Software Reproducibility: How to put it into practice?

Authors (in alphabetical order): Maria Cruz (TU Delft), Shalini Kurapati (TU Delft), Yasemin Türkyilmaz-van der Velden (TU Delft)

With contribution from workshop participants (in alphabetical order): Patrick Aerts (Netherlands eScience Center + DANS), Kees den Heijer (TU Delft), Jelle de Plaa (SRON), Jordi Domingo (KNMI), Martin Donnelly (University of Edinburgh), Raman Ganguly (University of Vienna), Rolf Hut (TU Delft), Karsten Kryger Hansen (Aalborg University), Carlos Martinez (Netherlands eScience center), Joakim Philipson (Stockholm University), Wessel Sloof (University Medical Center Groningen), Martijn Staats (Wageningen University & Research), Michael Svendsen (Royal Danish Library), Jan van der Ploeg (University Medical Center Groningen), Ronald van Haren (Netherlands eScience Center), Egbert Westerhof (DIFFER).

On 24 May 2018, Maria Cruz, Shalini Kurapati, and Yasemin Türkyilmaz-van der Velden led a workshop titled “Software Reproducibility: How to put it into practice?”, as part of the event [Towards cultural change in data management - data stewardship in practice](#) held at TU Delft, the Netherlands. There were 17 workshop participants, including researchers, data stewards, and research software engineers. Here we describe the rationale of the workshop, what happened on the day, key discussions and insights, and suggested next steps.

Rationale for the workshop

There is no denying about a reproducibility crisis in science. In some fields, [over half of published studies fail reproducibility tests](#). A [survey of 1576 scientists conducted by Nature in 2016](#) revealed that over 90% of the respondents agreed that there was some level of crisis and over 70% said they had tried and failed to reproduce another group's experiments. Given the ubiquitousness of software in many areas of contemporary scientific research, it could be argued that there can't be reproducibility in science without reproducible software.

In a recent [Comment](#) in *Water Resources Research*, in response to “[Most computational hydrology is not reproducible, so is it really science?](#)”, Hut, Van de Giesen and Drost (2017) 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 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.

Inspired by the article and its advice, the workshop aimed to explore the various topics of software reproducibility— how some of the advice could be put in practice, and what role could institutions, data stewards, and research software engineers play in this regard.

What happened on the day

The workshop session lasted one hour. It started with the moderators introducing themselves, followed by a short survey of the audience using [Mentimeter](#), led by Yasemin Türkyilmaz-van der Velden. Maria Cruz then gave a [presentation](#) setting the scene, providing information on reproducibility, and summarising the paper and the suggestions by [Hut, Van de Giesen and Drost \(2017\)](#). One of the authors of the paper, [Rolf Hut](#), attended the session and also said a few words about his paper and his ideas. Shalini Kurapati then moderated the main activity described below.

The participants

We asked a few questions to the audience to get familiar with their background and their experiences with research software. As seen in the responses below, there was an almost perfectly balanced audience formed by researchers, research software engineers, data stewards, and people in other research support positions. There was also a very good balance in terms of the participants' research backgrounds, which ranged from various disciplines in the physical sciences and medical research to intellectual history and information science. Almost all participants had experience with research software.

What is your research field?

17

What is your experience with research software?

The majority (65%) of the participants agreed that there is a reproducibility crisis in science. The reproducibility crisis was a hot topic during the main event ([Towards cultural change in data management - data stewardship in practice](#)) and had been already discussed comprehensively earlier in the programme, by the keynote speaker [Danny Kingsley](#). Therefore, a potential bias in

the responses of the participants cannot be excluded. Regardless, it was interesting to see that there is an increasing awareness of this important issue.

Is there a reproducibility crisis?

Before moving to the [presentation about software reproducibility](#), we asked the participants what came to their mind about this topic. The answers, which ranged from sustainability, preservation, and integrity to GitHub, Zenodo, containers, and Docker, clearly show that the audience was already very familiar with the topic of software reproducibility.

What comes to your mind when we talk about software reproducibility?

The activity

Prior to the workshop, we hoped to have a group of participants with diverse backgrounds and interests. Fortunately, that turned out to be the case, and we could form groups with the ideal representation from all stakeholders of interest. We divided the participants into 4 groups, each containing at least a data steward/research support staff, a research software engineer, and a researcher. The groups were invited to answer the following questions within a [collaborative google document](#):

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

The groups were allotted 20 minutes to discuss answers to the questions and record them in the google document. The workshop moderators were able to actively monitor the google document to steer the groups towards timely conclusion of their activity. After the activity was concluded, a representative from each group pitched their activity summary and their key findings for a minute. The contents of the google documents and the pitches, which were recorded live in the [workshop slides](#), provide us the insights on the challenges and corresponding solutions for software sustainability and reproducibility, that are reported in the next section.

Key discussion points and insights on the advice by Hut, van de Giesen & Drost

Lack of funding for Research Software Engineers

Lack of (sustainable) funding for hiring RSEs is one of the obstacles to putting the advice of [Hut, Van de Giesen and Drost](#) into practice. Larger projects typically already have RSEs on board, but for smaller projects this is not always possible. It is difficult to recruit and hire RSEs across disciplines. However, the [Netherlands eScience Center](#) is a good example of a way to centrally fund research software development and to pool developer expertise across disciplines.

Open source software is not always an option

Because of scientific competition, commercial and IP interests, it is not always an option to make research software available as open source software. Dockers (containers) are also not an option for commercial software.

Documentation

High-level documentation is very important. A good README file does part of the job, but documentation and a user manual are also important. Any information (e.g. equations, model) behind the software also needs to be shared.

Software validation

Lack of support for software validation is also a problem. As an addition to the advice by [Hut, van de Giesen and Drost](#), one of the groups suggested that support should also be provided for software validation (in-house code review). In cases where professional software support is limited, it would already be helpful if researchers would review each others' code, just like they would do with papers. If the goal is to make code understandable to other researchers, then their feedback will be paramount. Organizing code reviews in a research group could improve the quality of the code significantly with only a small time investment.

The role of data stewards, RSEs and researchers

Data stewards - the link between researchers and RSEs?

Two groups saw the role of data stewards as brokers between researchers and RSEs. It was acknowledged that researchers and RSEs should interact more to improve research codes (e.g. review of codes). Data stewards could be the link between the two. Data stewards could monitor possible synergy between projects and link researchers with specialist RSE expertise. One group felt that data stewards should provide the toolbox, with principles (e.g. [FAIR principles](#)) and guidance, and RSEs should help implement those principles, because they have the knowledge to do so.

Could RSEs do more to promote best practices?

Two groups thought that RSEs could take a more proactive role in providing training for researchers, promoting best practices, and generally propagating their knowledge. Without assigning roles, one of the groups felt that implementing the advice of [Hut, Van de Giesen and Drost](#) required programming courses, support staff to help out researchers at departmental level, and the breakdown of problems into smaller problems that could be solved with up-to-date techniques based on expert knowledge. Could RSEs also help with this?

Opportunities and barriers, and the role of institutions

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.

Sustainable funding is a challenge, so is the lack of recognition for developing research software in the current academic rewards system. There also needs to be a persuasive driver beyond just doing the right thing. This can come from funders, publishers and possibly institutions. Any driver will be most persuasive when it comes from the research community itself.

Universities and institutes should promote good practice for software engineering as part of open science.

Next steps

The short-term goal of this workshop was to start a conversation on the topic of software reproducibility between researchers, research support staff (data stewards and others with a similar role), and research software engineers, and to make the results of this discussion public via this forum.

The immediate next step is to bring the results of this interaction to the attention of the community [Working towards Sustainable Software for Science: Practice and Experiences \(WSSSPE\)](#), which includes researchers and research software engineers, but lacks a strong connection with data stewards and research data support. We plan to submit a paper for the [9th International Workshop on Sustainable Software for Science: Practice and Experiences](#), to be held in Amsterdam on 29 October 2018.

The time available for the workshop was limited and not all the issues were discussed or discussed in enough depth. For example, it would be interesting to discuss in more detail what training and resources researchers and research supports need to help software reproducibility become more of a reality and what role could data stewards and research software engineers play in this regard.

Institutions could certainly do more in terms of funding and rewards for software development, and promoting best practices. How to make this happen in a global and concerted manner?

In the long-term we will continue to engage with the necessary stakeholders to keep the discussion alive and to define operational solutions towards improving software reproducibility and sustainability.

Resources

- [Workshop slides](#)
- [Water Resources Research](#) paper by Hut, van de Giesen & Drost, in response to [“Most computational hydrology is not reproducible, so is it really science?”](#)
- [Participants’ contribution through collaborative google document](#)