Reporting Case Studies in Systematic Literature Studies – An Evidential Problem

Austen Rainer[a], Claes Wohlin[b]

[a]*Queen's University Belfast, 18 Malone Road, Computer Science Building, BT9 5BN, Belfast, Northern Ireland, UK*
[b]*Blekinge Institute of Technology, SE-371 79 Karlskrona, Sweden*

Abstract

Context: The term and label, "case study", is not used consistently by authors of primary studies in software engineering research. It is not clear whether this problem also occurs for systematic literature studies (SLSs). **Objective**: To investigate the extent to which SLSs in/correctly use the term and label, "case study", when classifying primary studies. **Method**: We systematically collect two sub-samples (2010-2021 & 2022) comprising a total of eleven SLSs and 79 primary studies. We examine the designs of these SLSs, and then analyse whether the SLS authors and the primary-study authors correctly label the respective primary study as a "case study". **Results**: 76% of the 79 primary studies are misclassified by SLSs (with the two sub-samples having 60% and 81% misclassification, respectively). For 39% of the 79 studies, the SLSs propagate a mislabelling by the original authors, whilst for 37%, the SLSs introduce a new mislabel, thus making the problem worse. SLSs rarely present explicit definitions for "case study" and when they do, the definition is not consistent with established definitions. **Conclusions**: SLSs are both propagating and exacerbating the problem of the mislabelling of primary studies as "case studies", rather than – as we should expect of SLSs – correcting the labelling of primary studies, and thus improving the body of credible evidence. Propagating and exacerbating mislabelling undermines the credibility of evidence in terms of its quantity, quality and relevance to both practice and research.

Keywords: Systematic Mapping Study, Systematic Review, Systematic Literature Review, Case Study, Credible Evidence

1. Introduction

For almost twenty years, the software engineering (SE) research community has noted that both the definition of the term "case study" and the label "case study" are not used consistently in software engineering research. Of particular concern is the observation that the label "case study" is not used in accordance with the established definitions for case study. This observation was made by Zannier et al. [1], more than 15 years ago, in their analysis of papers published at the International Conference on Software Engineering: "... our sample indicated a large misuse of the term case study." Not long after, in their guidelines, Runeson and Höst [2] conclude, "... the presented studies range from very ambitious and well organised studies in the field, to small toy examples that claim to be case studies."

Similar observations have been made, much more recently, by Wohlin [3], and Wohlin and Rainer [4]. These articles conclude that approximately 50% of so-called case studies are not actually case studies, again according to the established definitions. Whilst observations of mislabelling case studies are based on different definitions – Zannier et al. [1] and Runeson and Höst [2] base their observations on the definition by Yin [5], while the more recent studies use the definition by Wohlin [3] – the problem remains the same *across these definitions*. Thus, despite the problem being highlighted over the years, the problem of mislabelling *primary* studies as case studies is still far too common. Furthermore, this problem is not simply a matter of terminological precision or nicety: primary studies that are incorrectly labelled set erroneous expectations for the reader, and are both an *indicator* that the respective primary study may not have been designed or conducted properly, i.e., the primary study is unreliable, and also a *cause* to challenge the findings arising from that study, because they are based on incorrect foundations or at least an incorrect labelling.

To respond to these issues, Wohlin and Rainer [4] developed a checklist and a case study smell indicator to help ensure and assure a primary study as a case study. Rainer and Wohlin [6] report an evaluation of the smell indicator.

But as far as we are aware, there has been no investigation of the extent to which *secondary* studies (or indeed tertiary studies) misclassify primary studies as case studies. As with primary studies, secondary studies that incorrectly label primary studies set erroneous expectations for the reader, and are both an *indicator* that the secondary study may not have been

designed or conducted properly – i.e., again, a threat to reliability – and also a *cause* of invalid findings arising from that secondary study – i.e., again, a threat to the validity of the secondary study. There are more significant implications for software engineering research when a secondary study is unreliable or invalid, compared to a primary study. This is because the secondary study is attempting to aggregate or synthesis the state of research in an area. An unreliable or invalid synthesis misrepresents the body of knowledge and, as one implication, can provide incorrect recommendations for practice. We discuss the implications to reliability and validity in more detail later in the article.

We use the term, "systematic literature study" (SLS; [7, 8, 9]) as a collective term to cover different types of secondary study, such as systematic literature reviews (SLRs; [10]) and systematic mapping studies (SMSs; [11]). It is important to look at both SLRs and SMSs, and other types of systematic literature study too, because all such reviews and studies should label primary studies correctly and, if they do not, would suffer the problem of unreliability and invalidity stated above. We use the word "label" to refer to a primary study author's description of their own work or when discussing authors' descriptions of their studies in more general terms, including both primary study and SLS authors. We use the word "classify" to refer to an author's assignment of a label to someone else's work, based on a classification scheme, e.g., an SLS author classifies a primary study.

Investigating the extent to which SLSs misclassify primary studies as case studies is important because of the central contribution that SLSs make to evidence-based software engineering and to credible evidence. For example, practitioners are particularly interested in studies conducted in contemporary, real-life contexts because, amongst other reasons, the studies' findings are more likely to be relevant to practitioners' goals, challenges and decisions. Basili et al. [12] acknowledge the disconnect between practice and research: when research is not conducted in a real-life context, the research output is neither applicable nor scalable. An SLS that misclassifies primary studies as case studies can, however unintentionally, distort the perceived body of credible evidence and misinform practitioners on the applicability and scalability of recommendations arising from that evidence. Furthermore, an SLS that misrepresents the primary studies' research methods undermines our confidence in the SLS overall and in the body of evidence presented by the SLS.

Based on the above, we ask the following research question (RQ):

RQ For those SLSs that report the empirical research methods of primary studies, do those SLSs correct mislabelled case studies, propagate already mislabelled case studies, or exacerbate the problem by introducing new mislabels?

We consider four alternatives to how case studies are handled by SLS authors:

1. The SLS correctly accepts the labelling by the authors of the primary study, i.e., the primary study is a case study and the SLS correctly classifies it.
2. The SLS incorrectly accepts the labelling by the authors of the primary study, i.e., the primary study is not actually a case study, but the SLS accepts the respective authors' incorrect labelling of the primary study.
3. The SLS incorrectly classifies a primary study as a case study even though the respective authors' labelled their study as something other than a case study.
4. The SLS corrects an incorrectly labelled primary study, now correctly classifying the primary study as a case study. This situation can arise when the respective authors of the primary study had incorrectly labelled the primary study as something other than a case study. By making this correction, the SLS helps to reduce the problem of mislabelled case studies.

One consequence of alternative #2 is the propagation of an incorrect use of the label "case study", whilst alternative #3 exacerbates the situation by introducing further incorrect uses of the label "case study". Conversely, alternatives #1 and #4 result in the correct use of the label "case study", though sometimes alternative #4 my be a fortuitous outcome. We return to these alternatives in Section 6.

The remainder of this article is structured as follows: Section 2 presents background and related work; Section 3 explains how we investigated the RQ, including some preparatory analyses to address the RQ; Section 4 presents our analysis of the SLSs; Section 5 summarises our classification of primary studies from across the set of SLSs; Section 6 directly answers the RQ, and considers implications arising from our answer; Section 7 discusses the limitations of our research; finally, Section 8 concludes the article.

2. Background and related work

In this section, we review background and related work. In Section 2.1, we review definitions, guidelines and checklists for case study research, both generally and specific to software engineering. Then, in Section 2.2, we review problems of mislabelling constructs in research, and then the problems of the propagation and exacerbation of such mislabels. In Section 2.3, we suggest five actions to help address these problems, and position our current article relative to those actions. Finally, in Section 2.4, we summarise our review.

2.1. Definitions, guidelines and checklists

Many disciplines use case studies as a research method to investigate a phenomenon in its real-world context. Also, general guidelines for case study research have been published by several authors, the best known source probably being the book by Yin [13], with its first edition published in 1984. According to Runeson et al. [14], the definition of case study by Yin [13] fits particularly well in software engineering. Admittedly, researchers have different opinions on what constitute a case study, i.e., "one definition does not fit everybody". However, it is essential to use a definition and provide an appropriate reference. Given the applied nature of software engineering research, case study research has become an accepted method in the "research toolbox" for software engineering researchers. Discipline-specific guidelines for case study research in software engineering include the early guidelines by Kitchenham et al. [15] and, more recently, the detailed guidelines from, for example, Runeson and Höst [2], Verner et al. [16] and Runeson et al. [14]. Unfortunately, the SE-specific guidelines are primarily focused on the design, conduct and reporting of case studies, and do not emphasise the importance of ensuring the study is, in fact, a case study. Thus, although it may be possible to report a study as a case study, because the study "fits" the guidelines for *reporting* a case study, the study may not actually *be* a case study.

Considering the SE-specific guidelines more closely, we see that Runeson and Höst [2] refer to several prior definitions of case study research, including those by Benbasat et al. [17], Robson [18] and Yin [5]. Runeson and Höst [2] stress the real-world context: "Case studies are by definition conducted in real world settings, and thus have a high degree of realism, mostly at the expense of the level of control." In their book on case study research in software engineering, Runeson et al. [14] provide the following definition of case study for software engineering, which is based on the above sources:

> "Case study in software engineering is an empirical inquiry that draws on multiple sources of evidence to investigate one instance (or a small number of instances) of a contemporary software engineering phenomenon within its real-life context, especially when the boundary between phenomenon and context cannot be clearly specified."

Based on problems identified concerning the mislabelling of case study research, Wohlin [3] proposed a refined definition of *case study*, intended to *clarify* aspects of case study research.

> "A case study is an empirical investigation of a case, using multiple data collection methods, to study a contemporary phenomenon in its real-life context, and with the investigator(s) not taking an active role in the case investigated."

Wohlin's [3] definition clarifies the need to have more than a single data collection *method* in case study research. This clarification is added to avoid the possible misinterpretation of the concept of *multiple sources of evidence*, since multiple sources may simply imply, for example, the conduct of multiple interviews. Moreover, Wohlin's definition also differentiates case study research from action research through considering the role of the investigator.

In summary, the definition presents five essential components. These components were subsequently used to formulate a checklist [4], which we use in our analyses later in this article. The five components are ordered as follows based on the definition: the empirical investigation of a case; the use of multiple data collection methods; the study of a contemporary phenomenon; the real-life context; and the role of the investigator.

Furthermore, for our study, we define "case" as follows:

> "A case is a series of events or actions in a specific context, typically between different entities (e.g., people, teams, or organisational units) to accomplish an objective."

2.2. Mislabels, propagation and exacerbation

The problems of mislabelling, and of the propagation and exacerbation of mislabels, is not unique to the discipline of software engineering research or to the case study research method. We consider four complementary examples below.

In terms of mislabelling in other disciplines, Baker et al. [19] refer to "method slurring" in nursing, where researchers "... blur distinctions between the various qualitative approaches and combine their methodological prescriptions eclectically." Baker et al. contrast grounded theory and phenomenology along several dimensions of research, e.g., sources of data, sampling, and validity. They conclude that failure to align the chosen research method with the research question, and failure to define the method in the reporting, will severely undermine the credibility of the (in their case, qualitative) research.

Haslam [20] points to two forms of "concept creep" in psychology: "horizontal" and "vertical" concept creep. Haslam reviews six psychological concepts, synthesises from them, and then considers several benefits and drawbacks of concept creep. One concrete instance to demonstrate the benefit of concept creep is the expansion of the concept of *refugee* from the original conception of refugees displaced by conflict, to a refined conception of people also displaced by environmental catastrophe. One set of adverse effects of concept creep is that concepts might become confusing, semantically diluted, redundant, debased, or no longer meaningful. A second set of adverse effects concerns public perception of the discipline; in our context, this concerns software engineering professionals' perception of, for example, the relevance of research to practice.

Mislabelling occurs for other research methods in software engineering too. Ayala et al. [21] observe a similar situation with research that mines software repositories (MSR) and that uses the label "experiment". Ayala et al. recognise that experimental design decisions impact the respective study's ability to detect cause-effect relationships, which then has implications for internal validity and reliability. One consequence is that mislabelling a study misrepresents what the study is capable of doing, e.g., suggesting the study is capable of detecting cause-effect relationships when it is not.

Finally, Stol et al. [22] investigate the extent to which 98 articles, published across nine prominent SE journals, used any of the three main variants of Grounded Theory (GT), or techniques from those variants. Later in their article, they consider several implications of "method slurring" of GT: it undermines the legitimacy of GT; it undermines the legitimacy of *other* methods; it misrepresents the state of current research; and it violates a key principle of communicating science, i.e., accurately describing how data was collected and analysed.

2.3. Addressing the problems

We suggest five actions are needed to help address and improve the situation. First, we need appropriate definitions of different research methods. These definitions are, to a large extent, already available to the software engineering community. Second, authors of *primary* studies – and, as we will show, authors of *SLSs* too – should clearly cite the definitions used in their study, e.g., [13, 17, 18, 14, 3], when it comes to case study research. Third, authors of primary studies need to label their studies in accordance with the definitions cited. Fourth, authors of SLSs should ensure that they report research methods correctly, i.e., in accordance with correct definitions of different empirical research methods. Finally, reviewers and editors need to more carefully review manuscripts for these issues. Currently, the review process tacitly endorses low/er standards of quality assessment and mislabelling of research methods.

2.4. Summary

In summary, researchers in software engineering should correctly define and label the research methods they use in their primary studies, and when referring to others' studies. Furthermore, because of the standards expected of SLSs and of their potential impact on further research and practice, authors of SLSs have a particular responsibility to report empirical research methods correctly, including correcting mislabelled primary studies. In this article, we investigate how SLSs handle the labelling of primary studies as case studies.

3. Research approach

This section explains the approach we took to analysing the SLSs and the primary studies selected from those SLSs. We explain how we identified candidate SLSs (Section 3.1), categorised and selected SLSs (Section 3.2), analysed the SLSs (Section 3.3), selected and analysed primary studies from those SLSs (Section 3.4) and then answered our research question (Section 3.5).

3.1. Identifying candidate systematic literature studies

Given our focus on research methods and, in particular, on case study research, we wanted SLSs with results explicitly related to case studies. Furthermore, to mitigate the risk that the search strategy might affect the results, we used two complementary research strategies to collect two sub-samples: Sub-sample I and Sub-sample II. The two sub-samples cover two

different time spans as explained below. For both sub-samples, we used Scopus for our searches. The choice of Scopus was based on recommendations in the literature, e.g., [23, 24, 25]. We chose to use one database since our objective was to demonstrate a concern, and not to conduct a complete assessment of the literature. Moreover, given our objective to demonstrate a concern, rather than comprehensively assess the extent of the concern, we did not apply both search strings to both time spans. The searches for both sub-samples were limited to the "computer science" area and the document types "article" and "review" in Scopus.

Sub-sample I was conducted in May 2022 and covers SLSs published in the period 2010-2021. We used 2010 as the starting year since the guidelines of Runeson and Höst [2] and of Verner et al. [16] were published in 2009. After the publication of the guidelines for software engineering, SLS authors ought to be more aware of what constitutes a case study.

For Sub-sample I, we looked at those SLSs that explicitly refer to "case study" in their abstract, keywords or title. Our assumption was that explicit reference to "case study" in these ways would give an appropriate sample to investigate how case study research is treated in SLSs.

We used the following procedure to identify Sub-sample I:

1. We conducted three searches using Scopus to capture both systematic literature reviews and mapping studies, as well as studies describing structured reviews of the literature but using alternative phrasings.
2. We used the following three search strings, all limited to the "computer science" area and the document types "article" and "review":
   - #1 "systematic literature review" AND software AND (engineering OR development) AND "case stud*",
   - #2 "systematic review" AND software AND (engineering OR development) AND "case stud*",
   - #3 "mapping study" AND software AND (engineering OR development) AND "case stud*".
3. All searches were undertaken by the second author.
4. The searches returned 169 articles. The search strings resulted in an overlap of articles identified, i.e., the same articles appear in more than one of the searches. Thus, duplicates were removed from searches #2 and #3. Also, many articles contained the words used in the search strings, but were not necessarily SLSs. As the next step, therefore, non-SLSs were removed based on reviewing the abstracts. The filtering

of articles is summarised in Figure 1. 56 articles remain for further analysis.

Sub-sample II was collected in April 2023 and covers SLSs published only in 2022. We chose 2022 as a cutoff to include the full calendar year. For this sub-sample, we adopted a complementary strategy: we looked at all SLSs having variations of formulations related to "literature reviews" in their title.

We used the following procedure to identify Sub-sample II:

1. The search was limited to four general software engineering research journals ranked in the top 10 of Google Scholar when looking for top venues for "Software systems"[1]. The four journals investigated were: *IEEE Transactions on Software Engineering*, *Journal of Systems and Software*, *Journal of Information and Software Technology*, and *Journal of Empirical Software Engineering*.
2. We used the following search string, limited to the "computer science" area and the document types "article" and "review": "systematic literature review" OR "mapping study" OR "systematic review" OR "literature review" in the title of the articles.
3. All searches were undertaken by the second author.
4. The search returned 40 SLSs published in 2022.

In total we have 96 SLSs across the two sub-samples. As noted above, for Sub-sample I, only articles published in software engineering journals (see Table 1) are included, whilst for Sub-sample II, only articles published in four journals were included. A listing of the 96 SLS articles, with links to the articles, is available in Supplement 1 in the online supplementary material linked in Appendix A.

3.2. Categorising and selecting SLS

The articles identified through the search procedure were then categorised according to four criteria: whether the article is in fact an SLS; whether the article reports counts, or percentages, of research methods of the primary studies (so that we can examine the frequency of case studies reported in the

[1] `https://scholar.google.com/citations?view_op=top_venues&hl=en&vq=eng_softwaresystems`

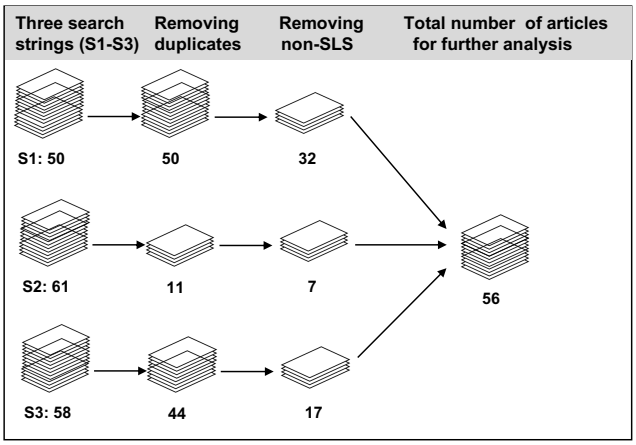

Figure 1: Search and selection process for Sub-sample I.

Table 1: Distributions of identified SLSs across journals.

Journals for Sub-sample I	**Count**
Information and Software Technology (IST)	24
Journal of Systems and Software (JSS)	11
Journal of Software Evolution and Process (JSEP)	5
Transactions on Software Engineering (TSE)	5
Software Engineering and Knowledge Engineering (SEKE)	3
Empirical Software Engineering (EMSE)	2
IET Software (IETS)	2
Requirements Engineering Journal (REJ)	2
Software Quality Journal (SQJ)	1
International Journal of Software Engineering Applications (SEA)	1
Sum	56
Journals for Sub-sample II	**Count**
Journal of Systems and Software (JSS)	17
Information and Software Technology (IST)	14
Empirical Software Engineering (EMSE	5
Transactions on Software Engineering (TSE)	4
Sum	40

literature); whether the article supports traceability from the SLS back to the source primary study (so that we can independently check the label of "case study"); and whether the article makes any explicit comment on the problem of labelling studies as "case studies". The definition of categories is summarised in Table 2.

The two authors independently categorised all articles for Sub-sample I, shared their independent categories and discussed them, arriving at a consensus decision for the category of each article. The categorisation of articles, and the process of filtering them, is summarised in Table 3. In the filtering

process, one article was identified in the search that did not meet the criteria in the search string. In other words, one article was incorrectly retained from the original searches, which was subsequently removed when cross-checking the identified articles with the search criteria.

Having categorised the articles, we removed articles in Categories A and B as they provide no meaningful information for our subsequent analysis. For Categories C, D and E, the lack of traceability means we cannot independently check how the SLSs classified each primary study, however articles in these categories will be useful for some of our analyses, concerning frequency counts and comments about the misuse of labels. Articles in Categories F and G will be the most valuable articles as they provide traceability to the source primary studies.

We calculated unweighted and weighted Kappa statistics for Sub-sample I to evaluate the agreement between the two authors' categorisations. The unweighted Kappa statistic assumes the categories are entirely independent whilst the weighted Kappa statistic takes account of the ordering of the categories. The Kappa statistics are reported in Table 3. Overall, we have substantial to near-perfect agreement for Sub-sample I. Given the high-level of agreement, the categorisation for Sub-sample II was done only by the second author, except for two SLSs where a second opinion was perceived as needed. The categorisation of the SLS, is available in Supplement 1 in the online supplementary material linked in Appendix A.

Table 2: Decision criteria for the categories of SLSs.

Cat.	Excluded	SLS	Counts	Traceability	Labelling
A	Yes	No	N/A	N/A	N/A
B	Yes	Yes	No	N/A	No
C		Yes	No	N/A	Yes
D		Yes	Yes	No	No
E		Yes	No	No	Yes
F		Yes	Yes	Yes	No
G		Yes	Yes	Yes	Yes

Notes:
Category A articles are not actually SLSs.
Category B articles are SLSs but lack necessary information.
SLS = Systematic literature study
Counts = SLS reports frequencies or percentages of research methods of the primary studies.
Traceability = SLS reports classification of research method for each primary study.
Labelling = SLS makes a comment on the misuse of the label "case study".

During our categorisation of Sub-sample I, we identified three articles

that are actually tertiary studies, rather than SLSs. Two of these articles are Category B articles and are removed almost immediately during the funneling process (see Table 3). The third article [26] is a Category G article. This article is separated out in Step 5 of Table 3. A second article [27] in Category G is also removed. These two articles are removed because they both comment on the misuse of the label "case study", but do not take this problem into account in their subsequent analysis. For a study in Category G to be included in our analysis, the article must comment on the mislabelling of case studies and also adapt their analysis accordingly. Thus, the two studies in Category G are removed from the analysis. We further discuss the exclusion of these two articles in Supplement 2 in the online supplementary material linked in Appendix A. The effect of these exclusions is that we did not identify a Category G article we could include in our analysis in the two sub-samples. Note that because the Category F articles do not comment on the misuse of labels, we cannot determine whether the authors know there is a problem, but do not report it, or do not know there is a problem. We assume the authors of SLSs in Category F do not know there is a problem.

Table 3: Identification and filtering of articles in different categories.

Step	**Description**	**Categories**	**Articles remaining for Sub-sample I**	**Articles remaining for Sub-sample II**
1	Input from search and screening	N/A	56	40
2	Remove Categories A and B	A & B	24	29
3	Remove Category D	D	16	7
4	Remove Categories C and E	C & E	11	1
5	Separate out remaining tertiary studies	N/A	10	0
6	Remove articles in Category G not needing a detailed analysis	N/A	9	0
7	After cross-check with search string, remove articles incorrectly included	N/A	1	0
	SLSs remaining for detailed analyses (all Category F)	N/A	8	3
Kappa statistics for Sub-sample I: Unweighted Kappa statistic: 0.796 (substantial agreement) Weighted Kappa statistic: 0.869 (near-perfect agreement)				

Table 4 presents summary information on the selected SLSs from the two sub-samples. Given the removal of two articles in Category G, all eleven SLSs selected for a detailed analysis are in Category F.

Table 4: Summary of SLSs selected for analysis.

ID	Cit.	Jnl	Title	Type
Sub-sample I				
SLS P2	[28]	IST	Software engineering research for computer games: a systematic review	SLR
SLS P11	[29]	IST	Past and future of software architectural decisions – a systematic mapping study	SMS
SLS P16	[30]	JSS	Feature extraction approaches from natural language requirements for reuse in software product lines: a systematic literature review	SLR
SLS P21	[31]	IST	The effects of test driven development on internal quality, external quality and productivity: a systematic review	SLR
SLS P33	[32]	IST	What software reuse benefits have been transferred to the industry? A systematic mapping study	SMS
SLS P36	[33]	IST	Empirical evidence in follow the Sun software development: a systematic mapping study	SMS
SLS P44	[34]	IETS	Impact of design patterns on software quality: a systematic literature review	SLR
SLS P45	[35]	IST	Empirical software product line engineering: a systematic literature review	SLR
Sub-sample II				
SLS P59	[36]	JSS	Revealing the state of the art of large-scale agile development research: A systematic mapping study	SMS
SLS P81	[37]	IST	Ambiguity in user stories: A systematic literature review	SLR
SLS P85	[38]	IST	Software security patch management - A systematic literature review of challenges, approaches, tools and practices	SLR

3.3. Analysis of the SLS articles

Before analysing the primary studies, we wanted to assess the SLSs. We are interested in two aspects of the SLSs:

1. The designs of the (retained) SLSs, e.g., what did the SLS consider when searching for, selecting and analysing the primary studies? To perform this analysis we used the eleven SLSs in Category F listed in Table 4. This analysis is presented in Section 4.1.
2. Comments made by the authors of the SLSs about the mislabelling of "case studies" in software engineering. To perform this analysis we used SLSs categorised as Category C and E according to Table 2. There are no SLSs available to consider from Category G, because we removed these from our analyses. Our analysis of the comments made by Category C and E articles is presented in Section 4.2.

3.4. Analysis of samples of primary studies from several SLSs

Having selected eleven SLSs and assessed them, we then turned to investigating a sample of primary studies from each SLS. We wanted to check

whether the SLS authors were classifying primary studies correctly. We investigated three aspects:

1. We compared the sample from each SLS's classification of primary studies against our classification of those studies. For our classification, we used the case study checklist briefly discussed in Section 2.1 and used in our previous analysis of primary studies [4]. As described in the checklist, the five components are investigated one at the time until one component does not fulfil the definition for being a case study. Each author of the current article first independently reviewed the primary studies. We then discussed our views and resolved disagreements. For those primary studies that we agreed were borderline "case studies", we took a generous attitude, i.e., to classify a study as a "case study". This analysis is presented in Section 5.1.
2. We examine those (few) studies correctly assessed as being case studies and present information about these studies in Section 5.2.
3. We also checked how the authors of the primary studies labelled their own study. This allows us to undertake a three-way comparison of the SLS authors' labels with the primary-study authors' labels with our labels. This analysis is most directly relevant to our research question and is presented in Section 6.

The analysis of primary studies reported later in this paper, e.g., in Section 5, is based on sampling six batches of primary studies from across our two sub-samples, with three batches per sample. For each batch, we sought to sample three primary studies per SLS, however as not all SLSs had nine primary studies classified as case studies we sometimes had to settle for less than nine case studies per SLS per batch. In total, 79 primary studies are analysed. For simplicity, we refer to the two samples rather than the six batches. A fuller explanation of the batches is provided in Supplement 3 in the online supplementary material linked in Appendix A.

3.5. Investigating the research question

As noted earlier, we collected three sets of labels: the SLS authors' labelling of primary studies, the primary-study authors' own labels, and our labels. We did this for primary studies from the two sub-samples. The three sets of labels allow us to assess the four alternative answers to our research question. We present the results of this analysis in Section 6, including a consideration of the implications of our results.

4. Analysis of the selected SLSs

In this section we first consider the designs of the eleven SLSs, as these designs provide insights into how the respective SLSs analysed the primary studies (Section 4.1). We then briefly discuss comments made by some of the SLSs on the way that the label "case study" has been misused in primary studies (Section 4.2).

4.1. The designs of the selected systematic literature studies

Our analysis of the designs of the eleven SLSs we studied is presented in Table 5. Additional information, summarising sources cited by the eleven SLSs, is presented in Table 6.

Table 5 analyses each of the eleven SLSs according to thirteen criteria. The criteria are numbered down the left-hand side of the table and summarised in the notes at the end of the table. The "answer" to each criterion may take several values, which are also summarised in the table's notes.

For each sub-sample, the right-most column presents a proportion. This proportion gives an *indication* of the extent to which the SLSs in the respective sub-sample considered the criterion. Then, in the last row of the main table, labelled "**Sum**", a total is given for most of the criteria in each column. This total gives an *indication* of the overall quality of the study design for that SLS. We emphasise these totals and proportions are approximations based our two sub-samples.

The proportions in Table 5 *suggest* that just over a half of the SLSs in Categories F (see item #2; 6/11 SLSs) had an explicit research question about research methods; that most SLSs explicitly analysed the research methods (item #3; 10/11); and that most SLSs commented on the most common research method (item #4; 8/11). Only *three* SLSs had an explicit definition for "case study" (item #5), though none of the three were entirely consistent with established definitions; just over a half of the SLSs (item #7; 6/11) used an existing classification (see Table 6 for more information) and for those that did not use an existing classification, no SLS explained how they developed their classification (item #8). Also, the SLSs varied in the number of research methods they considered, from two methods to ten (item #9).

In a previous study [4], we found that about 40% of studies reported as case studies by their authors were in fact better understood as small–scale evaluations. We base our definition of *small-scale evaluation* on Robson [39]:

> Small-scale evaluations aim to demonstrate, illustrate, or show the feasibility of a proposed solution, for example, a concept or tool. The evaluation is conducted with a single researcher or a small team of researchers. It is run over a short period, most often in a single site, for example, a laboratory.

Of the eleven SLSs, four SLSs have a method that appears similar to the small–scale evaluation (item #10). About a quarter of the SLSs explicitly cite guidance on the design of case studies in software engineering (item #11; see Table 6 for more information) and only three of the SLSs cite guidance on the design of empirical studies in software engineering (item #12' see Table 6 for more information). For items #11 and #12 in Table 5, there is no straightforward way to analyse the SLSs because the relevance of the citation can depend on context. For example, in Table 6, the sources cited for existing classifications include citations to generic guidance on software engineering research. For items #11 and #12, we therefore looked for citations made in the context of the SLS paper discussing the design or conduct of software engineering research. Finally, item #13 suggests that over a half (7/11) of the eleven SLSs explicitly included a Quality Assessment as part of the SLS.

Separate from Table 5, SLS P45 [35] is of particular interest since it provides a quality assessment of the reporting of the empirical methods used, including case study. The authors assess the quality of the case study reporting using ten criteria that case studies should fulfil. This in–depth analysis is valuable, although it focuses on what the authors report and not necessarily what the authors of the primary studies did. Furthermore, the authors also report the case studies in relation to characteristics that case studies ought to report. Their analysis is done based on the characteristics provided by Robson [18]. The assessments illustrate that the authors are concerned with how case studies are reported, although they do not directly discuss the case studies in relation to the definitions of case study research, and do not therefore explicitly consider whether a study has been *mislabelled* as a case study. The issue of the lack of use of (established) definitions may be exemplified with the authors of SLS P45 [35] reporting both *industrial* and *academic* case studies.

4.2. Comments on the case studies

Separate to our analysis of the 11 SLSs we selected for our main analysis (see Table 3 and Table 4), we wanted to see whether *other* articles from our

Table 5: Analysis of the SLS designs for the eleven Category F SLSs.

	Sub-sample I SLS #									Sub-sample II SLS #			
#	**2**	**11**	**16**	**21**	**33**	**36**	44	**45**	**Prp**	**59**	**81**	**85**	**Prp**
1	2010	2014	2015	2016	2018	2018	2020	2020	NA	2022	2022	2022	NA
2	Y	N	Y	N	Y	Y	N	Y	5/8	N	N	Y	1/3
3	Y	Y	N	Y	Y	Y	Y	Y	7/8	Y	Y	Y	3/3
4	N	Y	Y	Y	Y	Y	N	Y	6/8	Y	N	Y	2/3
5	Y	N	N	N	N	N	N	N	1/8	Y	N	Y	2/3
6	N	NA	NA	NA	NA	NA	NA	NA	0/8	P	NA	P	2/3
7	Y	Y	N	N	Y	Y	N	N	4/8	Y	N	Y	2/3
8	NA	NA	NS	NS	NA	NS	NS	NS	0/8	NA	NS	NA	0/3
9	3	4	2 x 2	5	4	9	3	3	NA	8	6	10	NA
10	N	Y	N	N	Y	P	N	N	3/8	N	N	Y	1/3
11	N	Y	N	N	N	N	Y	Y	3/8	N	N	N	0/3
12	Y	N	N	N	N	N	N	Y	2/8	Y	N	N	1/3
13	Y	N	Y	Y	N	N	Y	Y	5/8	N	Y	Y	2/3
Sum	6	5	3	3	5	5	3	6		6	2	8	

Notes:
Y = Yes; N = No; P = Potentially present; Prp = Proportion of Ys & Ps; NA = Not applicable
NS = Not clearly stated in the paper

Summary of criteria

1. Year published: In what year was the SLS published?
2. Specific RQ: Is there a research question (RQ) in the article that specifically asks about research methods?
3. Research methods analysed: Are research methods explicitly analysed?
4. Common method: Do the authors of the SLS explicitly discuss the most common, or predominant, research method?
5. Case study definition: Is "case study" explicitly defined in the paper?
6. Consistent definition: Is the case study definition consistent with established definition, e.g., [14]?
7. Existing classification: Do the authors use an existing, published classification of studies? The published classifications are summarised in Table 6.
8. Classification dev.: If the answer to Q#7 is "No", or NA, how was it developed?
9. Research methods count: Number of research methods used in the SLSs' classification?
10. Small-scale evaluation: Is there any type of suitable method related to small-scale evaluation?
11. SE case study design: Is a standard citation to case study design in SE used? Citations are summarised in Table 6.
12. Empirical studies: Is a citation to empirical studies used for research methods? Citations are summarised in Table 6.
13. Quality assessment: Does the paper report the quality assessment of primary studies?

Sum: Sum of Ys and Ps

broader dataset had recognised the problem of mislabelling. We identified six SLSs, all from Categories C and E and published between 2011 and 2022, that explicitly comment on the (mis)use of the label, "case study".

These six SLSs all demonstrate that, over the years, the problem of mislabelling primary studies as case studies, when they are not actually case studies, is well recognised and far too common. These six SLSs are in addition to other sources cited in Section 1. Yet although these SLSs recognise problems with the use of the label "case study" these SLSs, *themselves*, mislabel primary studies. One of these six SLSs (SLS P18) does however explicitly

Table 6: Sources cited by the SLSs.

ID	Sources
Sources cited for Q7: Existing classifications	
SLS P2	[40] [15]
SLS P11	[41] [42] [43] [44]
SLS P33	[45]
SLS P36	[46]
SLS P59	[47] [48] [49]
SLS P85	[50]
Sources cited for Q11: Case study design	
SLS P11	[51] [52] [53]
SLS P44	[14] [53]
SLS P45	[14] [2]
Sources cited for Q12: Design of empirical studies	
SLS P2	[54] [15] [40]
SLS P45	[18] [40]

introduce a distinct type, "Example", to help distinguish true case studies from other types of study. Comments on the mislabelling of case studies are further discussed in Supplement 4 in the online supplementary material linked in Appendix A.

5. Summary of analysis of primary studies

Having considered the SLSs in the previous section, we now turn to consider the primary studies. For conciseness we present a summary of our analysis in this section. A listing of the primary studies can be found in Supplement 5 and the full analysis of the primary studies is provided in Supplement 6. Links to the online supplementary material are provided in Appendix A. We first consider the classification of primary studies (Section 5.1) and then consider the set of studies we identified as correctly classified as case studies (Section 5.2).

5.1. Classifications of the primary studies

Based on our analysis of the 79 primary studies, and connecting our analysis back to the criteria identified in Wohlin's *refined* definition of a case study in Section 2.1, we conclude that:

1. The most common reason for deciding that a study was not a case study was that the study was not conducted in a real-life context. As Wohlin [3] has suggested before: it is a case and it is a study, but it is not a case study. Studies not conducted in a real-life context were

often small-scale evaluations in a laboratory environment, as identified in [3].

2. The second most common reason was that the study was not contemporary. Studies that were not of a contemporary phenomenon most commonly are archival studies, for example, of open source projects.
3. For some studies, they were not case studies, either because a *case* was not being studied (as required of a case study), or the study was better described as action research.

Across our sample of 79 primary studies, we found zero instances were the SLS authors classified a primary study correctly as a "case study" when the original authors mislabelled the study as not a case study. On the other hand, we found 29 instances (37% of the analysed primary studies) where the SLS classified a primary study as a "case study" despite the fact that the authors of the primary study did not label their study that way.

5.2. Investigation of the nineteen case studies

The nineteen actual case studies are published between 2003 and 2020. This means that the case study guidelines published in 2009 by Runeson and Höst [2] and Verner et al. [16] were not available for all primary-study authors, but these guidelines were available to the SLSs we studied.

About half of the nineteen case study articles refer to general references to empirical research and metrics, and slightly fewer refer specifically to sources concerning case study design.

Based on our comparison of primary studies between the two sub-samples, we conclude that too few primary studies refer to sources concerning the research method being used, even when they are correctly labelled.

The lack of citations to sources concerning research methodology may be a reason that too many studies are mislabelled, although other reasons may exist, for example, the different expectations at different publication venues.

Finally, it is worth noting that we, the authors of the current article, did not initially agree on the classification of "case study" for six of these nineteen case studies. This may appear surprising. We argue that the main reason for this disagreement is that the articles are often unclear about the components of the case study definitions (see Section 2.1) – in other words, authors do not clearly and transparently report the design etc of their study [55] – which is natural since many articles do not refer to sources for research methodology, in particular not to case study methodology.

6. The treatment of case studies by SLSs

Having completed several analyses, we are now in a position to directly consider our research question. In this section, we first discuss the frequency of case studies in software engineering. We present reported frequencies from the SLSs and compare those with our revised frequencies. We then use those frequencies to examine the degree to which SLSs have distorted, through propagation and exacerbation, the status of case studies in software engineering research.

6.1. The prevalence of case studies in SE

Table 7 presents counts and percentages of the prevalence of case studies in software engineering and of our estimated correctness of this prevalence, for the two sub-samples. As the table indicates, the full sample of eleven SLSs report a frequency of case studies in the range 10% (SLS P85) to 60% (SLS P59) with a mean average of 39%. For those studies classified by the SLSs as case studies, our estimates of correctness show that the SLS authors misreport the number of case studies in the literature. The degree of misreporting ranges from 33% (SLS P59) to 100% (SLSs P2, P16, P85). Based on our re-classification of primary studies, the average over-estimate for this set of SLSs is 76%, given that we estimate that 24% of primary studies are correctly classified.

Table 7: Prevalence of case studies and an estimated correctness rate.

ID	Jnl	Domain	# RM	Counts and % of case studies	Our estimated correctness
Sub-sample I					
SLS P2	IST	Games development	3	4/20 (20%)	0/4 (0%)
SLS P11	IST	Software architecture	4	31/144 (22%)	1/9 (11%)
SLS P16	JSS	Software product lines	2	2/13 (15%)	0/2 (0%)
SLS P21	IST	Test-driven development	4	8/27 (30%)	2/8 (25%)
SLS P33	IST	Software reuse	4	25/50 (50%)	2/9 (22%)
SLS P36	IST	Follow-the-sun software development	9	18/32 (56%)	1/9 (11%)
SLS P44	IETS	Design patterns	4	27/50 (54%)	1/9 (11%)
SLS P45	IST	Software product lines	3	35/62 (56%)	4/9 (44%)
Sub-total				150/398 (38%)	11/59 (19%)
Sub-sample II					
SLS P59	JSS	Large scale agile development	8	78/129 (60%)	6/9 (67%)
SLS P81	IST	User stories	6	4/17 (24%)	2/4 (50%)
SLS P85	IST	Security patch management	10	7/68 (10%)	0/7 (0%)
Sub-total				89/214 (42%)	8/20 (40%)
Total				239/612 (39%)	19/79 (24%)
Notes: RM = Research methods in the classification for each SLS.					

Our overall estimate of correctly labelled case studies is 24%. This estimate is based on the use of established definitions of case study research, and is derived only from the re-analysis of a sample of primary studies reported as case studies in the eleven SLSs. This estimate is *lower* than both the SLSs' overall estimate in relation to other research methods and the estimates we reported previously [3, 4], which were around 50% based on assessing primary studies claiming to present case studies. Our estimated corrections for the SLSs suggests that the authors of SLSs contribute to the over-reporting of the use of case study research in software engineering.

In 50 primary studies of the 79 primary studies in the overall sample, the *primary-study* authors claim that they present case studies. Based on our assessment, only 19 of these 50 primary studies present case studies. Thus, the correctness rate for *primary-study authors* self-reporting is 38% (19/50), which again is lower than our previous estimates in [3, 4].

In our previous analyses, the focus was either on case study articles published in journals (i.e., [3]) or on journal articles citing the case study guidelines by Runeson and Höst [2] (i.e., [4]). But in the current article, the primary studies included are those identified *by the SLS authors* when targeting a specific area of research in software engineering. Thus, we have three samples constructed using different sampling strategies.

The misclassifications by the SLSs is partly dependent on the classification scheme used by the SLS authors and hence the frequencies of different research methods. One SLS only classifies using two research methods (SLS P16), whereas another SLS (SLS P85) classifies using ten research methods, and the other SLSs are in between these two limits. The differences in classification schemes are an issue since, depending on the scheme used, all primary studies are "forced" into the classes in the scheme.

6.2. The mislabelling of case studies in SE

We now return to our RQ and specifically to the four alternative ways of labelling a study's research method, as discussed in Section 1. The four alternatives are concerned with how an SLS treats primary studies as case studies.

Table 8 summarises the four alternative ways of labelling case studies for the two sub-samples of 59 and 20 primary studies respectively.

Our overall conclusions from the analysis presented in Table 8 are averaged over the two sub-samples. Overall, the table indicates that authors of SLSs have a tendency to simply restate the label, "case study", used by

Table 8: SLS authors handling of case studies.

Alter-native	Description	Sub-sample counts and %s I Batches 1-3	II Batches 4-6	Total
1	Restate a correctly labelled primary study	11 (19%)	8 (40%)	19 (24%)
2	Restate an incorrectly labelled primary study	27[a] (46%)	4 (20%)	31 (39%)
3	Distort a correctly labelled primary study	21[a] (36%)	8 (40%)	29 (37%)
4	Correct an incorrectly labelled primary study	0 (0%)	0 (0%)	0 (0%)
Total		59	20	79

[a]due to rounding, 46% + 36% = 82% but (27+21)/59=81%

the authors of the respective primary study, independent of whether the label of "case study" is correct or not ((19+31)/79 instances; 63%). In about two-fifths of instances (31/79; 39%) the authors of the SLSs propagate an *incorrect* use of the term "case study". Furthermore, in over a third of instances (29/79; 37%), SLS authors exacerbate the situation: they incorrectly classify a primary study as a "case study", even when the primary study was *not labelled* as "case study" by the authors of the primary study. Taken together, propagation and exacerbation occur in 76% of the instances ((31+29)/79). By doing this, the SLS authors are distorting the body of evidence in the area of the respective SLS; or, in other words, by reporting *incorrect information* the SLS authors do not report *credible evidence*, one of the main objectives of SLSs.

Sub-sample II, drawn from 2022, does seem to present "better" results than Sub-sample I. Sub-sample II has a considerably lower percentage of re-stating incorrectly labelled primary studies (20% compared for 48% for Sub-sample I), a higher percentage of re-stating correctly labelled primary studies (40% compared to 19% for Sub-sample I) and a slightly lower percentage of simply restating the primary study's label (60% compared to 67%). Sub-sample II has a slightly higher percentage of distorting a correctly labelled primary study (40% to 36%). Like Sub-sample I, Sub-sample II does not correct any incorrectly labelled primary studies. However, Sub-sample II only includes three SLS, and one of them (SLS P59) is the SLS with the highest correctness, see Table 7. Thus, it is premature to assume that the situation is improving. Further studies are needed to be able to investigate any potential improvement over time.

An underlying, recurring problem seems to be that many researchers have a far too flexible interpretation of case study research or, alternatively, they lack sufficient knowledge concerning the definition of case study research. Researchers need to know and understand the definitions of research methods,

and use them correctly. Items #5 and #6 in Table 5 show that only three of the eleven SLSs explicitly defined the term, "case study", and *none* of the three SLSs presented a definition consistent with the established definitions, although two had partially consistent definitions. SLS authors need to be even more aware of these issues, compared to authors of primary studies, as SLSs attempt to gather and synthesise the available evidence in a research area. SLS authors therefore need to be careful gatekeepers, who ensure that research methods are correctly reported in SLSs. If not, we do not have *credible evidence*; we get the propagation of incorrect evidence or, even worse, the exacerbation of incorrect evidence.

A related issue, contributing to the problem, is *how* SLS authors choose to classify the research methods used in the primary studies. If authors develop their own classification scheme and decide *a priori* the research methods in their classification, then there is a risk that not all research methods reported across the primary studies will be properly represented: there may situations where a primary study is misclassified to fit the *a priori* classification. Item #7 in Table 5 shows that six of the eleven SLSs used an existing classification, however even then it is not clear that the existing classification would cover all types of primary study, as indicated by the range of research methods for item #9 of Table 5. For the other five SLSs, none explain how they developed their classification.

A potential way forward is to either use an already accepted and comprehensive classification scheme (preferred option) or formulate the classification based on the research methods stated by the authors of the primary study, whilst also ensuring that the methods stated in the primary studies adhere to the formal definitions of each research method. In other words, to take a bottom-up approach to classification, with careful checks on the labels allowed into the classification.

Relating these results back to our research question, presented in Section 1, we conclude that our analysis of primary studies corroborates prior research (the first item listed below), and, for SLSs, produces three novel findings (the subsequent three items listed below), i.e.:

1. Too many primary studies incorrectly present themselves as being case studies. This conclusion is supported by the literature, including, for example, [1, 2, 3, 4].
2. SLS authors frequently simply restate the incorrect case study label from the respective primary study.

3. In many cases, SLS authors incorrectly *change* the labels of primary studies to "case study". Thus, they make the incorrect labelling of case studies worse than if only re-stating the original label for the primary study.
4. Given our results, it seems very unlikely that SLS authors correct the labelling, i.e., correctly classify a primary study as a "case study" when the primary study is presented incorrectly as something else by the authors of the primary study.

In summary, based on our two sub-samples, SLS authors distort the labelling of primary studies as "case studies", through propagation and exacerbation, in 76% ((27+21)/59 = 81% for Sub-sample I and (4+8)/20 = 60% for Sub-sample II) of the instances of such labelling.

6.3. Implications

The mislabelling of primary studies as case studies raises several implications. We consider four broad implications here, supporting our discussion with illustrative examples from the eleven SLSs we analysed. Given publication constraints, we focus here on illustrative examples that can be concisely presented, which means the examples are structured or quantified. The principles behind our illustrative examples are intended to apply much more broadly across the different kinds of case study research. In their tertiary review of research synthesis in software engineering, Cruzes and Dybå [56] categorised the methods of synthesis into 15 categories, e.g., narrative, thematic, grounded, comparative, content, vote-counting, and quantitative (the 15^{th} was no method). We cite Cruzes and Dybå [56] to acknowledge that the broad implications we discuss here will manifest differently across different kinds of case study research and different methods of research synthesis. Also, our discussion here complements the consequences and implications already recognised in Section 2.2.

6.3.1. Labels set expectations

Consider an example where an SLS incorrectly labels a small-scale evaluation as a case study. Typically, small-scale evaluations occur in a laboratory environment. To label such a study as a case study creates the erroneous expectation for the reader that the study took place in a real, contemporary setting and was conducted to the standards of case study research. It also creates erroneous expectations about the way that the study's findings might

be generalised, e.g., through an analytical, or theoretical, mechanism, rather than through a statistical mechanism.

More generally, it is important to clearly and transparently report the actual research method used in the primary study so that the reader has a clear and transparent knowledge of the research design and the underlying research process [55]. Or, in other words, when we label a primary study as a case study, we create expectations for the reader about the aims, design, conduct, interpretation of results, and implications of results for professional practice.

The incorrect labelling of a primary study by an SLS is therefore both an *indicator* that the respective SLS may not have been designed, conducted etc. properly, and also a *cause* of invalid findings.

In terms of the indicator, the analysis and results of an SLS are dependent on the quality of the primary studies that are input to that SLS. If these inputs are mislabelled, this raises uncertainty about the general treatment of the primary studies as input to the SLS and, more generally, about the overall reliability of the SLS.

In terms of cause, if primary studies are mislabelled, this directly affects how each primary study is subsequently treated by the SLS and, by extension, how primary studies are synthesised. This raises concerns about the validity of the aggregated or synthesised findings.

6.3.2. Estimations and comparisons

The mislabelling of primary studies leads to difficulties estimating the number of genuine case studies conducted in SE in general or conducted for a particular domain in SE. This leads to difficulties comparing results both *within* and also *across* secondary studies and difficulties reporting trends over time, e.g., a supposed growth in the number of case studies being conducted. As one example, SLS P11 states that, "...surveys have similar distributions to case studies, but the number of surveys is about half the number of case studies..." Our findings challenge both SLS P11's observation on similar distributions as well as its observation on proportions. As another example of the problem of comparison, in Table 7 we sum and average the counts and percentages of case studies reported by SLSs, yet comparing across the SLSs is problematic because of the misclassification by the SLSs (and also because of the variation in number of research methods used by SLSs).

In terms of reporting trends, SLS P59 presents a bubble grid showing the number of studies per research approach over a thirteen year period. The

publication of case studies per year has grown from one in 2007 to seven in 2017 and then jumps to nineteen in 2018. Our estimates of correctness in Table 7 suggest that SLS P59 is the most correct, of the eleven SLSs, in its classification of case studies, though we estimate a 1/3 of SLS P59's studies are still incorrectly classified as "case study". This misclassification might have a particularly significant impact on the reported "growth" between 2007-2017.

6.3.3. Quality and quantity of evidence

Some SLSs organise their findings according to research methods. SLSs P11, P44 and P45 are particularly good examples. We use one main example, from SLS P44 [34], to illustrate the impact of organising findings by research method, and complement that example with brief comments about other examples. Before doing that, we first consider the nature of evidence presented in case studies.

The nature of evidence reported in case studies, even the "best" case studies, makes it hard for an SLS to perform any kind of *quantitative* synthesis other than vote-counting. It is also hard to perform analytical (also called theoretical) synthesis, particularly with case studies that do not present any kind of theoretical framework. A consequence is that evaluating the actual impact of the mislabelling of primary studies is affected by the limits of the (so-called) case studies. Incorporating studies that are not actually case studies further complicates the evaluation, since these studies are likely to have less rigorous evidence and theories. Mixing case studies and non-case studies further reduces/limits the kinds of qualitative or quantitative synthesis that might be properly conducted. Mislabelling therefore distorts the body of evidence, e.g. for vote-counting, "confuses" the quality of that evidence, and complicates how this mixture of evidence might be synthesised.

Table 9 presents a simplified version of a table, Table 5, found in SLS P44 [34]. The original table enumerates a list of case studies that each contain a dataset relating to the respective six systems presented in Table 9. We simplify our version of the table to only show the number of case studies reported in the SLS (see column, *SLS #CS*). We add into the table the number of studies that we found to not be case studies (see *Not CS (us)*), the number that the original authors of the primary studies considered were not case studies (see *Not CS (PS)*) and the resultant reduction in genuine case studies and, consequently, datasets (see *Reduction*). Because we only *sampled* from SLS P44, and did not assess all of the primary studies in that

SLS, the estimates we present in Table 9 are minimum estimates. In other words, *at least* four of the nine supposed case studies for JHotDraw are not actually case studies. JHotDraw also provides a concrete example where SLS authors are "distorting" the evidence: for four of the primary studies, their authors did not consider their studies to be case studies, and we conclude the same, yet the SLS authors have re-labelled these four studies as case studies. For JFreeCart, the sample of nine primary studies that we drew from SLS P44 did not include either of the two studies presented for JFreeCart, so we cannot assess this row of the table. Finally, for JRefactory, we consider one of the two case studies to not be a case study whilst the original authors presented it as a case study. We recognise this difference of opinion through approximating the reduction.

Overall, the table illustrates the distorting effect of mislabelling case studies on the body of evidence available for some item (in this example, software systems) both in quantitative terms (see *Reduction*), but also in qualitative terms, e.g., as the number of independent datasets reduces, so we have less opportunity for independent corroboration or for synthesis.

Table 9: Re-analysis of Table 5 from SLS P44 [34].

System	SLS #CS	Not CS (us)	Not CS (PS)	Reduction
JHotDraw	9	4	4	4/9 (44%)
Eclipse	5	3	3	3/5 (60%)
ArgoUML	5	1	1	1/5 (25%)
Xerces-J	3	2	2	2/3 (66%)
JFreeCart	2	NA	NA	NA
JRefactory	2	1	0	$\approx$ 1/2 (50%)

There are two other tables in SLS P44 (Table 4 and Table 10) that also present findings specific to case studies. The findings of these two tables are therefore also challenged by our analysis.

As another example, SLS P45 presents nine tables that all include information about case studies, of which three tables (i.e., Table 4, Table 12 and Table 13 in SLS P45) are all based only on the case studies.

Finally, SLS P33 states of their primary studies, "Remarkably, only 10 (40%) of the 25 papers that used the Case Study research method reported validity threats." We have estimated that approximately 22% in our sample of the supposed case studies in SLS P33 are genuine case studies. Further details are available through the tables in Supplement 6 of the online supplementary material linked in Appendix A. Thus, SLS P33's observation about the number of studies reporting validity threats is unreliable.

6.3.4. Decision making

The effects of mislabelling on the quality and quantity of evidence have consequences for decision making by both researchers and practitioners. Taking Table 9 and SLS P44 as a convenient example, there would be appear to be much less actual evidence to inform decision making on these six systems than might originally appear. As another example, SLS P59 presents a bubble grid that maps research approaches, contribution types and publication domains. Case studies make up 57% of the research approaches, with Systematic Reviews ranked second at 11%, and the remaining seven approaches each with 7% or less. Lessons learned constitute 78% of the contribution types, with guidelines (themselves appearing to be form of lesson learned) ranked second at 7%, and the other three contributions each, again, with less than 7%. The bubble grid is not a correlation, of course (and also presents three factors: approach, contribution and domain) but the frequency of case studies, lessons learned and guidelines again demonstrates the association of case studies with aspects of decision making, in this case lessons learned and guidelines. More generally, a distorted body of evidence can erroneously suggest there is more evidence, or more reliable evidence, for an intervention in software practice.

7. Limitations

In terms of limitations to our research, we consider four research-design decisions we made that introduce limitations to our study.

First, we did not use multiple databases in our search for SLSs. We only used one database for identifying a set of SLSs. Scopus was chosen given that it has good coverage of publications from different publishers. Several researchers (e.g., [23, 24, 25]) highlight Scopus as a viable option. We chose to use one database because our objective was to focus on a potential problem, i.e., the labelling of primary studies as "case study" by SLSs. Using multiple databases would most likely not affect the outcome of our research, since other databases are not likely to contain substantially different kinds of SLS. We chose different search strings for the two sub-samples to mitigate threats related to the search strategy, search engine, and the search string formulation.

Ultimately, we are interested in an indicative sample of SLSs in software engineering reporting primary studies being classified as case study research.

Since our focus is on having an indicative sample to demonstrate a concern, we did not apply both search strings to all years investigated.

Second, the designs of our search strategies may have affected the findings. In the first search, we looked for "case study" in the title, abstract or keywords for articles published between 2010-2021. The first search strategy may have selected only those articles that identified specific results relating to case studies, for example, the frequency of case studies in software engineering.

To investigate and mitigate this risk, we chose a second search strategy when investigating studies published in 2022. There is a difference between our two samples concerning the percentages of mislabelling, although the problem is the same independent of the search strategy. Thus, the percentages may be affected by our choices, but not the overall concern of case studies being mislabelled in SLSs.

Third, we did not systematically perform any quality assessment of the SLSs identified, though we have examined the *designs* of the SLSs, and thus considered some aspects of quality. Our objective was not to assess the SLSs of highest-quality, but instead to assess a sample of what is published in software engineering. All SLS articles analysed are published in well-established software engineering journals (nine are published in I&ST which explicitly encourages SLSs) and ought to be indicative of how case studies are treated in SLSs. Our categorisation of the SLSs worked as a filtering mechanism to ensure we looked at SLSs that included specific information concerning what the community refer to as being case studies. Where it was feasible to do so, we have analysed all of the categories of SLSs.

Fourth, we based our assessment on one definition of *case study*, and one definition of *case*. We recognise there are other definitions and opinions. Our starting point was Yin´s definition [13], which Runeson et al. [14] view as suitable for SE research. Based on that starting point, we chose to use the *refined* definition of Wohlin [3], given that it distinguishes five components for a case study, and given that a checklist has been formulated from the definition [4], which we used in our assessment.

Finally, we based our assessment on consensus discussions after independent review by the two authors. When we had different interpretations of an article, we chose to be generous and accept a primary study as presenting a case study. We have reported an agreement index for the categorisation of SLSs, but not for the assessment of the primary studies since the individual assessments were primarily seen as a basis for the consensus discussions. One

consequence of our generosity is that we may be *over-reporting* the frequency of actual case studies.

8. Conclusions

8.1. Summary of findings

It has been known for almost twenty years that authors of primary studies of software engineering do not consistently use the term and label, "case study", when reporting their studies. Systematic literature studies (SLSs), such as systematic literature reviews and systematic mapping studies, provide the opportunity to correct such misreporting, e.g., to re-label a primary study as not a "case study" when it does not conform to the established definitions for case study.

It was not clear whether SLSs do indeed make corrections to the labelling of primary studies. We therefore investigated the following research question: For those SLSs that report the empirical research methods of primary studies, do those SLSs correct mislabelled case studies, propagate already mislabelled case studies, or exacerbate the problem by introducing new mislabels?

Through a systematic search and categorisation of SLSs, we identified eleven SLSs and studied 79 primary studies drawn by stratified random sampling, from all eleven SLSs.

For the sample of primary studies, their authors do not claim that their study is a case study in 29 of the articles. However, they do claim incorrectly that it is a case study for 62% of the articles (31/50). This could be compared with previous studies reporting a mislabelling of about 50% [3, 4]. However, based on these 79 primary studies we find that SLSs restate the correct label of "case study" in 24% of instances, but misclassify studies in 76% of instances, either propagating an already incorrect label used by the respective primary study (39%) or introducing a new mislabel for a primary study (37%). 76% is a point estimate across all 79 studies. Considering averages *per sub-sample*, we find, respectively, 60% and 81% of the primary studies are mislabelled. Thus, we report a point estimate of 76% with a range estimate of between 60% and 81%.

We found eight of the eleven SLSs did not define the term, "case study". Of the three SLSs that provide a definition, none were entirely consistent with the established definitions for "case study" in software engineering research, though two were partially consistent. Thus, part of the explanation

for why SLSs misclassify a primary study may be that SLS authors are not considering accepted standards for research methods in their analyses.

We also briefly considered significant implications arising from the mislabelling: it leads to difficulties with estimations and comparisons; it distorts the body of evidence, "confuses" the quality of evidence, and complicates the synthesis of evidence; and it undermines the confidence we can have in recommendations arising from SLSs, e.g., for interventions in practice.

8.2. Recommendations and checklist

Credible evidence, i.e., the validity and relevance of the evidence, is essential for decision-making [57]. One vital aspect of credible evidence is how the evidence was acquired, for example, which research method was used to obtain the evidence.

On the basis of our findings and the need for credible evidence, we make several recommendations and propose a simple checklist. The checklist is intended to complement the recommendations, since the checklist can apply to editors, reviewers and readers, as well as SLS authors during the preparation of their articles.

In terms of the recommendations:

1. Those who conduct primary studies and SLSs need to better understand research methodology, and need to apply the terminology correctly, i.e., according to the definitions.
2. To support future syntheses of credible evidence in SLSs, primary study authors should write for synthesis [58], e.g., stating clearly their research method.
3. SLS authors must be evidence gate-keepers. It is essential that SLS authors ensure that evidence is presented in such a way that readers can determine the credibility of the evidence in their context, whether being for research or in practice. Thus, SLS authors should: a) check and report the research method claimed by the authors of the primary studies, as well as b) check and report their own assessment of the research method; and c) specify, or provide references to, the research method definitions they have used.
4. SLS authors should use a sufficiently comprehensive research method classification (preferably an already published one) that can accommodate the diversity of primary studies, and so avoid "forcing" studies into an overly simplistic classification.

5. Reviewers and editors need to more carefully review the manuscripts of both primary studies and SLSs concerning the research method claimed by the authors, as the current practices in the review process tacitly endorse low/er standards of quality assessment and mislabelling of research methods.
6. To reassure themselves of (but not guarantee) the reliability and validity of an SLS, readers of SLSs – which may include professional practitioners – should check, perhaps using the proposed checklist, whether the SLS authors have explicitly considered the problem of mislabelling.
7. Professional practitioners, and researchers working closely with industry should be particularly aware that any of the SLS's recommendations based on aggregations or syntheses of results from so-called case studies may not be reliable or valid, in particular such studies may not have been conducted in a contemporary, real-world setting.

In terms of the proposed checklist for checking whether SLSs have properly labelled primary studies:

1. Does the SLS report the research methods of each primary study?
2. Do the SLS authors report their own classification of the primary study's research methods?
3. Does the SLS formally assess the correctness of the labelling of research methods by the primary-study authors?
4. Does the SLS show a clear mapping of primary-study authors' and the SLS authors' classification of each primary study's research method, e.g., as a table or similar?
5. Does the SLS use a sufficiently comprehensive classification of research methods, and preferably an already-published classification?
6. Does the SLS provide a set of definitions of research methods, or clearly cite and apply a reference set of definitions?
7. Does the SLS explicitly recognise that there may be a difference of classification of research method, between the primary-study authors' and the SLS authors' classification of a primary study?
8. Does the SLS explain the reason, or reasons, for any differences in classification between primary-study authors' and the SLS authors' classification?

8.3. Further research

In terms of further research, we consider three directions here. First, our point estimate of 24% (19% and 40% for the two sub-samples) for correctly labelled case studies in software engineering is substantially lower than our previous estimate [3, 4] of 50%, where we directly sampled primary studies. The main reason for the differences in percentages is that in the current study we look at the misclassification by SLSs. The percentage of correctly labelled case studies by the primary-study authors is 38%. One direction for further research is therefore to better understand the reasons behind mislabelling by the primary-study authors, restatement of mislabelling by SLS authors and, probably most importantly, the misclassifications added by the SLS authors. A second direction is to replicate our study, both as a literal replication (do others find the same results with the same data?), and theoretical replication (do others find corroborating results with different data?). A third direction is to continue to examine whether the phenomenon of mislabelling occurs for different kinds of labels. An obvious example would be other research methods not investigated yet, e.g., Ayala et al. [21] observe a similar situation with the mis/use of the label "experiment", but this mislabelling may recur for other kinds of labels used by SLSs, such as different types of requirement or testing, or labels for agile practices, or types of defect.

8.4. Concluding remarks

Overall, we conclude that case studies are substantially over-reported in the literature, i.e., there are far fewer case studies actually conducted than are reported as being conducted. Our analysis is based on the definition by Wohlin [3]. As mentioned in Section 5.1, the two most common reasons for a study not being a case study are: the study is not conducted in a real-life context or it is not contemporary. These two aspects are common across the different definitions of case study research. Thus, the findings are not a consequence of our choice of definition. Furthermore, SLSs are both propagating and further exacerbating the problem of the mislabelling of primary studies as "case studies", rather than – as we should expect of SLSs – improving the labelling of primary studies, and thus improving the body of credible evidence.

Acknowledgement

We thank the anonymous reviewers and the editor handling the paper for their time and effort. We appreciate their constructive feedback, which helped us improve the article.

Appendix A. Supplementary material

The following is the supplementary material related to this article.

- Supplement S1 – SLSs categorised (**Link to be provided by the publisher**)
- Supplement S2 – Studies excluded from our SLS analyses (**Link to be provided by the publisher**)
- Supplement S3 – Explanation of batches (**Link to be provided by the publisher**)
- Supplement S4 – Comments on the mislabelling of case studies (**Link to be provided by the publisher**)
- Supplement S5 – Listing of primary studies (**Link to be provided by the publisher**)
- Supplement S6 – Full analysis of primary studies (**Link to be provided by the publisher**)

References

[1] C. Zannier, G. Melnik, F. Maurer, On the success of empirical studies in the international conference on software engineering, in: Proceedings International Conference on Software Engineering, 2006, pp. 341–350.

[2] P. Runeson, M. Höst, Guidelines for conducting and reporting case study research in software engineering, Empirical Software Engineering 14 (2) (2009) 131.

[3] C. Wohlin, Case study research in software engineering—it is a case, and it is a study, but is it a case study?, Information and Software Technology 133 (2021) 106514.

[4] C. Wohlin, A. Rainer, Is it a case study?—a critical analysis and guidance, Journal of Systems and Software 192 (2022) 111395.

[5] R. K. Yin, Case study research: Design and methods, 3rd Edition, Sage Publications, 2003.

[6] A. Rainer, C. Wohlin, Case study identification: A trivial indicator outperforms human classifiers, Information and Software Technology 161 (2023) 107252.

[7] S. Jalali, C. Wohlin, Systematic literature studies: database searches vs. backward snowballing, in: Proceedings of the ACM-IEEE international symposium on Empirical software engineering and measurement, 2012, pp. 29–38.

[8] C. Wohlin, Guidelines for snowballing in systematic literature studies and a replication in software engineering, in: Proceedings of the 18th international conference on evaluation and assessment in software engineering, 2014, pp. 1–10.

[9] D. Badampudi, C. Wohlin, K. Petersen, Experiences from using snowballing and database searches in systematic literature studies, in: Proceedings of the 19th international conference on evaluation and assessment in software engineering, 2015, pp. 1–10.

[10] B. A. Kitchenham, D. Budgen, P. Brereton, Evidence-based Software Engineering and Systematic Reviews, Vol. 4, CRC press, 2015.

[11] K. Petersen, S. Vakkalanka, L. Kuzniarz, Guidelines for conducting systematic mapping studies in software engineering: An update, Information and Software Technology 64 (2015) 1–18.

[12] V. Basili, L. Briand, D. Bianculli, S. Nejati, F. Pastore, M. Sabetzadeh, Software engineering research and industry: a symbiotic relationship to foster impact, IEEE Software 35 (5) (2018) 44–49.

[13] R. K. Yin, Case study research and applications: Design and methods, 6th Edition, Sage Publications, 2018.

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

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

[16] J. M. Verner, J. Sampson, V. Tosic, N. A. A. Bakar, B. A. Kitchenham, Guidelines for industrially-based multiple case studies in software engineering, in: Proceedings Third International Conference on Research Challenges in Information Science, 2009, pp. 313–324.

[17] I. Benbasat, D. K. Goldstein, M. Mead, The case research strategy in studies of information systems, MIS Quarterly 11 (3) (1987) 369–386.

[18] C. Robson, Real World Research : A Resource for Social Scientists and Practitioner-researchers, Wiley Publishing, 2002.

[19] C. Baker, J. Wuest, P. N. Stern, Method slurring: the grounded theory/phenomenology example, Journal of Advanced Nursing 17 (11) (1992) 1355–1360.

[20] N. Haslam, Concept creep: Psychology's expanding concepts of harm and pathology, Psychological Inquiry 27 (1) (2016) 1–17.

[21] C. Ayala, B. Turhan, X. Franch, N. Juristo, Use and misuse of the term "experiment" in mining software repositories research, IEEE Transactions on Software Engineering 48 (11) (2021) 4229–4248.

[22] K.-J. Stol, P. Ralph, B. Fitzgerald, Grounded theory in software engineering research: A critical review and guidelines, in: Proceedings of the 38th International Conference on Software Engineering, 2016, pp. 120–131.

[23] O. Dieste, A. Grimán, N. Juristo, Developing search strategies for detecting relevant experiments, Empirical Software Engineering 14 (5) (2009) 513–539.

[24] V. Garousi, M. V. Mäntylä, Citations, research topics and active countries in software engineering: A bibliometrics study, Computer Science Review 19 (2016) 56–77.

[25] E. Mourão, J. F. Pimentel, L. Murta, M. Kalinowski, E. Mendes, C. Wohlin, On the performance of hybrid search strategies for systematic literature reviews in software engineering, Information and Software Technology 123 (2020) 106294.

[26] D. Budgen, P. Brereton, N. Williams, S. Drummond, The contribution that empirical studies performed in industry make to the findings of systematic reviews: A tertiary study, Information and Software Technology 94 (2018) 234–244.

[27] S. M. Melo, J. C. Carver, P. S. Souza, S. R. Souza, Empirical research on concurrent software testing: A systematic mapping study, Information and Software Technology 105 (2019) 226–251.

[28] A. Ampatzoglou, I. Stamelos, Software engineering research for computer games: A systematic review, Information and Software Technology 52 (9) (2010) 888–901.

[29] D. Tofan, M. Galster, P. Avgeriou, W. Schuitema, Past and future of software architectural decisions–a systematic mapping study, Information and Software Technology 56 (8) (2014) 850–872.

[30] N. H. Bakar, Z. M. Kasirun, N. Salleh, Feature extraction approaches from natural language requirements for reuse in software product lines: A systematic literature review, Journal of Systems and Software 106 (2015) 132–149.

[31] W. Bissi, A. G. S. S. Neto, M. C. F. P. Emer, The effects of test driven development on internal quality, external quality and productivity: A systematic review, Information and Software Technology 74 (2016) 45–54.

[32] J. L. Barros-Justo, F. Pinciroli, S. Matalonga, N. Martínez-Araujo, What software reuse benefits have been transferred to the industry? a systematic mapping study, Information and Software Technology 103 (2018) 1–21.

[33] J. Kroll, I. Richardson, R. Prikladnicki, J. L. Audy, Empirical evidence in follow the sun software development: A systematic mapping study, Information and Software Technology 93 (2018) 30–44.

[34] F. Wedyan, S. Abufakher, Impact of design patterns on software quality: a systematic literature review, IET Software 14 (1) (2020) 1–17.

[35] A. E. Chacón-Luna, A. M. Gutiérrez, J. A. Galindo, D. Benavides, Empirical software product line engineering: a systematic literature review, Information and Software Technology 128 (2020) 106389.

[36] Ö. Uludağ, P. Philipp, A. Putta, M. Paasivaara, C. Lassenius, F. Matthes, Revealing the state of the art of large-scale agile development research: A systematic mapping study, Journal of Systems and Software (2022) 111473.

[37] A. R. Amna, G. Poels, Ambiguity in user stories: A systematic literature review, Information and Software Technology 145 (2022) 106824.

[38] N. Dissanayake, A. Jayatilaka, M. Zahedi, M. A. Babar, Software security patch management-a systematic literature review of challenges, approaches, tools and practices, Information and Software Technology 144 (2022) 106771.

[39] C. Robson, Small-scale evaluation: Principles and practice, Sage Publications Ltd, 2017.

[40] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, A. Wesslén, Experimentation in software engineering, Springer Science & Business Media, 2012.

[41] S. Easterbrook, J. Singer, M.-A. Storey, D. Damian, Selecting empirical methods for software engineering research, in: F. Shull, J. Singer, D. I. Sjøberg (Eds.), Guide to advanced empirical software engineering, Springer, 2008, pp. 285–311.

[42] Z. Li, P. Liang, P. Avgeriou, Application of knowledge-based approaches in software architecture: A systematic mapping study, Information and Software Technology 55 (5) (2013) 777–794.

[43] F. Elberzhager, A. Rosbach, J. Münch, R. Eschbach, Reducing test effort: A systematic mapping study on existing approaches, Information and Software Technology 54 (10) (2012) 1092–1106.

[44] J. Bailey, D. Budgen, M. Turner, B. Kitchenham, P. Brereton, S. Linkman, Evidence relating to object-oriented software design: A survey, in: First International Symposium on Empirical Software Engineering and Measurement, 2007, pp. 482–484.

[45] R. Wieringa, N. Maiden, N. Mead, C. Rolland, Requirements engineering paper classification and evaluation criteria: a proposal and a discussion, Requirements Engineering 11 (1) (2006) 102–107.

[46] K. Petersen, R. Feldt, S. Mujtaba, M. Mattsson, Systematic mapping studies in software engineering, in: 12th International Conference on Evaluation and Assessment in Software Engineering, 2008, pp. 1–10.

[47] V. Berg, J. Birkeland, A. Nguyen-Duc, I. O. Pappas, L. Jaccheri, Software startup engineering: A systematic mapping study, Journal of Systems and Software 144 (2018) 255–274.

[48] P. Rodríguez, A. Haghighatkhah, L. E. Lwakatare, S. Teppola, T. Suomalainen, J. Eskeli, T. Karvonen, P. Kuvaja, J. M. Verner, M. Oivo, Continuous deployment of software intensive products and services: A systematic mapping study, Journal of Systems and Software 123 (2017) 263–291.

[49] M. Unterkalmsteiner, T. Gorschek, A. M. Islam, C. K. Cheng, R. B. Permadi, R. Feldt, Evaluation and measurement of software process improvement—a systematic literature review, IEEE Transactions on Software Engineering 38 (2) (2011) 398–424.

[50] L. Chen, M. A. Babar, A systematic review of evaluation of variability management approaches in software product lines, Information and Software Technology 53 (4) (2011) 344–362.

[51] P. Brereton, B. Kitchenham, D. Budgen, Z. Li, Using a protocol template for case study planning, in: 12th International Conference on Evaluation and Assessment in Software Engineering, 2008, pp. 1–8.

[52] M. Höst, P. Runeson, Checklists for software engineering case study research, in: First International Symposium on Empirical Software Engineering and Measurement, IEEE, 2007, pp. 479–481.

[53] R. K. Yin, Case study research: Design and methods, 5th Edition, Sage Publications, 2009.

[54] R. L. Glass, I. Vessey, V. Ramesh, Research in software engineering: an analysis of the literature, Information and Software Technology 44 (8) (2002) 491–506.

[55] C. Wohlin, A. Aurum, Towards a decision-making structure for selecting a research design in empirical software engineering, Empirical Software Engineering 20 (2015) 1427–1455.

[56] D. S. Cruzes, T. Dybå, Research synthesis in software engineering: A tertiary study, Information and Software Technology 53 (5) (2011) 440–455.

[57] C. Wohlin, A. Rainer, Challenges and recommendations to publishing and using credible evidence in software engineering, Information and Software Technology 134 (2021) 106555.

[58] C. Wohlin, Writing for synthesis of evidence in empirical software engineering, in: Proceedings of the 8th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement, 2014, pp. 46:1–4.