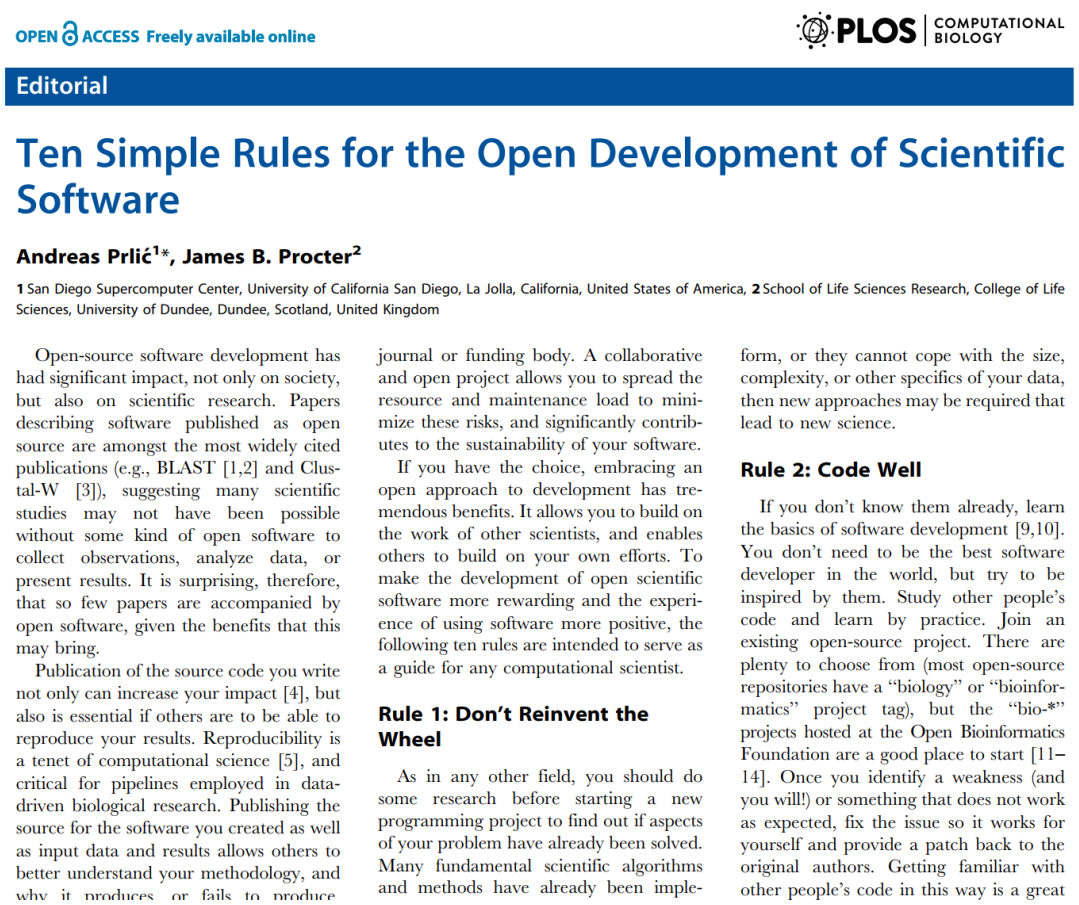
Towards Guidelines for Engineering of Energy Research Software

Stephan Ferenz[1,2], Emile Frost[1,2], Rico Schrage[1,2], Thomas Wolgast[1,2], Oliver Werth[1], Astrid Nieße[1,2]

[1] OFFIS – Institute for Information Technology, Oldenburg, Germany
[2] Carl von Ossietzky Universität Oldenburg, Oldenburg, Germany

© Carl von Ossietzky Universität Oldenburg

Guidelines for Research Software Engineering



OPEN ACCESS Freely available online

PLOS | COMPUTATIONAL BIOLOGY

Editorial

Ten Simple Rules for the Open Development of Scientific Software

Andreas Prlić[1]*, James B. Procter[2]

1 San Diego Supercomputer Center, University of California San Diego, La Jolla, California, United States of America, **2** School of Life Sciences Research, College of Life Sciences, University of Dundee, Dundee, Scotland, United Kingdom

Open-source software development has had significant impact, not only on society, but also on scientific research. Papers describing software published as open source are amongst the most widely cited publications (e.g., BLAST [1,2] and Clustal-W [3]), suggesting many scientific studies may not have been possible without some kind of open software to collect observations, analyze data, or present results. It is surprising, therefore, that so few papers are accompanied by open software, given the benefits that this may bring.

Publication of the source code you write not only can increase your impact [4], but also is essential if others are to be able to reproduce your results. Reproducibility is a tenet of computational science [5], and critical for pipelines employed in data-driven biological research. Publishing the source for the software you created as well as input data and results allows others to better understand your methodology, and why it produces, or fails to produce

journal or funding body. A collaborative and open project allows you to spread the resource and maintenance load to minimize these risks, and significantly contributes to the sustainability of your software.

If you have the choice, embracing an open approach to development has tremendous benefits. It allows you to build on the work of other scientists, and enables others to build on your own efforts. To make the development of open scientific software more rewarding and the experience of using software more positive, the following ten rules are intended to serve as a guide for any computational scientist.

Rule 1: Don't Reinvent the Wheel

As in any other field, you should do some research before starting a new programming project to find out if aspects of your problem have already been solved. Many fundamental scientific algorithms and methods have already been implemented, or they cannot cope with the size, complexity, or other specifics of your data, then new approaches may be required that lead to new science.

Rule 2: Code Well

If you don't know them already, learn the basics of software development [9,10]. You don't need to be the best software developer in the world, but try to be inspired by them. Study other people's code and learn by practice. Join an existing open-source project. There are plenty to choose from (most open-source repositories have a "biology" or "bioinformatics" project tag), but the "bio-*" projects hosted at the Open Bioinformatics Foundation are a good place to start [11–14]. Once you identify a weakness (and you will!) or something that does not work as expected, fix the issue so it works for yourself and provide a patch back to the original authors. Getting familiar with other people's code in this way is a great

- Just one of multiple papers focusing on simple rules to improve research software
- Some paper focus on general topics, e.g., open development [1]
- Some are domain-specific, e.g., bioinformatics [2]
- No energy-specific paper

[1] A. Prlić and J. B. Procter, "Ten Simple Rules for the Open Development of Scientific Software,"
[2] M. List, P. Ebert, and F. Albrecht, "Ten Simple Rules for Developing Usable Software in Computational Biology,"

Why Do We Need Guidelines for Engineering of Energy Research Software?

- Energy-related examples are more relatable for energy researchers than general papers
- Domain-specific guidelines get more attention by domain researchers than general guidelines
- Domain-specific problems can be addressed, e.g., high industry focus, interdisciplinarity, etc.

Energy Research Software

- Energy research software = software, that is
  - created during the energy research process or
  - created specially to the support the energy research process

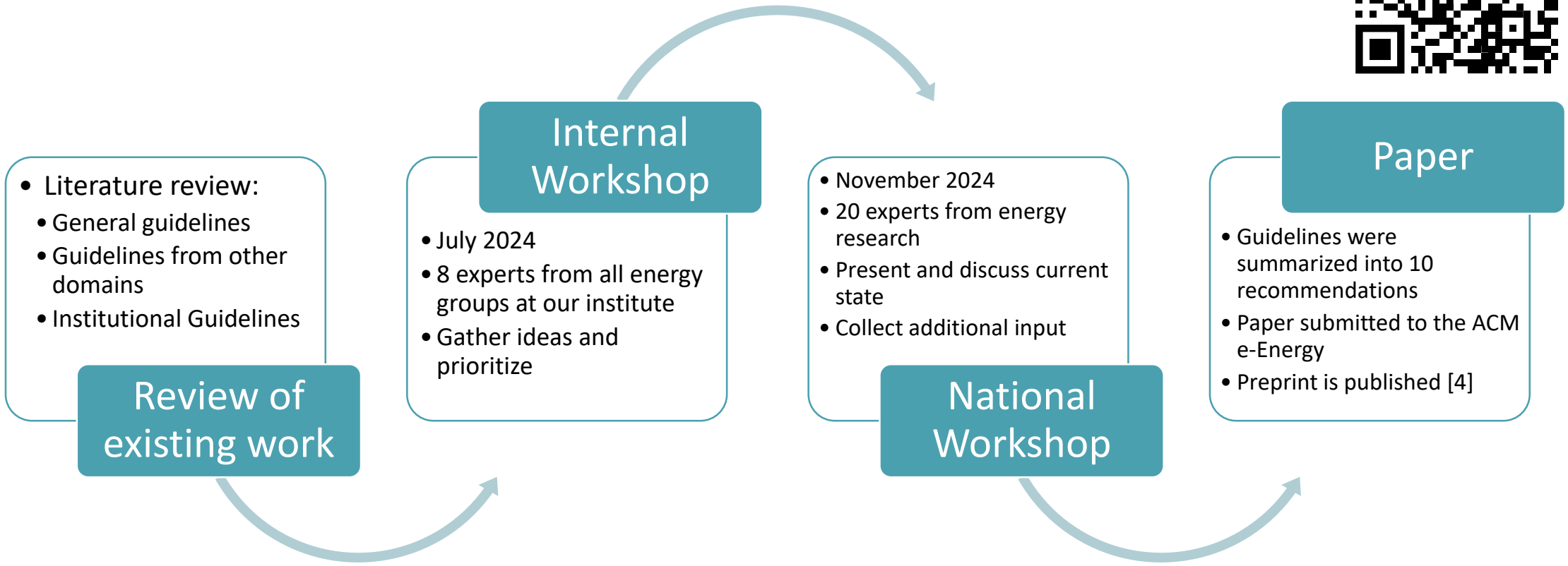
  - where the energy research process aims to understand, analyze, improve, and/or design energy systems or components specifically for energy systems.

What is special about Energy Research Software?

Interdisciplinary research	Applied research	Changing level of detail	Complexity
Data heterogeneity	Reliability	Changing time horizons	Coupled co-simulations

Towards Guidelines for Engineering of Energy Research Software

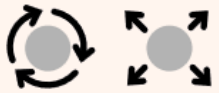
How to Develop Guidelines for Engineering of Energy Research Software?



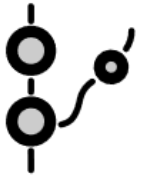
[4] S. Ferenz *et al.*, "Ten Recommendations for Engineering Research Software in Energy Research,"

10 Recommendations for Engineering Research Software in Energy Research

1. Reuse and extend other software if possible and useful.



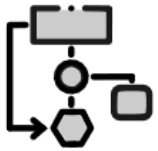
2. Use version control.



3. Develop open source and make your software findable.

4. Organize yourselfand your team.

5. Consider the reuse of your research software.



6. Keep the architecture as simple as possible (but as complex as necessary).

7. You are not done until the documentation is done.

8. Test your software based on a test strategy.

9. Grow your community.

10. Include Research Software Engineering (RSE) in your research project proposals.

How do the Recommendations Compare to Other Domains?

- Similar paper mainly exist in life science
- They do not cover:
  - Organization of software development
  - Research proposal
- They focus less on:
  - Collaboration with industry
  - Working in interdisciplinary teams

Additional Results of the Workshops

- Exchange between research software engineers is valuable
- Training course on the topics are helpful
- Good infrastructure is needed
- Research software should count as scientific contribution

Summary & Outlook

- Research Software Engineering is a rather new topic in energy research
- Recommendations on Research Software Engineering can help researchers to better develop research software
- Combining recommendations from more domains to general recommendations would be interesting

References

[1] A. Prlić and J. B. Procter, “Ten Simple Rules for the Open Development of Scientific Software,” *PLOS Computational Biology*, vol. 8, no. 12, p. e1002802, Dec. 2012, doi: 10.1371/journal.pcbi.1002802.

[2] M. List, P. Ebert, and F. Albrecht, “Ten Simple Rules for Developing Usable Software in Computational Biology,” *PLOS Computational Biology*, vol. 13, no. 1, p. e1005265, Jan. 2017, doi: 10.1371/journal.pcbi.1005265.

[3] M. Felderer, M. Goedicke, L. Grunske, W. Hasselbring, A.-L. Lamprecht, and B. Rumpe, “Investigating Research Software Engineering: Toward RSE Research,” *Commun. ACM*, vol. 68, no. 2, pp. 20–23, Jan. 2025, doi: 10.1145/3685265.

[4] S. Ferenz *et al.*, “Ten Recommendations for Engineering Research Software in Energy Research,” Feb. 19, 2025, *arXiv*: arXiv:2502.13510. doi: 10.48550/arXiv.2502.13510.