Replication studies considered harmful

Martin Shepperd
Brunel University London
Uxbridge
martin.shepperd@brunel.ac.uk

ABSTRACT

Context: There is growing interest in establishing software engineering as an evidence-based discipline. To that end, replication is often used to gain confidence in empirical findings, as opposed to reproduction where the goal is showing the correctness, or validity of the published results.
Objective: To consider what is required for a replication study to confirm the original experiment and apply this understanding in software engineering.
Method: Simulation is used to demonstrate why the prediction interval for confirmation can be surprisingly wide. This analysis is applied to three recent replications.
Results: It is shown that because the prediction intervals are wide, almost all replications are confirmatory, so in that sense there is no 'replication crisis', however, the contributions to knowledge are negligible.
Conclusion: Replicating empirical software engineering experiments, particularly if they are under-powered or under-reported, is a waste of scientific resources. By contrast, meta-analysis is strongly advocated so that all relevant experiments are combined to estimate the population effect.

CCS CONCEPTS

• **General and reference → Empirical studies**; **Experimentation**;

KEYWORDS

Software engineering, empirical study, replication, evidence

ACM Reference Format:
Martin Shepperd. 2018. Replication studies considered harmful. In *ICSE-NIER'18: 40th International Conference on Software Engineering: New Ideas and Emerging Results Track, May 27-June 3, 2018, Gothenburg, Sweden.* ACM, New York, NY, USA, Article 4, 4 pages. https://doi.org/10.1145/3183399.3183423

1 INTRODUCTION

The idea of replicating empirical studies in order to enhance the trustworthiness of an empirical result is widely seen as a central tenet of the scientific method [20]. This paper addresses the rather overlooked questions—certainly within empirical software engineering—of *(i) how do we know if a replication study confirms the original study* and *(ii) were it to do so, what would this tell us?* Likewise if the replication fails to confirm the original study, what can we learn from this?

Using empirical evidence, typically through experiment, observation and case study, to underpin software engineering has been gaining traction in recent years. This has been stimulated in part by the seminal paper promoting evidence-based software engineering from Kitchenham et al. [15]. Clearly it's desirable to understand which methods and techniques 'work', to what extent, and in what contexts? From this, emerged the idea of the community building knowledge through replication studies [26]. A mapping study by da Silva et al. [8] and extended by Bezerra et al. [1] found 135 articles reporting a total of 184 replications (1994–2012).

However, recently concerns have been expressed about the reliability of empirical findings both within software engineering [11] and beyond, e.g., in experimental psychology [20]. Consequently, replication studies have been seen as important for two reasons. First, in terms of their ability to increase our confidence in specific empirical findings via confirmation, or otherwise. Second, as a form of sample to estimate the reliability of software engineering empirical studies in general.

The remainder of this paper, briefly reviews the state of replication studies in software engineering focusing on experimentation. Next I show, by simulation, that simply through sampling error we can obtain considerably more diverse results than might be imagined. This is applied to a selection of published replication studies by formally computing prediction intervals. Finally I discuss the implications and argue that research effort would be far more usefully deployed performing meta-analyses.

2 RELATED WORK

A key work that sets out the generally accepted view of role of replication[1] studies in software engineering is by Shull et al. [26]. They state that:

> "if the results of the original study are reproduced using different experimental procedures, *then the community has a very high degree of confidence that the result is real*" [26] [my italics].

It would be fair to say that this represents the majority view in empirical software engineering and the paper is highly cited[2]. In other words, the primary purpose of replication is to increase confidence.

But how similar must results be to constitute confirmation? Curiously this has not been directly addressed, so whilst researchers

ICSE-NIER'18, May 27-June 3, 2018, Gothenburg, Sweden
© 2018 Association for Computing Machinery.
ACM ISBN 978-1-4503-5662-6/18/05…$15.00
https://doi.org/10.1145/3183399.3183423

[1] N.B., my focus is on 'conceptual' as opposed to 'exact' replications that deal with reproducibility questions and, in any case, can be problematic when using human participants.
[2] According to Google Scholar there are in excess of 200 citations (8.2.2018).

generally feel able to make judgements concerning a replication, I am unaware of any replication study in software engineering that has stated in some statistical sense how close a result must be to constitute a confirmation.

A range of approaches have been deployed to make comparisons. These include: comparison of (i) simple descriptors, e.g., means, (ii) goodness of fit measures, e.g., R-squared, (iii) correlations, (iv) null hypothesis significance testing (NHST) where both the original and replication study report statistical significance at an agreed α threshold and (v) standardised effect sizes such as Cohen's d or Cliff's δ. Of these, NHST is the dominant paradigm.

Unfortunately, NHST has come in for extensive criticism [6, 7]. For example, it has been argued that given the flexibility in choice of data and analysis methods the desire to have 'positive' findings is likely to substantially increase the likelihood of a false-positive above the nominal level set by α typically 5% [9, 27]. Another difficulty arising from the 'all or nothing' nature of NHST is publication bias due to the preference of authors, reviewers and editors for 'positive' results and the file-drawer problem [22]. Examples are reported from psychology [18] and software engineering [11] of the surprising prevalence of significant p values. Again this selectivity makes it more likely that a replication will fail to such a large effect as the original study [11].

A further problem is experimental power. For any experimental design, the power depends on sample size, measurement error, the number of comparisons being performed, and the effect size under investigation [9]. However, it is clear we work in a field that is dominated by low power studies [11, 13] and this is problematic in that it does not just mean a reduced likelihood of detecting true effects, it also implies increased likelihood of over-estimating the effect size or finding an effect that does not really exist [4]. Finally, NHST is impacted by sample size, so if the original and replication studies employ different numbers of experimental units this alone might lead to different values of p.

To summarise, although not generally made explicit, empirical software engineering researchers usually require both studies to report statistical significance (in the same direction) for the replication to be considered confirmatory. The meaning of neither study being significant is less self-evident [6]. This leaves a decision as to whether a replication 'confirms' the original study as largely being a subjective judgement.

3 SIMULATING REPLICATIONS

In order to give insight into the major role that sampling errors play on the variability of the experimental results, I use a simple Monte Carlo simulation[3]. Suppose we have two treatments X and Y and we want to compare them experimentally. Each experiment has 30[4] units, where a unit might be a participant, a data set, and so forth. Let's also suppose the experimental design is extremely simple and that the two samples are independent, as opposed to paired. We also assume the rather unlikely situation of no measurement errors, no publication bias and file drawer problems [27] and also that the underlying population is normally distributed.

Starting with the simplest case of no effect with $\mu = 100, \sigma = 20$, 10000 simulated experimental results behave as might expected from the Central Limit Theorem. However, as confirmed by Table 1 we actually observe a surprisingly wide range of possible effect sizes [5] with only just over half the experiments finding negligible or no effect [-0.2, +0.2]. Note that these simulation circumstances are considerably more propitious than we might expect in real life [11].

Next we contrast this simulation with a small positive effect using normal and rather more realistic mixed-normal distributions. These contaminated or mixed normal distributions generate heavy tails that differ from strictly normal distributions due to the presence of additional outliers [31, Chap. 1]. Figure 1 shows the boxplots of the experimental estimates of the effect size for these four cases (where None and Small are the true effect sizes and * denotes a distribution with outliers). It is clear that there is (i) a great deal of variability in all the results and (ii) small departures from normality greatly hinder our ability to detect a true small effect, exemplified by the very similar distributions for Small* and None*.

Figure 1: Boxplots of Random Experiments With and Without a True Small Effect

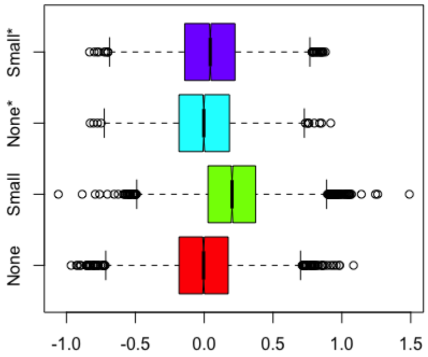

Table 1: Proportions of Different Effect Sizes (Using Cohen's d) for No True Effect

True Effect	Large -ve	Med -ve	Small -ve	None	Small +ve	Med +ve	Large +ve
None	0.1%	2.7%	19.5%	55.5%	19.7%	2.4%	0.20%
Small	0.02%	0.27%	5.79%	43.61%	37.78%	11.19%	1.34%
None*	0.05%	2.54%	20.0%	54.62%	20.17%	2.57%	0.05%
Small*	0.04%	1.76%	16.71%	52.11%	25.08%	4.25%	0.05%

Finally, we simulate the replication process by randomly drawing pairs of studies, without replacement, and observing the difference in results. Thus for each simulation of 10,000 experiments this yields 5,000 replications. Table 2 summarises the result in terms of confirmation, or Gelman's S-errors [9]. Unsurprisingly, when there is no true effect, the distribution of effect sign agreement is uniform. In the event of a normal distribution, and no other nuisance factors, something like 60% of replications will find an effect in the

[3] The R code, additional figures and associated materials are available from https://figshare.com/articles/_/5873754.

[4] Jørgensen et al. [11] report that in their survey of software engineering experiments 47% had a sample size of 25 or less.

same direction as the original study, but not necessarily with much concordance in terms of effect size. However, the presence of even a few outliers as per the mixed normal distributions reduces the number of replications that agree in direction to under one third. And this simulation simply focuses on sampling error. Introducing other sources of error e.g., measurement or excess researcher degrees of freedom [17] compound the situation.

Table 2: Counts of Simulated Replication Effect Direction Agreement

True Effect	- -	- +	+ -	+ +
None	1249	1246	1250	1255
Small	244	825	855	3076
None*	1241	1268	1246	1245
Small*	938	1201	1234	1627

Thus we can see that even in quite benign circumstances, replication is likely to be quite hit or miss simply because of sampling errors.

4 CONFIRMATION OF RESULTS IN SOFTWARE ENGINEERING

Next we consider the situation of replication studies within software engineering. As previously noted, two mapping studies [1, 8] have located well in excess of 100 replications, however, unfortunately careful examination reveals few report any measures of dispersion, e.g., variance of the measure of effect or response variable. This inhibits calculation of prediction intervals. Table 3 shows three examples that are selected because some calculations are possible.

As stated, replication studies in software engineering have not been in the habit of stating what range of values might be expected from a confirmatory replication. Formally we are asking what variation might arise just from sampling error. The Monte Carlo simulations from Section 3 indicate that this source of variability can be surprisingly large. The variability can be characterised as a prediction interval.

In this analysis the 95% prediction interval is reported using the approach due to Spence and Stanley [28] and implemented by the R package `predictionInterval`. Note a prediction interval differs from a confidence interval because we are concerned with the estimate from the specific study being replicated, as opposed to an estimate of the population effect size. Generally prediction intervals will be a little wider than confidence limits [29].

Table 3: Example Replications and Associated Prediction Intervals

Orig Study	Rep Study	Orig Eff Sz	Pred Int	Rep Eff Sz	Confirms?
[25]	[19]	0.101	[-0.33, 0.53][5]	-0.1[6]	Y
[12]	[24]	-0.176	[-0.84, 0.48]	0.122	Y
[3]	[2]	1.430	[0.05, 2.76]	1.090	Y

What is particularly noteworthy in Table 3 are the wide prediction intervals so, for instance in the second example, experiment [24] would confirm the original experiment [12] if anything from a large negative effect to a small-medium positive effect were detected. There are two contributory factors. First, small sample sizes, e.g., [3]. Second, small effect sizes, e.g., [12, 25] which are often driven down by high variance e.g., [19]. Thus, hugely varying results can simply be explained by sampling error. Of course this will in all probability be exacerbated by measurement error and publication bias so the foregoing prediction interval might be regarded as the best case scenario.

An alternative view is to regard studies that seek to answer the same research question as inputs to meta-analysis. To illustrate this, a simple meta-analysis is undertaken using the standardised mean difference effect size approach of Lipsey and Wilson [16] and implemented in the R package of `Metafor` from Viechtbauer [30]. The experimental results are pooled in order to estimate the population effect size.

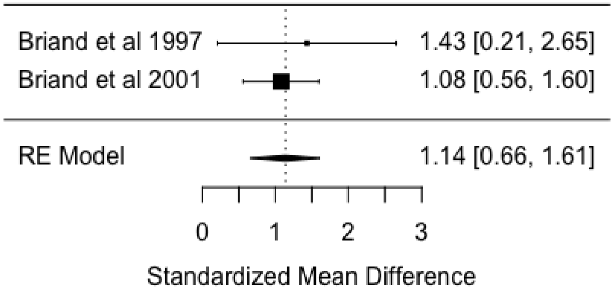

Figure 2: Forest plot of combining studies through meta-analysis

Fig. 2 shows a forest plot of the two studies of the effect of good design on comprehension for OO systems from [3] and [2]. The horizontal bars show the confidence intervals for the estimated effect sizes and the sizes of the centre points are proportional to the sample sizes. The standardised mean distance is a measure of mean difference normalised by standard deviation. Assuming a simple fixed effects model we obtain the an estimate of Cohen's d=1.14 [0.66, 1.61] denoted by the vertical dashed line. Note the limits are at least all in one direction and are narrower than either study individually (see Table 3). Thus we gain knowledge and precision as opposed to simply reporting that we can confirm the original weak finding.

5 DISCUSSION AND CONCLUSIONS

The title of this paper is intentionally provocative. The purpose, however, is to draw attention to the twin issues of how similar must a replication be to the original experiment to constitute confirmation and how effective is the process of replication for adding empirically-derived, software engineering knowledge? This paper has only considered experiments but in parallel, there is a case for developing, and applying, strong meta-analytic methods for qualitative studies e.g., case studies and action research.

[5] The interval is calculated by reasoning backwards from the replication study to the original study.

[6] The effect size is approximate due to the estimated pooled standard deviation.

Clearly, there is a place for us to consider reproducibility. That a study is reproducible should be considered minimally necessary [21]. But beyond this, we have shown there are two distinct difficulties with replication studies as practised in software engineering. First, when the prediction interval can be properly constructed [7, 28] what constitutes a confirmation is often a good deal broader than might be anticipated. This — as has been shown both by simulation and by example — could include a wide range of effect sizes, sometimes in *both* directions. Thus confirmation, particularly for low-powered experiments and small effect sizes, can be trivial.

In contrast, combining studies through meta-analysis enables all relevant studies to be combined so that we may generate our best estimate (complete with confidence interval) of the effect in question. This yields more nuanced information than reducing the matter to an all or nothing matter of confirmation or disconfirmation. Of course, meta-analyses cannot overcome the problems of poor quality primary studies or selective reporting and publication. However, techniques such as funnel plots can at least help highlight these problems [23]. Lack of heterogeneity (perhaps due to methodological differences or the existence of meaningful sub-populations) can also be detected and investigated [16, 23].

This implies the following recommendations for the empirical software engineering community:

(1) Properly report studies and in particular provide information on the *dispersion*, e.g., variance, of the response (dependent) variables. Without this information neither the prediction interval can be computed (for replication analysis) nor is meta-analysis possible. Given the current paucity of such information, this is the biggest single contributor to wasted research effort.
(2) Construct prediction intervals *prior* to conducting replication studies and understand that under-powered studies of small effects (i.e., much of empirical software engineering [11, 13]) can be trivially replicated, but the contribution to knowledge will be extremely small.
(3) Limit replications to matters of reproducibility (where warranted).
(4) Conduct, independent studies of important research questions where the effects may matter to practising software engineers and combine results using meta-analytic techniques. Avoid close replications since these may violate the independency assumption underlying meta-analysis [14]. Also, consider corrections to the meta-analysis [23] needed due to potential bias from inflated effect size estimates from the first study arising from publication bias [10].

ACKNOWLEDGMENTS

This work was partly funded by the EPSRC Grant EP/P025196/1.

REFERENCES

[1] R. Bezerra, F. da Silva, A. Santana, C. de Magalhães, and R. Santos. 2015. Replication of Empirical Studies in Software Engineering: An Update of a Systematic Mapping Study. In *Intl. Symp. on Emp. Softw. Eng. and Measurement*. 1–4.

[2] L. Briand, C. Bunse, and J. Daly. 2001. A controlled experiment for evaluating quality guidelines on the maintainability of object-oriented designs. *IEEE Transactions on Software Engineering* 27, 6 (2001), 513–530.

[3] L. Briand, C. Bunse, J.. Daly, and C. Differding. 1997. An Experimental Comparison of the Maintainability of Object-Oriented and Structured Design Documents. *Emp. Softw. Eng.* 2, 3 (1997), 291–312.

[4] K. Button, J. Ioannidis, C. Mokrysz, J. Nosek, B.and Flint, E. Robinson, and M. Munafò. 2013. Power failure: why small sample size undermines the reliability of neuroscience. *Nature Reviews Neuroscience* 14, 5 (2013), 365–376.

[5] J. Cohen. 1992. A power primer. *Pyschological Bulletin* 112, 1 (1992), 155–159.

[6] D. Colquhoun. 2014. An investigation of the false discovery rate and the misinterpretation of p-values. *Royal Society Open Science* 1, 3 (2014).

[7] G. Cumming. 2008. Replication and p intervals: p values predict the future only vaguely, but confidence intervals do much better. *Perspectives on Psychological Science* 3, 4 (2008), 286–300.

[8] F. da Silva, M. Suassuna, A. França, A. Grubb, T. Gouveia, C. Monteiro, and I. dos Santos. 2012. Replication of empirical studies in software engineering research: a systematic mapping study. *Emp. Softw. Eng.* 19, 3 (2012), 501–557.

[9] A. Gelman and J. Carlin. 2014. Beyond power calculations: Assessing Type S (sign) and Type M (magnitude) errors. *Perspectives on Psychological Science* 9, 6 (2014), 641–651.

[10] J. Ioannidis. 2008. Why most discovered true associations are inflated. *Epidemiology* 19, 5 (2008), 640–648.

[11] M. Jørgensen, T. Dybå, K. Liestøl, and D. Sjøberg. 2016. Incorrect Results in Software Engineering Experiments: How to Improve Research Practices. *J. of Syst. & Softw.* 116 (2016), 133–145.

[12] M. Jørgensen, U. Indahl, and D. Sjøberg. 2003. Software effort estimation by analogy and 'regression toward the mean'. *J. of Syst. & Softw.* 68, 3 (2003), 253–262.

[13] V. Kampenes, T. Dybå, J. Hannay, and D. Sjøberg. 2007. A systematic review of effect size in software engineering experiments. *Information and Software Technology* 49 (2007), 1073–1086.

[14] B. Kitchenham. 2008. The role of replications in empirical software engineering — a word of warning. *Emp. Softw. Eng.* 13, 2 (2008), 219–221.

[15] B. Kitchenham, T. Dybå, and M. Jørgensen. 2004. Evidence-based Software Engineering. In *26th IEEE International Conference on Software Engineering (ICSE 2004)*. IEEE Computer Society, 273–281.

[16] M. Lipsey and D. Wilson. 2001. *Practical meta-analysis*. Sage Publications.

[17] E. Loken and A. Gelman. 2017. Measurement error and the replication crisis. *Science* 355, 6325 (2017), 584–585.

[18] E. Masicampo and D. Lalande. 2012. A peculiar prevalence of p values just below .05. *Journal of Experimental Psychology* 65, 11 (2012), 2271–2279.

[19] I. Myrtveit and E. Stensrud. 1999. A controlled experiment to assess the benefits of estimating with analogy and regression models. *IEEE Transactions on Software Engineering* 25, 4 (1999), 510–525.

[20] Open Science Collaboration. 2015. Estimating the reproducibility of psychological science. *Science* 349, 6251 (2015), aac4716–3.

[21] R. Peng. 2011. Reproducible Research in Computational Science. *Science* 334, 6060 (2011), 1226–1227.

[22] R. Rosenthal. 1979. The "File Drawer Problem" and tolerance for null results. *Psychological Bulletin* 86, 3 (1979), 638–641.

[23] F. Schmidt and J. Hunter. 2014. *Methods of meta-analysis: Correcting error and bias in research findings*. Sage Publications.

[24] M. Shepperd and M. Cartwright. 2005. A Replication of the Use of Regression Towards the Mean (R2M) as an Adjustment to Effort Estimation Models. In *11th IEEE Intl. Softw. Metrics Symposium (Metrics05)*. Computer Society Press.

[25] M. Shepperd and C. Schofield. 1997. Estimating software project effort using analogies. *IEEE Transactions on Software Engineering* 23, 11 (1997), 736–743.

[26] F. Shull, J. Carver, S. Vegas, and N. Juristo. 2008. The role of replications in empirical software engineering. *Emp. Softw. Eng.* 13, 2 (2008), 211–218.

[27] J. Simmons, L. Nelson, and U. Simonsohn. 2011. False-positive psychology: Undisclosed flexibility in data collection and analysis allows presenting anything as significant. *Psychological Science* 22, 11 (2011), 1359–1366.

[28] J. Spence and D. Stanley. 2016. Prediction Interval: What to Expect When You're Expecting … A Replication. *PloS ONE* 11, 9 (2016), e0162874.

[29] D. Stanley and J. Spence. 2014. Expectations for replications are yours realistic? *Perspectives on Psychological Science* 9, 3 (2014), 305–318.

[30] W. Viechtbauer. 2010. Conducting meta-analyses in R with the metafor package. *Journal of Statistical Software* 36, 3 (2010), 1–48.

[31] R. Wilcox. 2012. *Introduction to Robust Estimation and Hypothesis Testing* (3rd ed.). Academic Press.