic literature review, Information & Software Technology 51 (2009) 7–15.

[13] B. Kitchenham, R. Pretorius, D. Budgen, P. Brereton, M. Turner, M. Niazi, S. Linkman, Systematic literature reviews in software engineering — a tertiary study, Information & Software Technology 52 (2010) 792–805.

[14] F. Q. da Silva, A. L. Santos, S. Soares, A. C. C. França, C. V. Monteiro, F. F. Maciel, Six years of systematic literature reviews in software engineering: An updated tertiary study, Information and Software Technology 53 (2011) 899–913.

[15] M. Ardis, D. Budgen, G. W. Hislop, J. Offutt, M. Sebern, W. Visser, SE2014: Curriculum Guidelines for undergraduate degree programs in software engineering, IEEE Computer (2015) 106–109.

[16] M. Jørgensen, Forecasting of software development work effort: Evidence on expert judgement and formal models, Int. Journal of Forecasting 23 (2007) 449–462.

[17] H. Petersson, T. Thelin, P. Runeson, C. Wohlin, Capture-recapture in software inspections after 10 years research—theory, evaluation and application, Journal of Systems and Software 72 (2004) 249–264.

[18] M. Jørgensen, Evidence-based guidelines for assessment of software development cost uncertainty, IEEE Transactions on Software Engineering 31 (2005) 942–954.

[19] P. Mohagheghi, R. Conradi, Quality, productivity and economic benefits of software reuse: a review of industrial studies, Empirical Software Engineering 12 (2007) 471–516.

[20] F. J. Pino, F. Garcia, M. Piattini, Software process improvement in small and medium software enterprises: a systematic review, Software Quality Journal 16 (2008) 237–261.

[21] M. Staples, M. Niazi, Systematic review of organizational motivations for adopting CMM-based SPI, Information and Software Technology 50 (2008) 605–620.

[22] S. Beecham, N. Baddoo, T. Hall, H. Robinson, H. Sharp, Motivation in software engineering: A systematic literature review, Information and Software Technology 50 (2008) 860 – 878.

[23] W. Azfal, R. Torkar, R. Feldt, A systematic review of search-based testing for non-functional system properties, Information and Software Technology 51 (2009) 957–976.

[24] E. Engström, P. Runeson, M. Skoglund, A systematic review on regression test selection techniques, Information and Software Technology 52 (2010) 14–30.

[25] J. Hannay, T. Dybå, E. Arisholm, D. Sjøberg, The effectiveness of pair programming. a meta analysis, Information & Software Technology 51 (2009) 1110–1122.

[26] J. S. Persson, L. Mathiassen, J. Boeg, T. S. Madsen, F. Steinson, Managing risks in distributed software projects: An integrative framework, IEEE Transactions on Engineering Management 56 (2009) 508–532.

[27] D. Smite, C. Wohlin, T. Gorschek, R. Feldt, Empirical evidence in global software engineering: a systematic review, Empirical Software Engineering 15 (2010) 91–118.

[28] L. B. Lisboa, V. C. Garcia, D. Lucrédio, E. S. de Almeida, S. R. de Lemos Meira, R. P. de Mattos Fortes, A systematic review of domain analysis tools, Information and Software Technology 52 (2010) 1 – 13.

[29] B. J. Williams, J. C. Carver, Characterizing software architecture changes: A systematic review, Information & Software Technology 52 (2010) 31–51.

[30] M. Turner, B. Kitchenham, P. Brereton, S. Charters, D. Budgen, Does the technology acceptance model predict actual use? A systematic literature review, Information and Software Technology 52 (2010) 463 – 479.

[31] M. S. Ali, M. A. Babar, L. Chen, K.-J. Stol, A systematic review of comparative evidence of aspect-oriented programming, Information and Software Technology 52 (2010) 871 – 887.

[32] O. Dieste, N. Juristo, Systematic review and aggregation of empirical studies on elicitation techniques, IEEE Transactions on Software Engineering 37 (2011) 283–304.

[33] A. H. Ghapanchi, A. Aurum, Antecedents to IT personnel's intentions to leave: A systematic literature review, Journal of Systems & Software 84 (2011) 238–249.

[34] K. Peterson, Measuring and predicting software productivity: A systematic map and review, Information & Software Technology 53 (2011) 317–343.

[35] T. B. C. Arias, P. van der Spek, P. Avgeriou, A practice-driven systematic review of dependency analysis solutions, Empirical Software Engineering 16 (2011) 544–586.

[36] J. Díaz, J. Pérez, P. P. Alarcón, J. Garbajosa, Agile product line engineering–a systematic literature review, Software — Practice and Experience 41 (2011) 921–941.

[37] C. Zhang, D. Budgen, What do we know about the effectiveness of software design patterns?, IEEE Transactions on Software Engineering 38 (2012) 1213–1231.

[38] T. Hall, S. Beecham, D. Bowes, D. Gray, S. Counsell, A systematic literature review on fault prediction performance in software engineering, IEEE Transactions on Software Engineering 38 (2012) 1276–1304.

[39] Y. Rafique, V. Misic, The effects of test-driven development on external quality and productivity: A meta-analysis, IEEE Transactions on Software Engineering 39 (2013).

[40] A. M. Magdaleno, C. M. L. Werner, R. M. de Araujo, Reconciling software development models: a quasi-systematic review, Journal of Systems & Software 85 (2012) 351–369.

[41] C. Pacheco, I. Garcia, A systematic literature review of stakeholder identification methods in requirements elicitation, Journal of Systems & Software 85 (2012) 2171–2181.

[42] Z. Li, H. Zhang, L. O'Brien, R. Cai, S. Flint, On evaluating commercial cloud services: A systematic review, Journal of Systems & Software 86 (2013) 2371–2393.

[43] N. B. Ali, K. Peterson, C. Wohlin, A systematic literature review on the industrial use of software process simulation, Journal of Systems & Software 97 (2014) 65–85.

[44] S. U. Khan, M. Niazi, R. Ahmad, Barriers in the selection of offshore software development oursourcing vendors: An exploratory study using a systematic literature review, Information & Software Technology 53 (2011) 693–706.

[45] R. Giuffrida, Y. Dittrich, Empirical studies on the use of social software in global software development–A systematic mapping study, Information & Software Technology 55 (2013) 1143–1164.

[46] D. Radjenović, M. Heričko, R. Torkar, A. Živkovič, Software fault prediction metrics: A systematic literature review, Information & Software Technology 55 (2013) 1397–1418.

[47] H. Munir, M. Moayyed, K. Peterson, Considering rigor and relevance when evaluating test driven development: A systematic review, Information & Software Technology 56 (2014) 375–394.

[48] N. Paternoster, C. Giardino, M. Unterkalmsteiner, T. Gorschek, Software development in startup companies: A systematic mapping study, Information & Software Technology 56 (2014) 1200–1218.

[49] U. Abelein, B. Paech, Understanding the influence of user participation and involvement on system success — a systematic mapping study, Empirical Software Engineering 20 (2015) 28–81.

[50] R. Jabangwe, J. Borstler, D. Smite, C. Wohlin, Empirical evidence on the link between object-oriented measures and external quality attributes: a systematic literature review, Empirical Software Engineering 20 (2015) 640–693.

[51] O. Al-Baik, J. Miller, The Kanban approach between agility and leanness: a systematic review, Empirical Software Engineering

20 (2015) 1861–1897.

[52] M. Zarour, A. Abran, J.-M. Desharnais, A. Alarifi, An investigation into the best practices for the successful design and implementation of lightweight software process assessment methods: A systematic literature review, Journal of Systems & Software 101 (2015) 180–192.

[53] A. Nguyen-Duc, D. S. Cruzes, R. Conradi, The impact of global dispersion on coordination, team performance and software quality – a systematic literature review, Information & Software Technology 57 (2015) 277–294.

[54] F. S. Silva, F. S. F. Soares, A. L. Peres, I. M. de Azevedo, A. P. L. F. Vasconcelos, F. K. Kamei, S. R. de Lemos Meira, Using CMMI together with agile software development: A systematic review, Information & Software Technology 58 (2015).

[55] M. Bano, D. Zowghi, A systematic review on the relationship between user involvement and system success, Information & Software Technology 58 (2015).

[56] A. Idri, F. A. Amazal, A. Abran, Analogy-based software development effort estimation: A systematic mapping and review, Information & Software Technology 58 (2015) 206–230.

[57] I. Steinmacher, M. A. G. Silva, M. A. Gerosa, D. F. Redmiles, A systematic literature review on the barriers faced by newcomers to open source software projects, Information & Software Technology 59 (2015).

[58] M. Brhel, H. Meth, A. Maedche, K. Werder, Exploring principles of user-centered agile software development: A literature review, Information & Software Technology 61 (2015) 163–181.

[59] E. Kupiainen, M. V. Mäntylä, J. Itkonen, Using metrics in agile and lean software development – a systematic literature review of industrial studies, Information & Software Technology 62 (2015) 143–163.

[60] S. Tiwari, A. Gupta, A systematic literature review of use case specifications research, Information & Software Technology 67 (2015) 128–158.

[61] D. Heaton, J. C. Carver, Claims about the use of software engineering practices in science: A systematic literature review, Information & Software Technology 67 (2015) 207–219.

[62] R. Rabiser, P. Grunbacher, D. Dhungana, Requirements for product derivation support: Results from a systematic literature review and an expert survey, Information & Software Technology 52 (2010) 324–346.

[63] E. Tüzün, B. Tekinerdogan, M. E. Kalender, S. Bilgen, Empirical evaluation of a decision support model for adopting software product line engineering, Information & Software Technology 60 (2015) 77–101.

[64] F. P. Brooks Jr., No silver bullet: essences and accidents of software engineering, IEEE Computer 20 (1987) 10–19.

[65] E. M. Rogers, Diffusion of Innovations, Free Press, New York, 5 edition, 2003.

[66] E. G. R. Barends, R. B. Briner, Teaching evidence-based practice: Lessons from the pioneers—An interview with Amanda Burls and Gordon Guyatt, Academy of Management Learning & Education 13 (2014) 476–483.

[67] P. Devanbu, T. Zimmermann, C. Bird, Belief & evidence in empirical software engineering, in: Proceedings 38th IEEE International Conference on Software Engineering (ICSE 2016), ACM Press, 2016, pp. 108–119.

[68] S. Oliver, K. Dickson, Policy-relevant systematic reviews to strengthen health systems: models and mechanisms to support their production, Evidence & Policy 12 (2016) 235–259.

[69] J. Lavis, G. Permanand, A. Oxman, S. Lewin, A. Fredheim, SUPPORT tools for evidence-informed health policy-making (stp) 13: Preparing and using policy briefs to support evidence-informed policymaking, Health Research Policy and Systems 7 (2009) S13.