Towards a Research Software Categorization

Wilhelm (Willi) Hasselbring

Software Engineering
http://se.informatik.uni-kiel.de

November 20th, 2023

Kiel University
Christian-Albrechts-Universität zu Kiel

Research Software

"Research Software includes source code files, algorithms, scripts, computational workflows and executables that were ***created during the research process or for a research purpose****."*

[Gruenpeter et al. 2021]

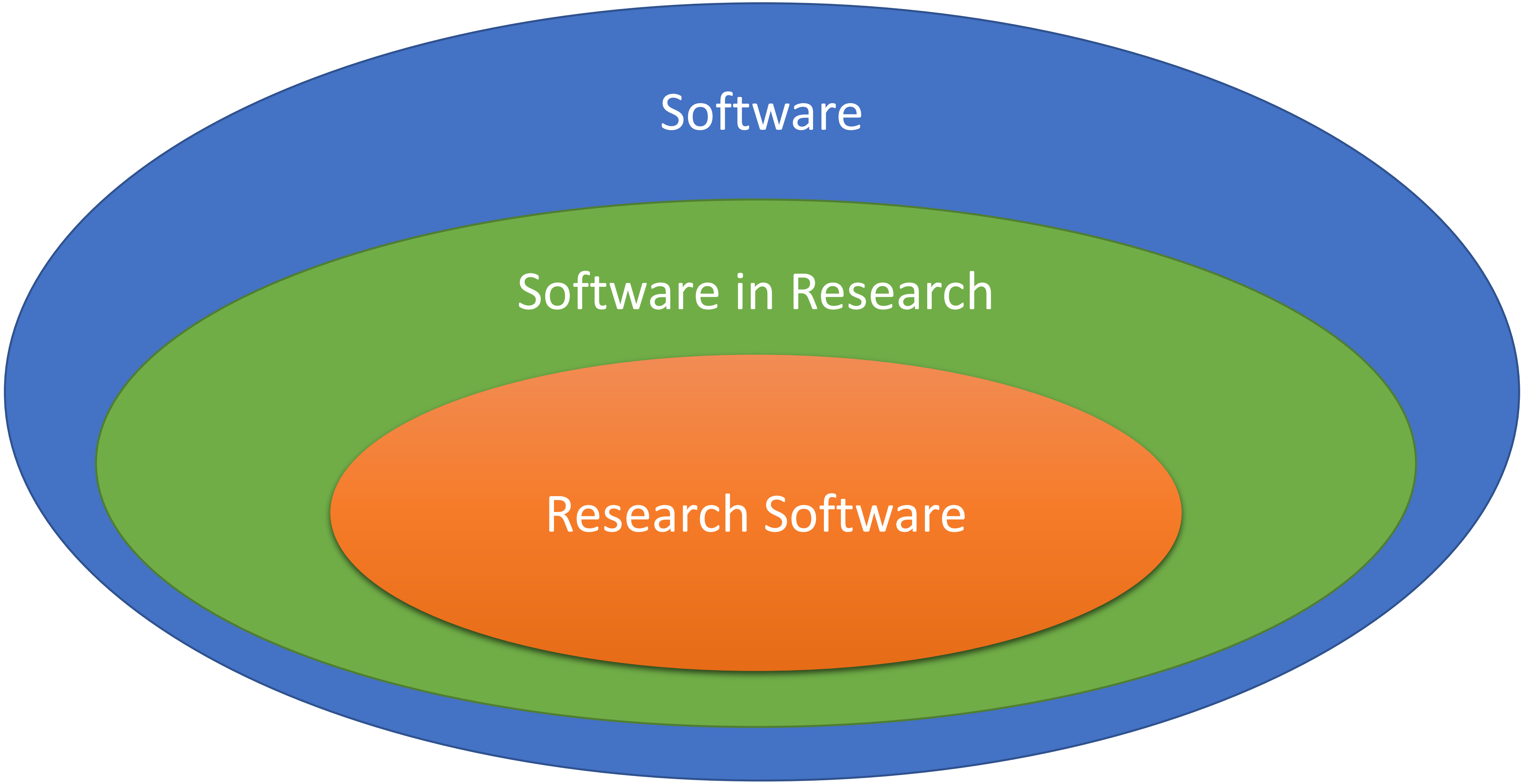
FAIR Research Software

RDA FAIR for Research Software (FAIR4RS) WG [Chue Hong et al. 2022] :

- Research software includes source code files, algorithms, scripts, computational workflows, and executables that are created **during** the research process or **for** a research purpose.
- Software components (e.g., operating systems, programming languages, libraries, etc.) that are used for research but were not created during or with a clear research intent should be considered `**software in research**´ and not `research software´.
- Thus, research software is a separate metaphor of software in research.

Research software should be **FAIR** [Hasselbring et al. 2020b, Lamprecht et al. 2020] and **open** [Hasselbring et al. 2020a].

Software Segmentation



We intend to further categorize the orange ellipse.

Categories of Research Software

Research software mainly falls into one of the following categories (and sometimes combinations):

1. **Modeling, Simulation and Data Analytics** of, e.g., physical, chemical, social, or biological processes in spatio-temporal contexts.
   - Numerical and agent-based modeling and simulation (in silico experiments)
   - Data-driven modeling
   - Data science and data engineering, incl. LLMs
   - Analytics pipelines
   - Data assimilation
2. **(Embedded) Control Software** for complex physical or chemical experiments and instruments, including many forms of sensor-based data collection.
3. **Proof-of-Concept Software Prototypes** in science and engineering research.
4. **Infrastructure** and platform software, such as research data and software management systems.

These categories have varying quality requirements!

[Felderer et al. 2023]

What could we / others do with a Research Software Categorization?

- Assign specific quality requirements to the individual categories
- Recommend appropriate software engineering methods for the individual categories
  - This is, for instance, relevant for institutional software engineering guidelines and checklists.
  - For instance, requirements engineering may be relevant for Category 4, but not for Category 1.
  - For instance, a safety analysis may be relevant for Category 2, but not for Category 1 and 3.
  - Good Practices for High-Quality Scientific Computing [Dubey 2022]
- Design appropriate teaching / education programs for the individual categories
- Explain the relation to stakeholders
- Rationale:
  - We need to understand what kinds of software we have to deal with, and their specific quality requirements

Category 1 in Earth System Sciences

Computational research in the Earth System Sciences

Simulation of Earth system processes by	▪ Earth system models (climate and weather models) and integrated assessment models ▪ sectoral models of, e.g., deep Earth processes, water on the continents, ocean processes, biogeochemical cycles and vegetation
Design, processing and analysis of	▪ Earth observations, e.g., • processing of GRACE satellite signals to derive time series of mass change • geomorphometric analyses of land surface elevations • object identification in satellite images ▪ lab and field observations and experiments, e.g., • luminescence dating • geostatistical analysis
Integrative analysis of	▪ simulation models and Earth observations by, e.g., data assimilation ▪ large databases using statistical analyses or machine learning ("big data" analyses) ▪ stakeholder knowledge by, e.g. multiple-criteria decision analysis or Bayesian networks

[Döll et al. 2023]

Refinement of Category 1

1. **Modeling, Simulation and Data Analytics** of, e.g., physical, chemical, social, or biological processes in spatio-temporal contexts.
    1. Numerical and agent-based modeling and simulation (in silico experiments)
    2. Data-driven modeling
    3. Observation data collection, related to Category 2 & 4
    4. Data science and data engineering, incl. LLMs and data generation
    5. Analytics pipelines for automation and integration, coupling of models, CI/CD
        1. This is related to Category 4 (Infrastructure)
    6. Data assimilation
    7. Scientific visualizations

Defining the roles of research software

[van Nieuwpoort 2022, van Nieuwpoort and Katz 2023]

Research software is a component of our instruments	Category 2
Research software *<u>is</u>* the instrument	Category 1 (& 3 ?)
Research software analyses research data	Category 1.4
Research software presents research results	Category 1.6
Research software assembles or integrates existing components into a working whole	Category 1.5 & 4
Research software is infrastructure or an underlying tool	Category 4
Research software facilitates distinctively research-oriented collaboration	Category 4

Category 3 not included. “proof of concept” is mentioned, but for simulations.

A National Agenda for Research Software
[Australian Research Data Commons 2022]

ANALYSIS CODE	capture research processes and methodology: the steps taken for tasks like data generation, preparation, analysis and visualisation	Category 1
PROTOTYPE TOOLS	demonstrate a new idea, method or model for research	Category 3 (?)
RESEARCH SOFTWARE INFRASTRUCTURE	capture more broadly accepted and used ideas, methods and models for research	Category 4

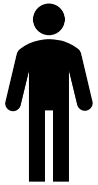
(Category 2 not included)

https://ardc.edu.au/article/research-software-a-first-class-research-output/

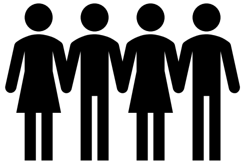
WHAT ARE THE CHALLENGES?

CREATORS OF	Analysis Code	**STRUGGLE WITH**	making their research practices more transparent
AUTHORS OF	Prototype Tools		enabling broad impact through quality software
MAINTAINERS OF	Research Software Infrastructure		lack of support for research software infrastructure

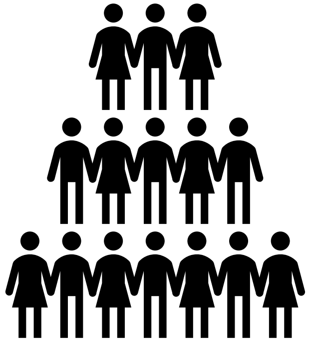
Another Categorization:
Stages of Research Software,
both for Developers and Users



Individual Researcher



Local Research Group



Community (incl. Non-Researchers)

Infrastructures for Quality Research Software Task Force

User stories for the software and research lifecycle:

1. Individual creating research software for own use (e.g. a PhD student)
2. A research team creating an application or workflow for use within the team
3. A team / community developing (possibly broadly applicable) open source research software
4. A team or community creating a research service

Source:

- [Courbebaisse et al. 2023]
- https://eosc.eu/advisory-groups/infrastructures-quality-research-software

Application classes (https://elib.dlr.de/148645/)

Application Class 0	Small scope, personal use	• Scripts to process data for a publication. • Simple administrative scripts to automate specific tasks • Software that demonstrates or tests certain functions
Application Class 1	Narrow scope, beyond personal use	• Software from Bachelor/Master/PhD theses • Software from smaller/shorter research projects
Application Class 2	Extended scope, wider use	• Software from longer-term research projects • Software libraries, frameworks
Application Class 3	Critical software, software products	• Mission-critical software • Software that is sold as a produt (with warranties) • Software that serves as research infrastructure

[Schlauch et al. 2018]
[Fritzsch 2023]

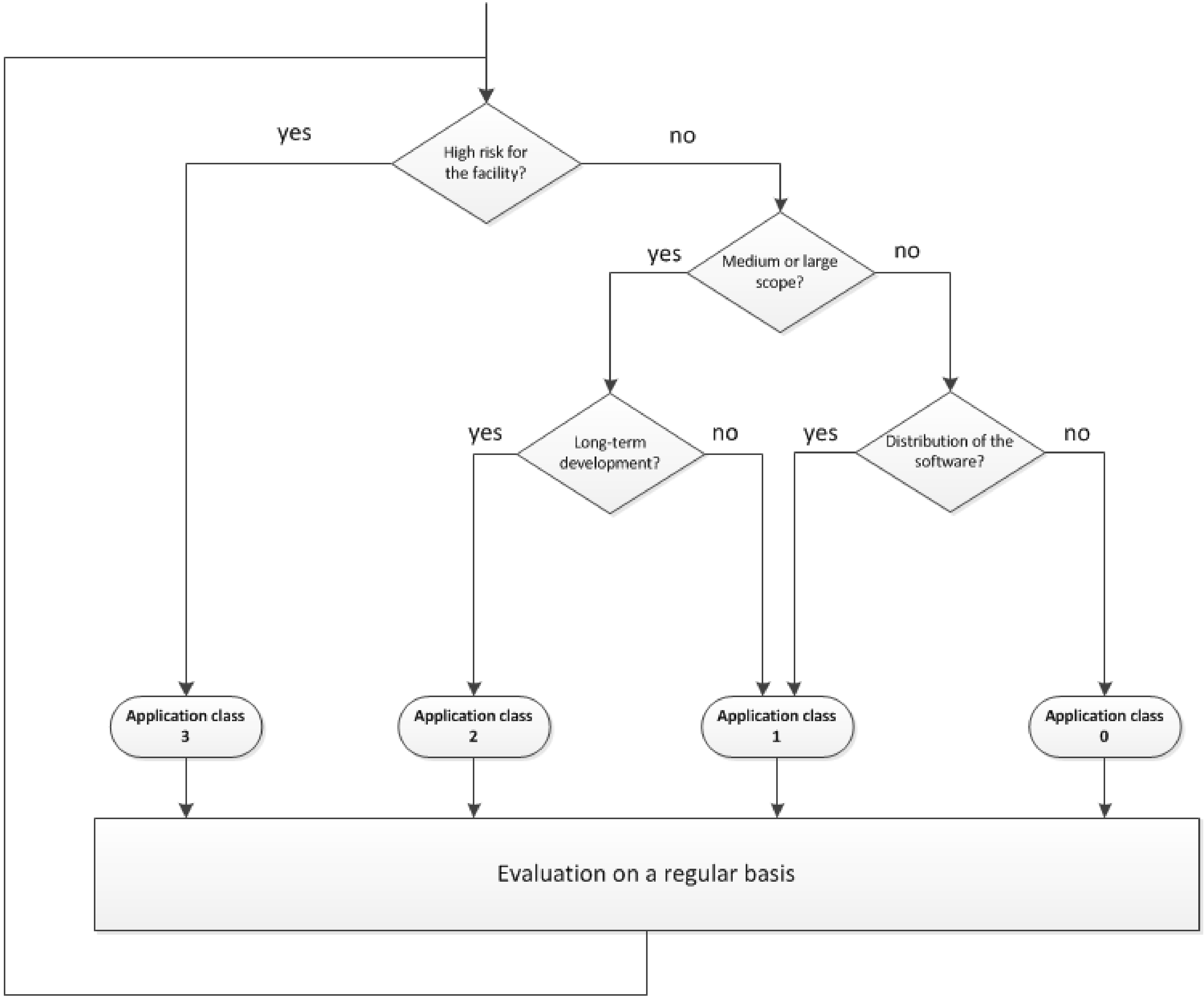
Categorization based on Criticality

- Safety-critical software
  - Failure results in loss of life, injury or damage to the environment;
  - Example: Railway interlocking system
- Mission-critical software
  - Failure results in failure of some goal-directed activity and/or loss of critical infrastructure;
  - Example: Spacecraft navigation system
- Business-critical software
  - Failure results in high economic losses or damage to reputation;
  - Example: Customer accounting system in a bank

⇒ Dependability

- Policy-critical software (?)

Potential risks, expected scope and lifetime determine the application class



[Schlauch et al. 2018]

Software Layers

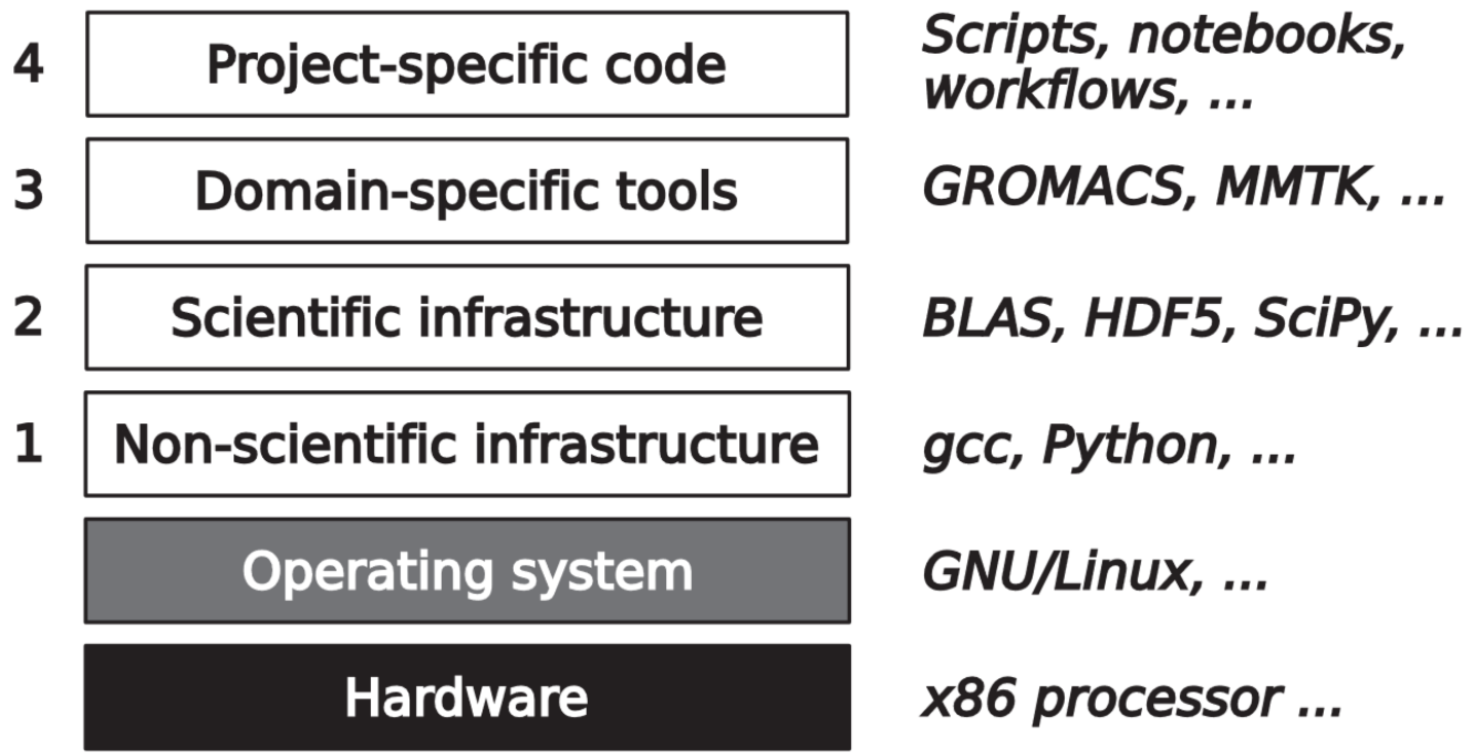


Figure 1. Typical scientific software stack.

Category 1

Category 1

Category 4

(Category 2 & 3 not included)

[Hinsen 2019]

The Research Software Encyclopedia´s Taxonomy

- Software to directly conduct research
  - Domain specific software
    * Domain-specific hardware (e.g., software for physics to control lab equipment, or embedded hardware)
    * Domain-specific optimized software (e.g., neuroscience software optimized for GPU)
    * Domain-specific analysis software (e.g., SPM, fsl, afni for neuroscience)
  - General software
    * Numerical libraries (includes optimization, statistics, simulation, e.g., numpy)
    * Data collection (e.g., web-based experiments or portals)
    * Visualization (interfaces to interact with, understand, and see data, plotting tools)
- Software to support research
  - Explicitly for research
    * Interactive development environments for research (e.g., Matlab, Jupyter)
    * Workflow managers
    * Provenance and metadata collection tools
  - Used for research, but not explicitly for it
    * Databases
    * Application programming interfaces
    * Frameworks (to generate documentation, content management systems, etc.)
  - Incidentally used for research
    * Operating Systems
    * Package Managers
    * Virtualization technologies
    * Formatting, indexing, or other small helper libraries
    * Scheduling and task management (for people)
    * Version Control
    * Text Editors and Integrated Development Environments (IDEs)
    * Communication tools or platforms (e.g., email, video-conferencing, etc.)
    * Infrastructure (e.g., on-prem or cloud servers used for services or research needs)
    * Testing or software libraries

[Sochat et al. 2022], https://rseng.github.io/rseng/

https://rseng.github.io/software/repository/github/containers/podman/annotate-taxonomy/index.html

Software to directly conduct research	Domain-specific software	☐ Domain-specific analysis software
Software to directly conduct research	Domain-specific software	☐ Domain-specific hardware
Software to directly conduct research	Domain-specific software	☐ Domain-specific optimized software
Software to directly conduct research	General software	☐ Data collection
Software to directly conduct research	General software	☐ Numerical libraries
Software to directly conduct research	General software	☐ Visualization
Software to support research	Explicitly for research	☐ Interactive development environments for research
Software to support research	Explicitly for research	☐ Provenance and metadata collection tools
Software to support research	Explicitly for research	☐ Workflow managers
Software to support research	Incidentally used for research	☐ Communication tools or platforms
Software to support research	Incidentally used for research	☐ Formatting, indexing, or other small helper libraries
Software to support research	Incidentally used for research	☐ Infrastructure
Software to support research	Incidentally used for research	☐ Operating systems
Software to support research	Incidentally used for research	☐ Package Management
Software to support research	Incidentally used for research	☐ Personal scheduling and task management
Software to support research	Incidentally used for research	☐ Testing
Software to support research	Incidentally used for research	☐ Text editors and integrated development environments
Software to support research	Incidentally used for research	☐ Version control
Software to support research	Incidentally used for research	☐ Virtualization technologies
Software to support research	Used for research but not explicitly for it	☐ Application Programming Interfaces
Software to support research	Used for research but not explicitly for it	☐ Databases
Software to support research	Used for research but not explicitly for it	☐ Frameworks

Upper Level Categorization

- Commercial Software
- Research Software
- System Software
- …

Application domains:

- Finance and Banking
- Healthcare
- Education
- Transportation and Logistics
- Retail and E-Commerce
- Manufacturing and Industrial
- Government and Public Sector
- Entertainment and Media

Research Software Examples

Example for Category 1 (Modeling and simulation): Modularization of Earth-system simulation software as basis for domain-specific languages

Software Modularization

How to

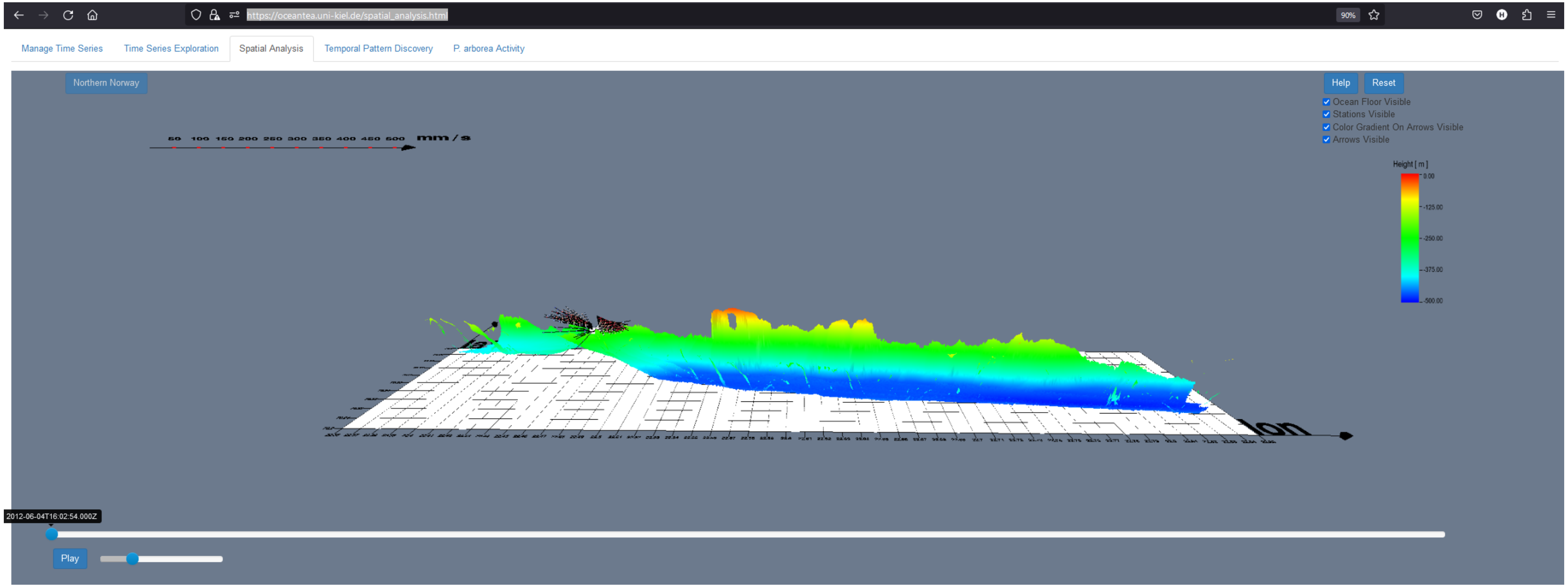
- improve maintainability, stability, reusability, reproducibility, … ?
- enable scalable execution in the Cloud?
- parallelize for high performance computing?
- test for higher quality?
- achieve higher flexibility?

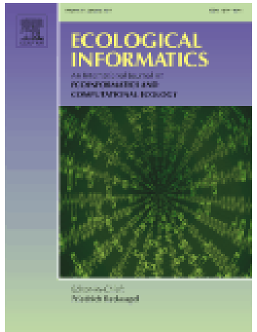
[Johanson & Hasselbring 2017, Claus et al. 2022, Jung et al. 2021, 2022a, 2022b]

OceanDSL

Funded by
DFG Deutsche Forschungsgemeinschaft
German Research Foundation

Example for Category 1 (Data analytics): OceanTEA



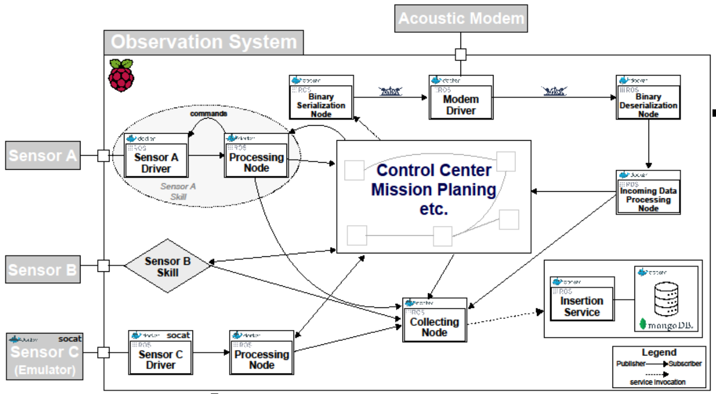


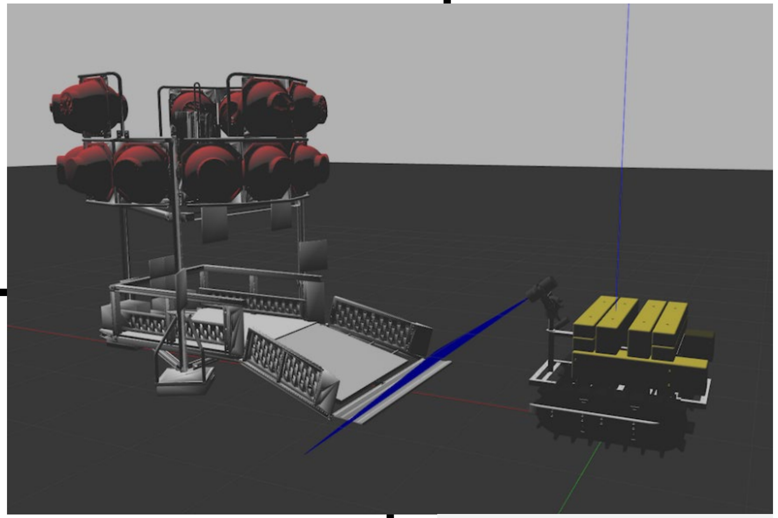
Paper on the analysis results: [Johanson et al. 2017]
Paper on the software architecture: [Johanson et al. 2016]
Code: https://github.com/cau-se/oceantea

ozean der zukunft
DIE KIELER MEERESWISSENSCHAFTEN

Example for Category 2 (Embedded control software): Entwicklung von Software für Unterwasser-Roboter

Digital Twin Prototype





Digital Twin

Physical Twin

[Barbie et al. 2021]

GEOMAR

Examples for Category 3 (Proof-of-Concept Software Prototypes): Software Impacts

Kieker

https://github.com/kieker-monitoring

Kieker: A monitoring framework for software engineering research
[Hasselbring and van Hoorn 2020]

ExplorViz

https://github.com/ExplorViz

ExplorViz: Research on software visualization, comprehension and collaboration
[Hasselbring et al. 2020c]

titan

https://github.com/cau-se/titan-ccp

The Titan Control Center for Industrial DevOps analytics research
[Henning and Hasselbring 2021]

Examples for Category 3 (Proof-of-Concept Software Prototypes):

Example from Pure Mathematics

- Arbitrary precision math in computer algebra systems
  - Goals for developing research software:
    - Proof of concepts
    - Find counter examples
    - Optimization
  - General purpose software
    - Example: Oscar.jl: https://github.com/oscar-system/Oscar.jl
    - Commercial Example: Mathematica: https://www.wolfram.com/mathematica/
  - Special purpose software
    - Example: polymake: https://polymake.org

(Contributed by Lars Kastner, TU Berlin)

Examples for Category 3 (Proof-of-Concept Software Prototypes):

Automated Theorem Proving

- Lean: https://leanprover.github.io
- KeY: https://www.key-project.org

(Contributed by Lars Kastner, TU Berlin)

Examples for Category 4 (Infrastructure):

D-GRID

nfdi

Nationale
Forschungsdaten
Infrastruktur

Outlook: RSE Research

Research Software Engineering

Software Engineering Research

Research Software Engineering Research
aims at understanding and improving how software is developed for research.

RSE Research, in short.

See also: https://github.com/NLeSC/RSE-research [Lamprecht et al. 2022]

References

[Australian Research Data Commons 2022] Australian Research Data Commons: “A National Agenda for Research Software”, Zenodo. DOI https://doi.org/10.5281/zenodo.4940273

[Barbie et al. 2021] Barbie, A., Pech, N., Hasselbring, W., Flögel, S., Wenzhöfer, F., Walter, M., Shchekinova, E., Busse, M., Türk, M., Hofbauer, M. und Sommer, S.: “Developing an Underwater Network of Ocean Observation Systems with Digital Twin Prototypes - A Field Report from the Baltic Sea.” IEEE Internet Computing. 2021. DOI https://doi.org/10.1109/MIC.2021.3065245

[Chue Hong 2022] N. P., Chue Hong, et al. (2022). FAIR Principles for Research Software version 1.0. (FAIR4RS Principles v1.0). Research Data Alliance. DOI https://doi.org/10.15497/RDA00068

[Claus et al. 2022] Claus, M., Gundlach, S., Hasselbring, W., Jung, R., Rath, W. und Schnoor, H.: “Modularizing Earth system models for interactive simulation.” Informatik Spektrum, 45. pp. 300-303. 2022, DOI https://doi.org/10.1007/s00287-022-01490-z

[Courbebaisse et al. 2023] G. Courbebaisse, B. Flemisch, K. Graf, U. Konrad, J. Maassen, R. Ritz: “Research Software Lifecycle”, Zenodo, 2023. DOI https://doi.org/10.5281/zenodo.8324828

[Döll et al. 2023] Döll, P., Sester, M., Feuerhake, U., Frahm, H., Fritzsch, B., Hezel, D.C., Kaus, B., Kolditz, O., Linxweiler, J., Müller Schmied, H., Nyenah, E., Risse, B., Schielein, U., Schlauch, T., Streck, T., Van den Oord, G.: “Sustainable research software for high-quality computational research in the Earth System Sciences: Recommendations for universities, funders and the scientific community in Germany”. 2023, DOI https://doi.org/10.23689/fidgeo-5805

[Dubey 2022] A. Dubey: “Good Practices for High-Quality Scientific Computing,” in Computing in Science & Engineering, vol. 24, no. 6, pp. 72-76, Nov.-Dec. 2022. DOI https://doi.org/10.1109/MCSE.2023.3259259

[Felderer et al. 2023] Felderer, M., Goedicke, M., Grunske, L., Hasselbring, W., Lamprecht, A. L. und Rumpe, B.: “Toward Research Software Engineering Research”. 2023. DOI https://doi.org/10.5281/ZENODO.8020525.

[Fritzsch 2023] Fritzsch, B.: „Richtlinie zur Entwicklung und zum Umgang mit Forschungssoftware am AWI“, EPIC, 2023. URL https://epic.awi.de/id/eprint/57800/

[Fuller and Millett 2011] S.H. Fuller and L.I. Millett, “Computing Performance: Game Over or Next Level?,” Computer, vol. 44, no. 1, 2011, pp. 31–38. DOI https://doi.org/10.1109/MC.2011.15

[Goltz et al.,2015] U. Goltz et al., “Design for Future: Managed Software Evolution,” Computer Science - Research and Development, vol. 30, no. 3, 2015, pp. 321–331. DOI https://doi.org/10.1007/s00450-014-0273-9

References

[Gruenpeter et al. 2021] Gruenpeter, M., Katz, D. S., Lamprecht, A.-L., Honeyman, T., Garijo, D., Struck, A., Niehues, A., Martinez, P. A., Castro, L. J., Rabemanantsoa, T., Chue Hong, N. P., Martinez-Ortiz, C., Sesink, L., Liffers, M., Fouilloux, A. C., Erdmann, C., Peroni, S., Martinez Lavanchy, P., Todorov, I., Sinha, M.: "Defining Research Software: a controversial discussion". 2021. DOI https://doi.org/10.5281/zenodo.5504016

[Hasselbring et al. 2020a] W. Hasselbring, L. Carr, S. Hettrick, H. Packer, T. Tiropanis: "Open Source Research Software". In: Computer, 53 (8), pp. 84-88. 2020. DOI https://doi.org/10.1109/MC.2020.2998235

[Hasselbring et al. 2020b] W. Hasselbring, L. Carr, S. Hettrick, H. Packer, T. Tiropanis: "From FAIR Research Data toward FAIR and Open Research Software", it - Information Technology, 2020. DOI https://doi.org/10.1515/itit-2019-0040

[Hasselbring et al. 2020c] Hasselbring, W., Krause, A., Zirkelbach, C.: "ExplorViz: Research on software visualization, comprehension and collaboration." In: Software Impacts, 6, 2020. DOI https://doi.org/10.1016/j.simpa.2020.100034.

[Hasselbring and van Hoorn 2020] Hasselbring, W., van Hoorn, A.: "Kieker: A monitoring framework for software engineering research." In: Software Impacts, 5, 2020. pp. 1-5. DOI https://doi.org/10.1016/j.simpa.2020.100019

[Henning and Hasselbring 2021] Henning, S., Hasselbring, W.: "The Titan Control Center for Industrial DevOps analytics research," In: Software Impacts, 7, 2021 . DOI https://doi.org/10.1016/j.simpa.2020.100050

[Hinsen 2019] K. Hinsen, "Dealing With Software Collapse," Computing in Science Engineering, vol. 21, no. 3, pp. 104 108, May 2019, DOI 10.1109/MCSE.2019.2900945

[Johanson et al. 2016] A. Johanson, S. Flögel, C. Dullo, W. Hasselbring: "OceanTEA: Exploring Ocean-Derived Climate Data Using Microservices". In: Sixth International Workshop on Climate Informatics (CI 2016), 2016, DOI http://dx.doi.org/10.5065/D6K072N6

[Johanson et al. 2017] A. Johanson, S. Flögel, C. Dullo, P. Linke, W. Hasselbring: "Modeling Polyp Activity of Paragorgia arborea Using Supervised Learning", In: Ecological Informatics, 39. pp. 109-118. 2017, DOI https://doi.org/10.1016/j.ecoinf.2017.02.007

[Johanson & Hasselbring 2017] A. Johanson, W. Hasselbring: "Effectiveness and efficiency of a domain-specific language for high-performance marine ecosystem simulation: a controlled experiment", In: Empirical Software Engineering 22 (8). pp. 2206-2236, 2017. DOI https://doi.org/10.1007/s10664-016-9483-z

[Johanson & Hasselbring 2018] A. Johanson, W. Hasselbring: "Software Engineering for Computational Science: Past, Present, Future", In: Computing in Science & Engineering, 2018. DOI https://doi.org/10.1109/MCSE.2018.021651343

References

[Jung et al. 2021] R. Jung, S. Gundlach, S. Simonov, W. Hasselbring: “Developing Domain-Specific Languages for Ocean Modeling”. In: Software Engineering 2021 Satellite Events, http://ceur-ws.org/Vol-2814/

[Jung et al. 2022a] R. Jung, S. Gundlach, W. Hasselbring: “Software development processes in ocean system modeling.” In: International Journal of Modeling, Simulation, and Scientific Computing, 13 (02). 2022, DOI https://doi.org/10.1142/S1793962322300023.

[Jung et al. 2022b] R. Jung, S. Gundlach, W. Hasselbring: “Thematic domain analysis for ocean modeling.” In: Environmental Modelling & Software, 150. p. 105323. 2022, DOI https://doi.org/10.1016/j.envsoft.2022.105323.

[Lamprecht et al. 2020] A.-L. Lamprecht et al.: “Towards FAIR principles for research software.” In: Data Science 3, 1 (June 2020), 37–59. DOI https://doi.org/10.3233/ds-190026

[Lamprecht et al. 2022] A.-L. Lamprecht et al.: “What Do We (Not) Know About Research Software Engineering?.” In: Journal of Open Research Software, 10(1). 2022. DOI https://doi.org/10.5334/jors.384

[Merali 2010] Z. Merali, “Computational Science: Error, Why Scientific Programming Does Not Compute,” Nature, vol. 467, no. 7317, 2010, pp. 775–777, DOI https://doi.org/10.1038/467775a

[van Nieuwpoort 2022] R. van Nieuwpoort: “What does Research Software look like?”, Zenodo, 2022. DOI https://doi.org/10.5281/zenodo.7347700

[van Nieuwpoort and Katz 2023] R. van Nieuwpoort and D.S. Katz: “Defining the roles of research software”, Front Matter, 2023. DOI https://doi.org/10.54900/9akm9y5-5ject5y

[Randell 2018] B. Randell: 50 years of Software Engineering - or - The View from Garmisch. May 2018, DOI https://doi.org/10.48550/arXiv.1805.02742

[Reussner et al. 2019] R. Reussner, M. Goedicke, W. Hasselbring, B. Vogel-Heuser, J. Keim, L. Märtin, L. (Eds.): “Managed Software Evolution”, Springer, 2019. DOI https://doi.org/10.1007/978-3-030-13499-0

[Schlauch et al. 2018] T. Schlauch, M. Meinel, C. Haupt: „DLR Software Engineering Guidelines", Zenodo, 2018. DOI https://doi.org/10.5281/zenodo.1344611

[Sochat et al. 2022] Sochat, V., May, N., Cosden, I., Martinez-Ortiz, C. and Bartholomew, S., 2022. The Research Software Encyclopedia: A Community Framework to Define Research Software. Journal of Open Research Software, 10(1). DOI https://doi.org/10.5334/jors.359