HELMHOLTZ ZENTRUM
DRESDEN ROSSENDORF

Workshop: Research Software Publication

Foundations of Research Software Publication + Software Publication with HERMES

David Pape (HZDR), Sophie Kernchen (DLR)

Department of Information Services and Computing · Computational Science · David Pape · d.pape@hzdr.de · https://www.hzdr.de/fwcc-d

DRESDEN concept

About the Material

“Foundations of Research Software Publication”

The first half of this workshop is based on the HIFIS workshop:

„Foundations of Research Software Publication“

- Extremely shortened version; full course spans two mornings (8h)
- Upcoming event with free spots in mid November: https://events.hifis.net/event/1345/
- More HIFIS courses available in the HIDA Course Catalog: https://www.helmholtz-hida.de/course-catalog
  - Git and GitLab, Continuous Integration, Python, Data Processing, …

Why Publish Your Software?

Software is a Research Product in Its Own Right

- Creating and maintaining research software is academic work
  - Requires domain knowledge and research
  - Should be credited as such
- Software should be cited
  - Important part of the provenance of your research outcomes
  - Enables reproducibility of the results
- Guidelines for citing research software: „Research software citation for researchers“

HELMHOLTZ ZENTRUM
DRESDEN ROSSENDORF

Put Your Code Under Version Control

To Share and Collaborate on the Code

DRESDEN
concept

Put Your Code Under Version Control

Using Git and the Helmholtz Codebase GitLab Instance

Git allows you to:

- Store snapshots of your software
- Navigate between snapshots
- Incrementally improve your software
- Many more things

GitLab allows you to:

- Backup your code
- Share your code within your team
- Use builtin collaboration workflows (issues, merge requests, comments, code review)
- Utilize CI/CD infrastructure (automated testing, builds, deployment)

https://codebase.helmholtz.cloud

Put Your Code Under Version Control

What to Store in a Git Repository?

- Everything required to run your project
- Application/library code, configuration files, test cases, examples
- Project management and build system files
  - `pyproject.toml`, `pom.xml`, `Cargo.toml`, `CMakeLists.txt`, `Makefile`, …
- (Essential) documentation files
  - `README`, `LICENSE`, `CHANGELOG`, `CITATION`
  - User/contributor/maintainer/administrator docs
  - `CONTRIBUTING`, `CODE_OF_CONDUCT`
- References to literature (but not PDFs of the papers!)
- No build artifacts (object files, executables, …)

HELMHOLTZ ZENTRUM
DRESDEN ROSSENDORF

Make Sure Your Code Is in a Shareable State

DRESDEN
concept

Make Sure Your Code Is in a Shareable State

General Tips

- Explicitly declared dependencies and requirements
- Runnable without access to internal infrastructure
  - No hard-coded paths, names, credentials
- Don’t share secrets or internals
  - Usernames, passwords, keys, IP addresses, …
- Follow good practices of your community / programming language / domain
- Use automated code checks and testing
  - Static code analysis / linters / code checkers help find poor code and enforce rules
  - Automated tests prevent broken code to be merged into mainline branch

HELMHOLTZ ZENTRUM
DRESDEN ROSSENDORF

Add a License

IMPORTANT:

I am not a lawyer and this is not legal advice!

If in doubt, ask your supervisor and consult the HZDR legal department!

DRESDEN concept

Add a License

Software Is Subject to Copyright

- Your employer is the copyright holder and has the economic rights to the software you write at work
  (§ 69b UrhG, German copyright law, „Verwertungsrechte“)
- You, as the author, retain rights such as attribution
  (§§ 12 – 14 UrhG, German copyright law, „Urheberpersönlichkeitsrechte“)
  - Authorship can not be waived or transferred (no „public domain“)

Without a license, third parties don’t have the right to (re-)use your software!

- How to handle licensing of your software?

Add a License

HZDR Software Policy (Regulation B230)

DOI 10.14278/rodare.2748

HZDR Regulation B230 controls development, documentation, and **transfer** (including licensing) of software developed at HZDR.

- Publication under open source licenses is encouraged
  - Guidance for choosing a suitable license is included (Annex A2)
- Proprietary licenses or dual licensing may be viable options
  - Consider which transfer route will have the best impact on society and economy
- Ultimately, the institute decides
- **Reminder:** All publications at HZDR must be approved in Robis **before** going public!
- Additional info: https://www.hzdr.de/research-data, https://www.hzdr.de/software-development

Add a License

Practical Task

Examine HZDR Software Policy Annex A2.

DOI 10.14278/rodare.2748

- Which open source licenses are recommended by HZDR?
- How can these licenses broadly be categorized?
- Which license fits your software best?
- Bonus: Find the license text and add it as a `LICENSE` file to your project directory!

You have until XX:XX!

HELMHOLTZ ZENTRUM
DRESDEN ROSSENDORF

Make Your Code Citable

DRESDEN
concept

Make Your Code Citable

Providing Citation Metadata

- Metadata files for software metadata
  - `codemeta.json`: Software metadata including authors, version, release date, …
  - `CITATION.cff`: Human/machine-readable citation metadata
  - Read by Rodare/Zenodo/HERMES/…
- Manual management of metadata on publication repository (Rodare, Zenodo, …, Robis)
- Consider different roles for authorship
  - Programmers, technical writers, software architects, UI designers, …
  - Contributors (issue reporters, typo fixers, „evangelists“) are not authors

Make Your Code Citable

Practical Task

Visit the cffinit website at https://citation-file-format.github.io/cff-initializer-javascript

- Create a basic `CITATION.cff` for your project
  - For the sake of time: Completeness not required
- Download the file and add it to your project directory

You have until XX:XX!

HELMHOLTZ ZENTRUM
DRESDEN ROSSENDORF

Release Your Code

→ Research Software Publication with HERMES

DRESDEN
concept