

# PEER REVIEWING DATA AND SOFTWARE: A PILOT PROJECT

---

Author: Saba Sharma

Contributors: Frédérique Belliard, Yan Wang

DOI : **10.5281/zenodo.11234594**

Citation: Sharma, S. (2024). PEER REVIEWING  
DATA AND SOFTWARE: A PILOT PROJECT (1.0).  
Zenodo. <https://doi.org/10.5281/zenodo.11234594>

PROJECT REPORT  
2023-24

---



Funded by the Open Science Programme TU Delft



---

## Contents

Acknowledgements .....	3
Background .....	4
Rationale and previous iterations of the project .....	4
Approach to Present Pilot Project.....	4
Summary of project activities .....	6
Peer review of datasets.....	6
Peer review of software .....	7
Challenges and Emerging Questions.....	9
Recommendations for future action.....	10
APPENDIX A: DATA PEER REVIEW GUIDELINES.....	12
APPENDIX B: CODECHECK HACKATHON FEEDBACK.....	14

---

## Acknowledgements

This project has benefitted immensely from discussions and advice from a number of stakeholders, whose existing work and experience in data, software and peer review and helped shape the project's outcomes. In addition to collaborators we worked with extensively during the project who are named in the project report, other people who have provided valuable contributions and who we would like to thank are: Marta Teperek, Connie Clare, Nicoleta Nastase and Alenka Prinčič (initial project team at TU Delft), Carlos Martinez (eScience Center), Julie Beardsell, Maurtis Kok, Raul (and the Digital Competence Center at TU Delft), Paula Martinez Lavanchy, Carlos Utrilla Guerrero (and the RDS team), Claudiu Forgaci (Faculty of Architecture, TU Delft), Madeleine de Smaele, Roel Janssen and Jan van der Heul (4TU Research Data).

---

# Background

## Rationale and previous iterations of the project

In late 2021, the Research Data Services<sup>1</sup> and TU Delft OPEN Publishing teams (both based at the TU Delft Library) discussed a potential initiative towards promoting data peer review within TU Delft. The project stemmed from identifying gaps in the research and peer review process, namely – the lack of recognition of data as a valuable research output in itself, and how to introduce incentives that could encourage both the publication and peer review of datasets by researchers. As academic outputs become increasingly diverse, with datasets and data description papers becoming common research outputs, peer review too must expand beyond the review of articles and books.<sup>2</sup>

It was also highlighted that there were clear advantages to publishing and peer reviewing datasets, namely

- Optimising datasets for reuse
- Making the methodology of data collection and publication more transparent
- Recognising data as a valid research output
- Enhancing reproducibility
- Promoting the principles of Open Science

As software increasingly becomes a common research output, the present pilot project considered both data and software within the purview of innovations in peer review.

The current iteration of the pilot project is a collaboration between the present RDS team and TU Delft Open Publishing, funded by the TU Delft Open Science programme. The pilot ran for the duration of 1 year, from April 2023-March 2024, as an Innovation project housed in the RDS team.

## Approach to present pilot project

As data and software both became part of the peer review pilot, the project took the approach to treat data peer review and software peer review as two distinct ideas, and approached each differently. This decision was taken after consultation with data and software experts (such as data stewards at TU Delft, and experts from the eScience centre in the Netherlands), as well as following the work of scholars like Katz (2016), who argue that both outputs differ significantly in that software is more dynamic, executable, and generally has a shorter lifespan than data.<sup>3</sup>

---

<sup>1</sup> Since then, this team has now been split into two – the Research Data and Software (RDS) team, and the 4TU team (a federation of four technical universities in the Netherlands).

<sup>2</sup> <https://blog.mdpi.com/2023/09/27/data-peer-review/>

<sup>3</sup> Katz DS, Niemeyer KE, Smith AM, Anderson WL, Boettiger C, Hinsén K, Hooft R, Hucka M, Lee A, Löffler F, Pollard T, Rios F. 2016. Software vs. data in the context of citation. *PeerJ Preprints* 4:e2630v1 <https://doi.org/10.7287/peerj.preprints.2630v1>

---

The project undertook an experimental approach by deciding on two ‘test’ cases for data and software peer review, with goal of understanding the complexities involved in each, the resulting questions and practical consequences in operationalising data and software peer review, and sustainable ways to take them forward.

It is worth noting that several journals already conduct some form of data or software peer review, and these processes were also guides for us in experimenting with peer review. For instance, journals such as MDPI’s *Data*, *Geoscience Data Journal*, *Scientific Data (Nature)*, *Journal of Open Source Software*, *Software-X* conduct data and software peer review of data or software papers, following guidelines specific to each journal. In the context of TU Delft, we were open to using this as one model of data/software peer review (that is, hosting the data/software peer review through a journal, as is common with other forms of peer review). However, we also wanted to experiment with other data and software peer review processes – for instance, at the level of the 4TU repository,<sup>4</sup> where datasets are submitted, and certain standardisation and quality control measures are followed. Equally, we were curious to explore whether there were other models of peer review that could help achieve our overall objectives of enhanced reproducibility and transparency.

A key principle of the approach to data/software peer review was to consider open peer review as the default in all modes of experimentation during the project. This ensures that transparency extends from the research outputs to the review process as well. This approach is particularly relevant following recent calls for more experimentation in open peer review, to explore how different forms of openness could be applied in peer review.<sup>5</sup>

Therefore, the goal was to engage communities engaged in data, software, publishing and open science, to explore what the existing best practices were (both within TU Delft and beyond), and how they could be adapted for TU Delft.

The core project team consisted of Saba Sharma (project lead and part of the RDS team) and Frédérique Belliard (senior publishing officer, TU Delft OPEN Publishing), with Yan Wang (head of RDS) in an advisory capacity. In addition, Heather Andrews Mancilla, Esther Plomp and Junzi Sun from TU Delft were closely involved during activities for data peer review, and Daniel Nuest (TU Dresden), Stephen Eglen (University of Cambridge) and Jeremy Cohen (Imperial College) were closely involved in the software peer review activities. A future iteration of the software peer review project has been launched in March 2023, with the help of an NWO Open Science grant.<sup>6</sup> The project aims to conduct codecheck workshops in four universities, based on the model trialled at TU Delft in September 2023, as part of this present pilot project. This future iteration also involves previous project participants Saba Sharma, Frédérique Belliard, Junzi Sun, Daniel Nuest, Stephen Eglen, along with Daniela Gawehns (Reproducibility Network) and Frank Ostermann of UTwente as PI.

---

<sup>4</sup> 4TU Research Data Repository <https://data.4tu.nl/>

<sup>5</sup> ‘Lack of experimentation has stalled the debate on open peer review’, *Impact of Social Science LSE Blog*, 21 March 2024, <https://blogs.lse.ac.uk/impactofsocialsciences/2024/03/21/lack-of-experimentation-has-stalled-the-debate-on-open-peer-review/>

<sup>6</sup> CHECK-NL <https://www.nwo.nl/projecten/osf232063>

---

## Summary of project activities

As outlined in the approach section of this report, the activities for peer review of datasets and software were segregated, and are detailed in separate sections. Nonetheless, a common feature of both was an attempt to build on existing practices in the field and ideas that have some traction in the scientific community. In addition, defining data and software peer review was also a work in progress, as the terms do not yet have universal understanding in the way that peer review of articles or books does. Nonetheless, defining and understanding the scope of peer review in these research outputs was also part of the pilot project's goal, and we document it to the best of our understanding here.

### Peer review of datasets

The previous iteration of the pilot project had begun work on a data peer review template, which provided broad guidance on how different parameters of a dataset (such as metadata, file formats, licenses, and so on) could be elements of review. During the course of the project, the guideline was reworked with the help of two data stewards at TU Delft. In creating the template, we referred to existing data peer review guidelines at journals that publish data papers, as well as existing practices in research teams. Such guidelines broadly cover the editorial, metadata, and methodology review of the datasets, while some also go into details about licenses, data descriptions, links to repositories, and so on.<sup>7</sup>

Based on previous work done in the project, data peer review guidelines at several open access data journals, and with inputs from two data stewards (Heather Andrews Mancilla and Esther Plomp) and two editors of the Journal of Open Aviation Science (Xavier Olive and Junzi Sun), we developed a template for data peer review (see Appendix A of this report). The template distinguishes between 'technical' and 'scientific' checks. Technical checks are concerned with the completeness and FAIRness of a dataset, for instance, whether there is a README file, an adequate description, metadata, license, and so on. These could be relevant in the context of the 4TU repository. It would be useful to tally these against the findings of another upcoming report from the Open Science programme, which looks at standardisation practices in the 4TU data repository.

The additional scientific checks look at the quality of data and methodology of the data collection and whether supporting experiments are adequately described and research questions addressed. This would require some accompanying documentation to review, such as a short data paper or journal article. Such a review would be best suited, as with other data peer review practices, at a journal where datasets and accompanying data papers (or articles) are published.

---

<sup>7</sup> Todd Carpenter, 'What constitutes peer review of data? A survey of peer review guidelines', 11 April 2017, <https://scholarlykitchen.sspnet.org/2017/04/11/what-constitutes-peer-review-research-data/>; Mayernik, M. S., Callaghan, S., Leigh, R., Tedds, J., & Worley, S. (2015). Peer Review of Datasets: When, Why, and How. *Bulletin of the American Meteorological Society*, 96(2), 191-201. <https://doi.org/10.1175/BAMS-D-13-00083.1>

---

The ideas of the project and template were presented at the Open Science Festival in Rotterdam in August 2023, with a team from the project. The session was conducted as a workshop, with participants providing feedback on the general concept of data peer review through interactive activities and sessions. Some key points of relevant feedback from the session:

- Data peer review has value has a concept: it improves the quality of datasets, makes them more reusable and fit for repurposing, can mitigate publication bias by encouraging publication of negative results, helps recognise data as a valid research output, and is a timely initiative which is already overdue in the scientific community.
- Some best practices were also suggested (or pointed out, if they existed): automated checks, explicit guidelines and standards shared in advance of data collection, using a rating system, 'Code buddies' system (in Leiden), hiring a reproducibility statistician for a research group (from a psychology research group based in the UK), open reviewing processes like sharing code and receiving comments; peer review of data as something that becomes part of a research support service, rather than an additional task for supervisors and researchers.
- Some nuances and challenges to consider: making the requirements too strict, lack of protocols for erroneous data, a lack of reviewers to undertake data peer review, a lack of incentives for reviewers, data and software are dynamic and would require a more dynamic review process, using terms like 'peer review' or 'requirements' could create an issue of trust with researchers, or as if an extra layer of evaluation is being added, traditional publishing models may not be well suited to data peer review, data peer review might be too subjective to work.

## Peer review of software

One of our key collaborations in the project was with the team at CODECHECK,<sup>8</sup> a group that promotes code reproducibility. The codecheck process is a light touch code review of computational processes underlying research outputs, and involves constant communication between reviewers and authors.<sup>9</sup> In speaking with the team, we understood codechecking to be an existing best practice in the field of code review, and decided to collaborate with the team for our project. While codechecking does not concern itself with the scientific validity of the code and whether it answers the given research question, it provides a mechanism to identify issues in code execution that authors can address.

---

<sup>8</sup> CODECHECK <https://codecheck.org.uk/>

<sup>9</sup> For more details about CODECHECK and how the process works, see: Nüst D and Eglen SJ. CODECHECK: an Open Science initiative for the independent execution of computations underlying research articles during peer review to improve reproducibility [version 2; peer review: 2 approved]. *F1000Research* 2021, **10**:253 (<https://doi.org/10.12688/f1000research.51738.2>)

---

In September 2023, TU Delft and CODECHECK organised a hackathon in TU Delft. Three codecheckers — Daniel Nüst, Stephen Eglén, and Jeremy Cohen — teamed up with participants from TU Delft and the Netherlands to share their expertise and insights on codechecking, an open and collaborative code review process. We decided to conduct this part of the project in workshop format, rather than working on guidelines, as the CODECHECK team suggested that showing and doing code review provided clearer understanding of the process. In addition, we hoped that bringing participants from TU Delft together for the workshop would help generate interest in codechecking and sow the seeds for forming a community around codecheck.

In advance of the workshop, we put out a call for papers/projects and code, inviting researchers and support staff to share code and related documentation that they were willing to have codechecked live during the session.<sup>10</sup> This helped us set up the session in advance and plan the breakout groups and other activities during the session. Simultaneously, we also invited participants to register to be codecheckers at the hackathon, and learn how to codecheck another person's work. Being able to offer 0.5 GS credits from the TU Delft Graduate School meant that we were able to have many PhD candidates join and make the event part of their doctoral educational programme. A pre-workshop survey was sent out to determine the level of programming experience among participants, which programming languages they were familiar with, and whether they could install new software on their computers during the workshop.

The first half of the workshop was dedicated to introducing the concept of codecheck to participants, with Daniel expanding on the principles of this Open Science initiative that aims to check the computational workflows in a research project, enhancing its reproducibility. A live demo followed the introductory presentation, where a project submitted from a TU Delft participant was successfully live-codechecked.

In the second half of the workshop, participants were divided into breakout groups, with each group codechecking a project previously submitted by TU Delft researchers, most of whom were also present at the event. The breakout groups helped participants get hands-on experience of what codechecking entails, and the skills you need and acquire in the process. For participants who submitted their code, codechecking provided important feedback on how they could make their code more reproducible.<sup>11</sup>

In a final reflection session that wrapped up our workshop, participants gave consideration to the concepts they learned, and whether codechecking fit into their own research workflows. Additionally, we also collectively reflected on the challenges of codechecking, including how to reward or recognise codechecking efforts, how to ensure there are enough codecheckers out there to make the practice more widespread, and how to think about making codecheck a more sustainable practice, including at TU Delft.<sup>12</sup> More details about feedback from this session are linked in Appendix B.

Through brainstorming with participants at the workshop and the codecheck team afterwards, we identified some key skills acquired through participating in the workshops. Given the skill development involved, we suggest a more regular programme of such workshops (detailed in the recommendations section).

Identified areas of skill development in codecheck workshops:

---

<sup>10</sup> The call text is available here: <https://openpublishing.tudl.tudelft.nl/codecheck-and-tu-delft-hackathon/>

<sup>11</sup> An example codecheck certificate from the workshop is available here: <https://zenodo.org/records/8359200>

<sup>12</sup> Blog post about the hackathon, also available at: <https://openpublishing.tudl.tudelft.nl/tu-delft-codecheck-hackathon-some-perspectives/>



---

### *Code reproducibility-related skills*

- Running code workflows of other researchers
- Publishing and citing code and data
- Reproducing existing scientific studies and research papers
- Troubleshooting and solving errors in executing code workflows
- Learning to write and publish a codecheck report/certificate
- Working with version control software (typically Git + Github)
- Gaining practical experience of what high-quality sustainable research projects look like

### *Other transferable skills and benefits*

- Learning to communicate with authors in the process of reviewing (peer reviewing skills)
- Application of technical programming expertise to other research projects (collaborative skills)
- Becoming part of open science communities, such as codechecking communities
- Gain an understanding of good practices for sharing code and data, which helps own work become more useful
- Getting to know new methods and tools from the reproduced workflows
- Following and understanding the reasoning of another person's methodological workflow ("project set up and organisation")

As mentioned above, four more codecheck workshops will be conducted across the Netherlands as part of an NWO-funded Open Science project. These workshops will be discipline-specific, with the first one scheduled to take place on 30<sup>th</sup> May 2024 in TU Delft for engineering sciences. With the continuation of the codecheck workshops we hope to make strides in building a community around codecheck in the Netherlands, one of the objectives we hoped to advance during the pilot project.

## Challenges and emerging questions

While the pilot project was able to experiment with some new ideas, there are also some challenges, both conceptual and logistical, to implementing these on a more sustainable scale:

- Resources: Our pilot project was staffed by one 0.5FTE position, with other contributions coming from existing time commitments or on a voluntary basis. To implement peer review in a more sustainable format, either at institutional or publication level, it would require person hours and resources to be dedicated towards it. The question of the increasing burden of peer review in the context of scholarly publications is also a current discussion, and is among the challenges we face as well.
- Rewards and recognition: In the course of the project, we discussed several ideas for recognition for reviewers, such as linking reviews to ORCIDs, or having badges/certificates for peer reviewed datasets. But fundamentally, the questions remained.
  - How do we reward work done by peer reviewers?
  - What are the incentives for researchers/authors

- Defining the terms: Data and software peer review are not established practices with a common understanding across academic contexts. Different disciplines or research groups, for instance, may have differing approaches to what counts as data peer review for the types of data used in their field, and practices may be difficult to standardize. Similarly, while CODECHECK helps improve transparency and reproducibility around code, it may not equate to peer review in every context.
- Sustainability: How can sustainable communities be cultivated in TU Delft around the ideas of data or software peer review. Our suggestions below are largely related to aligning with existing initiatives as the best possible option.

## Recommendations for future action

There are a few ideas for future steps within the context of TU Delft emerging from the pilot project. Just as the initial ideas for the project attempted to draw from existing practices, the recommendations also try to align with existing initiatives within TU Delft, and complement work already being undertaken.

There are some existing policies, guidelines, and other ongoing projects at TU Delft around the publication and review of data. These include:

- TU Delft Research Data Framework<sup>13</sup>
- TU Delft OPEN Publishing Data policy<sup>14</sup>
- The supervisor's guideline, which is guidance for PhD supervisors on the publication of data and code being developed by data stewards at TU Delft.<sup>15</sup> The supervisor's guide and the guide for data peer review in this project contain many overlaps, for instance, checking that the data is in an appropriate repository, contains a PID, has proper file structure, and explanations about data ownership. Therefore, some of the 'technical' checks about datasets can (and do) occur at the level of the research group or department/faculty.
- Review process at 4TU data repository (including an ongoing project in the Open Science team on streamlining and standardising some aspects of depositing datasets at 4TU, to ensure consistency. This report, written by two data stewards at TU Delft, is likely to also cover many of the 'technical' dataset checkpoints in the guideline, and therefore 4TU

---

<sup>13</sup> TU Delft Research Data Framework <https://zenodo.org/records/4088123>

<sup>14</sup> TU Delft OPEN Publishing Data Policy <https://www.tudelft.nl/library/actuele-themas/open-publishing/about/policies>

<sup>15</sup> Checklist Guidance for PhD Supervisors <https://filelist.tudelft.nl/Library/Themaportalen/RDM/ResearchDataArchivingChecklist-GuidanceForPhDSupervisors.pdf>, which can be found under <https://www.tudelft.nl/en/library/research-data-management/r/publish/publish-research-data/publish-your-phd-data/guidance-for-doctoral-candidates-completing-their-studies>

---

would be an ideal environment to conduct these checks (and many of these checks already occur at the level of the repository).

- An idea from the previous iteration of the pilot project was to suggest adding **badges or micro-credentials** to datasets that undergo some basic level of technical review at the repository.
- This would, however, require resources to assign more people to conduct such checks. Who would conduct the additional review, or could it be incorporated into existing dataset review processes?
- In looking at the ‘scientific’ elements of the data peer review guideline, one suggestion for the **Open Publishing team** is to consider whether some journals in TU Delft OPEN Publishing would be open to experimenting with data peer review through publishing associated datasets with journal articles, or publishing short data descriptors/papers (such as in the MDPI journal *Data*.<sup>16</sup>
  - Some possible suggestions during discussions were: *The Evolving Scholar* or one of the journals in the field of Architecture. This would bring up the issue of who would do the peer review, and whether drawing from the existing pool of reviewers is an option.
  - Another idea was to involve the *Journal of Open Aviation Sciences*, who already practice open peer review with the code and data submitted for their journal.<sup>17</sup>
- For the peer-review of software, we hope to try and build on the existing practice of codecheck, and incorporate it into existing training and open science initiatives at TU Delft. As suggested above, we hope that the identified skills and benefits to PhDs in particular could help to build a more regular programme of recurring codecheck workshops, in collaboration with the **RDS training team** and **TU Delft DCC** (who have the relevant expertise to lead such workshops) and the **Graduate School** (who could help award the right GS credits for PhDs and incorporate them into the regular schedule of graduate school courses).
- One component of the NWO project is also to develop a tool/plugin that could automate some parts of the codechecking process (although, as the founders of codecheck emphasise, the entire process cannot be automated). However, it could be something that the open access journals within **TU Delft OPEN Publishing** (for instance, the Journal of Open Aviation Sciences, who have also been collaborating with us on the pilot project) could find useful to incorporate.
- In addition to organising workshops to promote more codechecks and code reproducibility, supervisors or research teams could also consider organising ‘mini codechecks’ within their teams, which could be smaller scale and more informal. Codecheck workshops can be easily adapted to different contexts.<sup>18</sup> In a research group, for instance, a call for papers or projects would not be necessary, as members would already have projects, and could work in groups or pairs to review each other’s code.

---

<sup>16</sup> *Data* <https://www.mdpi.com/journal/data>

<sup>17</sup> *Journal of Open Aviation Science* <https://journals.open.tudelft.nl/joas>

<sup>18</sup> Examples of how code reproducibility practices can be adapted to different contexts, and related templates are available at the Reprohack website: <https://www.reprohack.org/ways-to-reprohack>

# APPENDIX A: DATA PEER REVIEW GUIDELINES

The template is aimed at a data reviewer who may be from a trusted repository, a journal, or a university/research institution. By 'dataset' we mean 'data' only (and not code). We do not assume the data is always related to an article nor that the reviewer has access to such article (except where specified).

The following suggestions are based on the distinction between raw, processed and finalized data; where:

- Raw data correspond to the data obtained directly from an instrument; a (simulation) code; the raw answers from a questionnaire; etc.
- Processed data correspond to any modification (in shape or content) made to the raw data following a given methodology for the purpose of doing research. That is: cleaned up raw data; anonymized data; etc.
- Finalized data correspond to the tabular data and images corresponding to the tables and the plots presented in an article, thesis, etc.

**'Technical' checks** of the dataset (applicable to repositories, institutions, journals)

General assessment	<ul style="list-style-type: none"> <li>• Is the dataset findable: data registered/indexed with a unique persistent identifier <ul style="list-style-type: none"> <li>▪ <i>Are there any related identifiers to the dataset (like sub datasets or whether data of other groups has been reused)</i></li> </ul> </li> <li>• Is the dataset cited in the paper (if associated with a research paper/publication)?</li> <li>• Is there a contact address listed for further queries?</li> <li>• Are all contributors appropriately listed/acknowledged?</li> </ul>
Metadata	<ul style="list-style-type: none"> <li>• Is the title of the dataset (as stated in citation metadata of the repository and the supporting documentation presented in the dataset) understandable and consistent with what the dataset contains?</li> </ul> <p>Citation metadata (of the repository where the data is being published):</p> <ul style="list-style-type: none"> <li>• Is the information of all authors correct?</li> <li>• Is the description of the dataset clear and consistent as to what the dataset contains?</li> <li>• This information should be in the description of the dataset in both, the citation metadata and the supporting documentation (README). In the citation metadata this information should be summarized, while in the supporting documentation it should be provided in more detail. Based on whether the data is raw, processed or finalised; <ul style="list-style-type: none"> <li>▪ <i>If it is raw data: is there a clear explanation of how it was obtained (such as instruments, facilities, dates, software).?</i></li> <li>▪ <i>If it is processed data: is there information about the raw data from which the processed data is derived? Is there information about the processing steps applied to the raw data (including software, methodology, caveats)?</i></li> <li>▪ <i>If it is finalized data: is there information about the raw and processed data from which the finalized data is derived?</i></li> </ul> </li> </ul>

	<ul style="list-style-type: none"> <li>• Are there any articles related to the dataset? If yes, are they included in the citation metadata of the dataset?</li> <li>• Are there any other references mentioned in the dataset that should also be included in the citation metadata? Other references can include code repositories, standard protocols, funding agencies.</li> </ul>
Files	<ul style="list-style-type: none"> <li>• Are the files in open or standard formats?</li> <li>• Is there a README file? <ul style="list-style-type: none"> <li>▪ <i>Does it contain information about how the data was collected/processed?</i></li> <li>▪ <i>Does it contain information about naming of variables, units and any extra disciplinary metadata?</i></li> <li>▪ <i>Does it contain a description of the dataset structure?</i></li> <li>▪ <i>Does it contain a description of caveats, errors and assumptions considered while collecting/processing the data?</i></li> <li>▪ <i>Are all references properly indicated?</i></li> <li>▪ <i>Is the methodology well described?</i></li> <li>▪ <i>Are all the files described?</i></li> </ul> </li> <li>• Is there extra supporting documentation (such as logs, manuals)?</li> <li>• Are there extra supporting discipline-specific metadata (such as XML files)?</li> <li>• Can the files be opened without problems? Is there sufficient information about what the files contain?</li> <li>• Do the files contain errors/missing values not explained in the supporting documentation?</li> </ul>
License	<ul style="list-style-type: none"> <li>• Is there a single license or are there files licensed under different licenses? If it is the latter, how are these specified?</li> <li>• Is the license consistent with the related data and the concept of Open Science?</li> </ul>

**‘Scientific’ checks** of the dataset (additional checks, applicable for journals or where additional documentation [like a paper/preprint] is also available)

Dataset	<ul style="list-style-type: none"> <li>• Is the dataset appropriate for answering the research questions posed in the accompanying publication? (if there is an accompanying publication)</li> <li>• Is the dataset complete? (is validation information available)</li> <li>• Is the dataset error/missing-values free?</li> </ul>
Methodology	<ul style="list-style-type: none"> <li>• Is the methodology for data collection properly described (for instance, how samples were collected or how interviews were conducted)?</li> <li>• Are the methodology and study design replicable (for instance, are all variables defined, and is there enough detail so that others could reproduce or follow the steps taken)?</li> <li>• Are the equipment and supporting experiments described?</li> </ul>

# APPENDIX B: CODECHECK HACKATHON FEEDBACK

Reflections from hackathon participants during and after the workshop session

Mentimeter

## Why bother?

18 responses

Yes. It is the only way of making sure that whatever is put out there, can be actually reused.

Great! But I would not have time to do it really...

yes but it is time consuming!

Yes, because it makes sure that our output is of use to others. This is important because many of us are tax funded.

yes, very good idea, but I'm biased! But there are lots of approaches to consider, of which this is just one approach. Stephen

Yes, it is as important as peer review

Yes. Having someone else check your paper outputs or code is really helpful.

Absolutely helpful to increase the ability of or overall enable researchers to reproduce. "Forces" you to think of how to have others check your work.

Yes: it's pragmatic. No: concerned that the code is less reusable because no attention to code quality

1 16

Mentimeter

## Why bother?

18 responses

Yes, if combined with training and agreed upon requirements

It is but how do we ensure there are enough code checkers? It's already difficult to get a peer review

Yes, because you actually recreate the workflow

Difficult because science is 'rewards the first' and not the most thorough.

Yes. In theory peer-reviewers should be doing this, but they don't. Sometimes it's time-constraints, sometimes knowledge constraints. Good to have a separate specialised thing for this.

Yes. But I think this should be a step in the process, which is taken by journals before sending the paper to reviewers.

Yes, I do. I think that peer reviewing code is very helpful, for the researcher's whose code is being reviewed, and for the community. The main issue I see is scalability: it is a big ask on the code reviewers

I think it's an excellent approach! One of my frustrations is people reinventing the wheel continuously, and this seems like an excellent solution. Also very helpful to avoid a 'reproduction crisis'.

It is a great approach, but just hard to apply it for researchers if not set an integrated part of the publishing process

1 16

## What's missing from codecheck?

19 responses

sustainable routes to codechecking at scale. How to get beyond 'pilot projects'?

Seems like code check is really for fleshed out libraries/projects. Is this of value for little projects as well? Thinking of something that goes on posters etc. Like a 3-4 file project .

"Recipe" steps that are standardised - and can mindfully be deviated from maybe

Computation and code (at least in my field), is rarely the only important thing, e.g. behavioural assumptions that go into the code/model matter as well. Is this template-able for diff. disciplines ?

The process seems great, but more structure to enable more people to get involved could be good.

Peer review is already becoming slower and slower because reviewers are overworked. Codecheck may easily follow the same path.

experts

Should you be expert of the field in which you do the codecheck...? Just wondering if this would this be beneficial or not

Should we want to check code more regularly?



## What's missing from codecheck?

19 responses

A website with a tutorial and a guideline on best-practices and/or links to other great sources for how to setup a README, common licenses practices, perhaps also a good/bad repo to look at

Are there funding agencies which take notice of projects which are codechecked? If so that would be great! If not, that might be an important missing part.

CodeCheck award maybe for the best reviewed paper ? Awards are good incentives for researchers.

As mentioned by a fellow participant: materials that help checkability (like docs for setting up that computational environment). This would require earlier interventions though

I cannot think of anything

As mentioned in your paper: more automation. Quite challenging though, prob more useful (and also mentioned in your paper) in more domain-specific CODECHECK workflows

I can reproduce everything just clicking a button. But does this mean that the results are really reproducible?

Taylor-made processes/certificates for specific journals

collaborative review (code checking)?



### What's missing from codecheck?

19 responses

GS credits



### How can we incentivise or reward codecheckers?

18 responses

receive a code review  
permanent contract  
monetary compensation  
maybe gs credits improve tu delft output  
awards award for best paper  
specific job position  
quid pro quo job policy if hard a co-authorship  
official award for cv  
gs credits acknowledgement  
acknowledgement in paper  
graduate school points



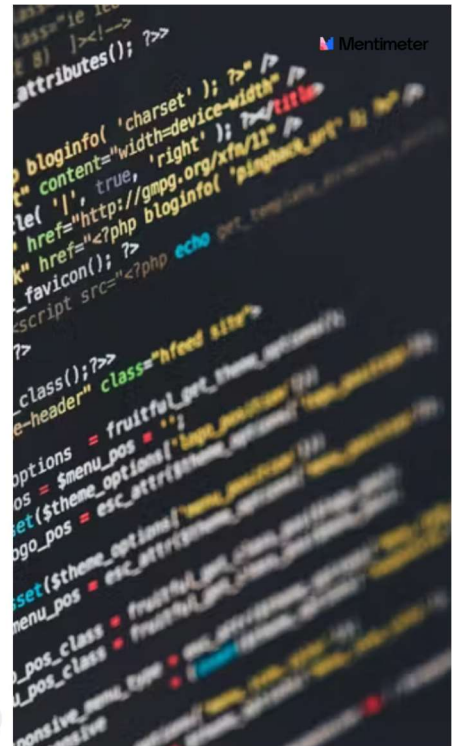


Would you be interested in attending more codecheck-related events?

  
Yes, maybe once a year

  
Yes, regular events a few times a year

0  
No, it's not relevant for me



#### *Feedback from participants after the workshop:*

“Early on, it was stated that it is hard to give generic advice for a coder that is in between the beginner and advanced level, and that makes a lot of sense to me. I am currently there and find it hard to get my hands on new tips, tricks and courses. This workshop actually hit the sweet spot, so I was impressed. I really appreciated and agreed with the sentiment that science should be a ‘show me’ world and that the code is equally, if not more, important than the paper itself is.”

“The hands-on experience of having someone else’s code and research project codechecked was incredibly beneficial. It allowed me to see firsthand the potential pitfalls and challenges in ensuring the reproducibility of my work. The constructive feedback and guidance provided by the codecheckers were instrumental in improving the quality and transparency of code and data.”