Reproducible Research – What, Why and How

Lech Madeyski*, Barbara Kitchenham†

Abstract The minimum criterion for valid research and data analyses in software engineering and other disciplines is reproducibility. The aim of this paper is to explain the concept of Reproducible Research (RR), embracing reporting modern data analyses in a reproducible manner, its importance, and how researchers and data analysts can use current technology to adopt RR and increase reproducibility of their data analysis tasks.

Keywords reproducible research, Literate Programming, R, LaTeX, knitr

1 Introduction

Recent results in empirical software engineering have cast some doubts on the validity of our software engineering research practices. For example, Martin Shepperd, David Bowes and Tracy Hall[1] analyzed the results of 42 papers reporting studies comparing methods for predicting fault-proneness. They found that the explanatory factor that accounted for the largest percentage of the differences among studies (i.e. 30%) was research group. In contrast prediction method, which was the main topic of research, accounted for only 1.3% of the variation among studies. They commented that "*It matters more who does the work than what is done.*" and "*Until this can be satisfactorily addressed there seems little point in conducting further primary studies*". We recently undertook a replication of a meta-analysis study of code reading methods and found that 70% of the variation among the individual studies was explained by design type[2]. Thus, after 18 studies performed over a period of 18 years, we are unable to confirm whether one reading method is better than another.

Both these studies suggest that there is a lot of wasted effort in empirical software engineering. Furthermore these results are probably only the tip of the iceberg. Barbara Kitchenham and Emilia Mendes point out that reported accuracy statistics for cost estimation studies claiming to use the same method on the same dataset are inconsistent[3]. In addition, Dag Sjøberg and his colleagues reported that replications performed by researchers who undertook the initial study were more likely to find the same result than replications undertaken by independent researchers[4].

Reproducibility of data analyses and ability to repeat easily exactly the same analytic procedures, with the same set of parameters, on new data sets are also extremely important from researchers' and data analysts' point of view. Reproducibility problems have, however, serious consequences in a wide range of disciplines. For example, in the context of drug trials, Lev Osherovich reports that "*an 'unspoken rule' among early stage VCs [Venture Capitals] is that at least 50% of published studies, even those in top-tier academic journals, 'can't be repeated with the same conclusions by an industrial lab.'* "[5] John Ioannidis and his colleagues reported that only 2 of 18 research papers published by Nature Genetics journal (one of the highest ranked journals in the world, with an $Impact\ Factor > 35$) could be fully reproduced[6]. The reasons for this included data sets and home made software disappearing, or the specification of data processing and analysis being incomplete.

In the field of psychology, Harold Pashler and Eric-Jan Wagenmakers[7] report "a crisis of confidence in psychological science reflecting an unprecedented level of doubt among practitioners about the reliability of research findings in the field". Some of the problems they report will have a familiar ring to many software engineering researchers:

- Psychologists are often unwilling or unable to share their published data for reanalysis.
- Scientists were having difficulties replicating well-known results.
- Researchers admitted to using "questionable practices", such as exploring multiple dependent variables or covariates and only reporting those that deliver significant results.

They point out that these problems have been associated with an increasingly "hypercompetitive academic climate and an incentive scheme that provides rich rewards for overselling one's work and few rewards at all for caution and circumspection". Furthermore, replications, which remain the basic scientific mechanism to ensure incorrect results are detected, are being undertaken less frequently "presumably reflecting an incentive scheme gone askew".

2 The Goal of Reproducible Research

It is beyond the ability of individual researchers to change the social context in which scientific research takes place, but in this paper, we advocate data analysts and researchers like ourselves reporting our analyses using a reproducible research (RR) approach. The goal of RR is to publish research papers that incorporate the basic data and the specific methods used to analyze that data. We need to be clear at this point that this will not solve the problem of lack of replications of empirical studies nor combat outright fraud. It only ensures that given the data reported by the authors and the data analysis they used, an independent researcher will obtain the analysis results reported in the paper. This can be contrasted with replication where independent researchers run the same basic experiment again with different subjects and/or experimental

materials, and thus, generate a new data set to test the original experimental hypothesis. However, in our view, RR is the minimum requirement to ensure our empirical studies are contributions to scientific progress rather than just elaborate anecdotes.

In the following sections, we define RR and report the origins of the idea, explaining its importance. We identify the basic mechanisms needed to make research outcomes and data analyses reproducible and provide a brief worked example of how the approach can be used in practice. We consider legal and contractual issues, and summarize motivations for and barriers to adopting RR.

3 The Origins and Definition of Reproducible Research

The term reproducible research is attributed to Professor Jon Claerbout of Stanford University, who in 1990, imposed the standard of make files for all the figures and computational results published by the Stanford Exploration Project. Furthermore, in 2000, he and his students shared their experience of creating a reproducible research environment[8].

Reproducible research refers to the idea that the ultimate product of research is the paper plus its computational environment. That is, a reproducible research document, incorporates the textual body of the paper plus any necessary supplementary materials including protocols, the data used by the study, and the analysis steps (algorithms) used to process the data, in the context of an open access environment that is used to compile these pieces of information into the resulting document. Thus, an independent researcher or data analyst can reproduce the results, verify the findings, and create new work based on the original research, for example, conduct alternative analyses of the same data, or replicate the original analysis on an updated or new data set.

The standard of reproducibility is weaker and less demanding than full replication. However, Roger Peng notes that limited data and code exploration may be sufficient to verify the quality of the presented claims bridging the gap between no replication and full replication in the evidence-generating process[9].

Within the domain of software engineering, there has been some discussion of “laboratory packages”, which were pioneered by Victor Basili with the aim of assisting replications by providing additional detailed information about specific experiments, such as experimental materials and detailed instructions related to the experimental process.

4 The Value of RR

In 2005, John Ioannidis used simple simulations to prove that most claimed research results are false[10]. Only well-conducted randomized controlled trials, and meta-analyses of such trials, are likely to be true about 85% of the time. He also reported factors that decrease the likelihood of research finding being true, including some that are particularly likely to affect the software engineering domain:

- Excessive flexibility in designs, definitions, outcomes, and analytic modes which enable questionable research practices. Our own meta-analysis of inspection studies provides additional evidence that supports this factor[2].
- Extensive financial and other interests and prejudices which undermine researchers' independence.
- Hot topics with many scientific teams involved which fuel the "hypercompetitive academic climate".

Scientific claims that cannot, at the very least, be reproduced undermine the basic principles of science and hinder the uptake of research results by industry. In software engineering, it is generally accepted that replication is an important part of the scientific process, but it is unreasonable to expect researchers to replicate studies that cannot even achieve the minimum requirement of reproducibility. Thus, RR provides the minimum basis for good scientific practice. Furthermore, if full replication is not possible (due to cost or time constraints), reproducibility becomes the only standard for judging scientific claims.

Sidebar 1. Literate Programming

If we accept that the output of research is not just a paper, but also the full computational environment, then we can use Donald Knuth's concept of Literate Programming to achieve RR. Literate Programming treats a program as a piece of literature addressed to human beings rather than a computer. The key assumptions are:

- A program should have plain language explanations interspersed with source code.
- The source code, data and plain language explanations are combined together.
- Results of program or code chunks are automatically included when document is created (so no exporting is needed).
- After recompilation, changes are automatically incorporated if code or data sets change.
- Tools are available to make this simple to achieve.

5 Methods for Reproducible Research

If we follow the principles of Literate Programming as outlined in Sidebar 1, our research or data analysis should be easily reproducible, allowing independent professionals to understand how we obtained our results. In practice, we used the following methods and tools to support production in our meta-analysis paper as well as this paper:

- The R programming language was used for all data analyses (http://www.r-project.org/).
- All references were stored in pure BibTeX and managed using BibDesk.
- The paper was written in LaTeX and incorporated the R code using an R package called **knitr** (http://yihui.name/knitr), which is a replacement for Sweave. Several other R packages were employed as well.

We used R because it is one of the most popular languages for data analysis. It is a mature language derived from S-plus, as well as being open source and free to use. The R language is important for RR because the use of a statistical language provides more traceability to the details of the statistical

analysis than a statistical package that includes various built-in defaults that are not explicitly reported. In addition, typing an R script is more reproducible and easier to communicate than using the point-and-click user interface often adopted in other statistical packages. It is also the case that new statistical and data analysis methods tend to appear in R far sooner than in other statistical tools, partly because it is the developers of the methods who often produce the supporting R packages. Other commercial packages simply do not provide all the capabilities provided by R.

We used BibDesk because:

- It is based on BibTeX and can import and populate BibTeX fields from the clipboard, files and digital libraries.
- It is simple to use but very powerful. Papers and e-books can be attached to their BibTeX descriptions. It also provides good search capabilities based on BibTeX fields (e.g. keywords, titles, or authors) and attachments (e.g. papers and books in pdf).
- It is maintained, frequently updated and free. BibDesk is available on Mac OS X. A cross-platform (Java-based) alternative is JabRef (http://jabref.sourceforge.net/).

The **knitr** package provides the mechanism for linking R-code into basic LaTeX documents. In fact, all basic document production can be done using R Studio. We found **xtable** package particularly useful for generating, on the fly, nicely styled tables in LaTeX or HTML from data structures produced using R. In addition, graphical capabilities of R are available to construct high quality figures using, e.g., **ggplot2** package. Furthermore, LaTeX is an ideal language for representing mathematical and statistical equations.

Other useful packages like **repmis** or **packrat** include a collection of features for reproducible research with R, for example, the ability to install specific R package versions. This is important functionality because changes between packages can impede reproducibility. This is a particular problem since changes between packages can be subtle, for example, related to default parameters, which may not be specified explicitly in the code responsible for analysis (even though good practice suggests they should be).

6 Example of the RR Process

Figure 1 visually presents the workflow of the RR process inspired by Roger D. Peng. To streamline the uptake of RR we present steps needed to create a simple example of RR:

1) Install R, LaTeX (e.g., TeX Live or Miktex) and RStudio
2) Optionally launch RStudio (find the RStudio Preferences window, select the "Sweave" option and make sure the "Weave Rnw file using:" option is set to "knitr", the "Typeset LaTeX into PDF using:" option is set to "pdfLaTeX").
3) Install and load the **knitr** R package:

```
install.packages('knitr', dependencies = TRUE)
library('knitr')  # Load knitr
```

4) Create an `Example.Rnw` file (in RStudio using the "File" - "New file" menu option with the "R Sweave" submenu option) containing text (with LaTeX markup) and data analysis (with R code chunks, i.e., sections of R code). Each chunk can have its own options to configure how it is rendered. `Example.Rnw` file:

```
\documentclass{article}
\begin{document}
\title{Reproducible Research -- What, Why and How}
\maketitle
<<setup, include=FALSE>>=
#Use devtools to install the development version from my web
      site:
# install.packages("devtools", dependencies = T, repos = "http
      ://cran.rstudio.com/")
# library(devtools)
# devtools::install_url("http://madeyski.e-informatyka.pl/
      download/R/reproducer_0.1.2.tar.gz")
# library(reproducer)
#Or install the stable version from CRAN:
install.packages('reproducer', dependencies = TRUE)
library(reproducer)
library(ggplot2)
@
Text...
<<myDataAnalysis, include=TRUE, warning=FALSE, message=FALSE,
      results='markup', cache=FALSE, tidy=TRUE, tidy.opts=list(
      blank=FALSE, width.cutoff=50), out.height="0.49\\
      textheight">>=
# descriptive analysis
summary(Madeyski15SQJ.NDC$simple)
summary(Madeyski15SQJ.NDC$advanced)
reproducer::boxplotAndDensityCurveOnHistogram(Madeyski15SQJ.NDC,
       "simple", 0, 100)
reproducer::boxplotAndDensityCurveOnHistogram(Madeyski15SQJ.NDC,
       "advanced", 0, 100)
# inferential analysis using Wilcoxon test
stats::wilcox.test(Madeyski15SQJ.NDC$simple,Madeyski15SQJ.
      NDC$advanced,paired=TRUE)
@
Text...
\end{document}
```

5) Compiling the .Rnw to .tex and .pdf in one of the following ways:

- Press "Compile PDF" button in RStudio
- Type into the R console:

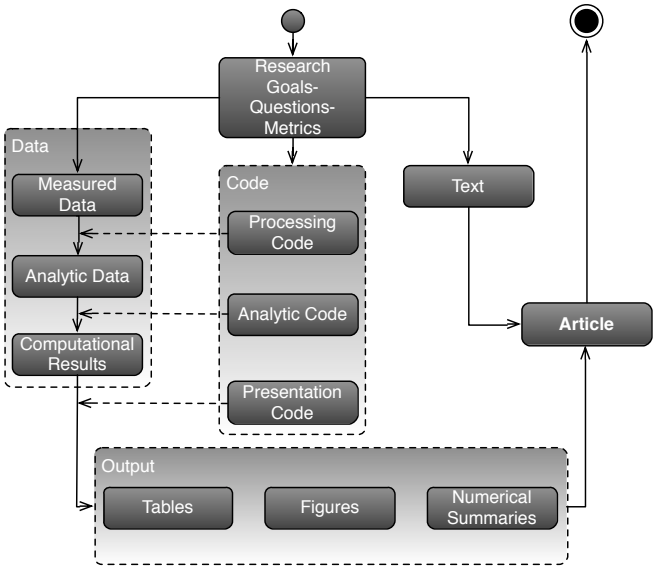


Fig. 1. Reproducible Research in action

```
knit2pdf('./Example.Rnw')
```

- Type into terminal/OS shell:

```
Rscript -e "library(knitr);
knit2pdf('./Example.Rnw')"
```

This article will serve as a working example of the presented RR approach. We will use a real data set recently analysed by Madeyski and Jureczko [11]. The analytic example presented in this section is based on empirical comparison of simple and advanced software defect prediction models performed on thirty-four (15 industrial and 19 open source) versions of software projects. The advanced models, using not only software product metrics but also the NDC (Number of distinct committers) process metric, outperform the simple ones, using only product metrics, in terms of percentage of classes that must be tested in order to find 80% of software defects.

We will use this real data set and perform a simple descriptive and inferential statistical analysis for illustration purposes.

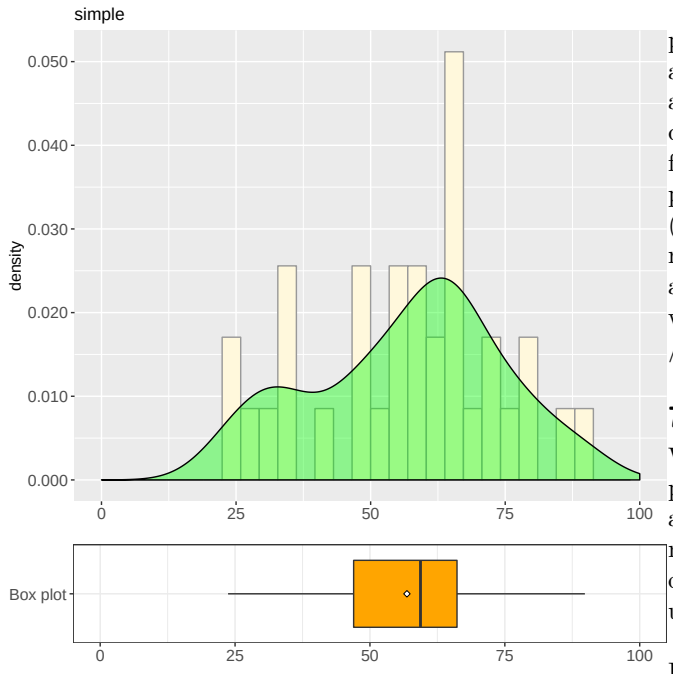
```
summary(Madeyski15SQJ.NDC$simple)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  23.71   46.98   59.37   56.86   66.14   89.84
```

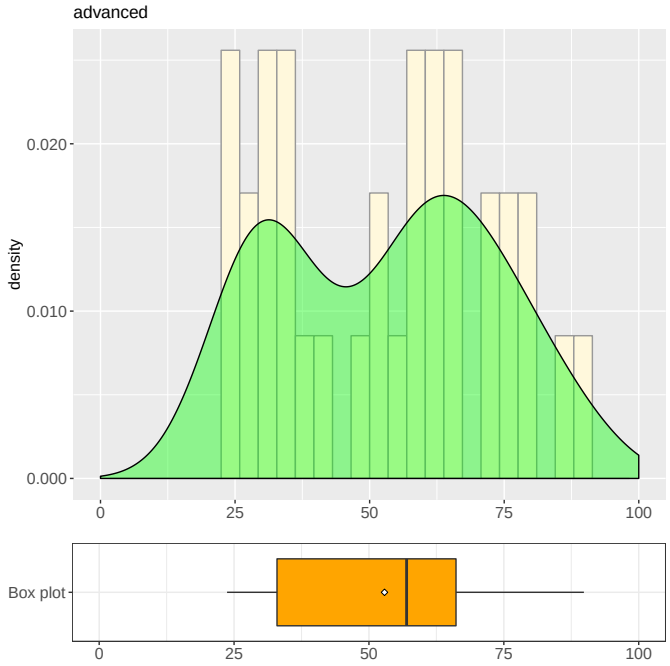
```
summary(Madeyski15SQJ.NDC$advanced)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  23.71   32.96   56.98   52.88   66.14   89.84
```

```
reproducer::boxplotAndDensityCurveOnHistogram(Madeyski15SQJ.NDC, "simple",
      0, 100)
```




```
reproducer::boxplotAndDensityCurveOnHistogram(Madeyski15SQJ.NDC, "advanced
      ", 0, 100)
```



As the data are non-normal, t-tests may not be appropriate and an alternative is Wilcoxon paired test.

```
stats::wilcox.test(Madeyski15SQJ.NDC$simple,Madeyski15SQJ.NDC$advanced,
      paired=TRUE)
```

```
        Wilcoxon signed rank test with continuity
             correction

data:  Madeyski15SQJ.NDC$simple and Madeyski15SQJ
    .NDC$advanced
V = 128, p-value = 0.01577
alternative hypothesis: true location shift is
    not equal to 0
```

Readers are invited to reproduce our small working example of reproducible research. All of our analyzes as well as data are encapsultated in the **reproducer** R package we created and made available from http://madeyski.e-informatyka.pl/download/R/reproducer_0.1.2.tar.gz as well as from the official CRAN repository[12]. We prepared this paper using the presented approach and tools as well as will make the paper (if accepted) available at http://madeyski.e-informatyka.pl/reproducible-research/ to stimulate the uptake of the RR approach by the software engineering community. Our simple working example (`Example.Rnw` file) is now available at http://madeyski.e-informatyka.pl/download/R/Example.Rnw.

7 Copyright and Licensing Issues

Works created by individuals (including papers, books, talks, posters, course syllabi, lecture notes, data and software) are automatically copyrighted, even without a formal copyright notice. If you work in industry it is likely that your employer owns the copyright, however, in academic settings, researchers usually retain copyright on anything they produce.

At the International Workshop on Empirical Software Engineering Issues held at Dagstuhl Castle, Germany in 2006, participants discussed data sharing and concluded that there were areas of dissent among the software engineering community. Some people wanted licensing others were opposed,

and there was no agreement as to who owns the data. Basili and colleagues[13] published a proposal for a data and artifact sharing agreements in software engineering research. However, another approach that is being widely considered, is easy to adopt, and matches scientific community norms is CC-BY which says, use my work however you wish, but make sure you credit me in the manner I specify. Additionally, Open Data Commons has produced three open solutions specifically for data, datasets and databases. Open Data Commons Attribution Licence ODC-BY licence (http://opendatacommons.org/licenses/by/summary/) is compatible with CC-BY. The popular DBLP Computer Science Bibliography database (http://www.informatik.uni-trier.de/~ley/db/) released data under this license.

8 Motivations for, and Barriers to Adopting RR

In our paper on the use of RR in Security and Privacy[14], we present a detailed SWOT analysis of RR. Here we summarize the main motivations for, and barriers to, adopting RR. The most important reasons for adopting the use of RR are that:

- RR actively supports a more rigorous approach to scientific research, and may hopefully reduce the number of research findings or data analysis outcomes that are later found to be incorrect.
- For a research group or industrial lab, RR supports the preservation of group knowledge in terms of long-term conservation of both reference lists and experimental data together with its supporting data analysis. This means that knowledge of current research is not lost when researchers or analysts move on, and it is easier for new members of a group to understand and build on previous research.
- Funding agencies in the US and UK and journals are increasingly adopting open research policies. If they start to enforce such polices, we will have no choice but to comply with basic RR principles.

The most significant barriers to the adoption of RR are:

- In software engineering, it is often the case that data cannot be distributed due to confidentiality issues. It may be that, in the future, we need to reconsider the scientific value of confidential data.
- In our experience, adopting an RR approach requires additional effort from individual researchers. Some researchers could argue that, although this might be necessary for researchers aiming at internationally leading journals in the critical disciplines such as medicine, it is too much of an overhead for software engineering researchers.
- Other researchers may be able to pre-empt our future research plans by making use of our data or analysis methods before we are able to complete our planned research programme. Earl Barr and colleagues[15] discuss a range of possible solutions to this problem including giving researchers exclusive rights to their data for a limited period of time.

9 Conclusions

Evidence of bad practice, and even misconduct, continue to surface in a range of scientific disciplines, including software engineering. These problems are causing other disciplines, such as psychology and medicine, to review and improve their scientific practices. Since methods such as Literate Programming and many of the tools needed to support reproducible research originated in the Computer Science domain, it seems appropriate for researchers in software engineering and computer science to think seriously about adopting RR.

It may be that RR turns out to be short-term fad, although personally we doubt it, but adopting some of the basic principles is not too difficult, and whatever happens, must be judged to be good scientific practice. In the worst case, some of us will have to change our established methods of working: learning or returning to LaTeX (which requires some getting used to) and adopting the R language for our statistical analyses (which we would recommend to all empirical researchers, irrespective of its support for RR). For those of us who, like Kitchenham, are getting near retirement, it may seem to be a lot of effort for little return. However, while we are still supervising the next generation of research students, we owe it to them to make them aware both of the problems with current research practices, and of the current range of possible solutions to them. Furthermore, it is likely that our younger colleagues will find adopting these techniques relatively easy and, perhaps, more interesting that their existing techniques.

As we have mentioned, RR cannot solve all the problems that plague scientific research. It supports the minimum level of validity that we should expect of research outcomes. Nonetheless, we believe that RR incorporates good scientific practice, and we strongly advocate the adoption of RR by software engineering and computer science researchers and data analysts.

References

[1] M. Shepperd, D. Bowes, and T. Hall, "Researcher Bias: The Use of Machine Learning in Software Defect Prediction," *IEEE Transactions in Software Engineering*, vol. 40, no. 6, pp. 603–616, 2014. DOI: 10.1109/TSE.2014.2322358.

[2] L. Madeyski and B. Kitchenham, "How variations in experimental designs impact the construction of comparable effect sizes for meta-analysis," Wroclaw University of Technology, PRE W08/2015/P-019, 2015.

[3] B. Kitchenham and E. Mendes, "Why comparative effort prediction studies may be invalid," in *Proceedings of the 5th International Conference on Predictor Models in Software Engineering*, ser. PROMISE '09, ACM, 2009, 4:1–4:5. DOI: 10.1145/1540438.1540444.

[4] D. I. K. Sjøberg, J. E. Hannay, O. Hansen, V. B. Kampenes, A. Karahasanovic, N.-K. Liborg, and A. C. Rekdal, "A survey of controlled experiments in software engineering.," *IEEE Transactions on Software Engineering*, vol. 31, no. 9, pp. 733–753, 2005. DOI: 10.1109/TSE.2005.97.

[5] L. Osherovich, "Hedging against academic risk," *Science–Business eXchange*, vol. 4, no. 15, 2011. DOI: 10.1038/scibx.2011.416.

[6] J. P. A. Ioannidis, D. B. Allison, C. A. Ball, I. Coulibaly, X. Cui, A. C. Culhane, M. Falchi, C. Furlanello, L. Game, G. Jurman, J. Mangion, T. Mehta, M. Nitzberg, G. P. Page, E. Petretto, and V. van Noort, "Repeatability of published microarray gene expression analyses," *Nature Genetics*, vol. 41, pp. 149–155, 2009. DOI: 10.1038/ng.295.

[7] H. Pashler and E.-J. Wagenmakers, "Editors' Introduction to the Special Section on Replicability in Psychological Science: A Crisis of Confidence?" *Perspectives on Psychological Science*, vol. 7, no. 6, pp. 528–530, 2012. DOI: 10.1177/1745691612465253.

[8] M. Schwab, M. Karrenbach, and J. Claerbout, "Making Scientific Computations Reproducible," *Computing in Science and Engineering*, vol. 2, no. 6, pp. 61–67, Nov. 2000. DOI: 10.1109/5992.881708.

[9] R. D. Peng, "Reproducible research in computational science," *Science*, vol. 334, no. 6060, pp. 1226–1227, 2011. DOI: 10.1126/science.1213847.

[10] J. P. A. Ioannidis, "Why Most Published Research Findings Are False," *PLoS Medicine*, vol. 2, no. 8, pp. 696–701, 2005. DOI: 10.1371/journal.pmed.0020124.

[11] L. Madeyski and M. Jureczko, "Which Process Metrics Can Significantly Improve Defect Prediction Models? An Empirical Study," *Software Quality Journal*, vol. 23, no. 3, pp. 393–422, 2015. DOI: 10.1007/s11219-014-9241-7.

[12] L. Madeyski, *reproducer: Reproduce Statistical Analyses and Meta-Analyses*, R package version 0.1.4 (http://CRAN.R-project.org/package=reproducer), 2015.

[13] V. R. Basili, M. V. Zelkowitz, D. I. Sjøberg, P. Johnson, and A. J. Cowling, "Protocols in the use of empirical software engineering artifacts," *Empirical Software Engineering*, vol. 12, no. 1, pp. 107–119, Feb. 2007. DOI: 10.1007/s10664-006-9030-4.

[14] L. Madeyski, B. A. Kitchenham, and S. L. Pfleeger, "Why reproducible research is beneficial for security research," *IEEE Security and Privacy (under review)*, 2015, preprint: http://madeyski.e-informatyka.pl/download/MadeyskiKitchenhamPfleeger15.pdf.

[15] E. Barr, C. Bird, E. Hyatt, T. Menzies, and G. Robles, "On the shoulders of giants," in *Proceedings of the FSE/SDP Workshop on Future of Software Engineering Research*, ser. FoSER '10, ACM, 2010, pp. 23–28. DOI: 10.1145/1882362.1882368.

Lech Madeyski is an Associate Professor at Wroclaw University of Technology, Poland. His research has focused on empirical software engineering including statistical analysis and meta-analysis of empirical studies. Contact him at Lech.Madeyski@pwr.edu.pl

Barbara Kitchenham is a Professor of Quantitative Software Engineering at Keele University in the UK. Her most recent research has focused on the application of evidence-based practice to software engineering. Contact her at b.a.kitchenham@keele.ac.uk.