



RSE Winter of Code Seminar - Introduction

Applying good practices for research software

Robert Speck | Jülich Supercomputing Centre at Forschungszentrum Jülich GmbH | 2025

When code goes rogue...

illegal Conversion between
data types caused Arianne
5 rocket to explode 430
seconds after lift off.



Has a software bug really called decades
of brain imaging research into question?

The Guardian, 2016

Science
A Scientist's Nightmare: Software Problem Leads to
Five Retractions

illegal mixing of different
units of measurement (SI,
Imperial, and US) caused
the Mars Climate Orbiter
to be lost on entering orbit
around Mars in 1999.

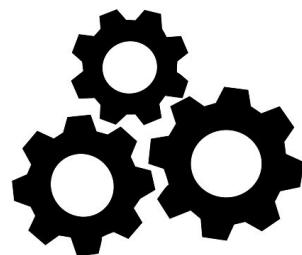
F
indable



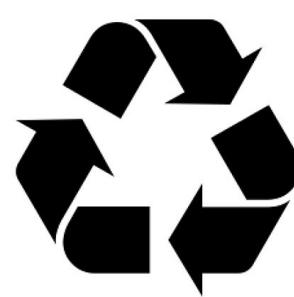
A
ccessible



I
nteroperable



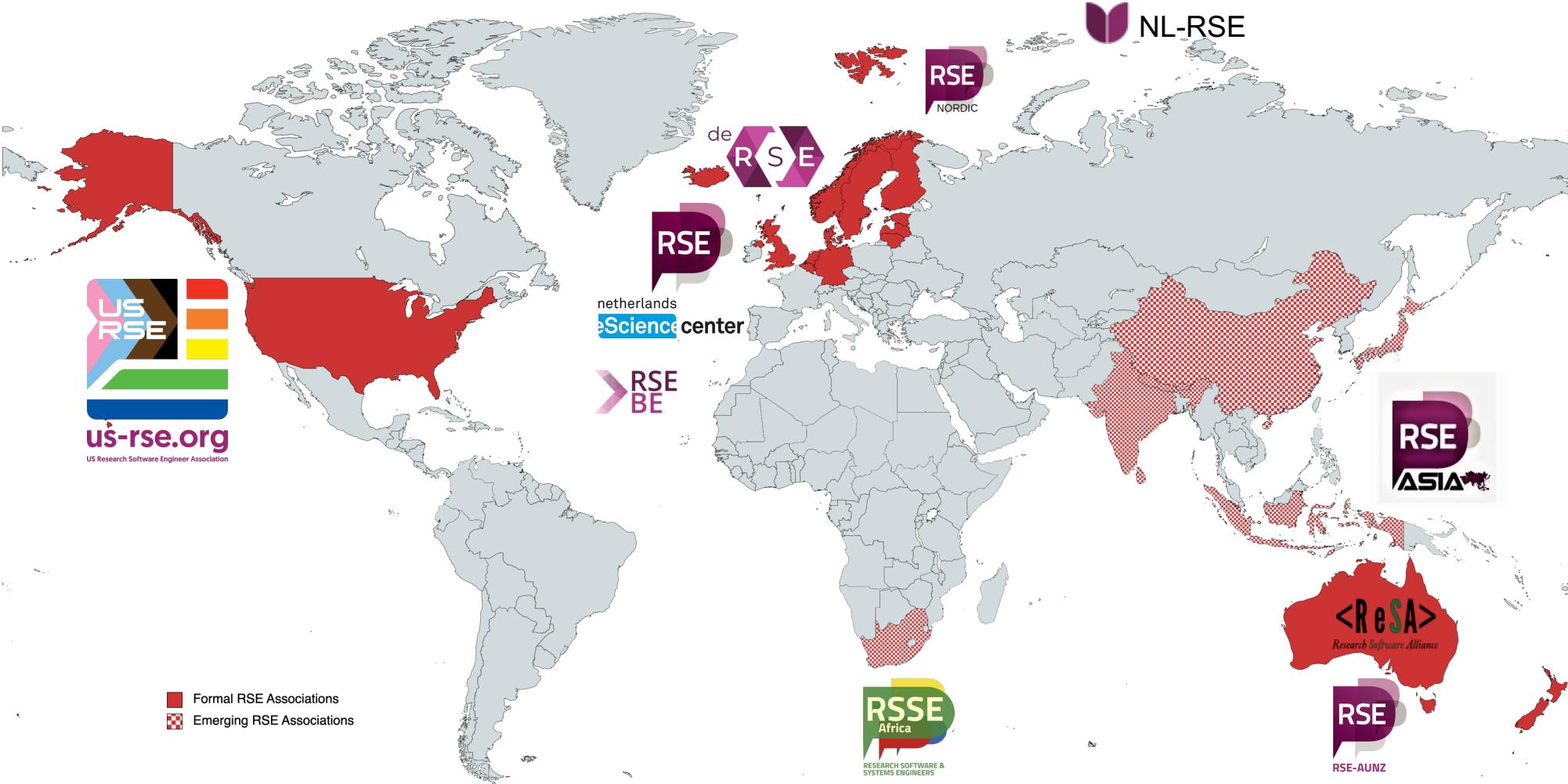
R
eusable



Part I: Introduction to RSE

Robert Speck | Jülich Supercomputing Centre at Forschungszentrum Jülich GmbH | 2025

The worldwide RSE movement



What is research software?

A strict/exclusive definition of research software

- Well identified software that is part of the research discovery process, which might require specialized domain knowledge and is by itself a contribution to science and research
- Software that was developed with the intention of being part of research

Examples of research software

- An application software, serial, parallel, GPU, CPU, big machine, small machine, ...
- A software other people use for their scientific work: a benchmarking tool, a profiler, a plotting script, ...

Examples of non-research software (in the exclusive sense)

- Software used for science, but not developed for it (Python, Excel, ...)
- Software used in the scientific workflow, but not for scientific discovery (e.g., Linux)

⚠ Did I mention Excel? Excel is NOT a research software.

Defining Research Software: a controversial discussion
Summary Report of FAIR4RS Subgroup 3 activity and discussion

Morane Gruenpeter (Inria, Software Heritage), Daniel S. Katz (University of Illinois), Anna-Lena Lamprecht (Utrecht University), Tom Honeyman (Australian Research Data Commons), Daniel Garijo (Information Sciences Institute), Alexander Struck (Cluster of Excellence Matters of Activity, de-RSE), Anna Niehues (Radboud university medical center), Paula Andrea Martinez (Research Software Alliance), Leyla Jael Castro (ZB MED Information Centre for Life Sciences), Toivo Rabemananjara (French National Research Institute for Agriculture, Food and Environment), Esther Plomp (Delft University of Technology - Faculty of Applied Sciences), Neil Chue Hong (Software Sustainability Institute / EPCC, University of Edinburgh), Carlos Martinez-Ortiz (Netherlands eScience Center), Laurens Sesink (Leiden University Libraries), Matthias Liffers (Australian Research Data Commons), Anne Claire Fouilloux (University of Oslo, Department of Geosciences), Chris Erdmann (American Geophysical Union), Silvio Peroni (University of Bologna), Paula Martinez Lavanchy (TU Delft Library), Ilian Todorov (UKRI STFC), and Manodeep Sinha (Swinburne University of Technology)

<https://zenodo.org/records/5504016>

Difference between research and industry software

Industry software

- 🏭 Result/behavior usually clearly defined
- 🏭 Should follow a **clear development plan**
- 🏭 Significant impact on budgets
- 🏭 Mostly written by **specialized software engineers**
- 🏭 Usually a clearly defined target
- 🏭 **Profit-oriented** with a target market
- 🏭 Should follow good software engineering practices

Research software

- 👷 Has usually **unknown results**
- 👷 Can evolve from a PoC to a full application
- 👷 Usually **not part of any budget**
- 👷 Rarely written by specialized software engineers
- 👷 Changing target, **changing requirements**
- 👷 Usually not profit-oriented
- 👷 Should follow **good software engineering practices**

Research software application classes

Following the FZJ software guidelines: <http://hdl.handle.net/2128/33259>

<i>Class</i>	<i>Use</i>	<i>Example</i>
0	Personal and in-house within project team	Code to a minimum extent, individual functions, simple scripts
1	At the institute	Software resulting from doctoral theses with focus on demonstration
2	Planned as result of externally funded projects, to be exploited longer term, planned as a product	Software publications, software developed and used in cooperation with partners
3	Product characteristics	Software for commercial exploitation (e.g. part of a spin-off) or developed as part of large open source project

Important aspects of RSE

The FAIR4RS way

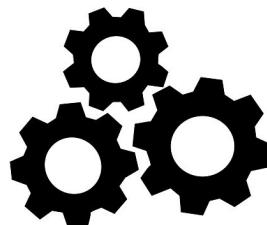
F
indable



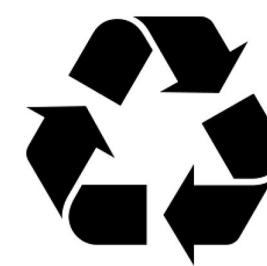
A
ccessible



I
nteroperable



R
Reusable



From:
Barker, M., Chue
Hong, N.P., Katz, D.S.
et al. **Introducing the
FAIR Principles for
research software.**
Sci Data 9, 622 (2022).
[https://doi.org/10.1038/
s41597-022-01710-x](https://doi.org/10.1038/s41597-022-01710-x)

- F:** Software, and its associated metadata, is easy for both humans and machines to find.
- A:** Software, and its metadata, is retrievable via standardized protocols.
- I:** Software interoperates with other software [...] via APIs, described through standards.
- R:** Software is both usable and reusable.

FAIR4RS: application of the FAIR principles to research software

⚠ Note: research software is NOT (only) data!

From FAIR to RSE

Making/keeping research software FAIR

We can use the **FAIR4RS principles** to derive good practices for research software:

- Documentation from: Interoperable, Reusable
- Community from: Accessible, Findable
- Versioning from: Findable, Accessible
- Automation from: Interoperable, Reusable

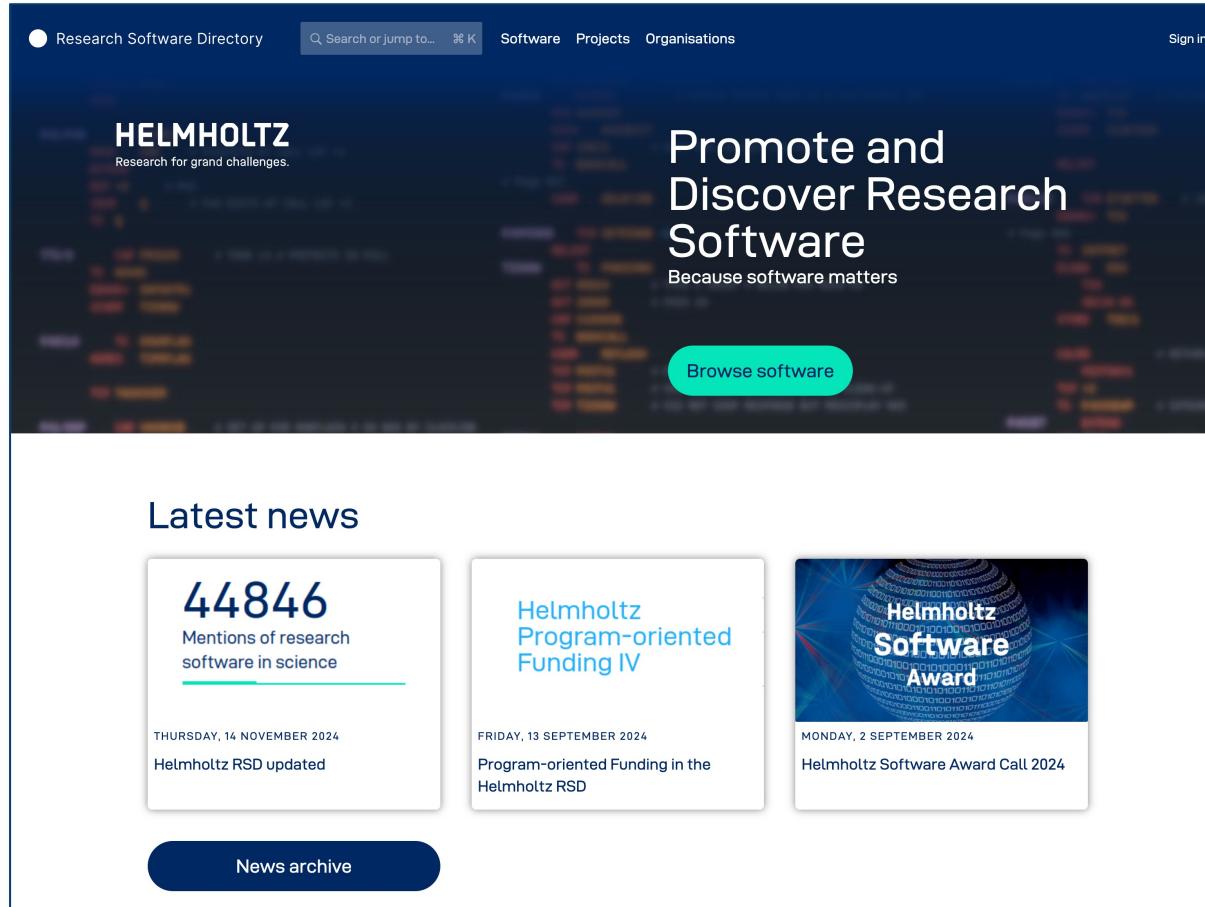
Note: There are many more approaches to “define” what RSE comprises, e.g.:

- [Best Practices for Scientific Computing](#), Wilson et al., 2014, PLOS
 - [Good Enough Practices in Scientific Computing](#), Wilson et al., 2017, PLOS
 - de-RSE position paper and resource collection on [RSE competencies](#)
 - [Four simple recommendations to encourage best practices in research software](#), Jimenez et al., 2017
- + an unknown number of other publications, websites, opinions, ...

⚠ Bottom line: there is no single standard definition or curriculum of RSE techniques!

The Helmholtz Research Software Directory

Making software findable: <https://helmholtz.software/>



The screenshot shows the homepage of the Helmholtz Research Software Directory. At the top, there's a dark header with the Helmholtz logo and a search bar. Below the header, a large banner features the text "Promote and Discover Research Software" and "Because software matters". A teal button labeled "Browse software" is centered in the banner. The main content area has a white background. On the left, a section titled "Latest news" displays three cards: one showing "44846 Mentions of research software in science" (updated on Thursday, 14 November 2024), another about "Helmholtz Program-oriented Funding IV" (due Friday, 13 September 2024), and a third about the "Helmholtz Software Award" (due Monday, 2 September 2024). A blue button at the bottom left says "News archive".

For Researchers who code

- Show the impact of their software
- Show relations to organisations, research projects and other software
- Guide visitors to codebase

For Researchers

- Discover software they need in their research field
- Get help for citing code they use

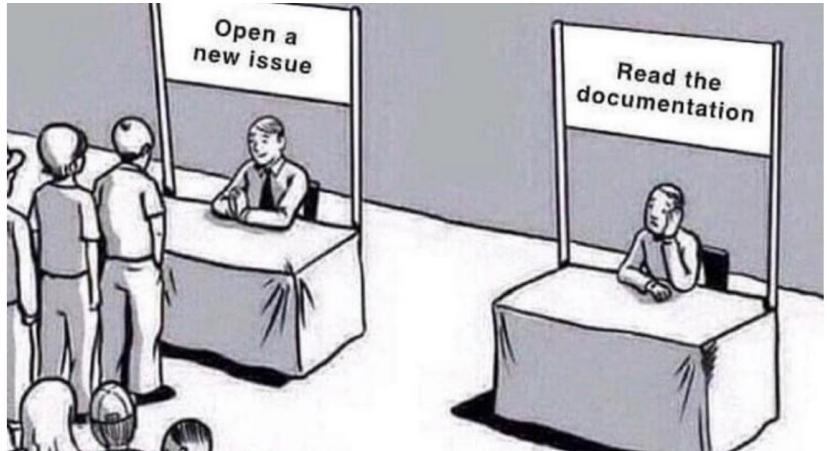
For Organisations

- Keep track of software
- Metrics and evaluation

FAIR4RS in practice: Documentation

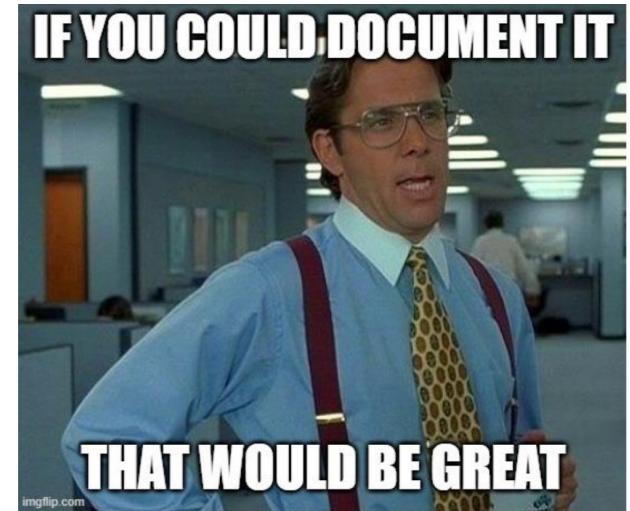
What?

- Describe what the code is doing and how to install it
- Have a license that comes with your code



Why?

- User want to understand how to work with your code
- Also, Future You will thank Present You for this



How?

- Use established tools/conventions, do not re-invent the wheel
- Try to be consistent and consequent

<https://zenodo.org/records/7260347>

FAIR4RS in practice: Community

What?

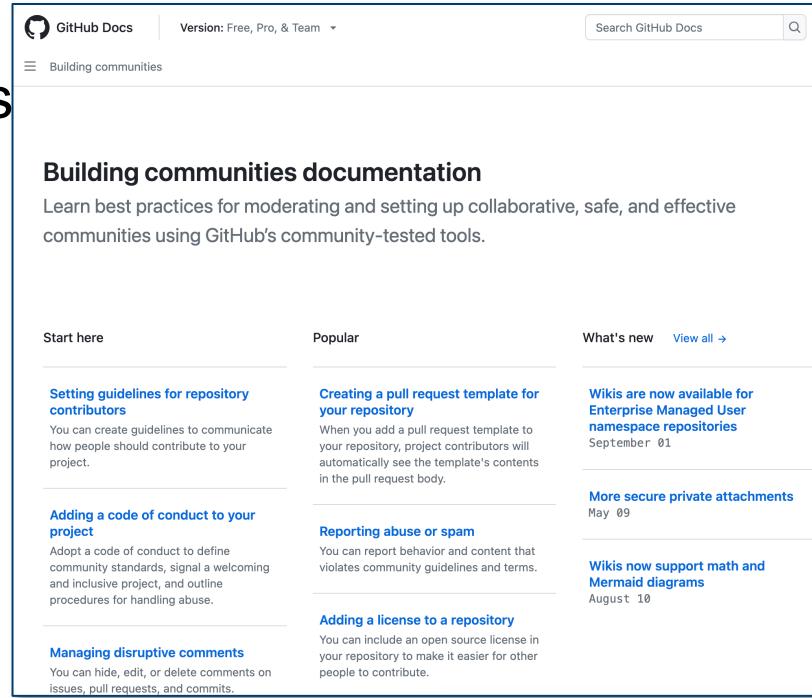
- Easy: Have issue tracker, contribution guidelines
- Pro: Run a mailing list, meet people/present at conferences

Why?

- Users should know how to get in touch and contribute
- Being open for contributions is the only way to get them

How?

- Follow established standards/guidelines (e.g., at GitHub)
- “Treat others how you want to be treated”



The screenshot shows the GitHub Docs page for "Building communities". The header includes the GitHub logo, "GitHub Docs", "Version: Free, Pro, & Team", and a search bar. Below the header, there's a sidebar with a "Building communities" section. The main content area is titled "Building communities documentation" and describes best practices for moderating and setting up collaborative, safe, and effective communities. It features several cards: "Start here" (link to Setting guidelines for repository contributors), "Popular" (link to Creating a pull request template for your repository), "What's new" (link to Wikis are now available for Enterprise Managed User namespace repositories, dated September 01), "Setting guidelines for repository contributors" (link to Adding a code of conduct to your project), "Creating a pull request template for your repository" (link to Reporting abuse or spam), "Wikis are now available for Enterprise Managed User namespace repositories" (link to More secure private attachments, dated May 09), "Adding a code of conduct to your project" (link to Wikis now support math and Mermaid diagrams, dated August 10), and "Reporting abuse or spam" (link to Managing disruptive comments). The "Wikis are now available for Enterprise Managed User namespace repositories" card also links to "More secure private attachments".

<https://docs.github.com/en/communities>

FAIR4RS in practice: Versioning

What?

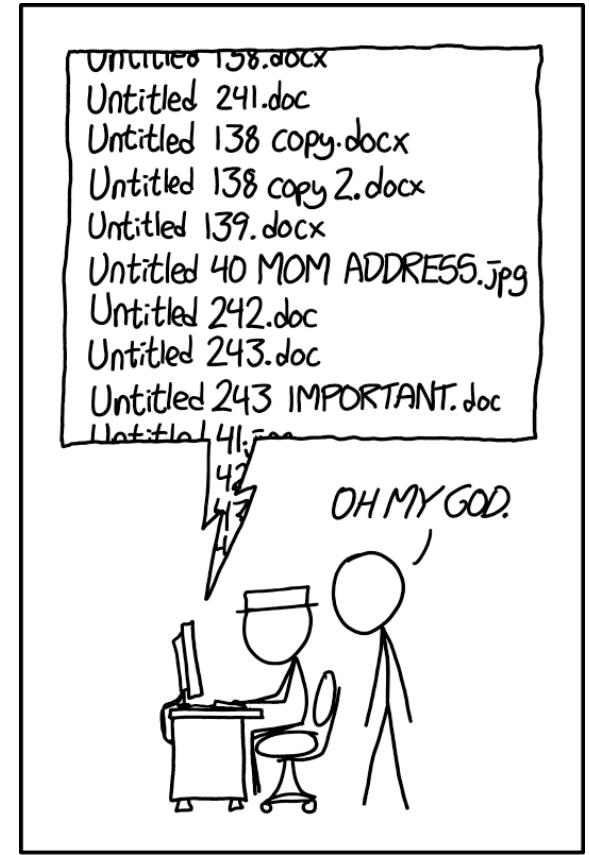
- Make sure your code is version controlled
- Each version should have a unique identifier

Why?

- Users need to know what version they work with
- Future You will thank Present You once you track changes

How?

- Use a service like gitlab or GitHub, gives you commit hashes
- Write clear commit messages and release notes



PROTIP: NEVER LOOK IN SOMEONE ELSE'S DOCUMENTS FOLDER.

<https://xkcd.com/1459/>

FAIR4RS in practice: Automation

What?

- Have tests for your code showing correctness and usability
- Have an automated way to run these tests

Why?

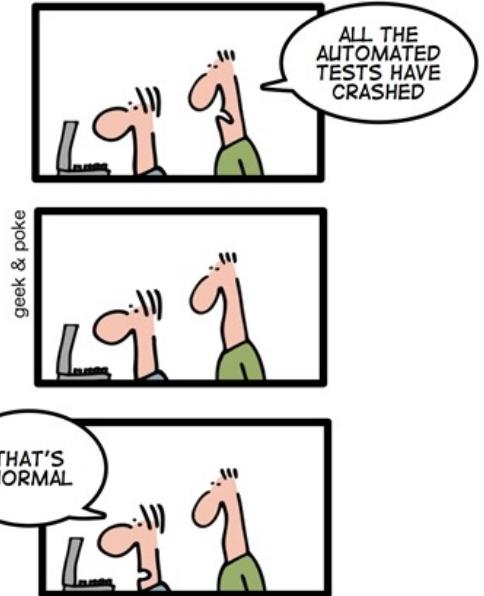
- Tests strengthen the trust in your work, for users and yourself
- Automation reduces the human factor

How?

- Use continuous integration pipelines offered e.g. by GitHub
- Follow established standards, do not re-invent the wheel

GEEK & POKE'S LIST OF
BEST PRACTICES

TODAY: CONTINUOUS INTEGRATION
GIVES YOU THE COMFORTING
FEELING TO KNOW THAT
EVERYTHING IS NORMAL



<https://geek-and-poke.com>



Technische Universität Hamburg

Part II: The RSE Seminar

Robert Speck | Jülich Supercomputing Centre at Forschungszentrum Jülich GmbH | 2025

The seminar

Goals and details

Winter (of Code) is coming

- Select from various real-life research software teams and tasks
- Work with them to see how they do research software engineering in their daily life
- Contribute to open-source research software (and be proud of it)
- Learn to talk about research software, learn to write about it

On the iron throne

- Try to wrap your contribution into one or more pull/merge requests with a clear description of your work
- Give a 20 minute talk about your contribution, the software, and the team
- Write up the story of your contribution in a ~4-page report

The seminar

The website/repository: https://github.com/pancetta/RSE_seminar_TUHH_2

The screenshot shows the GitHub repository page for 'RSE_seminar_TUHH_2'. The repository is public and has 1 branch and 0 tags. It contains several files: 'Update README.md' by pancetta (42f3fd6 · 27 days ago), '.github/ISSUE_TEMPLATE' (Update proposal-for-an-rse-challenge.md · 3 months ago), 'README.md' (Update README.md · 27 days ago), 'assignment.md' (Update assignment.md · 3 months ago), and 'timeline.md' (Create timeline.md · 3 months ago). The 'About' section indicates no description, website, or topics provided. The 'Readme' section contains the following text:

Research Software Engineering Seminar: Winter of Code

- Place: TU Hamburg & online
- Time: Winter semester 2025/2026, see [timeline](#) for more details
- Host: [Robert Speck](#)

How does research software look in practice? How do researchers write and maintain their codes? What obstacles do they face? These questions will be addressed in this seminar.

In this "Winter of Code", participating students will get hands-on experience with cutting-edge research software, giving them a unique glimpse of how research software is written, used, maintained, and extended in practice. Supervised by researchers from the Helmholtz Association, the students will be working on the researcher's software, a particular application area, and a [seminar task](#) from the field of [research software engineering](#). This could range from adding a new feature or repairing a long-standing issue to establishing a new continuous integration workflow, improving test coverage or usability.

⚠ Important: If you consider taking this seminar, please make sure you find at least one suitable assignment in the [list of issues](#) before signing up! ⚠

Next steps

Workflow:

1. Pick a task from the list of issues (done)
2. Await assignment as Github Issue in the seminar's repository
3. Await connecting email to the code owners (today!)
4. Code owners will get in touch on discuss the work with you
5. Work with them on the task, meet with them regularly
6. Attend mandatory milestone meetings (Dec and Jan, more information soon)
7. Think about your presentation
8. Finish task, create presentation, think about the short report
9. Hold presentation (~20 + 10 mins), finish short report (~4 pages)
10. Profit!

More resources and references

- HiRSE’s “Things to check out”: <https://www.helmholtz-hirse.de/resources.html>
- HiRSE Seminar Series YouTube Channel:
https://www.youtube.com/@Helmholtz_Platform_for_RSE
- HiRSE Zenodo community (slides): <https://zenodo.org/communities/hirse>
- Barker, M., Chue Hong, N.P., Katz, D.S. et al. **Introducing the FAIR Principles for research software**. Sci Data 9, 622 (2022). <https://doi.org/10.1038/s41597-022-01710-x>
- Helmholtz Research Software Directory: <https://helmholtz.software>
- The seminar on GitHub: https://github.com/pancetta/RSE_seminar_TUHH_2