

The Role of Data Stewardship in Software Sustainability and Reproducibility

Maria J. Cruz
TU Delft Library
Delft University of Technology
Delft, The Netherlands
m.j.marquesdebarrosacruz@vu.nl

Shalini Kurapati
Technology, Policy and Management
Delft University of Technology
Delft, The Netherlands
s.kurapati@tudelft.nl

Yasemin Turkyilmaz-van der Velden
Applied Sciences/ Mechanical,
Materials, and Maritime Engineering
Delft University of Technology
Delft, The Netherlands
y.turkyilmaz-vandervelden@tudelft.nl

Abstract—Software and computational tools are instrumental for scientific investigation in today’s digitized research environment. Despite this crucial role, the path towards implementing best practices to achieve reproducibility and sustainability of research software is challenging. Delft University of Technology has begun recently a novel initiative of data stewardship — disciplinary support for research data management, one of the main aims of which is achieving reproducibility of scientific results in general. In this paper, we aim to explore the potential of data stewardship for supporting software reproducibility and sustainability as well. Recently, we gathered the key stakeholders of the topic (i.e. researchers, research software engineers, and data stewards) in a workshop setting to understand the challenges and barriers, the support required to achieve software sustainability and reproducibility, and how all the three parties can efficiently work together. Based on the insights from the workshop, as well as our professional experience as data stewards, we draw conclusions on possible ways forward to achieve the important goal of software reproducibility and sustainability through coordinated efforts of the key stakeholders.

Keywords—Software Sustainability, Software Reproducibility, Data Stewardship, Research Software Engineering

I. INTRODUCTION

Scientific advances are predicated on research results that are robust and reliable. They usually serve as a solid foundation on which further advances can be built [1]. Unfortunately, there is compelling evidence that in some fields of science the majority of discoveries will not stand the test of time [1], [2].

Methodologically speaking, the two branches of scientific investigation have been deductive (for example, mathematics, philosophy, social sciences) and empirical (for example, controlled experiments and measurements in biology, chemistry, astronomy, etc.) [3]. Since the past two decades, scientific computation has become absolutely central to the scientific method [3]. Naturally, most scientists have accepted computation (simulations, model generation, etc.) as the third branch of scientific investigation [3]. Furthermore, even if the scientific method

is not computational, pervasive digitization of the research process means that data collection, storage, analysis, and the sharing of both the data and the results are often done through software tools and applications. Therefore, software and computational tools not only facilitate scientific investigation, but also help verify, extend, and understand scientific results [4].

Although software and related tools are attributed this powerful status in science, it has been argued that computation as a research method is far from being scientific “because current computational science practice doesn’t generate routinely verifiable knowledge” [3]. Identifying these serious issues, a number of computational scientists and researchers, not limited to [1], [3]–[5], have highlighted the ‘crisis’ science is facing due to lax standards in documentation, verification, and reproducibility in the development and use of software and computational tools.

In parallel, the discussion on reproducibility in science in general and the role of research data management have both gathered a critical mass to put into motion policy directives, especially in Europe, towards open data and open science. Initiatives such as data stewardship programmes have been implemented at least in some European countries, such as the Netherlands, to support researchers towards achieving reproducible and transparent science. The authors of this paper are or have been part of the data stewardship team in place at the Delft University of Technology (TU Delft) in the Netherlands since August 2017 [6]. The focus of this initiative has been mostly on supporting best practices in research data management, initially without much acknowledgement to research software. However, it quickly became clear that software sustainability and reproducibility were important topics for at least some of the faculties [7] and research institutes at TU Delft [8].

The challenge now is to understand, from the perspective of data stewardship, the level of support and resources required by researchers to adopt best practices in research software development for sustainability and reproducibility

of the code and the associated data. This is the main goal of this paper.

II. BACKGROUND

A. A brief introduction to the reproducibility crisis and the link to data stewardship

In 2016, the scientific journal *Nature* conducted a survey among 1576 researchers about the state of reproducibility in science [9]. Strikingly, more than 70% of the respondents have tried and failed to reproduce another scientist's experiments. And more than 50% have failed to reproduce their own experiments. Among the key reasons behind the reproducibility crisis, according to the survey respondents, were:

1. Selective reporting of results.
2. Pressure to publish.
3. Unavailability of related research data and software code.

Such stark revelations have at least been partially responsible to bring key shift in funding policies. Indeed, 80% of the survey respondents thought that funders should do more to improve reproducibility [9].

Funding bodies such as the European Commission and the Netherlands Organisation for Scientific Research (NWO), among others, have already implemented substantial policy changes to promote open science and the FAIR data principles. FAIR data is connected to improving reproducibility of research results and refers to a dataset being Findable, Accessible, Interoperable, and Reusable according to a set of 15 guiding principles [10]. The objectives of the open science policy of the European Commission are four-fold [11].

1. Opening access to publications and the underlying research data to improve the transparency of the research process.
2. Development of new research methods for big data management and analysis, simulations, and remote instrumentation.
3. Engagement of citizens in the scientific process.
4. Improving collaboration in research by facilitating data sharing.

Yet, "it requires effort and skills to make research open, reusable, and discoverable by others" [12]. The substantial efforts are often related and not limited to the following challenges [13]:

1. The increase of data volumes and new methods for extracting information and knowledge from data.
2. Like we just discussed, policies by research funders requiring that data should be open and FAIR over long periods of time.
3. Privacy, Intellectual Property and export regulations on data that are beyond the role of the researchers to ensure compliance.

4. The growing need to situate data and computational resources together to enable researchers to develop scientific applications.

To help researchers navigate through these important but challenging prerequisites to ensure FAIR and open science, several research institutions and universities in Europe, particularly in the Netherlands, have set up dedicated research data management support services, and in some cases hired data stewards. At TU Delft, the main objective of the data stewardship programme is to provide all the available support and resources to researchers through the entire research cycle of planning, conducting, and publishing research [6]. The main tasks of a data steward at TU Delft include:

- Analysing research data management needs.
- Helping set up secure data storage for research projects.
- Providing advice on good research data management practices.
- Conducting training sessions and workshops on data sharing and research data management.
- Reviewing and providing advice on data management plans.
- Advising on data privacy issues.
- Developing faculty-specific research data policies.

These initiatives appear promising towards supporting researchers to improve the transparency and credibility of their research. However, so far, the discussion around the reproducibility crisis, the open science initiative, and data stewardship have all largely focused on research data. Yet software is fundamental to research [14] and key for ensuring research reproducibility, and FAIR and open science.

B. The importance of software in research and the link to research transparency

"It's impossible to conduct research without software" say 7 out of 10 UK researchers surveyed in a study by the UK Software Sustainability Institute in 2014 [15]. In this study, 92% of academics stated that they used software at some point in their research and 69% said that their research would not be practical without it. The survey results are based on the responses of 417 researchers selected at random from 15 Russell Group universities, considered the UK's top higher education institutions. According to the authors of the survey, the responses, which are statistically significant in their number, represent the views of researchers in research-intensive universities in the UK [15].

These views are shared by postdoctoral researchers in the US. A March 2017 survey of the members of the US National Postdoctoral Association showed that 95% of respondents use research software [16]. Of all the respondents, 63% stated they could not do their research without research software, 31% could do it but with more effort, and 6% would not find a significant difference in their research without research software.

Counterintuitively, in today's digitized research environment, the research process tends to become more opaque with the increasing use of computational tools. Software underlying research publications is commonly not cited or mentioned; and even when it is mentioned, it is very often unavailable [14]. As explained in [17], "documenting and sharing the input/output data without the accompanying software process only makes the process translucent, not transparent". Achieving full transparency requires, according to [18], "to process and manage data and software on equal footing, policy-wise and practically". In the next section, we briefly sketch the evolution of the best practices and recommendations for software sustainability and reproducibility, both at the practical and policy level.

C. *Short overview of current best practices and recommendations for software sustainability and reproducibility*

One of the earliest directives that lead to a discussion on best practices can be traced to a 2003 report of the US National Research Council [19], focused on the life sciences. It essentially states a 'quid pro quo' reasoning for sharing related research materials, i.e., "in exchange for the credit and acknowledgement that come with publishing in a peer-reviewed journal, authors are expected to provide the information essential to their published findings".

Based on this directive, [4] developed a list of best practices for computational science that includes:

1. Version control for software and code. Some of the suggestions include the use of GitHub and BitBucket.
2. Citation standards for code, including a persistent identifier. Alternatively, a Github link could be provided.
3. Standards for reuse: To reproduce the same results, code needs to be re-used exactly as is. Therefore the terms of use and licensing should be specified.

Much of the widely known advice for research software sustainability and reproducibility is often practical and pragmatic and it focuses mostly on good code writing, version control, and documentation.

For example, highly, practical suggestions from [20] include:

1. Placing an exploratory comment at the start of the program or code that explains how the code is used.
2. Breaking the code into functions to make the functionality of the program clearer to the user.
3. Not reinventing the wheel: [20] suggests to reuse existing libraries and/or elegantly use variables, and data structures such as lists to avoid variable overload.
4. Giving functions meaningful names, to properly document their purpose in the program and also to understand their functionality.

5. Creating clear documentation on requirements and dependencies for the entire project, rather than piecemeal for every program within the project.
6. Using control statements (if/else) instead of commenting code sections to control program behaviour.
7. Providing a sample dataset that users can run to test whether the program is functional.
8. Storing/archiving code in a reputable research repository that can also assign a persistent identifier to the code.

Reference [5], reiterating [4] and [20], strongly advocates the need to write readable and reusable code, create good documentation, use a persistent identifier such as a Digital Object Identifier (DOI) to help find and cite code, and assign appropriate licenses. In addition, [5] suggests to make the code workflows and related metadata available through repositories. They also call for new procedures to evaluate scientific rigour, particularly in cases where reproducing large-scale studies is computationally expensive and time-consuming.

A recent report on research software sustainability by key European stakeholders [21] supports the need and feasibility of such a practical approach to software sustainability and reproducibility, but it raises concerns on the long-term sustainability of software with such approaches alone. The report cites a number of technical barriers, such as software decay and the fast pace of change in the field of computer science, as well as important societal or cultural barriers. The latter include the lack of: awareness of the role of software in research; identification and citing of software; understanding of licensing and ownership; clear incentives and impact; software skills; and career path for software experts. Thus, the path towards software sustainability and reproducibility stretches beyond software alone and it requires change across the entire research community, from funders, research organisations, publishers, and researchers [21], [22].

Echoing the sentiments of [21] and acknowledging the need for collaboration between different partners in the research community, in a direct response to [5], [23] argue that documenting and archiving code and data is not enough to guarantee the reproducibility of computational results. They suggest the use of software containers and open interfaces, and crucially that researchers work more closely with research software engineers (RSEs) to learn best practices in software design. This advice is presented in the context of hydrology, but it could be applied more generally.

There has been criticism regarding the use of containers for research software in that they provide bitwise reproducibility, but as black boxes, do not support reusability, in cases where a different group wants to change the experiment and setup [24]. Still, bearing these limitations in mind, the recommendations by [23] triggered our interest to investigate a holistic approach towards

software sustainability and reproducibility, as we will explain in the next section.

III. RESULTS FROM A WORKSHOP ON SOFTWARE REPRODUCIBILITY AT TU DELFT

A. State of affairs at TU Delft

In addition to analysing the best practices and related challenges for software reproducibility within the wider scientific community, we wanted to understand the state of affairs in our home institution. As part of a previous initiative by the TU Delft data stewards, a survey was sent to researchers at TU Delft to understand their generic research data management practices and requirements [25], [26]. The researchers were asked a variety of questions ranging from their data backup strategy to their level of awareness about data ownership and the FAIR data principles [26]. This survey allowed us to get a comprehensive overview of the current research data management practices at TU Delft and the overall levels of awareness in this area [25].

The full survey and the data are currently available for the faculties of Electrical Engineering, Mathematics & Computer science, Civil Engineering & Geosciences, and Aerospace Engineering [27]. The survey was sent to the research staff mailing lists, from the dean's office and faculty communications team, facilitated by the data stewards of the respective faculties. To encourage participation, two gift cards were offered per faculty through lottery as a reward. There were 426 responses from the said faculties over a period of 2 months. Among the respondents, 226 (53.3%) were PhD candidates, 60 (14.2%) were postdoctoral researchers, 61 (14.4%) were assistant professors, 20 (4.7%) were associate professors, 20 (4.7%) were full professors, and the remaining respondents comprised MSc students and support staff.

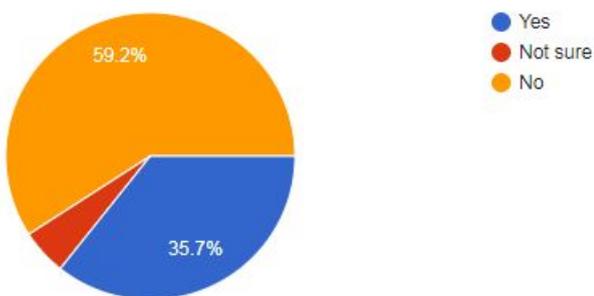


Fig 1. Responses (total of 426) on the use of dedicated tools for research data management, including electronic lab notebooks, or version control systems, as part of a general survey on research data management at the TU Delft [25], [27]. Among the 35.7% respondents who answered positively to this question, 70% said they used a git based system, such as GitHub and BitBucket for version control.

Related to research software, one of the clear questions posed to the researchers was whether they used dedicated tools for research data management, for example, an electronic lab notebook, or a version control system (such as Git). Of all the respondents, 35.7% stated that they used one or more dedicated tools for their data and/or software management (Fig. 1). Among the respondents who answered positively, 70% used a git based system, such as GitHub and BitBucket for version control.

Researchers were also asked whether they would be interested in any potential training on research data management, with training on using version control software and software carpentry being two of the options. Of all the respondents, 32% stated they were interested in training on version control and 18% on software carpentry.

We did not gather through the survey any further information on how research software is being produced, documented, maintained, published and shared at TU Delft. However, in our day-to-day experience as research data management professionals, we notice that some of the cultural barriers mentioned in [21] are valid for our institution. Although it is encouraging that many researchers at TU Delft practise version control or are interested to learn more about it, our daily practice tell us that there is still a lot of confusion and lack of awareness among researchers about software licenses, standards for citation and publication of software, and the use of persistent identifiers for software.

We also know, from a series of sandbox sessions held at TU Delft during the second half of 2017 [28], that some of the researchers who produce research software at our institution are concerned about the lack of recognition for software development. They would welcome not only training and support, particularly in the areas of documentation, archiving, and citation of code, but also more clarity at the policy level and the building of a community around open source research software.

B. Workshop rationale

Given this information, we wanted to measure the pulse of the current state of affairs in a way that captured the richness and complexity attached to software sustainability and reproducibility.

Our starting point was the paper by Hut, van de Giesen and Drost [23], where they rightfully point that the burden of achieving software sustainability and reproducibility is not limited to researchers alone. And that researchers may require support from research software engineers to ensure that the latest methods and technologies are in place to make reproducible and sustainable code. Because TU Delft is in the right direction of offering research data management support through data stewards, we wanted to also explore how this support can be tapped and extended towards research software sustainability and reproducibility.

Taking advantage of an international event on data stewardship in practice held at TU Delft in May 2018 [29], we seized the opportunity to bring the principal stakeholders (i.e. researchers, data stewards, and research software

engineers) together to discuss what support and coordination is possible from data stewards and beyond within a workshop setting [30].

C. Workshop Set-up

The workshop took place at TU Delft on 24 May 2018 [29], [30]. It attracted 17 participants, including 3 researchers, 5 research software engineers, and 5 data stewards; 4 participants either had a mix of these roles or did not identify with them. This meant that we could form groups with the ideal representation from all stakeholders of interest. There was also a very good balance among the workshop participants in terms of their research backgrounds, which ranged from various disciplines in the physical sciences and medical research to intellectual history and information science [30]. Almost all participants had experience with research software and they demonstrated some prior knowledge and interest around the topics of software sustainability and reproducibility [30].

The workshop lasted an hour. After a brief survey of the audience (using the interactive presentation software Mentimeter) and a short presentation setting the scene [30], we divided the participants into four groups, trying as much as possible that each group contained at least a data steward/research support staff person, a research software engineer, and a researcher. The groups were then given 20 minutes to answer the following questions within a collaborative google document [30]:

- What do you think about the advice of Hut, van de Giesen and Drost [23], i.e., to use containers (e.g. Docker), to use open interfaces, and to closely collaborate with Research Software Engineers to improve software reproducibility?
- Any additional advice to Hut, van de Giesen and Drost [23], to improve software reproducibility?
- How can researchers, RSEs, and data stewards work together towards implementing this advice?

Afterwards, each group was asked to pitch their key findings for a minute.

D. Insights from the workshop

Here we gather the main insights from the workshop participants, who are representatives of the most important stakeholders in the field of software reproducibility and sustainability in the context of research software. These results have been described before in [30] and are based on the content of the collaborative google document and the group pitches mentioned above.

In response to the first question, the workshop participants identified the lack of sustainable funding for hiring RSEs and the difficulty of recruiting RSEs across disciplines as the main obstacles to putting the advice of Hut, Van de Giesen and Drost [23] into practice. They also felt that open source software is not always an option (due to scientific competition or commercial interests) and that documentation is still king. In particular, documentation

should not be limited to README files: a good user manual and any information (e.g. equations, model) behind the software should also be included.

As additional advice, the participants identified the need for support in terms of software validation. In cases where professional support is not possible, the participants suggested that researchers review each others' code. Organizing code reviews in a research group could lead to improved code quality with only a small time investment.

Regarding the roles of data stewards and RSEs, opinions were split. Agreeing that researchers and RSEs should work more closely together, some of the participant groups felt that data stewards should provide the link between researchers and RSEs. One group felt that data stewards should provide the toolbox, with principles and guidance, and RSEs should help implement those principles.

The need for training (e.g. programming courses) and the option that support staff help researchers with programming was also acknowledged by the participants as part of the resources needed to implement the advice of Hut, Van de Giesen and Drost [23]. RSEs, in particular, could take a more proactive role in providing training for researchers, promoting best practices, and generally propagating their knowledge.

Lack of sustainable funding was seen as a challenge, and so was the lack of recognition for developing research software in the current academic rewards system. There needs to be a persuasive driver beyond just doing the right thing. Any driver will be most persuasive when it comes from the research community itself, but it could also come from funders, publishers, and institutions.

Universities and research institutes, in particular, should promote good practice for software engineering as part of open science. Within this domain, integrated teams working across university faculties, departments, and institutes, with a single point of contact, could provide a way for researchers, data stewards, and RSEs to work together. Fear of stepping into others' "working areas" and different working cultures may create barriers, as well as the potential lack of scientific/research expertise from RSEs and software developers.

IV. DISCUSSION & CONCLUSIONS

In this section, we try to compare and contrast the insights from the workshop to some of the best practices as well as shortcomings of implementing theory into practice that we discussed in the background section. More importantly, we also analyse and comment on the supporting role of data stewards based on our observations from the workshop as well as the reflection of the current practices and resources in the context of our home institution.

A brief recap on the motivation behind the discussions and initiatives to achieve software sustainability and reproducibility can be neatly captured in the quote of Jonathan Buckheit and David Donoho, paraphrasing Jon Claerbout [31]: "An article about computational science in a

scientific publication is not the scholarship itself, it is merely advertising of the scholarship. The actual scholarship is the complete software development environment and the complete set of instructions which generated the figures.”

So far this has been the foremost and strongest rhetoric for software reproducibility and sustainability — reproducible code environment enriched with sufficient documentation. Indeed, the importance of documentation while developing code was strongly resonated in the opinions of the participants during the workshop. They not only acknowledged its importance but also opined that all possible additional information should be added as a part of documentation.

In contrast to the literature cited in the background section, e.g. [4], the participants did not discuss version control, citation, and licensing issues, and they also did not directly focus on the practical aspects of writing good code as described in [20]. This may be because the group was generally well versed with these topics and may not have considered them an issue. However, it is quite likely that these topics were left out simply due to the time constraints of the workshop, and also because the participants were asked to focus on the additions by [23], which go beyond just providing advice for documenting, writing, and archiving code.

Crucially, the participants did acknowledge the need for training researchers. As highlighted in [23], many researchers who develop research software do not necessarily have the skills, background, and training to do so. This has been identified as a key barrier to research software sustainability [21], an issue that can certainly be extended to research software reproducibility. In the recent years, initiatives such as software carpentries and code refining have been aiming at improving not only the computational skills of researchers but also the overall quality of code developed for research purposes. However, such training initiatives not only require financial resources and institutional support, but also capable trainers to cater to researchers’ needs. Facilitating such training opportunities, at both the practical and strategic level, could be a task for data stewards.

Seeing that the workshop participants agreed with [23] that researchers and RSEs should work more closely together and thought that data stewards could provide a connection between the two, training is an obvious area of common interest for all these key stakeholders and where all partners could see tangible benefits. Through programmes such as Software Carpentry, some RSEs already provide training on software development to researchers. Data stewards could help accelerate this effort by connecting RSEs to researchers, bringing much needed expertise to the research communities they support. By sharing their knowledge and expertise in this way, RSEs could increase their impact and influence in the research community. With the help of RSEs, data stewards could also become trainers themselves. They could also identify researchers interested to become trainers. Many of these researchers will have

formal teaching responsibilities at undergraduate and doctoral level, which will get enriched by their participation in software carpentry or code refinement.

We are cautiously optimistic that such a value based approach might garner the support of researchers to attract the attention of the university executives and may have a good chance to lead to better institutional approaches to funding and recognition for sustainable research software development and training. This approach could potentially combat some of the issues that were identified by the workshop participants as barriers to research software reproducibility.

The participants also mentioned the need for support during the verification and validation of code, where RSEs could potentially be suitable consultants, particularly for large research projects. Data stewards could act as a bridge between RSEs and researchers here, too.

Another key challenge mentioned in the workshop was related to persuasive drivers and incentives towards software sustainability and reproducibility. Because of the lack of adherence and recognition of citation and publication standards for research software [14], unlike with scientific publications and datasets, it is currently very difficult to measure the impact of research software. Thus, researchers do not always see the tangible benefits of following best practices for software sustainability and reproducibility. As they already do for research data, data stewards can raise awareness of the best practices for research software management, sharing, and publication. Another interesting solution, at least in the Netherlands, is to work together with RSEs to promote awareness and use of research software directories, such as that provided by the Netherlands eScience Center.

There are many other areas, which were not explored during the workshop, where data stewards and research software engineers could collaborate for the benefit of the research community and the entire research enterprise. For example, as they do with research data management plans, data stewards could help with software management plans and with developing disciplinary protocols for software management. This may help with bringing software and data management on equal footing, at least practically, as suggested by [18].

At the policy and strategic level, data stewards could help ensure that institutional policies on research data management do not neglect research software. At TU Delft, the data stewards have been key partners in the preparation of the recently approved Research Data Framework Policy, in which research software is included as part of the definition of research data [32]. According to this policy, all researchers are expected to ensure that research data, code, and any other materials needed to reproduce research findings are appropriately documented and shared in a research data repository in accordance with the FAIR principles for at least 10 years from the end of the research project, unless there are valid reasons not to do so.

The TU Delft data stewards are now actively developing faculty-specific policies together with the deans and faculty management teams. The results of the research data management survey conducted at TU Delft [25, 26, 27] play a key role in informing the development of these policies. It would be interesting to run a dedicated survey about software management practices such as software documentation, licensing, citation, and publication. Results of such a survey in combination with the outcomes of the workshop would prove useful in fine-tuning the expected roles and responsibilities at both the University and faculty level with respect to research software management.

Recently, as a direct result of the workshop, we have been also involved in talks with representatives from The Carpentries and in establishing closer connections with the RSE community in the Netherlands. These discussions have led already to the 4TU.Centre for Research Data [8] becoming a member organisation of The Carpentries. The 4TU.Centre for Research Data is the trusted repository for the technical sciences in the Netherlands, encompassing the four technical universities in the Netherlands, including TU Delft. This is an important step towards fostering exchange of knowledge and best practices not only within the local communities at the technical universities in the Netherlands, but also at a national and international level.

While strengthening our national and international partnerships, we also reached out internally to the TU Delft ICT innovation department who offer trainings to researchers on advanced topics, such as High Performance Computing, Big Data Analytics, Machine Learning using Python etc. We agreed to schedule our individual trainings such that the advanced trainings offered by them will be preceded by functional software trainings offered by The Carpentries. We are also currently supporting a TU Delft researcher to set up a community of researchers, similar to the study group model of The Carpentries, to bring together researchers of varying level of expertise to share expertise and organically create a strong culture of good research software practices.

We hope that our inter-organizational and intra-organizational approach towards software sustainability and reproducibility will set up a good example for other institutions and universities to consider. It is also an important first step towards developing a coordinated effort of all the stakeholders within the institutional, financial, and operational structures at the levels of university, research community, and national and international funding bodies. We believe this is crucial in working towards a future where reproducibility in science, software or otherwise, becomes a norm and not the exception.

ACKNOWLEDGMENT

We are extremely grateful to all 17 participants of the workshop on software reproducibility held at TU Delft on 24 May 2018, some of whom accepted to be acknowledged here: Patrick Aerts, Kees den Heijer, Jelle de Plaa, Jordi Domingo, Martin Donnelly, Raman Ganguly, Rolf Hut,

Karsten Kryger Hansen, Carlos Martinez Ortiz, Joakim Philipson, Ronald van Haren, Martijn Staats, Michael Svendsen, and Egbert Westerhof. We are also grateful for the encouragement and feedback we received from Niels Drost and Marta Teperek on the original idea for the workshop. We also thank Carlos Martinez Ortiz, Romulo Gonçalves, Ronald van Haren, and Jason Maassen for interesting and fruitful discussions we had at the Netherlands eScience Center on data stewardship and research software engineering, part of which inspired some of the conclusions presented here. We learned about the eScience Center's Research Software Directory through this meeting.

REFERENCES

- [1] C. G. Begley and J. P. A. Ioannidis, "Reproducibility in science: improving the standard for basic and preclinical research.," *Circ. Res.*, vol. 116, no. 1, pp. 116–126, Jan. 2015.
- [2] Open Science Collaboration, "PSYCHOLOGY. Estimating the reproducibility of psychological science.," *Science*, vol. 349, no. 6251, p. aac4716, Aug. 2015.
- [3] D. L. Donoho, A. Maleki, I. U. Rahman, M. Shahram, and V. Stodden, "Reproducible Research in Computational Harmonic Analysis," *Comput. Sci. Eng.*, vol. 11, no. 1, pp. 8–18, Jan. 2009.
- [4] V. Stodden and S. Miguez, "Best Practices for Computational Science: Software Infrastructure and Environments for Reproducible and Extensible Research," *J. Open Res. Softw.*, vol. 2, no. 1, p. e21, Jul. 2014.
- [5] C. Hutton, T. Wagener, J. Freer, D. Han, C. Duffy, and B. Arheimer, "Most computational hydrology is not reproducible, so is it really science?," *Water Resour. Res.*, vol. 52, no. 10, pp. 7548–7555, Oct. 2016.
- [6] M. Teperek, M. J. Cruz, E. Verbakel, J. K. Böhmer, and A. Dunning, "Data Stewardship – addressing disciplinary data management needs," 29-Jan-2018. Open Science Framework. <https://doi.org/10.17605/OSF.IO/MJK9T>.
- [7] "EEMCS – 4 months' Data Stewardship Progress Report | Open Working." [Online]. Available: <https://openworking.wordpress.com/2017/11/28/eemcs-4-months-data-stewardship-progress-report/>. [Accessed: 08-Jul-2018].
- [8] M. J. Cruz and A. Dunning, "Research Data Management within the 4TU Research Centres," 20-Mar-2018. Open Science Framework. <https://doi.org/10.31219/OSF.IO/SGFTW>.
- [9] M. Baker, "1,500 scientists lift the lid on reproducibility.," *Nature*, vol. 533, no. 7604, pp. 452–454, May 2016.
- [10] M. D. Wilkinson, M. Dumontier, I. J. J. Aalbersberg, G. Appleton, M. Axton, A. Baak, N. Blomberg, J.-W. Boiten, L. B. da Silva Santos, P. E. Bourne, J. Bouwman, A. J. Brookes, T. Clark, M. Crosas, I. Dillo, O. Dumon, S. Edmunds, C. T. Evelo, R. Finkers, A. Gonzalez-Beltran, and B. Mons, "The FAIR Guiding Principles for scientific data management and stewardship.," *Sci. Data*, vol. 3, p.

- 160018, Mar. 2016.
- [11] “Open Science | Digital Single Market.” [Online]. Available: <https://ec-europa-eu.tudelft.idm.oclc.org/digital-single-market/en/open-science>. [Accessed: 08-Jul-2018].
- [12] “Could Open Research benefit Cambridge University researchers? | Unlocking Research.” [Online]. Available: <https://unlockingresearch-blog.lib.cam.ac.uk/?p=817>. [Accessed: 08-Jul-2018].
- [13] “Data Stewardship in the Age of Big Data.” [Online]. Available: <https://ercim-news.ercim.eu/en89/special/data-stewardship-in-the-age-of-big-data>. [Accessed: 08-Jul-2018].
- [14] J. Howison and J. Bullard, “Software in the scientific literature: Problems with seeing, finding, and using software mentioned in the biology literature,” *J. Assoc. Inf. Sci. Technol.*, vol. 67, no. 9, pp. 2137–2155, Sep. 2016.
- [15] “It’s impossible to conduct research without software, say 7 out of 10 UK researchers | Software Sustainability Institute.” [Online]. Available: <https://www.software.ac.uk/blog/2016-09-12-its-impossible-to-conduct-research-without-software-say-7-out-of-10-uk-researchers>. [Accessed: 08-Jul-2018].
- [16] U. Nangia and D. S. Katz, “Track 1 Paper: Surveying the U.S. National Postdoctoral Association Regarding Software Use and Training in Research,” 19-Jun-2017. Zenodo. <http://doi.org/10.5281/zenodo.814220>.
- [17] “Beyond Data: Reproducibility in Scientific Software and the Role of Digital Preservation - Council on Library and Information Resources.” [Online]. Available: <http://connect.clir.org/blogs/fernando-rios/2016/04/28/beyond-data>. [Accessed: 08-Jul-2018].
- [18] P. K. Doorn and P. Aerts, “A Conceptual Approach To Data Stewardship and Software Sustainability : Scientists in charge, with a little help from their friends,” p. p.1-12, 2016.
- [19] National Research Council (US) Committee on Responsibilities of Authorship in the Biological Sciences, *Sharing Publication-Related Data and Materials: Responsibilities of Authorship in the Life Sciences*. Washington (DC): National Academies Press (US), 2003.
- [20] G. Wilson, J. Bryan, K. Cranston, J. Kitzes, L. Nederbragt, and T. K. Teal, “Good enough practices in scientific computing,” *PLoS Comput. Biol.*, vol. 13, no. 6, p. e1005510, Jun. 2017.
- [21] S. Hettrick, “Research Software Sustainability: Report on a Knowledge Exchange Workshop,” *Copyright, Fair Use, Scholarly Communication, etc.*, Feb. 2016.
- [22] M. Katerbow and G. Feulner, “Recommendations on the development, use and provision of Research Software,” 16-Mar-2017. Zenodo. <http://doi.org/10.5281/zenodo.1172988>.
- [23] R. W. Hut, N. C. van de Giesen, and N. Drost, “Comment on ‘Most computational hydrology is not reproducible, so is it really science?’ by Christopher Hutton et al.: Let hydrologists learn the latest computer science by working with Research Software Engineers (RSEs) and not reinvent the waterwheel our,” *Water Resour. Res.*, vol. 53, no. 5, pp. 4524–4526, May 2017.
- [24] “Is software reproducibility possible and practical? | Software Sustainability Institute.” [Online]. Available: <https://www.software.ac.uk/blog/2017-02-20-software-reproducibility-possible-and-practical>. [Accessed: 09-Jul-2018].
- [25] “Do as you preach: results of 2017 data management survey now published | Open Working.” [Online]. Available: <https://openworking.wordpress.com/2018/02/07/do-as-you-preach-results-of-2017-data-management-survey-now-published/>. [Accessed: 09-Jul-2018].
- [26] M. Teperek, J. Krause, N. Lambeng, E. Blumer, J. van Dijck, R. Eggermont, M. van der Kruijk, K. den Heijer, H. Andrews, J. K. Böhmer, Maria J. Cruz, and Marta Busse-Wicher, “Survey: research data management practice,” Open Science Framework. OSF.IO/54EM6.
- [27] T. D. D. Stewards, EPFL Library Research Data Team, J. Krause, N. Lambeng, H. Andrews, J. Boehmer, M. Cruz, J. van Dijck, K. den Heijter, M. van der Kruijk, and M. Teperek, “Quantitative assessment of research data management practice,” Zenodo. <http://doi.org/10.5281/zenodo.1164398>.
- [28] “Open Source Software.” [Online]. Available: <https://www.tudelft.nl/library/actuele-themas/tu-delft-open-science/os/open-source-software/>. [Accessed: 10-Jul-2018].
- [29] “Event report: Towards cultural change in data management – data stewardship in practice | Open Working.” [Online]. Available: <https://openworking.wordpress.com/2018/06/12/event-report-towards-cultural-change-in-data-management-data-stewardship-in-practice/>. [Accessed: 10-Jul-2018].
- [30] M. J. Cruz, S. Kurapati, and Y. T. der Velden, “Software Reproducibility: How to put it into practice?,” 06-Jul-2018. Open Science Framework. <https://doi.org/10.31219/OSF.IO/Z48CM>.
- [31] J. B. Buckheit and D. L. Donoho, “Wavelab and reproducible research,” in *Wavelets and Statistics*, vol. 103, A. Antoniadis and G. Oppenheim, Eds. New York, NY: Springer New York, 1995, pp. 55–81.
- [32] “TU Delft Research Data Framework Policy now approved | Open Working.” [Online]. Available: <https://openworking.wordpress.com/2018/06/28/tu-delft-research-data-framework-policy-now-approved/>. [Accessed: 11-Jul-2018].