The role and value of replication in empirical software engineering results

Martin Shepperd and Steve Counsell

Brunel Software Engineering Lab (BSEL)
Dept. of Computer Science
Brunel University London
UB8 3PH, UK

Nemitari Ajienka

Dept. of Computer Science
Edge Hill University
Ormskirk, L39 4QP, UK

Abstract

Context: Concerns have been raised from many quarters regarding the reliability of empirical research findings and this includes software engineering. Replication has been proposed as an important means of increasing confidence.
Objective: We aim to better understand the value of replication studies, the level of confirmation between replication and original studies, what confirmation means in a statistical sense and what factors modify this relationship.
Method: We perform a systematic review to identify relevant replication experimental studies in the areas of (i) software project effort prediction and (ii) pair programming. Where sufficient details are provided we compute prediction intervals.
Results: Our review locates 28 unique articles that describe replications of 35 original studies that address 75 research questions. Of these 10 are external, 15 internal and 3 internal-same-article replications. The odds ratio of internal to external (conducted by independent researchers) replications of obtaining a 'confirmatory' result is 8.64. We also found incomplete reporting hampered our ability to extract estimates of effect sizes. Where we are able to compute replication prediction intervals these were surprisingly large.
Conclusion: We show that there is substantial evidence to suggest that current approaches to empirical replications are highly problematic. There is a consensus that replications are important, but there is a need for better reporting of both original and replicated studies. Given the low power and incomplete reporting of many original studies, it can be unclear the extent to which a replication is confirmatory and to what extent it yields additional knowledge to the software engineering community. We recommend attention is switched from replication research to meta-analysis.

Keywords: software engineering, experiment, reliability, replication,

meta-analysis.

> "no single study is a pure reflection of the underlying truth" – Spence and Stanley [102]

1. Introduction

Concerns about the reliability of empirical research results are fast becoming endemic and software engineering is no exception [94, 43]. Central to this has been the seminal paper of Ioannidis [38] in bringing the question of false discoveries to our attention and by illustrating how likely published experiments report erroneous results. Additionally, other researchers have questioned the prevalence of reported p-values just below the customary threshold of significance [105]. Still others have expressed concern about experimental design influencing results [61] and the variability of results depending upon which research team performs the work [95]. The situation is exacerbated by publication bias and the 'file drawer' problem [87] when studies are selectively published based on preferences for particular results (usually 'positive' ones). Finally, there is the concern that many software engineering experiments are seriously under-powered [25, 43], that is there is both a low probability of discovering a true effect and the parameter of interest has high variance[1].

To remedy the uncertainty surrounding false discoveries, researchers in software engineering have advocated the use of replications for over three decades [10, 81]. The idea is that an empirical result that can be replicated is more trustworthy. This has gained considerable traction and a series of systematic reviews [99, 62] have located 135 articles, just up until 2012, describing replication experiments within the domain of empirical software engineering.

Nevertheless, in part influenced by problems such as the perceived "replication crisis"[2] in psychology [78], and the seeming low levels of agreement between primary studies and replication studies in software engineering (e.g., Sjøberg et al. report that barely 50% of differentiated replications were confirmatory [101]) there is now considerable concern about the state of health of our empirical research, particularly experimental research.

Therefore the goal of this study is to provide evidence concerning the contribution of replication studies, the level of agreement or confirmation between replication and original studies and what confirmation means in a more objective and statistical sense. The underlying proposition is that confirmation implies greater reliability and confidence in the result. However, if we find that

[1] It is the high variance associated with under-powered studies, coupled with selective reporting that can lead to systematic over-estimation of effect sizes[39].

[2] The Open Science Collaboration reported in *Science* [78] the lack of reproducible results in experimental psychology after conducting 100 replications of experiments published in leading psychology journals during 2008; only 39% of effects were subjectively rated to have been replicated.

whether the replication is conducted by an independent team of researchers or not, modifies this relationship (cf. [101, 99]) then it might suggest there are additional sources of bias which unfortunately then reduce our confidence in the empirical results. Of course there is natural variability[3] between studies and effect sizes that are found. However, a pattern might be a cause for concern, hence our systematic review which endeavours to find all relevant replication studies in the fields of software project effort prediction and pair programming.

To explore these issues in detail, we consider two problem domains within software engineering. These are software project effort prediction where the experimental units are software project data and then, pair programming where the experimental units are human participants, specifically programmers. Effort prediction was chosen as a well defined sub-domain of empirical software engineering based on computational experiments, something that is growing in importance with the advent of modern machine learning algorithms. Pair programming was included as a clearly defined sub-domain and one that introduces non-trivial experimental design challenges such as learning effects and ordering effects. Finally, pair programming provides a contrast with the effort prediction by offering a human-centric replication area.

In our review we include both experiments and quasi-experiments that apply treatments (although the treatment may not be randomly allocated) to experimental units. We focus on replication studies where the research question is fixed but aspects of the experiment such as participants may be changed. We exclude reproducibility studies, where the purpose is principally to check for errors of commission for two reasons. First, because the concept cannot be applied to experiments using human participants. Second, because the issues surrounding reproducibility concern such matters as sharing scripts, providing stable software platforms and so forth. In contrast we are interested in assessing the reliability of experimental results. See Section 2 for a fuller discussion.

This paper makes the following contributions:

1. We conduct systematic reviews (up until August 2017) that aim to locate all published, refereed articles that replicate (a) software project effort prediction and (b) paired programming experiments.
2. We compare the outcomes of internal replications (i.e. where one or more authors is in common with the original study) and external replications (where the authors are independent) and show how this is related to the outcome (confirmation or disconfirmation).
3. We show there is a good deal of diversity in how replications are conducted, and widespread problems with reporting, not least in terms of replication goals and expectations. This reduces the value of the replication.
4. Next we show how prediction intervals are larger than might be expected, with the implication that original studies may be both easier to replicate

[3]Note that this natural variability due to sampling and measurement error can be considerably greater than researchers often appreciate [23].

than might be expected but the meaning, in terms of contribution to knowledge, less than one might hope.

5. Finally we argue that empirical software engineering would benefit from better reported studies and that meta-analysis is more likely to contribute to our understanding than replication, particularly when the original study is under-reported or under-powered.

The remainder of the paper is organised as follows. The next section considers the problem of how to define a replication study and then goes onto address how replication research is undertaken within software engineering. Then Section 3 describes our systematic review and how it was conducted. Section 4 describes the overall results from the review and then applies various meta-analyses to explore factors (type of replication, year and publication venue) that determine replication outcomes. This is followed by a more general discussion in Section 5 and we conclude in Section 6 by summarising our findings and making some recommendations about the conduct of future replications. The appendix contains descriptions of the included Replication and Original Study articles.

2. Related Work

2.1. Replications and Reproductions

Defining what constitutes a replication study is central to our analysis, but unfortunately is not straightforward. A comprehensive review of replication classification schemes across many disciplines by Gómez et al. identified 20 different taxonomies and approximately 70 replication types [28, Table 2]. These were grouped into three categories, namely:

Group I: which essentially involve faithful replication of the original experiment with no, or minimal, changes.

Group II: for these replications there will be some variation from the original experiment which could include "measurement instruments, metrics, protocol, populations, experimental design or researchers" [28].

Group III: here the "theoretical structure, i.e. they share the same constructs and hypotheses" [28] is all that is in common between the original and replicated experiment. These are sometimes referred to as 'conceptual' replications.

In our view Groups II and III lie upon a continuum and there is no obvious rule to determine when a Group II replication becomes a Group III replication. Nor is it clear how useful such a distinction would be. Discussing how faithful a replication study must be, Miller states:

> "replication in software engineering is fated to be further removed from exact duplication than traditional sciences. Is this important?

> The short answer is 'no'. What is important is that the replication examines the same (or a generalised or specialised version of the same) hypothesis." [73]

This highlights the distinction between reproducing an experiment, which should be as faithful as possible, with replication. In the case of the former, the goal is determining whether there have been errors in commission and/or reporting. For the latter, the goal is addressing confidence and generalisability.

Beyond software engineering, Peng [80] considers some of the specifics of computational experiments and also makes this distinction between reproducibility and replication[4]. Therefore, we simply distinguish between reproducing an experiment and replicating an experiment. Reproducibility concerns the validity of the original experiment including data and algorithm correctness. By contrast, for an experiment to be considered a replication we require the following:

- The authors must explicitly state which original experiment is being replicated.
- The purpose of the replication study includes extending the external validity of the experiment (i.e., adding to our understanding of how the results generalise).
- Both experiments must have research questions or hypotheses in common (i.e., there are shared constructs and interventions).
- The analysis must contain a comparison of original and new results with a view to confirming or disconfirming the original experiment. Note that we intentionally avoid judgements such as 'successful'.

Replications may be categorised as internal (where the replication team includes members from the original experiment and could be published in a single article or several over time) and external (where the entire replication team are independent of the original experiment) [16, 73]. A number of commentators indicate a preference for external replications as being more independent, e.g., [16, 45], however, there is the potential downside in that the replication may be unintentionally less exact [98].

2.2. What Constitutes a 'Successful' Replication?

An important question—but not one that seems to have been widely considered in software engineering—is what constitutes a 'successful' or the less value-laden

[4] Although there are dissenting viewpoints, e.g., the debate between Shull et al. [98] and Kitchenham [45] concerning the benefits of 'exact' replications, we adopt the more pragmatic approach of focusing on both Group II and Group III replications where it is the research hypothesis or question that is being replicated. To do otherwise severely restricts the number of such studies and also has implications upon external validity.

term confirmatory replication? It may be that researchers consider the answer is self-evident hence objective decision processes are seldom articulated. Be that as it may, this section explores how researchers from other problem domains have addressed this question.

Except in the case of reproducing the results of a previous study, where one might hope to find complete agreement, researchers do not expect to find identical results [102]. So how much might a replication deviate from the original study and still be considered a confirmation? As Spence and Stanley put it "the question is not if deviation across studies is permissible, but instead 'how much deviation is permissible?' " [102].

An obvious and common approach is to use p-values and null hypothesis significance testing (NHST). If the calculated p-value falls below a threshold (commonly this is $\alpha = 0.05$ possibly with correction for multiple tests) then the effect is deemed to be 'statistically significant' so one would expect the replication to have similar findings if it is 'successful'. Unfortunately this is mistaken. As Amrhein et al. state "significance ($p \leq 0.05$) is hardly replicable: at a good statistical power of 80%, two studies will be 'conflicting, meaning that one is significant and the other is not, in one third of the cases if there is a true effect" [5]. An additional, and not widely appreciated, problem is that if the null hypothesis is true then p becomes a random variable following a uniform distribution [76] which means all values of p are equally likely.

In addition, it has been widely conjectured that the all or nothing nature of NHST contributes to reporting biases [43]. This is compounded by the problem that "researchers typically have so many 'researcher degrees of freedom' — unacknowledged choices in how they prepare, analyse, and report their data — that statistical significance is easily found even in the absence of underlying effects" [60].

Instead, Jørgensen et al. [43] suggest researchers should focus on the replication of effect size rather than statistical significance, a sentiment with which we agree. It is well known that effect size and significance are not necessarily related and from a practical perspective the effect is what is of interest [26]. The questions remains how similar must effect sizes be? And indeed how immune are they from some of the biases and problems of selective reporting we have already discussed.

Researchers are generally encouraged to report confidence limits around effect sizes since we still expect sampling error [5] and measurement error [60] (assuming all other sources of bias are dealt with). However, in the context of replication we actually require the *prediction interval* [23, 102] since a confidence interval relates to the estimate of the population effect size whereas we are concerned with the estimate from the specific study being replicated. Note also that whilst the potential sources of error in research are many and varied (e.g., measurement error and publication bias), as Spence and Stanley [102] observe, sampling error will always be present.

A prediction interval is the range of results that might be expected in a replication due to chance from sampling error. The idea is the original study contributes the first n_1 observations of the effect. Using their known variance

how would we expect the next n_2 observations contributed by the replication study to behave assuming they are all sampled from the same underlying population. Generally prediction limits tend to be wider than confidence intervals and can be considerably wider than might be expected. The key point is they may be calculated using the n_1 observations and therefore *before* the replication is conducted. This should then influence how we interpret the outcome of a replication since a confirmatory replication study should yield a result that falls within a prediction interval computed from the original study. Normally the 95% prediction interval should be employed.

It has been argued that this has contributed to the so-called replication crisis and that we have good reason to expect a wider spread of results than has been customary [104]. Patil et al. [79] found that although it was reported that whilst only 36% of the Open Science initiative replications [78] were considered to be confirmatory, they showed that 77% of the replication effect sizes reported were within a 95% prediction interval calculated using the original effect size. Similarly Spence and Stanley discuss the frequently underestimated impact of sampling [102] and measurement error [103]. Therefore we can see the great importance of properly defining replication expectations as a more statistically based view leads to a doubling in the number of confirmations due to small sample sizes, large variances and generally small effect sizes.

Helpfully Spence and Stanley have created a prediction interval package for `R` called `predictionInterval` [102] and provide online calculators, the most useful one for our purposes being for d-values, i.e., standardised mean differences ($(m_1 - m_2)/s$ where is s is some pooled estimate of the standard deviation) at `https://replication.shinyapps.io/dvalue/`.

2.3. Replications in Software Engineering

Similar to many other empirical disciplines, replication has been seen as an important means of assessing reliability and confidence in empirical findings [81, 82, 11, 73]. For this reason da Silva et al. [99] conducted a comprehensive mapping study of software engineering replication studies that identified 96 articles based upon replication studies. This has been extended by Bezerra et al. [12] which covered replication studies up until 2012.

The dominant paradigm for determining whether a replication confirms the original study is null hypothesis significance testing (NHST). In other words a replication is concordant if both it and the original study report a statistically significant effect, presumably in the same direction. Additionally it would be expected that the effect is broadly similar. However, the basis for comparison is seldom articulated and finding some operational definition for similarity, in particular *in advance*, has not been articulated in any software engineering replication study to the best of our knowledge.

However, despite much agreement on the importance of replication and a growth in the number of such studies some challenges have been identified. For example, Gómez et al. [28] discuss some of the problems of reaching a consensus on terminology and what actually constitutes a replication. Mende [63]

describes two case studies where he re-visits two published, software defect prediction experiments and sought to reproduce (as well as extend) their results. He identified a number of minor details that rendered reproduction of the results challenging particularly for one study and concluded that full publication of scripts as well as data is important. Likewise, our two systematic reviews described in Section 3 reveal problems of under-reporting which restricts the opportunities for further analysis.

3. Systematic Review

Systematic reviews have been widely adopted in software engineering over the past decade. The goal is to locate *all* articles relevant to the given research question. The process is guided by an explicit protocol which enables the repeatability of the review. An in depth description of the methods and techniques can be found in the recent handbook authored by Kitchenham, Budgen and Brereton [46].

Thus a systematic review is an appropriate means of finding studies relevant to the following questions:

1. To what extent do original and replication studies agree?
2. How do researchers interpret agreement and how does this align with a statistical view?
3. Do internal/external replication, date and publication venue influence replication outcomes?

As indicated in the Introduction, given the breadth and diversity of research in empirical software engineering, we restrict the Review to experiments relating to (i) software project effort prediction and (ii) pair programming. This enables a more in depth analysis and comparison of two sub-areas. Table 1 provides a summary of the conduct of the two reviews; Sections 3.1, 3.2 and 3.3 provide more details.

3.1. Searching for effort prediction primary studies

Our first task was to locate candidate studies. Note that our search was for published *articles* that might contain one or more *replications*. A replication is applied to an *original* study. We distinguish between *internal* and *external* replications where an internal replication has at least one author in common with the original study. An internal replication can be within a single article or refer to a previously published article (we refer to the former as *Internal-SameArticle*).

We are fortunate in that a comprehensive mapping study of software engineering replication studies has been published by da Silva et al. [99] which covers replication studies up until 2010. They identified 96 articles that reported on 133 replication studies based on 72 original studies. Almost 70% of the replications were published after 2004 and of these 70% were internal replications. This was subsequently updated to cover 2011-12 in which a further 39 articles were identified yielding a total of 135 articles [12].

Systematic Review Characteristic	**Description**
Research question	RQ1: To what extent do original and replication studies agree? RQ2: What does confirmation mean in a statistical sense? RQ3: Do internal/external replication, date and publication venue influence replication outcomes?
Motivation	To understand the effectiveness of replications to contribute to empirical knowledge and to explore ways to improve our research methods and experimental design
Target audience	Empirical software engineering researchers
Period covered	unbounded - August 2017
Quality	Only demonstrably refereed articles
General inclusion criteria	(i) An experiment (ii) Explicitly references study being replicated (iii) Comparison of results (iv) Article written in English and available (v) Describes one or more replications *not* reproductions
Protocol and raw data	`https://figshare.com/s/d1f2a035c3f62168b48a`
Effort Prediction Systematic Review	
Population	Experiments that apply treatments (i.e., different software project effort predictors) to experimental units i.e., software projects contained in datasets. The response variable is an estimate of prediction performance on unseen projects.
Sources used	(i) Previous mapping studies [99, 12] (ii) Google Scholar (iii) Scopus
Specific inclusion criteria	(i) Manipulates predictors (ii) Response variable = prediction performance
Pair Programming Systematic Review	
Sources used	(i) Google Scholar (ii) Scopus
Specific inclusion criteria	(i) Response variable = pair programming performance

Table 1: Systematic Review Overview

There are two main differences between the research of da Silva et al. and Bezera et al. and ours. First, they conducted mapping studies [22] which report on research activity in a given area rather than research findings, whereas we explore all relevant scientific evidence that bears upon our research question i.e., a systematic review [47]. Second, their focus covers all software engineering, whereas we are interested in software project effort prediction and pair programming experiments.

As has been noted by others such as de Magalhães et al. [62], one of the difficulties we encountered was that there is no consistent interpretation of the notion of replication. For inclusion in our review we required four things (as discussed in Section 2 on page 4). First, the authors must be explicit that they are replicating another study. Second, the purpose must be to extend our knowledge of the phenomenon under investigation (as opposed to verifying the correctness of the original study). Third, the basic experimental questions or hypotheses must remain unchanged[5], so this precludes introducing new or updated treatments (i.e., types of predictor), however this may be applied to new data (i.e, experimental units which are either software projects or programmers). Fourth, there must be an explicit comparison of results.

Although the initial search was performed by MS, this was independently checked by SC and NA. All contentious decisions were discussed amongst all authors. In detail our search was conducted in the order as follows:

1. we extracted all relevant articles from the da Silva et al. [99] mapping study of replications as a starting point **(9 articles)**
2. we then searched the 2015 update from Bezera et al. [12] **(1 article)**
3. forward chained (one level) from [99] to locate more recent replication articles **(no articles)**
4. backward chained (one level) from hit list derived from Steps 1 and 2 to identify any articles missed by [99, 12] **(3 articles)**
5. performed a full document google scholar search based on the first 200[6] results from 2013 onwards using the search string: `replication (cost OR effort) (prediction OR estimation) "software project"` **(7 articles)**
6. performed a Scopus search of title and abstract only from 2013 onwards using the search string: `"software project" AND replic* AND (prediction OR estimat*) AND (effort OR cost)` **(2 articles)**

The figures in parentheses are counts of new articles, over and above, those found by the previous steps. The above strategy yielded a total of $9 + 1 + 0 + 3 + 7 + 2 = 22$ articles. Since some of these 22 articles contained more than one replication study they pointed to 29 original articles that described the studies

[5]Note that a study might be included where some questions remain unchanged and new questions introduced in which case for the purposes of our review we ignore the new questions.

[6]Clearly the decision of how many results to examine is a subjective judgement, however, the fact that an exhaustive Scopus search only located an additional two articles provides support that 200 was a reasonable decision.

being replicated (plus one replication article contained its own original study). These are detailed in Tables B.9 and B.10.

3.2. Searching for pair-programming primary studies

Again the search commenced with the two mapping studies on replications [99, 12]. There is also a useful systematic review and meta-analysis [31] although this is limited to studies up and until 2007. A Scopus search (title, abstract and keywords) on "pair-programming" or "pair programming" gave 662 sources since 1999. A search of Scopus and Google scholar of title, abstract and keywords on "pair-programming" or "pair programming" gave 949 results. On the refined search string "pair-programming" or "pair programming" and "replicat*", seventeen documents in Scopus and zero citations on Google Scholar were returned.

Three of the seventeen sources were not considered because they were part of an introduction to proceedings and contained a list only of the topics in the set of proceedings (and included the word replication). One further paper was an extension of a Working Group Report containing no actual replication and was thus removed from consideration. One paper was a chapter in a volume of papers and only discussed the topic of replication. Of the 12 remaining papers, one was a was a Journal extension of another, leaving a total of 11 papers, of which five were not replications (again, they only mentioned the topic of replication without actually replicating anything). The distribution of the remaining six papers across the period 1999-present studied was as follows: a single study in each of 2008 and 2014 and two studies each in 2006 and 2012. A total of just six studies from a total of 949 means that only 0.6% of studies in pair-programming (PP) were actual replications (i.e., actually cited and repeated an earlier experiment). Each of the six documents was then analysed according to the extraction criteria previously described and checks made on each paper to ensure other replication references had not been missed.

3.3. Data extraction

Once the target articles had been located we then identified the number of replication studies and associated original studies for each article.

In terms of counting and analysis we noted a number of researchers have conducted chains or families of replications, $S_{t+1} \mapsto S_t$ denoting that study S_{t+1} replicates study S_t, however, we do not assert or count transitivity so for the study $S_{t+2} \mapsto S_{t+1}$ we do *not* infer $S_{t+2} \mapsto S_t$ even if explicitly stated by the authors since the separate study S_{t+1} has already elsewhere been included within our meta-analysis. Thus, for example, in [85] we only analyse one replication of the seven reported since the other six are previously described in [84]. Where a Replication article addresses more topics than replicating an Original study we only extract those aspects relevant to the replication. The details of all articles located are to be found in the Appendix as Table B.9 for the replications and Table B.10 for the articles containing original studies.

We then extracted the following information:

- Bibliographical information e.g., title, authors, etc
- Year
- Journal or conference?
- Type (internal, internal-same-paper or external)
- Article(s) replicated
- Number of questions / hypotheses being assessed
- Comparison procedure(s) e.g., NHST
- Comparison conclusion (Y, N, ?)
- Comparison text, i.e., text fragments that support the foregoing
- Other relevant notes e.g., where one replication duplicates part of another replication article

4. Results and Meta-analysis

4.1. Review Summary Data

Table 2 provides a summary of the results from our review. The replication studies located range from 1999 to 2017 and the original studies from 1993-2015. One study (O007) was independently replicated three times otherwise no other study was replicated more than once. This is a little surprising, although it could be the research community believes there is diminishing value in successive validations of the same original study. In contrast, a number of replication studies tackle multiple original studies within a single article.

Note that a replication study can also be the target of a subsequent replication and so appears as both a replication and an original study, e.g., R012 also serves as O019 and in one case a single article replicates studies contained within it hence it counts both as a replication and an original study i.e., R007. Also be aware that more than 29 original articles were cited by the replication studies (and the 30th study is where the replicated studies were contained within the replication itself). On occasions, however, careful reading revealed that one original article was subsumed by another more extensive or detailed article, in which case we only count the latter.

Another consideration is what proportion of all published experiments have been replicated? Although it is hard to give a definitive estimate of the number of software project effort prediction experiments that have been published, we attempted to corroborate this with a general Scopus search:

```
ALL ("software project" AND (experiment OR empirical) AND
 (effort OR cost) AND (predict* OR estimat* ))
```

Category	Effort Review Count	PP Review Count	Total Count
Relevant replication articles	22	6	28
Journal articles	5	3	8
Conference articles	17	3	20
Original articles replicated	30	5	35
Original articles - journals	7	1	8
Original articles - conference	23	4	27
Individual research questions / hypotheses replicated	60	15	75
External replication	8	2	10
Internal replication	13	2	15
Internal-same paper replication	1	2	3
Replication articles published	1999–2017	2006–2014	1999–2017
Original article published	1993–2015	2005–2009	1993–2015

Table 2: Systematic Review Summary

that retrieved in excess of 4600 articles. So the implication is of the order of 1% of relevant effort prediction experiments are replicated. A similar analysis for pair-programming yields 6/949 suggests that only $\approx 0.6\%$ of studies have been replicated. This is an even lower proportion than for effort prediction. Whilst we would not wish to claim much precision in the foregoing speculative calculations, it does indicate a very low rate.

The original and replication studies have deployed a range of analytic techniques the most common being null hypothesis significance testing (NHST) based by the use of p-values associated with the null hypothesis and stated or implied acceptance thresholds. These are summarised in Table 3. Note the proportions sum to more than 100% since some replications use multiple techniques. It is clear that the dominant comparison paradigm (for assessing the degree of confirmation from the replication) is null hypothesis significance testing (NHST) and in all cases the significance threshold was $\alpha = 0.05$. We also observed that of the 21 articles using NHST only three made any correction to α despite multiple tests, in some studies of the order of hundreds (see Table 4). Generally, no correction will be most permissive (in terms of Type I errors) and Holm-Bonferroni most stringent.

4.2. Meta-analysis

Note that for this analysis we combine both systematic reviews since the Pair-Programming review only located 6 replications. Our basic coding of replication results is to classify the replication researchers' verdicts as either confirmatory

Table 3: Analysis techniques used by replication articles

Comparison techniques used by replication articles	**Count**	**% of articles**
Comparison of descriptors, e.g., means	6	21%
Comparison of goodness of fit, e.g., R-squared and ANOVA	4	14%
Comparison of correlations	3	11%
NHST	21	75%
Comparison of clusterings from a win-draw-loss and Scott-Knot procedure	1	4%

Table 4: Approaches to correcting α in NHST

Correction procedure	**Count**
None	18
Holm-Bonferroni	2
Control false discovery rate	1

('Y'), contradictory ('N') or undecided ('?') where the authors were unable to reach a clear verdict, e.g. stating that the results were "mixed". Given the wide range of types of analysis and varying levels of detail provided by the replication studies we have used text analysis to classify replication outcomes[7]. Since replications, by definition, relate their findings to the original study this proved relatively straightforward and the results are summarised in Table 5.

Of course the process whereby the authors determined their conclusion text is itself inexact; there may be additional unreported factors that are relevant, however one has to question the reliability of subjective judgement when the range of replication outcomes considered confirmatory are not articulated [103].

	N	**?**	**Y**	**Total**
Counts	19	5	51	75
Percentage	25	7	68	100

Table 5: Replication Study Text-based Results

The first thing to note from Table 5 is that almost 7 out of 10 replications yield confirmatory results. This compares with 39%[8] from the Open Science [78]

[7] All raw data may be found at `https://figshare.com/s/d1f2a035c3f62168b48a`.
[8] The 39% is from subjective rating and 36% from tests of statistical significance [78].

replication in psychology so, on the face of it, software engineering has a higher replication rate. Interestingly, a 7% of replication comparisons were uncertain whether the new results supported the original study or not.

Next we break down the replication findings by replication type (External, Internal), publication venue (Journal, Conference) and by date (before 2011, 2011 onwards) based on a median split. We report effects in terms of odds ratios since the outcomes are treated as dichotomous in this analysis, i.e., confirmatory or otherwise (see Haddock et al. [29] for a discussion on their use as measures of effect size).

Replication Type	**N**	**?**	**Y**	**Total**	**Y/N**	**Y/(Y+N)**
External	11	0	7	18	0.64	0.39
Internal	8	5	44	57	5.5	0.85
Total	19	5	51	75	8.64	2.18

Table 6: Replication Findings by Replication Type

It is clear from the contingency table (see Table 6) that there is a substantial difference in the replications that are external (i.e., conducted by researchers that do not overlap with those responsible for the original study) and internal studies. This can be expressed as an odds ratio, i.e., $0.64 : 5.5 \approx 8.64$ with a 95% confidence interval of $(2.58, 29.0)$ [2]. In other words, a replication conducted by researchers who had some involvement with the original study are of the order of eight times as likely to find confirmatory evidence from their replication. Of course a wide confidence interval such as this implies considerable uncertainty but we note that it does not straddle unity[9] suggesting that we can have some confidence there is an effect. If the undecided results ('?') are factored in as being unsupportive or non-confirmatory the findings change slightly. The odds ratio becomes 5.32 and the 95% confidence interval narrows slightly to to (1.72,16.49) but our conclusions do not materially change.

More than a decade ago a review of software engineering experiments by Sjøberg et al. [101] similarly reported a ratio of 1:6 of confirmation from external replications contrasting with a ratio of 7:1 from internal replications. This yields a remarkable odds ratio of 42! Given the small sample size of 15 the confidence interval is extremely broad $(2.1, 825.8)$, however it again doesn't cover unity. It is also interesting to note that the external confirmation rate is 7/18 or 39% which is similar to the 39% reported by the Open Science group who were also conducting external replications.

Combining our findings with those of previous studies provides a consistent picture that external replications are less likely to be confirmatory than internal replications. This points to problems with researcher bias [95, 43] and thus may be a reason to be cautious about the independence of internal replications. A specific example is the review and meta-analysis of perspective based reading

[9] AN odds ratio of one indicates no effect or no difference between the alternatives.

experiments by Ciolkowski [21] who commented that he found "strong indicators of researcher bias".

Replication Type	N	?	Y	Total	Y/N	Y/(Y+N)
Journal	10	1	12	23	1.20	0.55
Conference	9	4	39	52	4.33	0.81
Total	19	5	51	75	3.61	1.49

Table 7: Replication Findings by Replication Venue

Next, we consider whether there are differences between journal and conference published replications (see Table 7). For example one could hypothesise that journals are more rigorously reviewed and so there might be fewer false positives and false negatives. We see that the odds ratio is ≈ 3.6 and the 95% confidence interval is $(1.19, 10.95)$ which of course falls just outside unity and so we conclude there is some evidence for a relationship between publication venue and replication outcome. Specifically replication studies published in a journal are more likely to be disconfirmatory than in a conference.

Finally, we consider date of the replication to investigate if there are any trends over time. We performed a median split to allocate replications into before-2011 and 2011-onwards categories (see Table 8). There is a tendency for more recent replications to be confirmatory as evidenced by an odds ratio of 3.61, however, the confidence interval (1.05 , 12.36) is again just outside unity so again this must be viewed as some evidence for an increasing proportion of confirmatory replications. The relative risk ratio is a modest 1.36. Factoring in the uncertain ('?') replication outcomes leads to a diminution of this effect, so we need some caution in arguing for a time based effect. Two possible considerations are (i) years and publication dates are highly arbitrary and (ii) it may be with growing emphasis upon better reporting of empirical studies, the opportunities are increased for improved replications, i.e., ones that are more exact.

Replication Type	N	?	Y	Total	Y/N	Y/(Y+N)
≤ 2010	15	3	26	44	1.73	0.63
≥ 2011	4	2	25	31	6.25	0.86
Total	19	5	51	75	3.61	0.73

Table 8: Replication Findings by Replication Date

Ideally we would now compare all the studies in terms of effect size and where appropriate convert into a single standardised measure using the formulae provided in Borenstein et al. [13, Chapter 7]. This requires the mean difference between treatments, sample size n and the two standard deviations s_1 and s_2. Again where standard deviations are not provided it would in principle be possible to determine an estimate using alternative measures of dispersion such

as the inter-quartile range using the procedure suggested by Wan et al. [106].

Unfortunately, estimating this proved to be extremely challenging since almost no studies provided descriptive statistics for the dispersion of the response variables, although in most cases detailed descriptions were provided for the explanatory variables. Additionally, only a few studies provided direct estimates of effect sizes and often where these were provided they tended to be with respect to a benchmark such as a random prediction procedure or a relative naïve approach such as using the sample mean or median for all predictions. Since this is generally not the research question being replicated this has not helped our meta-analysis. Consequently, whilst we consider it would be ideal to compare replications and original studies in terms of effect sizes and confidence limits this has not proved possible with the current level of reporting.

Finally we turn to the research question of what does confirmation mean, particularly in the statistical sense of prediction intervals discussed in Section 2.2. It is clear there is a good deal of informality and diversity in determining what constitutes a 'successful' replication or how similar results should be. An example of the inherent complexity can be seen in the family of replications performed by Quesada-López and Jenkins where for example the range of reported R-squared coefficients vary from 0.36 to 0.94 with a value of 0.68 for the replication study [84, Table 9]. Although all models are 'significant' with $p < 0.01$ the effect sizes are clearly somewhat different and this is likely to have some practical consequences from a practitioner perspective. Consequently it is a little unclear what meaning to attach to these replications. Interestingly, Quesada-López and Jenkins somewhat hedged their bets by simply indicating that the results presented "support *some* of the findings of the original studies" [our italics].

Attempting to apply these ideas to a software engineering replication we use R001 and O004 (since Myrtveit and Stensrud [77] usefully provide standard deviations for the response variable MMRE where the subscript denotes the two treatments, A=analogy and R=regression. We need to reason backwards since [97] does not provide this information. So if we pool the standard deviation from R001 we obtain $s_r = 265$ and a mean effect m_R of $MMRE_A - MMRE_R = -27$ and $n_R = 68$ (see [77, Table 8] and we pool the 8 comparable datasets from [97, Table 2] so $n_O = 254$ we compute a 95% prediction interval of [-99.22 , 45.22]. Computing a weighted mean of $m_O = MMRE_A - MMRE_R$ we obtain 43.2 which is just within the prediction interval despite the seemingly quite distinct results. If this seems surprising recall that the variance or standard deviation is high.

To interpret this analysis more clearly, the original study was based upon 254 projects and the replication on 68 projects. The original study reported a mean difference in MMRE between regression and analogy-based prediction of 43.2%. The replication found a difference in the *opposite* direction of -27%. Yet even this non-trivial difference based on many software projects would fall within our computed 95% prediction interval and so in principle should be accepted as a confirmation, not a failure to replicate. Such broad prediction intervals are due to the very high levels of variance in the predictions. In terms of generat-

ing useful knowledge for the real world they are extremely problematic since a project manager would certainly differentiate between this range of performance since they encompass technique being substantially better and worse than the comparison technique.

Another even more extreme example of small and underpowered studies is [O010] [44] and the replication [R005] [96]. The raw results are provided in [R005] which enables us to again reason backwards. The approximate effect size is given by the mean difference in absolute residual between the two prediction techniques (analogy and regression-to-the-mean) normalised by the pooled standard deviation. This yields an effect size of ~ 0.11 which is below the threshold of small suggested by Cohen [26]. The sample sizes are also small with data sets of 18 and 16 and 20 and 16 respectively. From this we can construct the prediction intervals which are approximately [-0.78, 1.02]. Put simply a replication study that found a large effect in *either direction* could be considered to be confirmatory! The reason again is the small effect size and high variance such that, without even any other issues, hugely varying results can simply be explained by sampling error. In these circumstances a replication is simply not useful.

5. Discussion

Clearly this review exposes that there are issues concerning what we mean by replication and what we might expect to learn from a completed replication study. These are not unique to software engineering. Since the widely discussed 100 replications study in psychology [78] there has been a good deal of reflection from this discipline. Anderson and Maxwell [6, Table 1] identify six distinct replication goals, and it is probable that their list is not exhaustive given the particular needs and goals of software engineering. Certainly it would be helpful for replication studies to be more explicit about what their purpose.

As a number of researchers have remarked e.g., [78, 102] there is no single agreed way to define replication confirmation. Moreover, unlike the Open Science initiative, which completed 100 replications of published studies in psychology this is not a prospective study and hence far less control. In addition, their original studies that were the target of their replications more consistently relied on the null hypothesis significance testing (NHST) paradigm than is the case for our systematic review where it was only adopted by two thirds of the replication articles (see 3). Furthermore we found there is often some imprecision in terms of exactly what hypotheses were tested in the original study and the extent to which they were determined *post hoc*.[10] There also seems considerable flexibility with the direction of hypotheses and also the false implication

[10] As an example of *post hoc* selection of hypotheses [9] state "we discarded all those hypotheses that turned out to be clearly unsupported by the data analysis, and we kept all those that were supported (at the 0.05 significance level) and those data with a p-value close to 0.05, to test them again.". We respect their candour. It is most probable other studies have done similarly but neglected to report the fact.

that failure to reject the null hypothesis implies that there is strong evidence of no effect.

So we may have the unintuitive situation that an underpowered original experiment showing a small effect may be a good deal easier to 'confirm' than one might imagine! Of course the confirmation may not be tremendously valuable or yield much useful new knowledge! Alternatively it could guide the design of more powerful replications that are more able to yield useful results, e.g., by increasing the number of experimental units, using simpler experimental design or by seeking to reduce heterogeneity hence the variance.

5.1. Threats to Validity

Internal validity or the degree to which the conclusions derived from the study are justified. We identify three potential threats.

First, three replications e.g., R012 replicating R011/O020 used the same but extended data set, in other words there is a common subset of projects. This clearly impacts the independence of the replication study. However we have included such studies since the authors interpret these as replication studies.

The second threat lies in the potential inexactitude of the replications particularly when these are conducted externally. For example, some replications follow similar statistical procedures whilst other replications are loosely inspired by the general research question, but thereafter details differ considerably. However, there is variation in the clarity of the experimental description contained in the Original Study and potential help between teams so for example, R015 remarked that they "were fortunate to be able to consult with the first author of [8] and be able to use the same code for the nine learners and most of the pre-processors" [7]. By contrast Lokan and Mendes [55] stated that many aspects of the Original Study they were seeking to replicate were unclear. Without this information it is difficult to compute, for example, prediction intervals [102]. Mindful of these difficulties we have focused on major results and report confidence intervals wherever possible.

Third, the data extraction and coding was not always straightforward. Articles were written in different styles for varying audiences and purposes over a period of more than 20 years. We also assume that a single common author with the original study is sufficient to classify a replication as internal. However, the influence of a sole author may not be decisive. Nevertheless, the difference between internal and external replication study confirmation rates is so pronounced we doubt this threat will make much difference. Different statistical analyses have been deployed and differing levels of detail reported. This has certainly hindered our meta-analysis as not all our judgements have been made with confidence. We make the raw data available and invite other researchers to reproduce or revisit this analysis.

external validity or the degree to which it is reasonable to generalise from the study results to other contexts, in particular to other sub-fields within software engineering. Here we identify two potential threats.

First, the community comprises a relatively small number of researchers of whom a couple dominate, namely Emilia Mendes is associated with eight replications and Chris Lokan with six. Thus this, combined with our small sample of replications, might impact our confidence in any generalisation. However, wherever possible we have compared our results with other published meta-analyses, particularly the recent work by Jørgensen et al. [43] that also addresses empirical software engineering studies and also from the field of experimental psychology.

Second, we have only considered experiments and only from the specific sub-fields of project effort prediction and pair-programming. The study on researcher bias for defect prediction [95] highlighted the likelihood of problems in this area too so we don't see any compelling reason to believe effort prediction is unrepresentative. Of course further studies could better address this question.

6. Conclusions

First, we wish to reiterate that we agree with the view that replication is a fundamental part, indeed a "cornerstone of science" [100]. The ability of software engineers to reproduce empirical results will add to our body of knowledge and our confidence in it. This must be to the benefit of researchers and practitioners alike. However, we consider that meta-analysis [32] is far more likely to prove fruitful than the notion of replicating individual studies many of which may be under-powered or under-reported. The philosophy of replication is to ask whether the results of a particularly study can be confirmed by another study. In other words is it 'true'? Meta-analysis is concerned with pooling results to better estimate the population parameter(s) of interest and understand the uncertainty that surrounds these estimates. This allows the possibility of working with multiple studies rather than singling out individual studies and the pooling of all relevant evidence. Of course, meta-analysis is not without problems such as dealing with low quality studies. Nevertheless, we believe this is preferable to arbitrarily picking an original study (typically underpowered) and seeking to confirm the results via a replication study.

Our findings fall into five groups.

1. Experiments in software project effort prediction are replicated relatively infrequently. We located 28 replication articles. The total population of target experiments is not easy to estimate but it is likely to be order of magnitude thousands rather than hundreds. This implies only a few percent are replicated. We saw little evidence of any time based effect.
2. Approximately half of all replications are conducted internally. The odds ratio of an internal replication confirming the original study is 8.64. This

is of a concern. It is also in line with the systematic review of Sjøberg et al. [101], da Silva et al. [99] and Jørgensen et al. [43]. Two possible explanations are researcher bias (possibly subconscious) or incomplete reporting leading to inexact replication. Neither possibility is ideal.

3. There was great variety between the replications despite focusing on a relatively small domain. This included the level of granularity and the exactness of the replication of the original experimental details and analysis. Although the dominant form of reasoning was null hypothesis significance testing (75%) details concerning the specification of hypotheses and corrections for multiple tests varied considerably (both in the original and replication studies).
4. The variety described above coupled with incomplete reporting meant that determining effect sizes and associated confidence limits was generally not possible. Most notably absent were measures of dispersion for the response variables.
5. For two replications that reported standard deviation data we could estimate the 95% prediction interval for the replications (which were deemed to contradict the original studies). Despite the seeming substantial difference in results and effect direction and quite large sample sizes the prediction interval was surprisingly wide and therefore both replications instead of being contradictory should be considered as confirmatory of the original study. The issue is one of low power in a setting of high variance. Unfortunately, there is a good deal of evidence to suggest that many studies are under-powered [25, 8]. Very little useful additional knowledge is obtained when such variation in effect size and direction are permitted. The question then arises as to what steps can be taken to narrow prediction intervals such that replication studies yield more useful information. We believe the research comm should move from the idea of replicating individual (possibly interesting studies) to meta-analysis of interesting questions. Of course reproducibility might remain an issue on occasions.

Reporting guidelines have been proposed by Carver [20] however our meta-analysis has been considerably hindered by incomplete and unclear information, notably the lack of details concerning the dispersion of the response variables. The lack of this kind of information substantially detracts from the usefulness of an empirical study.

To conclude, we would strongly urge those considering replications to carefully specify exactly what is being replicated and what results they expect, what would constitute a confirmation and what would constitute a disconfirmation. Where bounds seem unacceptably wide steps may be taken to address the problem such as (i) reducing the heterogeneity of the experimental units, (ii) simplification of the experimental design or increasing the sample size *before* conducting the actual replication. Alternatively we recommend meta-analysis.

Acknowledgements

We would like to thank both the guest editors and the reviewers for their detailed and constructive comments on two earlier versions of this paper. Martin Shepperd was supported by the EPSRC Grant EP/P025196/1. Steve Counsell was supported by the EPSRC Grants EP/M024083/1 and EP/N011627/1.

Appendix A. Detailed comments on the pair programming review

Appendix A.1. Pair Programming Replications

A set of general observations can be distilled from those six studies. To begin with, only one study of the six by Lemos et al. [S4] used industrial developers as a basis. Even then, only seven developers were used, as opposed to eighty-five students as part of the same experiment. While of course, from a pedagogical standpoint, these are valuable studies, the validity in, and transferability to, industrial settings of student-based studies could be questioned; moreover, the small sample size of industrial developers in this case (and in many other empirical studies) also poses an external validity threat. In the paper by Lemos et al. two research questions related specifically to PP were tested. Firstly, could PP help obtain more correct implementations than when those implementations were individually programmed? Secondly, did pair programmers spend more time [coding] than individual programmers? Results showed that PP tended to increase reliability in terms of correctness, vis-a-vis defects. However, PP tended to increase development time with compared with individually programmed code.

A further observation of the extracted studies is that only two of the six were external replications of previous studies [S2, S3]. The other four used specific cohorts on undergraduate CS programmes and internally replicated a previous study. The first of the external replications was by Hanks [S3], a study which partially replicated the work of Robins et al. [SR4]. In the original study, the work explored the process by which student cohorts learnt a first programming language and the problems the students faced in the process (it did not specifically explore PP per se).

The later study explored two hypotheses related to programming task completion comparing paired versus individual programmers (students in this case). The first hypothesis was that the proportion of problems encountered in the allocated tasks would be smaller for paired students when compared to individuals. The problems was a set of acknowledged issues that students faced in the original study and borrowed for the later study. The second hypothesis explored whether paired students required assistance on fewer problems. In terms of results, the types of problems encountered by the paired programmers were similar to those of individual students. However, the number of problems requiring assistance was much smaller for the paired students, suggesting that they were able to resolve more on their own. The study therefore supported the use of PP compared to individual program development. In the other external replication by Canfora et al. [S2], an analysis of the work of Al-Kilidar et

al. [SR2] was conducted. That earlier study suggested that pair design quality was higher than individual design quality in terms of functionality, usability, portability and maintenance compliance. Canfora et al.'s study explored two hypotheses related to effort and quality of pair-programmed code versus individual code. The key conclusion was of evidence that pair programming made effort and quality more predictable than traditional solo programming.

The fact that only two of the six studies externally replicated a study potentially highlights another generic problem with empirical studies and observation. It is difficult enough to internally replicate a study where there is usually significant control over the experimental instrumentation; the majority of PP replications seem to use subjects that are relatively easy to control and are largely homogenous in nature (i.e., student cohorts). The extent to which these studies can be transferred to industrial practice is questionable and so it has to be asked whether the ease with which students can be used in experiments compromises progress in understanding actual PP practice.

A final observation and particularly relevant to the overall aim of the paper is that only one of the six studies reported effect sizes and statistical power [S6]. In Salleh et al. [S6] a replication of an earlier study using a cohort of Year 1 students was made using Year 2 students at the University of Auckland. The authors undertook a set of five experiments (including the one replicated experiment), between 2009 and 2010, exploring the effects of personality traits on PP. The replicated study explored the personality trait of 'conscientiousness', defined as: "concerned with one's achievement orientation. Those who have a high score tend to be hardworking, organised, able to complete tasks thoroughly and on time, and reliable". The study did not find any significant correlation between Conscientiousness and academic performance.

Moreover, only one of the studies by Sfetsos et al. [S5] reported a level of significance other than at the 5% level, reporting at the 1% level. Unsurprisingly, at least one of the studies justified the use of a 5% level by claiming a "traditional" confidence level of 95%. In Sfetsos et al. [S5] the emphasis was on pair performance in terms of communication, time to complete assignments and overall score. Differences between two personality groups were studied: a control group with homogeneous personalities in pairs and an experimental one with heterogeneous personalities. The study was based on two prior experiments [SR3, SR5] using seventy and one hundred and sixty students, respectively. Three hypotheses were explored related to a mix of personality types and temperaments. Results showed that pairs comprising heterogeneous personalities and temperaments performed better than pairs with the same personality and temperament type; the time spent completing assignments was not found to be statistically significant between groups.

Finally, the study by Mendes et al. [S1] replicated the work originally described in [SR1]. The study in [S1] compared the performance of students carrying out PP in a controlled environment of students at the University of Auckland with those doing individual work in a similar setting. In total, 190 second-year students on software design and construction course took part, 78 of which were paired and 112 worked individually. The effect the PP experience

appeared to improve both the quality of work produced and the enthusiasm and motivation for doing more supporting the earlier work in [SR1]. Table B.11 shows the set of studies that replicated (either internally or externally) another study and Table B.12 the original set of studies.

Appendix B. Articles located by the systematic review

Table B.9: Replication Study Details

ID	Citation	Year	Article Title	Rep Type	Article Type	Orig Article Id
R001	[77]	1999	A Controlled Experiment to Assess the Benefits of Estimating with Analogy and Regression Models	Ext	J	O004
R002	[15]	2000	A replicated Assessment and Comparison of Common Software Cost Modeling Techniques	Int	C	O005
R003	[72]	2003	A Replicated Assessment of the Use of Adaptation Rules to Improve Web Cost Estimation	Int	C	O009
R004	[69]	2005	A Replicated Comparison of Cross-company and Within-company Effort Estimation Models using the ISBSG Database	Ext	C	O007
R005	[96]	2005	A Replication of the Use of Regression Towards the Mean (R2M) as an Adjustment to Effort Estimation Models	Ext	C	O010
R006	[55]	2006	Cross-company and Single-company Effort Models Using the ISBSG Database: a Further Replicated Study	Ext	C	O007
R007	[9]	2007	Three Empirical Studies on Estimating the Design Effort of Web Applications	Int-Same Article	J	R007
R008	[67]	2008	Replicating studies on cross- vs single-company effort models using the ISBSG Database	Ext / Int	J	O007, O014

Continued on next page

Table B.9 – *Continued from previous page*

ID	Citation	Year	Article Title	Rep Type	Article Type	Orig. Article Id
R009	[70]	2008	Cross-company vs. single-company web effort models using the Tukutuku database: An extended study	Int	J	O013
R010	[68]	2009	Investigating the Use of Chronological Splitting to Compare Software Cross-company and Single-company Effort Predictions: A Replicated Study	Int	C	O015
R011 (= O019)	[27]	2010	Estimating web application development effort using COSMIC: Impact of the base functional component types	Ext	C	O016
R012	[17]	2010	Which cosmic base functional components are significant in estimating web application development? - a case study	Int	C	O019
R013	[24]	2011	Using web objects for development effort estimation of web applications : a replicated study	Ext	C	O011, O012
R014	[37]	2013	Software cost estimation by classical and Fuzzy Analogy for Web Hypermedia Applications : A replicated study	Int	C	O008, O020
R015	[7]	2013	Using ensembles for web effort estimation	Ext	C	O021
R016	[52]	2015	Functional Size Measures and Effort Estimation in Agile Development : A Replicated Study	Int	C	O025
R017	[74]	2015	How to Make Best Use of Cross-Company Data for Web Effort Estimation?	Int	C	O026
R018 (= O029)	[84]	2015	An empirical validation of function point structure and applicability : A replication study	Ext / Int	C	O001, O002, O003, O006, O022, O027

Continued on next page

Table B.9 – *Continued from previous page*

ID	Citation	Year	Article Title	Rep Type	Article Type	Orig. Article Id
R019	[36]	2015	RBFN networks-based models for estimating software development effort : A cross-validation study	Int	C	O023
R021	[4]	2016	A replication study on the effects of weighted moving windows for software effort estimation	Int	C	O024, O028
R020	[85]	2016	Function Point Structure and Applicability : A Replicated Study	Int	J	O029
R022	[59]	2017	Investigating the use of moving windows to improve software effort prediction: a replicated study	Int	J	2017

Table B.10: Original Study Details

Id	Citation	Year	Article Title	Article Type	Rep Article Id
O001	[48]	1993	Inter-item correlations among function points	C	R018
O002	[40]	1993	A comparison of function point counting techniques	J	R018
O003	[42]	1996	Function point sizing: structure, validity and applicability	J	R018
O004	[97]	1997	Estimating Software Project Effort Using Analogies	J	R001
O005	[14]	1999	An Assessment and Comparison of Common Software Cost Estimation Modeling Techniques	C	R002
O006	[54]	1999	An empirical study of the correlations between function point elements	C	R018
O007	[41]	2001	Using Public Domain Metrics to Estimate Software Development Effort	C	R004, R006
O008	[33]	2002	Estimating software project effort by analogy based on linguistic values	C	R014

Continued on next page

Table B.10 – *Continued from previous page*

Id	Citation	Year	Article Title	Article Type	Rep Article Id
O009	[71]	2003	Do Adaptation Rules Improve Web Cost Estimation?	C	R003
O010	[44]	2003	Effort Estimation: Software Effort Estimation by Analogy and "Regression Toward the Mean"	J	R005
O011	[88]	2003	Cost estimation for web applications	C	R013
O012	[89]	2003	Using Web Objects for Estimating Software Development Effort for Web Applications	C	R013
O013	[66]	2004	Further comparison of cross-company and within-company effort estimation models for web applications	C	R009
O014	[55]	2006	Cross-company and single-company effort models using the ISBSG database: a further replicated study	C	R008
R007	[9]	2007	Three Empirical Studies on Estimating the Design Effort of Web Applications	J	R007
O015	[56]	2008	Investigating the Use of Chronological Splitting to Compare Software Cross-company and Single-company Effort Predictions	C	R010
O016	[18]	2008	Impact of Base Functional Component Types on Software Functional Size Based Effort Estimation	C	R011
O017	[57]	2009	Applying moving windows to software effort estimation	C	R022
O018	[17]	2010	Which cosmic base functional components are significant in estimating web application development? - a case study	C	R012
O019 (= R011)	[27]	2010	Estimating web application development effort using COSMIC: Impact of the base functional component types	C	R012
O020	[34]	2012	Software cost estimation by fuzzy analogy for ISBSG repository	C	R014
O021	[49]	2012	On the Value of Ensemble Effort Estimation	J	R015

Continued on next page

Table B.10 – *Continued from previous page*

Id	Citation	Year	Article Title	Article Type	Rep Article Id
O022	[50]	2013	Towards a simplified definition of Function Points	J	R018
O023	[35]	2013	Assessing RBFN Based Software Cost Estimation Models	C	R019
O024	[3]	2013	The Evaluation of Weighted Moving Windows for Software Effort Estimation	C	R021
O025	[53]	2014	Can Functional Size Measure Improve Effort Estimation in SCRUM?	C	R016
O026	[75]	2014	How to make best use of cross-company data in software effort estimation?	C	R017
O027	[83]	2014	Function point structure and applicability validation using the ISBSG dataset: a replicated study	C	R018
O028	[58]	2014	Investigating the use of duration-based moving windows to improve software effort prediction: A replicated study	J	R021
O029 (= R018)	[84]	2015	An empirical validation of function point structure and applicability: A replication study	C	R020

References

[1] H. Al-Kilidar, P. Parkin, A. Aurum, R. Jeffery, Evaluation of effects of pair work on quality of designs, in: Australian Software Engineering Conference.

[2] D. Altman, Practical statistics for medical research, Chapman and Hall, London, 1990.

[3] S. Amasaki, C. Lokan, The evaluation of weighted moving windows for software effort estimation, in: 2013 International Conference on Product Focused Software Process Improvement, Springer, pp. 214–228.

[4] S. Amasaki, C. Lokan, A replication study on the effects of weighted moving windows for software effort estimation, in: Proceedings of the 20th International Conference on Evaluation and Assessment in Software Engineering, ACM, pp. 40:1–40:9.

Table B.11: Pair Programming (PP) Replication Study Details

ID	Citation	Year	Article Title	Rep Type	Article Type	Orig Article Id
S1	[64]	2006	A replicated experiment of pair-programming in a 2nd-year software development and design computer science course	Int	C	SR5
S2	[19]	2006	Performances of pair designing on software evolution: A controlled experiment	Ext	C	SR3
S3	[30]	2008	Problems encountered by novice pair programmers	Ext	J	SR4
S4	[51]	2012	Development of auxiliary functions: Should you be agile? An empirical assessment of pair programming and test-first programming	Int Same Article	C	S4
S5	[91]	2012	Investigating the impact of personality and temperament traits on pair programming: A controlled experiment replication	Int	C	SR1 and SR2
S6	[90]	2014	Investigating the effects of personality traits on pair programming in a higher education setting through a family of experiments	Int Same Article	J	S6

[5] V. Amrhein, F. Korner-Nievergelt, T. Roth, The earth is flat (p ¿ 0.05): significance thresholds and the crisis of unreplicable research, PeerJ 5 (2017) e3544.

[6] S. Anderson, S. Maxwell, There's more than one way to conduct a replication study: Beyond statistical significance, Psychological Methods 21 (2016) 1–12.

[7] D. Azhar, P. Riddle, E. Mendes, N. Mittas, L. Angelis, Using ensembles for web effort estimation, in: 2013 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement, IEEE, pp. 173–182.

[8] M. Bakker, A. van Dijk, J. Wicherts, The rules of the game called psychological science, Perspectives on Psychological Science 7 (2012) 543–554.

Table B.12: Pair Programming (PP) Original Study Details

Id	Citation	Year	Article Title	Article Type	Rep Article Id
SR1	[65]	2005	Investigating Pair Programming in a 2nd year Software Development and Design Computer Science Course	C	S5
SR2	[1]	2005	Evaluation of Effects of Pair Work on Quality of Designs	C	S5
SR3	[92]	2006	Investigating the Impact of Personality Types on Communication and Collaboration- Viability in Pair Programming-An Empirical Study	C	S2
SR4	[86]	2006	Problem distributions in a CS1 course	C	S3
SR5	[93]	2009	An Experimental Investigation of Personality Types Impact on Pair Effectiveness in Pair Programming	J	S1

[9] L. Baresi, S. Morasca, Three empirical studies on estimating the design effort of web applications, ACM Transactions on Software Engineering and Methodology 16 (2007).

[10] V. Basili, R. Selby, D. Hutchens, Experimentation in software engineering, IEEE Transactions on Software Engineering 12 (1986) 733–743.

[11] V.R. Basili, F. Shull, F. Lanubile, Building knowledge through families of experiments, IEEE Transactions on Software Engineering 25 (1999) 456–473.

[12] R. Bezerra, F. da Silva, A. Santana, C. de Magalhães, R. Santos, Replication of empirical studies in software engineering: An update of a systematic mapping study, in: International Symposium on Empirical Software Engineering and Measurement (ESEM 15), IEEE, pp. 1–4.

[13] M. Borenstein, L. Hedges, J. Higgins, H. Rothstein, Introduction to Meta-Analysis, Wiley, Chichester, West Sussex, UK, 2009.

[14] L. Briand, K. El Emam, D. Surmann, I. Wieczorek, K. Maxwell, An assessment and comparison of common software cost estimation modeling techniques, in: International Conference on Software Engineering, 1999, IEEE, pp. 313–323.

[15] L. Briand, T. Langley, I. Wieczorek, A replicated assessment and comparison of common software cost modeling techniques, in: Proceedings

of the 22nd International Conference on Software Engineering, ACM, pp. 377–386.

[16] A. Brooks, J. Daly, J. Miller, M. Roper, M. Wood, Replication of experimental results in software engineering, International Software Engineering Research Network (ISERN) Technical Report ISERN-96-10, University of Strathclyde (1996).

[17] L. Buglione, F. Ferrucci, C. Gencel, C. Gravino, F. Sarro, Which cosmic base functional components are significant in estimating web application development?: A case study, in: 20th International Workshop on Software Measurement (IWSM)/Metrikon/MENSURA Joint Conference, Shaker Verlag.

[18] L. Buglione, C. Gencel, Impact of base functional component types on software functional size based effort estimation, in: International Conference on Product Focused Software Process Improvement, Springer, pp. 75–89.

[19] G. Canfora, A. Cimitile, F. Garcia, M. Piattini, C. Visaggio, Performances of pair designing on software evolution: A controlled experiment, in: European Conference on Software Maintenance and Reengineering, pp. 197–205.

[20] J. Carver, Towards reporting guidelines for experimental replications: A proposal, in: 1st International Workshop on Replication in Empirical Software Engineering (2010).

[21] M. Ciolkowski, What do we know about perspective-based reading? an approach for quantitative aggregation in software engineering, in: 3rd International Symposium on Empirical Software Engineering and Measurement (ESEM 2009), IEEE, pp. 133–144.

[22] I. Cooper, What is a "mapping study?", Journal of the Medical Library Association 104 (2016) 76–78.

[23] G. Cumming, Replication and p intervals: p values predict the future only vaguely, but confidence intervals do much better, Perspectives on Psychological Science 3 (2008) 286–300.

[24] S. Di Martino, F. Ferrucci, C. Gravino, F. Sarro, Using web objects for development effort estimation of web applications: a replicated study, in: International Conference on Product Focused Software Process Improvement, Springer, pp. 186–201.

[25] T. Dybå, V.B. Kampenes, D.I. Sjøberg, A systematic review of statistical power in software engineering experiments, Information and Software Technology 48 (2006) 745–755.

[26] P. Ellis, The Essential Guide to Effect Sizes: Statistical Power, Meta-Analysis, and the Interpretation of Research Results, Cambridge University Press, 2010.

[27] F. Ferrucci, C. Gravino, L. Buglione, Estimating web application development effort using cosmic: Impact of the base functional component types, in: Software Measurement European Forum (SMEF), pp. 103–116.

[28] O. Gómez, N. Juristo, S. Vegas, Understanding replication of experiments in software engineering: A classification, Information and Software Technology 56 (2014) 1033–1048.

[29] C. Haddock, D. Rindskopf, W. Shadish, Using odds ratios as effect sizes for meta-analysis of dichotomous data: a primer on methods and issues., Psychological Methods 3 (1998) 339–353.

[30] B. Hanks, Problems encountered by novice pair programmers, ACM Journal on Educational Resources in Computing 4 (2008).

[31] J. Hannay, T. Dybå, E. Arisholm, D. Sjøberg, The effectiveness of pair programming: A meta-analysis, Information and Software Technology 51 (2009) 1110–1122.

[32] L. Hedges, I. Olkin, Statistical methods for meta-analysis, Academic Press, London, 1985.

[33] A. Idri, A. Abran, T. Khoshgoftaar, Estimating software project effort by analogy based on linguistic values, in: 8th IEEE Symposium on Software Metrics, 2002, IEEE, pp. 21–30.

[34] A. Idri, F. Amazal, Software cost estimation by fuzzy analogy for ISBSG repository, in: Uncertainty Modeling in Knowledge Engineering and Decision Making, World Scientific, 2012, pp. 863–868.

[35] A. Idri, A. Hassani, A. Abran, Assessing rbfn-based software cost estimation models (2013).

[36] A. Idri, A. Hassani, A. Abran, Rbfn networks-based models for estimating software development effort: A cross-validation study, in: 2015 IEEE Symposium Series on Computational Intelligence, IEEE, pp. 976–983.

[37] A. Idri, A. Zahi, Software cost estimation by classical and fuzzy analogy for web hypermedia applications: A replicated study, in: 2013 IEEE Symposium on Computational Intelligence and Data Mining (CIDM), IEEE, pp. 207–213.

[38] J. Ioannidis, Why Most Published Research Findings Are False, PLoS Medicine 2 (2005) e124–6.

[39] J. Ioannidis, Why most discovered true associations are inflated, Epidemiology 19 (2008) 640–648.

[40] R. Jeffery, G. Low, M. Barnes, A comparison of function point counting techniques, IEEE Transactions on Software Engineering 19 (1993) 529–532.

[41] R. Jeffery, M. Ruhe, I. Wieczorek, Using public domain metrics to estimate software development effort, in: 7th IEEE Intl. Metrics Symposium, 2001, IEEE Computer Press, 2001, pp. 16–27.

[42] R. Jeffery, J. Stathis, Function point sizing: structure, validity and applicability, Empirical Software Engineering 1 (1996) 11–30.

[43] M. Jørgensen, T. Dybå, K. Liestøl, D. Sjøberg, Incorrect results in software engineering experiments: How to improve research practices, J. of Systems & Software 116 (2016) 133–145.

[44] M. Jørgensen, U. Indahl, D. Sjøberg, Software effort estimation by analogy and regression toward the mean, Journal of Systems and Software 68 (2003) 253–262.

[45] B. Kitchenham, The role of replications in empirical software engineering — a word of warning, Empirical Software Engineering 13 (2008) 219–221.

[46] B. Kitchenham, D. Budgen, P. Brereton, Evidence-Based Software engineering and systematic reviews, CRC Press, Boca Raton, Fl, US, 2015.

[47] B. Kitchenham, T. Dybå, M. Jørgensen, Evidence-based software engineering, in: 27th IEEE Intl. Softw. Eng. Conf. (ICSE 2004), IEEE Computer Society, 2004.

[48] B. Kitchenham, K. Kansala, Inter-item correlations among function points, in: 1st International Symposium on Software Metrics, IEEE Computer Society Press, 1993.

[49] E. Kocaguneli, T. Menzies, J. Keung, On the value of ensemble effort estimation, IEEE Transactions on Software Engineering 38 (2012) 1403–1416.

[50] L. Lavazza, S. Morasca, G. Robiolo, Towards a simplified definition of function points, Information and Software Technology 55 (2013) 1796–1809.

[51] O. Lemos, F. Ferrari, F. Silveira, A. Garcia, Development of auxiliary functions: Should you be agile? An empirical assessment of pair programming and test-first programming, in: 34th IEEE International Conference on Software Engineering, pp. 529–539.

[52] V. Lenarduzzi, I. Lunesu, M. Matta, D. Taibi, Functional size measures and effort estimation in agile development: A replicated study, in: International Conference on Agile Software Development, Springer, pp. 105–116.

[53] V. Lenarduzzi, D. Taibi, Can functional size measure improve effort estimation in scrum, in: ICSEA-International Conference on Software Engineering and Advances, Nice, France.

[54] C. Lokan, An empirical study of the correlations between function point elements [software metrics], in: Sixth International Software Metrics Symposium, 1999, IEEE, pp. 200–206.

[55] C. Lokan, E. Mendes, Cross-company and single-company effort models using the ISBSG database: a further replicated study, in: Proceedings of the 2006 ACM/IEEE International Symposium on Empirical Software Engineering, ACM, pp. 75–84.

[56] C. Lokan, E. Mendes, Investigating the use of chronological splitting to compare software cross-company and single-company effort predictions., in: EASE 2008.

[57] C. Lokan, E. Mendes, Applying moving windows to software effort estimation, in: 3rd International Symposium on Empirical Software Engineering and Measurement (ESEM), IEEE Computer Society, 2009.

[58] C. Lokan, E. Mendes, Investigating the use of duration-based moving windows to improve software effort prediction: A replicated study, Information and Software Technology 56 (2014) 1063–1075.

[59] C. Lokan, E. Mendes, Investigating the use of moving windows to improve software effort prediction: a replicated study, Empirical Software Engineering 22 (2017) 716–767.

[60] E. Loken, A. Gelman, Measurement error and the replication crisis, Science 355 (2017) 584–585.

[61] L. Madeyski, B.A. Kitchenham, S.L. Pfleeger, Why reproducible research is beneficial for security research, 2015. Unpublished paper.

[62] C. de Magalhães, F. da Silva, R. Santos, M. Suassuna, Investigations about replication of empirical studies in software engineering: A systematic mapping study, Information & Software Technology 64 (2015) 76–101.

[63] T. Mende, Replication of defect prediction studies: Problems, pitfalls and recommendations, in: 6th International Conference on Predictive Models in Software Engineering, PROMISE '10, ACM, New York, NY, USA, 2010, pp. 5:1–5:10.

[64] E. Mendes, L. Al-Fakhri, A. Luxton-Reilly, A replicated experiment of pair-programming in a 2nd-year software development and design computer science course, in: Proceedings of the 11th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education, pp. 108–112.

[65] E. Mendes, L. El-Fakhri, A. Luxton-Reilly, Investigating pair programming in a 2nd year software development and design computer science course, in: Proceedings of ITiCSE 2005, pp. 296–300.

[66] E. Mendes, B. Kitchenham, A further comparison of cross-company and within-company effort estimation models for web applications, in: 10th International Symposium on Software Metrics, IEEE, pp. 348–357.

[67] E. Mendes, C. Lokan, Replicating studies on cross-vs single-company effort models using the ISBSG database, Empirical Software Engineering 13 (2008) 3–37.

[68] E. Mendes, C. Lokan, Investigating the use of chronological splitting to compare software cross-company and single-company effort predictions: A replicated study., in: EASE 2009.

[69] E. Mendes, C. Lokan, R. Harrison, C. Triggs, A replicated comparison of cross-company and within-company effort estimation models using the ISBSG database, in: 11th IEEE International Symposium on Software Metrics, 2005, IEEE.

[70] E. Mendes, S. di Martino, F. Ferrucci, C. Gravino, Cross-company vs. single-company web effort models using the tukutuku database: An extended study, Journal of Systems and Software 81 (2008) 673–690.

[71] E. Mendes, N. Mosley, S. Counsell, Do adaptation rules improve web cost estimation?, in: 14th ACM Conference on Hypertext and Hypermedia, 2003, ACM, 2003, pp. 173–183.

[72] E. Mendes, N. Mosley, S. Counsell, A replicated assessment of the use of adaptation rules to improve web cost estimation, in: 2003 International Symposium on Empirical Software Engineering, IEEE, pp. 100–109.

[73] J. Miller, Replicating software engineering experiments: a poisoned chalice or the holy grail, Information & Software Technology 47 (2005) 233–244.

[74] L. Minku, F. Sarro, E. Mendes, F. Ferrucci, How to make best use of cross-company data for web effort estimation?, in: 2015 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM), pp. 1–10.

[75] L. Minku, X. Yao, How to make best use of cross-company data in software effort estimation?, in: Proceedings of the 36th International Conference on Software Engineering, ACM, pp. 446–456.

[76] D. Murdoch, Y. Tsai, J. Adcock, P-values are random variables, The American Statistician 62 (2008) 242–245.

[77] I. Myrtveit, E. Stensrud, A controlled experiment to assess the benefits of estimating with analogy and regression models, IEEE Transactions on Software Engineering 25 (1999) 510–525.

[78] Open Science Collaboration, Estimating the reproducibility of psychological science, Science 349 (2015) aac4716–3.

[79] P. Patil, R. Peng, J. Leek, What should researchers expect when they replicate studies? a statistical view of replicability in psychological science, Perspectives on Psychological Science 11 (2016) 539–544.

[80] R. Peng, Reproducible research in computational science, Science 334 (2011) 1226–1227.

[81] S. Pfleeger, Experimental design and analysis in software engineering, Annals of Software Engineering 1 (1995) 219–253.

[82] L. Pickard, B. Kitchenham, P. Jones, Combining empirical results in software engineering, Information and Software Technology 40 (1998) 811–821.

[83] C. Quesada-López, M. Jenkins, Function point structure and applicability validation using the ISBSG dataset: a replicated study, in: Proceedings of the 8th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement, ACM.

[84] C. Quesada-López, M. Jenkins, An empirical validation of function point structure and applicability: A replication study, in: Proceedings of the XVIII Ibero-American Conference on Software Engineering, CIBSE, 2015.

[85] C. Quesada-López, M. Jenkins, Function point structure and applicability: A replicated study, Journal of Object Technology 15 (2016) 2:1–26.

[86] A. Robins, P. Haden, S. Garner, Problem distributions in a cs1 course, in: Proceedings of the 8th Australian Conference on Computing Education, pp. 165–173.

[87] R. Rosenthal, The file drawer problem and tolerance for null results., Psychological Bulletin 86 (1979) 638–641.

[88] M. Ruhe, R. Jeffery, I. Wieczorek, Cost estimation for web applications, in: 25th International Conference on Software Engineering, IEEE Computer Society, 2003, pp. 285–294.

[89] M. Ruhe, R. Jeffery, I. Wieczorek, Using web objects for estimating software development effort for web applications, in: 9th IEEE International Software Metrics Symposium, pp. 30–37.

[90] N. Salleh, E. Mendes, J. Grundy, Investigating the effects of personality traits on pair programming in a higher education setting through a family of experiments, Empirical Software Engineering 3 (2014) 714–752.

[91] P. Sfetsos, P. Adamidis, L. Angelis, I. Stamelos, I. Deligiannis, Investigating the impact of personality and temperament traits on pair programming: A controlled experiment replication, in: 8th International Conference on the Quality of Information and Communications Technology, pp. 57–65.

[92] P. Sfetsos, I. Stamelos, L. Angelis, I. Deligiannis, Investigating the impact of personality types on communication and collaboration-viability in pair programming–an empirical study, Extreme programming and agile processes in software engineering (2006) 43–52.

[93] P. Sfetsos, I. Stamelos, L. Angelis, I. Deligiannis, An experimental investigation of personality types impact on pair effectiveness in pair programming, Empirical Software Engineering 21 (2009) 187–226.

[94] M. Shepperd, How do I know whether to trust a research result?, IEEE Software 32 (2015) 106–109.

[95] M. Shepperd, D. Bowes, T. Hall, Researcher bias: The use of machine learning in software defect prediction, IEEE Transactions on Software Engineering 40 (2014) 603–616.

[96] M. Shepperd, M. Cartwright, A replication of the use of regression towards the mean (R2M) as an adjustment to effort estimation models, in: 11th IEEE Intl. Softw. Metrics Sympsium (Metrics05), Computer Society Press, 2005.

[97] M. Shepperd, C. Schofield, Estimating software project effort using analogies, IEEE Transactions on Software Engineering 23 (1997) 736–743.

[98] F. Shull, J. Carver, S. Vegas, N. Juristo, The role of replications in empirical software engineering, Empirical Software Engineering 13 (2008) 211–218.

[99] F. da Silva, M. Suassuna, A. Frana, A. Grubb, T. Gouveia, C. Monteiro, I. dos Santos, Replication of empirical studies in software engineering research: a systematic mapping study, Empirical Software Engineering 19 (2012) 501–557.

[100] D. Simons, The value of direct replication, Perspectives on Psychological Science 9 (2014) 76–80.

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

[102] J. Spence, D. Stanley, Prediction interval: What to expect when you're expecting ... a replication, PloS ONE 11 (2016) e0162874.

[103] D. Stanley, J. Spence, Expectations for replications are yours realistic?, Perspectives on Psychological Science 9 (2014) 305–318.

[104] W. Stroebe, F. Strack, The alleged crisis and the illusion of exact replication, Perspectives on Psychological Science 9 (2014) 59–71.

[105] I. Vermeulen, C.J. Beukeboom, A. Batenburg, A. Avramiea, D. Stoyanov, B. van de Velde, D. Oegema, Blinded by the light: how a focus on statistical significance may cause p-value misreporting and an excess of p-values just below. 05 in communication science, Communication Methods and Measures 9 (2015) 253–279.

[106] X. Wan, W. Wang, J. Liu, T. Tong, Estimating the sample mean and standard deviation from the sample size, median, range and/or interquartile range, BMC Medical Research Methodology 14 (2014).