CODE CHECK ✓
https://codecheck.org.uk/

RSEng expertise for scientific peer review with CODECHECK

Daniel Nüst, Technische Universität Dresden
Stephen J. Eglen, University of Cambridge

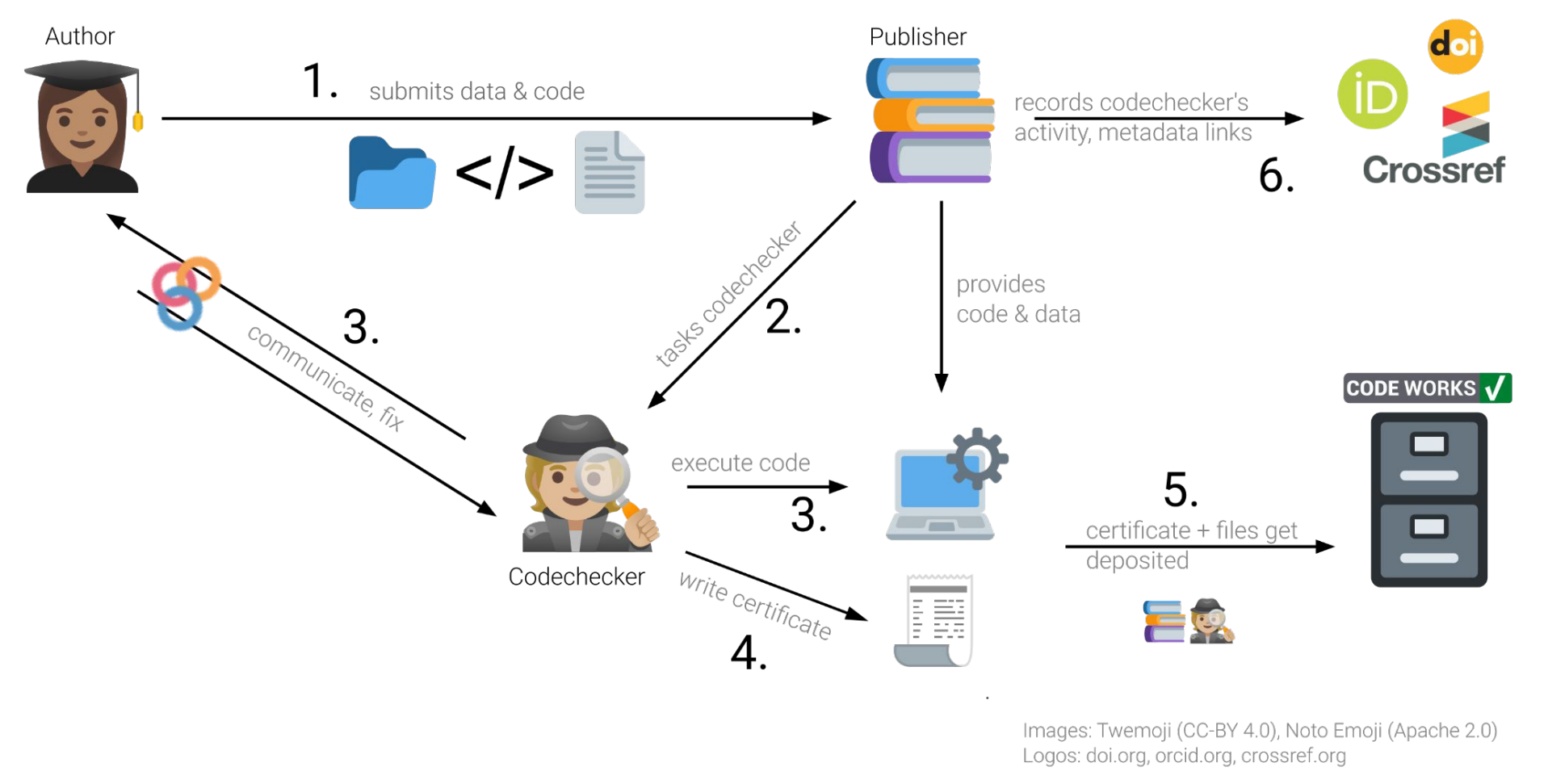
daniel.nuest@tu-dresden.de

Data and software are the foundation for a vast variety and volume of scientific research. Computational research is used in most scientific disciplines to make sense of small or large datasets using everything from one-off scripts to high-performance computing infrastructures. At some point, all these works are presented to a scientific community and in the current academic reality, the publication of a research paper is still a crucial step for the recognition of research outputs and career advancement. While research papers are increasingly accompanied by data and software to ensure transparency, reproducibility, and reusability, the inspection of these building blocks is not a common part of the publication and peer review process. The CODECHECK initiative tries to make code execution standard practice in peer review. It is based on five principles and introduces the role of codechecker. Having codecheckers can introduce required expertise and capacity into peer review if students and RSEs can get involved to improve quality, FAIRness, and openness of research outputs.

CODECHECK principles

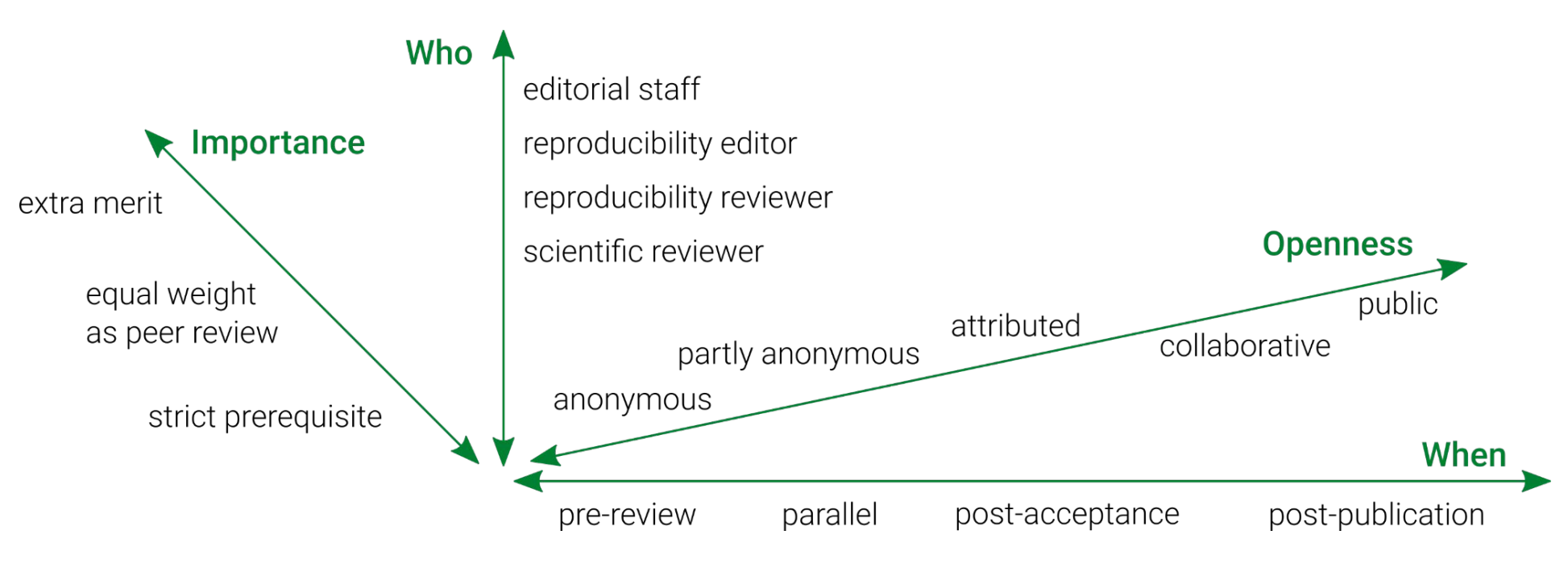
1. **Codecheckers record but don't investigate or fix.**
2. **Communication between humans is key.**
3. **Credit is given to codecheckers.**
4. **Workflows must be auditable.**
5. **Open by default and transitional by disposition.**

A CODECHECK workflow & roles


Author
1. submits data & code
Publisher
records codechecker's activity, metadata links
6.
Crossref
tasks codechecker
2.
provides code & data
3.
communicate, fix
Codechecker
execute code
3.
write certificate
4.
5.
certificate + files get deposited
CODE WORKS ✓
Images: Twemoji (CC-BY 4.0), Noto Emoji (Apache 2.0)
Logos: doi.org, orcid.org, crossref.org

Codecheckers act as detectives: They investigate and record, but do not fix issues. They check "Is it a car with wheels and a startable engine?" but not "Does it carry the advertised load at the promised apex speed without the passengers getting wet?". A CODECHECK is based on the same data and code that the author used. At the end stands a certificate of computational reproducibility.

CODECHECK variants



CODECHECKs can happen at different times and codecheckers can have diverse roles. The flexibility towards welcomes communities, disciplines, conferences, and journals at all levels of transitioning to Open Scholarship and works with preprints, fine-grained publishing of outputs, commercial tools, and HPC.

Academic peer review & contribution

… is not as set in stone as it might seem, and has not always been around. It must be questioned and evolve. Reasons for anonymity exist, but codechecking works best if codechecker and author can communicate and even collaborate. Very similar to academic peer review, **a CODECHECK certifies that one person at one point in time declares that they were able to execute a workflow or computational pipeline without significant errors.** It is not a high bar, but it is more than nothing. It ensures that "everything is there" for transparency, reusability, and future quality control.

RSEs can & must save scholarly communication so that we publish better research.

Become a CODECHECKER: codecheck.org.uk/ get-involved

One-line advice for reproducibility

Have a README:everything else is details.
Inspired by Greg Wilson's first rule of Teaching Tech Together.

CODECHECK ❤️ RSEs

You have the **knowledge** and **expertise** to spin up a computational workflow underlying a research article more quickly than less code-savvy researchers (who also need to understand the research).

Academic peer review is flooded with submissions and getting authors to put more work into the data and software publication will only work if these efforts are **acknowledged**. Your codecheck can do that.

We need **human interactions** to initiate 1 a **cultural change** towards more FAIRness and Openness of research data and research software, not just rules and regulations. Science needs you to put excellent examples in the spotlight.

Your experience with programming languages and complex workflows goes beyond what regular researchers can, and should, know; finding and **matching required expertise** for a review is a lot easier if scientific knowledge and workflow/tool knowledge can be provided by different individuals.

You are **a friendly face** in the review and authors are much more likely to improve their documentation if you ask for it than for anonymous possibly non-existing users or automated checks.

You are the one you can tell if the CODE WORKS ✓

RSEs ❤️ CODECHECK

Why should you get involved in academic peer review as a codechecker? Besides of it being fun and educational.

You can **help** people to write code that can be understood by others, can be reused, and can be improved.

It **connects** you with a journal's community, gives **recognition** by peers.

You gain experiences in academic peer **review**.

You get to know all the **package managers**. All of them.

It gives you peer reviewer **credit** and can help your **evaluation** as a researcher.

You can give back to the **community**.

It trains your **skills** to understand other people's code and ultimately helps you to improve your coding skills.

You get to know possibly useful tools and practices for your **own work**.

You **meet** possible collaborators.

Seriously, get to know package managers., But also new languages or libraries, if you like. Jumpstart your next project by **learning** the tools around the code.

You help to increase **availability** and **usefulness** of code and data in your discipline and others.

You can **reduce** people's **pain** by helping to improve "academic" codebases.

It let's you take an interesting **peek** into completely new fields of research outside of your own.

It helps to spread **good practices** in academic software use and development.

You can **contribute to Open** Science & Open Scholarship.

It **reduces** the amount of bad code in the world and fosters collaboration & reuse over **reinventing** the wheel.

You see the **latest** research.

Your feedback will be **welcome** and valued by researchers.

You can **push** publishing practices in the right direction and contribute to establishing a more open and friendly **culture** of reproducible research.

These statements are inspired by motivations shared by the awesome volunteer codechecker community: https://github.com/codecheckers/codecheckers

🤝 Private CODECHECKing

Ask a colleague to do a CODECHECK before you publish a preprint or submit a manuscript for peer review. Return the favour. Establish CODECHECKing in your lab and in your institute to make your research better.

CODECHECK is not code review

rOpenSci pyOpenSci The Journal of Open Source Software

Want to learn more? Me too!

Tell me about your experiences and opinions, please. Are you active as a reviewer? Is participation in academic peer review a part of your job? Are you interested in a metascience study on code execution in peer review? Visit https://osf.io/x32nc/.

All the details / citation

Nüst, Daniel. 2023. RSEng expertise for scientific peer review with CODECHECK. Poster at deRSE23 - Conference for Research Software Engineering in Germany, 20-22 Feb 2023, Paderborn, Germany. Zenodo. **https://doi.org/10.5281/zenodo.7642836**

Nüst, Daniel and Eglen, Stephen. 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