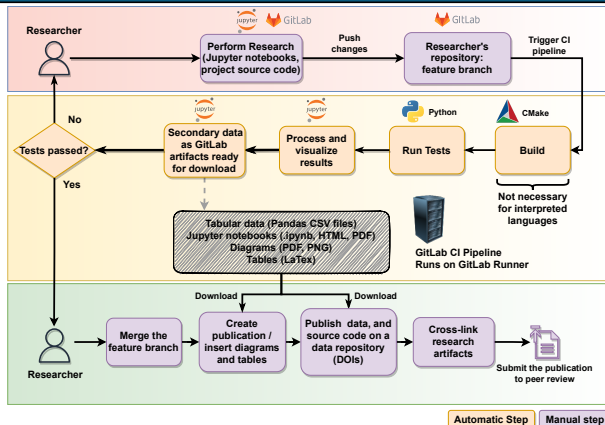


A Research Software Engineering Workflow for Computational Science and Engineering



TECHNISCHE
UNIVERSITÄT
DARMSTADT

deRSE23 - Conference for Research Software Engineering in Germany



NFDI4ing

Computational Science and Engineering software in university research groups

Boundary and initial conditions



TECHNISCHE
UNIVERSITÄT
DARMSTADT

- Publish or perish 🎓¹ prioritizes publications over scientific software.
- Dedicated resources for increasing software quality are usually not available.
- Ph.D. students rotate every 3-5 years, postdocs every 1-2 years.
 - ▣ Little or no overlap between successors and predecessors.
- Large-scale software design is not a mandatory part of the CSE curriculum.
 - ▣ Different CSE background: (Applied) Mathematics, Mechanical Engineering, Physics, Informatics.

¹Symbol of a publish-or-perish simplification of the workflow :)

Computational Science and Engineering software in university research groups

The chaos scientific legacy code



TECHNISCHE
UNIVERSITÄT
DARMSTADT

BETTY



engineering
research software

Betty is a CSE researcher, working with a legacy research code.
Why is Betty so (rightfully) angry?

- Betty inherited a **research software that is only partially tested**.
- Betty inherited a **research software that isn't automatically tested**.
 - Betty changes one part of the code and gets her model running, only to see 10 other things fail, after days of manually running tests.
- Betty's software has **no documentation of the scientific workflow**.
 - Betty doesn't know how to use existing scripts to run simulations and analyze (reproduce) results.
- Betty's software has **disjoint (diverging) versions** - that she can't integrate.
- **Betty can't even find code versions used to generate results in the publications from her research group.**

Computational Science and Engineering software in university research groups

Continuous integration and cross-linking to the rescue



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Automated testing (verification and validation), **version control**, and **cross-linking** reports, source code and research data increase Findability, Accessibility and Reproducibility (**FAIR**) and **speed up research**.

- **Continuous Integration (CI) = automatic testing + version control.**
- CSE research requires **scientific workflows**: initialize simulations, run parameter variations, agglomerate data, visualize, and check results.
- CI can be used to **automate and document scientific workflows**.
- CI ensures that the **integration of new changes does not break existing functionality**.
- Once the changes are integrated, the publication, the source code and the data are published on pre-print and data repositories and **cross-linked** using git tags and DOIs.

A Research Software Engineering Workflow

Collecting the components: Automation I



TECHNISCHE
UNIVERSITÄT
DARMSTADT

while Results are unsatisfactory **do**

Work on algorithms.

(Compile the code.)

for All studies **do**

Prepare the study.

Run the study.

Analyze results.

Move results to a report.

end for

Compare old and new results.

end while

A Research Software Engineering Workflow

Collecting the components: Automation I



TECHNISCHE
UNIVERSITÄT
DARMSTADT

while Results are unsatisfactory **do**

Work on algorithms.

(Compile the code.)

for All studies **do**

Prepare the study.

Run the study.

Analyze results.

Move results to a report.

end for

Compare old and new results.

end while

while Results are unsatisfactory **do**

Work on algorithms.

(Compile the code.)

Run initialization scripts (jobs).

Run simulation scripts (jobs).

(Run postprocessing scripts (jobs)).

Visualize results live in Jupyter notebooks.

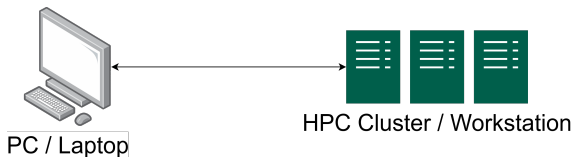
end while

A Research Software Engineering Workflow

Collecting the components: Automation II



TECHNISCHE
UNIVERSITÄT
DARMSTADT



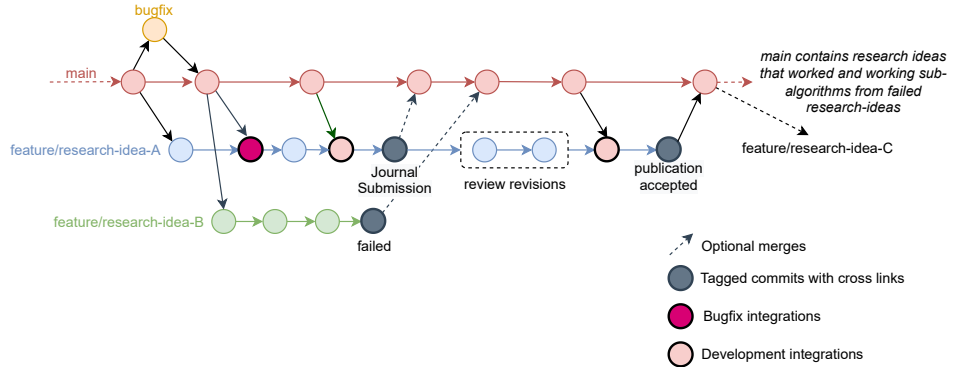
- **Manual testing takes a lot of time.**
- **Manual testing** of all previous tests **is prone to error** - even if V&V scripts do not require metadata.
- **Only the researcher knows the details** behind the initialization, running and post-processing scripts
- Relevant **V&V tests are automated using Continuous Integration (CI)**.
 - ▣ The remote repository starts the so-called **CI** test pipeline (a sequence of tests).
 - ▣ All tests are automatically run, processed and visualized.

A Research Software Engineering Workflow

Collecting the components: Version Control



TECHNISCHE
UNIVERSITÄT
DARMSTADT



The branching model increases reproducibility.



- A **text (YAML) file** is added to a repository, that **specifies the tests** (jobs) in a CI pipeline.
- When the YAML file is pushed to an upstream git repository (GitLab), **GitLab creates a CI pipeline from the YAML file**.
- The CI pipeline needs a **machine for running tests - the GitLab runner**.
 - ▢ Shared runners on gitlab.com have limited capacity.
 - ▢ We can install and register our own GitLab runner.
- A **Docker image encapsulates the computing environment**.
 - ▢ **Virtualization/Containerisation** increases reproducibility and simplifies testing.
- The Docker image must be publicly accessible for it to be used by a shared runner.

A Research Software Engineering Workflow

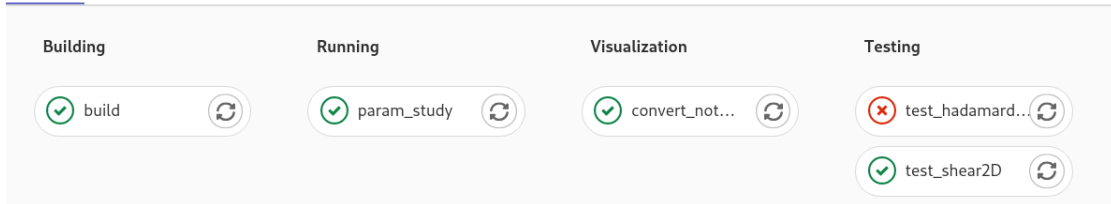
Collecting the components: Continuous Integration II



TECHNISCHE
UNIVERSITÄT
DARMSTADT

An example CI pipeline

Pipeline Needs Jobs 5 Failed Jobs 1 Tests 0

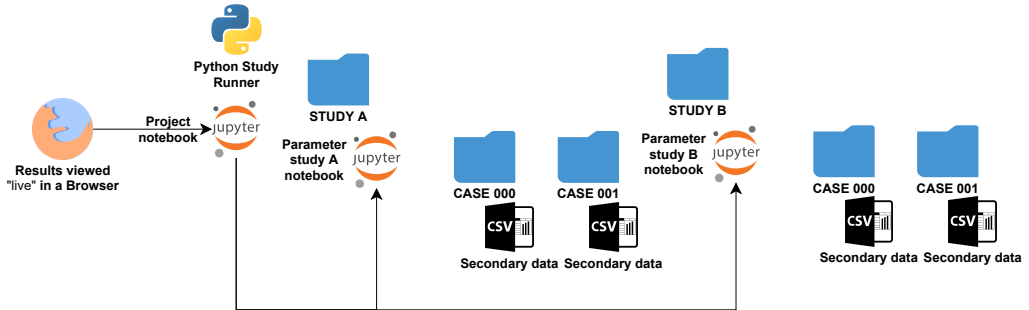


A Research Software Engineering Workflow

Collecting the components: Parameter studies I



TECHNISCHE
UNIVERSITÄT
DARMSTADT



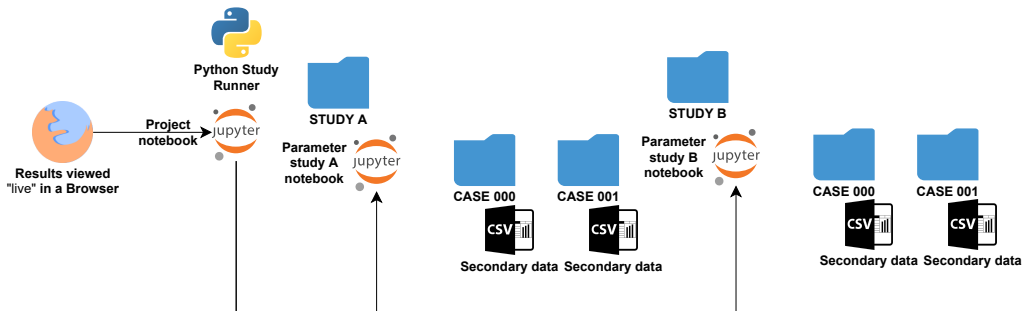
Organize your simulation studies.

A Research Software Engineering Workflow

Collecting the components: Parameter studies I



TECHNISCHE
UNIVERSITÄT
DARMSTADT



- Associate simulation cases with their metadata.
- `{case000 : {N_CELLS: 32, MODEL : shear2D}}`
- Store this information using a standard open-source format (**Interoperability** in **FAIR**).



Use Jupyter notebooks² and pandas³ for

- **Documentation:** geometry, initial and boundary conditions, error norms, comparison data.
- **Data processing:** verification errors (conservation, convergence, stability), validation errors
- **Result analysis:** interactive and remote, while simulations are running!

²<https://jupyter.org/>

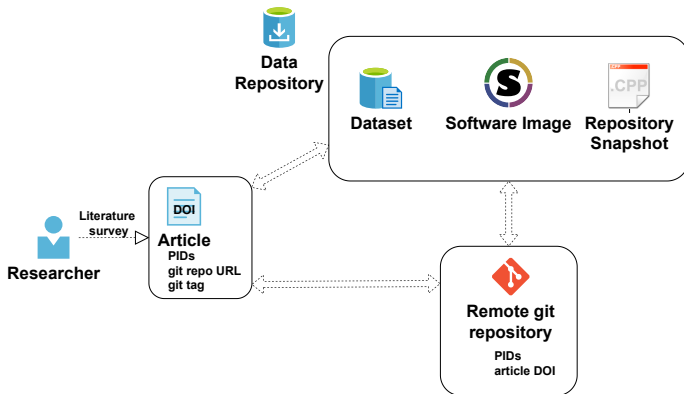
³<https://pandas.pydata.org/>

A Research Software Engineering Workflow

Collecting the components: Cross-linking



TECHNISCHE
UNIVERSITÄT
DARMSTADT





We did not cover in this talk:

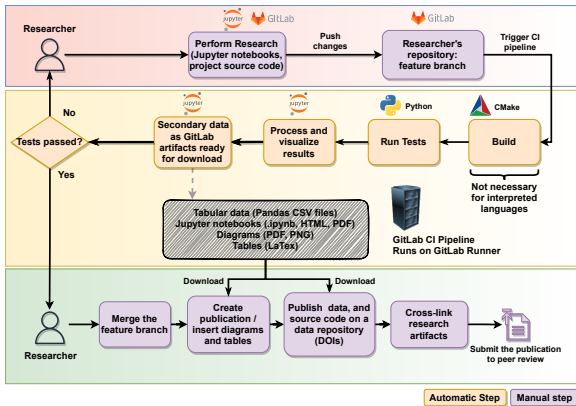
- Build System and Containerization
- Test Driven Development (TDD)
- Continuous Benchmarking
- Regression Tests
- Parameter space exploration
- Metadata standard for secondary data
- Issue Tracking

A Research Software Engineering Workflow

Assembling the Workflow



TECHNISCHE
UNIVERSITÄT
DARMSTADT

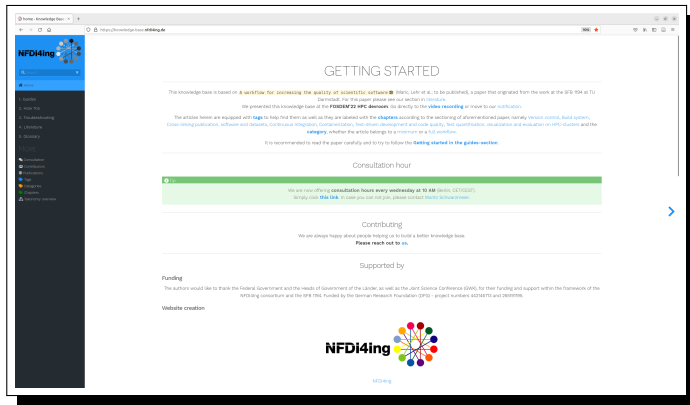


- 1: Track changes using version-control.
- 2: **while** Milestone not reached **do**
- 3: **for** study in studies **do** ▷ On an HPC cluster.
- 4: Automate data processing and visualization.
- 5: Run study.
- 6: Check results and apply code changes.
- 7: **end for**
- 8: **if** results are improved on the HPC cluster **then**
- 9: Push changes to the remote repository.
- 10: **if** CI pipeline tests pass **then**
- 11: Milestone reached.
- 12: Add new tests to the CI pipeline,
- 13: Merge feature into development branch.
- 14: Cross-link publication, data, and source code.
- 15: **end if**
- 16: **end if**
- 17: **end while**

We run a Knowledge Base that contains much more information



TECHNISCHE
UNIVERSITÄT
DARMSTADT



knowledge-base.nfdi4ing.de

We just added interactivity

by offering a weekly consultation hour



TECHNISCHE
UNIVERSITÄT
DARMSTADT

The consultation hour is **every wednesday at 10 AM**.

- Information is displayed on the **landing page**.
- It is a **small initial investment**,
- but it **scales badly**.

We are curious, whether this will work out as:

- super-**interactive** for the users and
- super-direct **feedback** for us.

Let us know your opinion in the afterwards discussion!





- This presentation
is available at <https://doi.org/10.5281/zenodo.7649162>
- The **Knowledge Base**
is available at <https://knowledge-base.nfdi4ing.de>.
- **Preprint** describing the workflow
A Research Software Engineering Workflow for Computational Science and Engineering;
Marić, Gläser, Lehr et al., 2022, <https://doi.org/10.48550/arXiv.2208.07460>
- **Slides** about the workflow **at full length including hands-on**
"Continuous" Integration of Scientific Software (in Computational Science and Engineering);
Marić et al., 2021,
<https://zenodo.org/record/5522820.YnTOvnVByXI><https://www.overleaf.com/project/63b5541709ab687c017>



Interaction between
Transport and Wetting Processes

Funded by the German Research Foundation (DFG) – Project-ID 265191195 –
CRC 1194 : Z-INF



NFDI4ing

Funded by the German Research Foundation (DFG) – Project-ID 442146713 –
NFDI4ing : Base Service S-2 *Research software development*