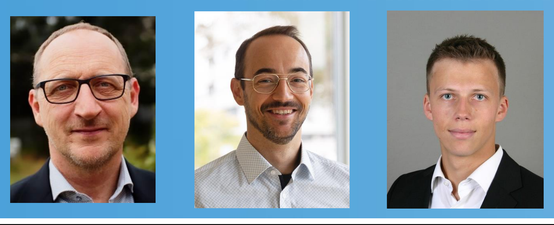
A Research Software Engineering Workflow for Computational Science and Engineering

PASC23, 2023-06-26, Davos, Switzerland

Bothe **Maric** **Schwarzmeier**



Funded by the German Research Foundation (DFG)
Project-ID 265191195 – CRC1194: Z-INF and
Project-ID 442146713 – NFDI4Ing: Base Service S-2

NFDi4ing

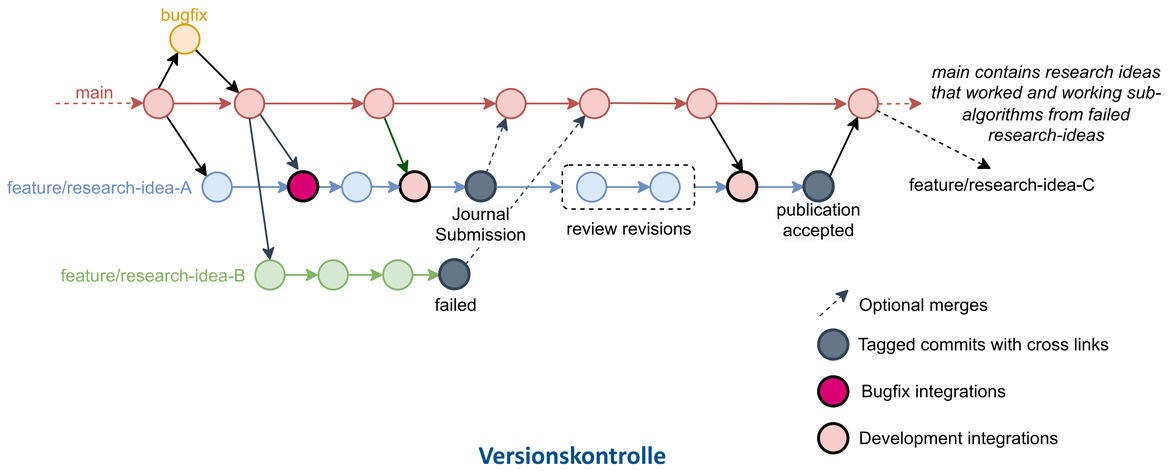
SFB 1194 Wechselseitige Beeinflussung von Transport und Benetzungsvorgängen

TECHNISCHE UNIVERSITÄT DARMSTADT · h_da HOCHSCHULE DARMSTADT UNIVERSITY OF APPLIED SCIENCES · Clean Circles · KIT Karlsruher Institut für Technologie · JGU JOHANNES GUTENBERG UNIVERSITÄT MAINZ · DLR Deutsches Zentrum für Luft- und Raumfahrt

Open Data Concept enabling Traceability

Version Control and Branching Model

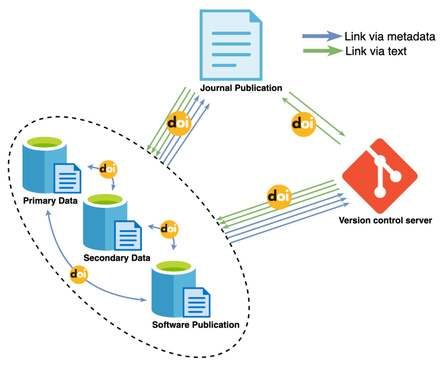
- Tracking of changes is possible → Findability
- Enables continuous development of a single main version→ Sustainability



Versionskontrolle

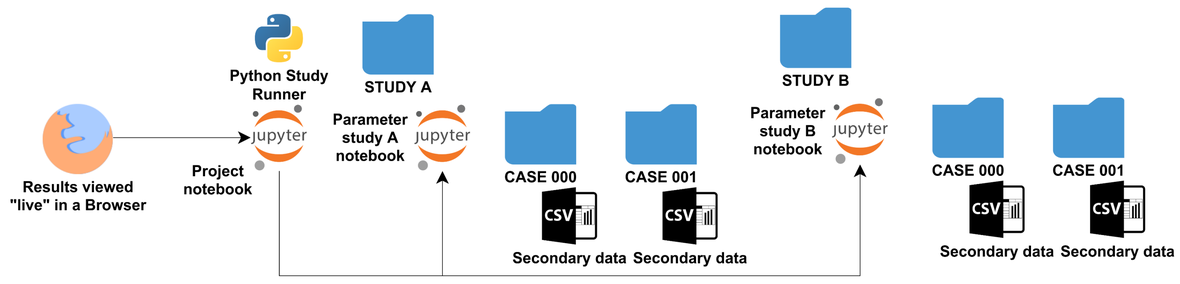
Cross-linking and Data Repositories

- Linking of research data, articles and software
- Retrievable, unique and persistent



Cross-linking

Folder Structure ensures Clarity and Readability

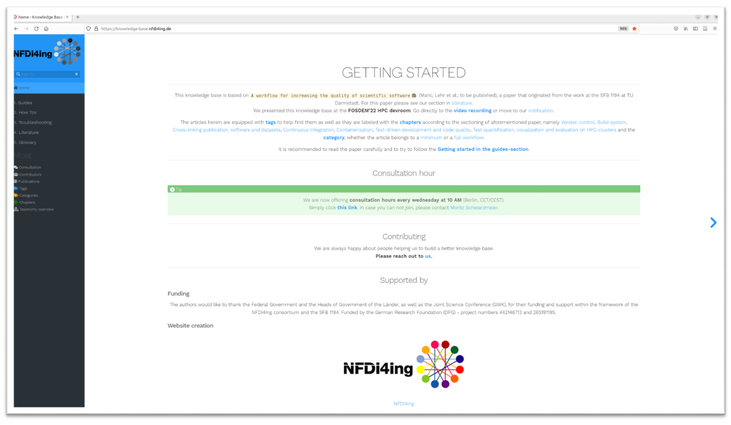


Parameter study structure

- Folder structure ensures clarity and machine readability:
  - One top level folder per parameter study and
  - One low level folder per case
- Parameter study notebook describes parameter study and displays secondary data

Knowledge Base

We run a Knowledge Base in Collaboration with NFDI4Ing



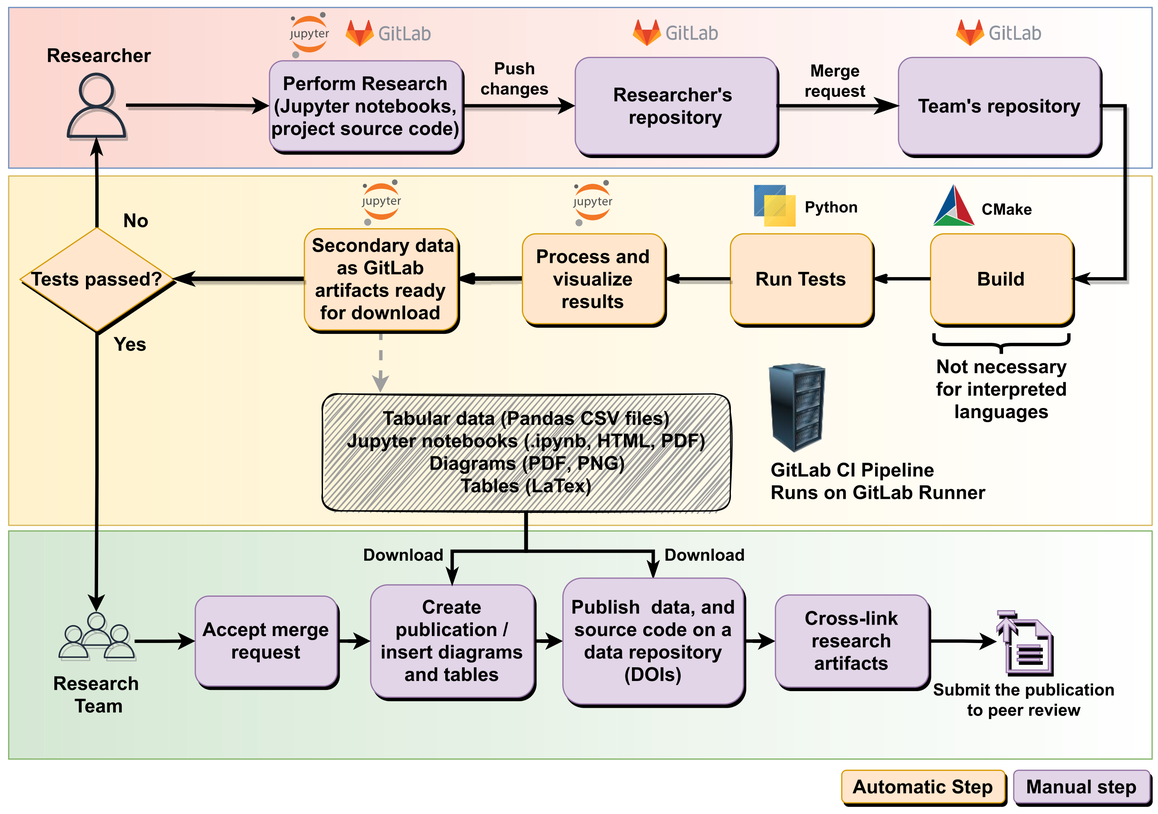
https://knowledge-base.nfdi4ing.de

Tooltips help with unknown terms

We offer consultation hours every wednesday.

- Topics: Version Control, CI, Build Systems, Cross-linking, Container, TDD, Test evaluation
- Guidelines
- Guides with code examples
- Literature
- Peer-reviewed
- Under constant revision
- Taxonomies (tags, chapters)
- Participants from
  - Software development
  - Engineering
  - Research data management

Continuous Integration (CI)



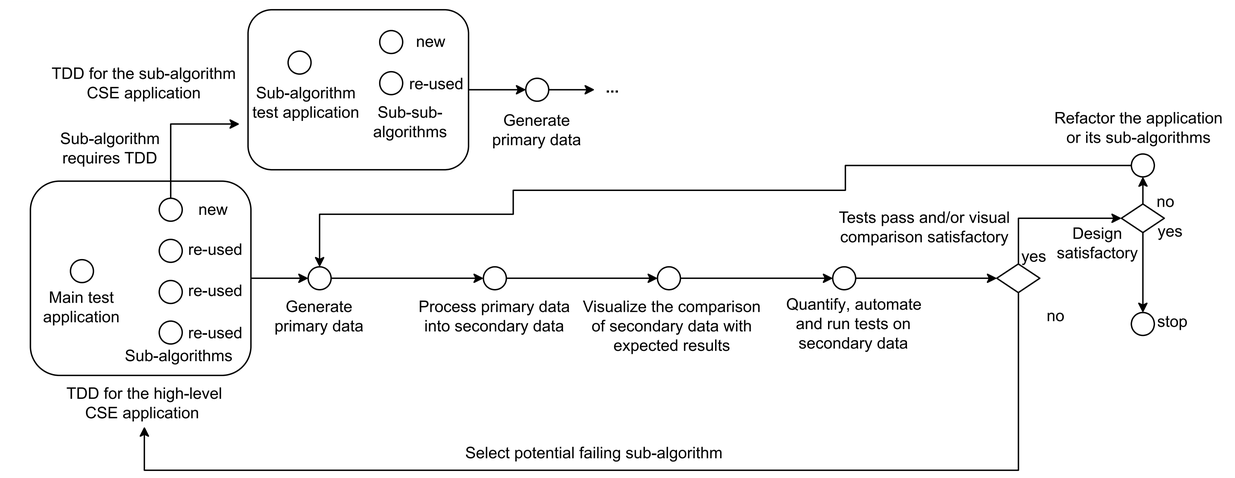
Continuous Integration

- **Integration successfull only, if there are existing tests.**
  - ➢ Reproducibility
- **CMake**: cross-platform build system
  - ➢ **Simplified dependencies** between software projects
    - → Sustainability
- **Jupyter Notebooks**:
  - ➢ **Detailed test documentation** with integrated results/comparison data
- **GitLab CI artifacts**: download of the test results
- **Python test suite** classifies the test results for GitLab CI
- **CI documents workflow** that is guaranteed to be up to date

Challenges

- **Docker (/Apptainer) images** → Reproducibility of primary data
- **Convergence studies might require HPC resources**
  - Legal, security, funding, portability, ... concerns.
  - Reduction of (convergence) testing?

Top-down Test-Driven Development



Testing approach for complex, open source and modular software (like OpenFOAM)

Test functionality, not implementation!

- Assume all sub-algorithms are running.
- Break down non-working algorithms into sub-algorithms.
- Refactor only the running algorithms.

Further Information

- **Knowledge Base:** https://knowledge-base.nfdi4ing.de
- **Preprint** about this 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 with exercises** regarding this workflow:
  "Continuous" Integration of Scientific Software (in Computational Science and Engineering);
  Marić et al., 2021, https://zenodo.org/record/5522820.YnTOvnVByXI
- **This poster:** https://doi.org/10.5281/zenodo.7930299