

GI- UND DE-RSE MUSTER-LEITLINIE ZUR

EFFIZIENTEN ENTWICKLUNG VON FORSCHUNGSSOFTWARE

JULI 2024

DE-ESE.DRG / GLDE

DOI: 10.18420/2025-gi_de-rse



Time to launch: Making the "Standard Template for the Efficient Development of Research Software" accessible

Bernadette Fritzsch, Carina Haupt, Sebastian Nielebock, Bernhard Rumpe, Alexander Struck, Inga Ulusoy

Outline

0 min	30 min	70 min	90 min
 Presentation and Q&A: Target Audience Purpose of the guidelines Content of the guidelines Steps to adoption 	 Breakout sessions: Incentives for decision makers and researchers 	Reporting back Summary and call for act	tions

- Dissemination, incentives and feedback channels

Muster-Leitlinie für die effiziente Entwicklung von Forschungssoftware

Focus Group RSE Software Development Guidelines (special interest group of GI topic area RSE / of de-RSE)

The authors come from an ...



Authors:

Andreas Czerniak, Universität Bielefeld Adrian Ehrenhofer, TU Dresden Bernadette Fritzsch, Alfred-Wegener-Institut, Bremerhaven Maximilian Funk, Max-Planck-Gesellschaft e.V., München Florian Goth, Universität Würzburg Reiner Hähnle, TU Darmstadt Carina Haupt, Deutsches Zentrum für Luft- und Raumfahrt (DLR) Marco Konersmann, RWTH Aachen Jan Linxweiler, TU Braunschweig Frank Löffler, Friedrich-Schiller-Universität Jena Alexander Lüpges, RWTH Aachen Sebastian Nielebock, OVGU Magdeburg Bernhard Rumpe, RWTH Aachen Ina Schieferdecker. TU Berlin Tobias Schlauch, Deutsches Zentrum für Luft- und Raumfahrt (DLR) Robert Speck, Forschungszentrum Jülich GmbH Alexander Struck, Humboldt-Universität zu Berlin Jan Philipp Thiele, Weierstrass Institut Berlin Matthias Tichy, Universität Ulm Inga Ulusov, Scientific Software Center, Universität Heidelberg

... and many more commentators 💪 🙌

Target audience



Purpose of the guidelines

- Requirements of the development, management, and distribution of research software
- Foundation for researchers for research software development
- Practical and organizational processes in RSE
- Decision basis for choosing licenses
- Research Software Engineering Support for scientists

Content of the guidelines

Template-Guideline for Efficient Development of Research Software



Available (in German): https://dl.gi.de/handle/20.500.12116/45663 and https://fg-rse.gi.de/ Table of Contents:

- 1. Executive Summary (for decision makers)
- 2. Introduction
- 3. Research Software Development
 - Categorization of Research Software
 - Minimal Requirements for Research Software Development
- 4. Licensing
- 5. Research Software Engineering Support
- Technical Support
- Technical Services
- Appreciation
- Financial Support

Chapter 3: Research Software - Categorization

Categorization Options



• Other influences (e.g., community, criticality,...)

Chapter 3: Research Software Development - Minimum Requirements

Categorization as foundation for Methodology Choices for Research Software Development

Technology Readiness Levels (TRLs)	Description				
TRL 1-3	Exploration and "Prototyping"				
TRL 4-6	Development and Hardening	SWEBOK - Activities	TRL 1-3	TRL 4-6	TRL 7-9
TRL 7-9	Deployment and Usage	Software Requirements	+	+	+
		Software Design	+	+	++
		Software Construction	++	++	++
		Software Testing	+	++	++
		[]	[]	[]	[] 11

11

Chapter 3: Research Software Development - Foundations



- Software Development Process
- Quality Management (Testing, Validation, etc.)
- Requirements
- Software-Architecture
- Software Modelling
- Versioning
- Concept and Automation of Testing
- Management of software-related data and data foundation
- Best Practices, Design Pattern,
 Issue-Tracking, Coding Guidelines

Technical Foundations

- Git Version Control System
- Continuous Integration / Continuous Delivery
- Testing Frameworks
- Dissemination
- Software Discovery

Chapter 4: Licensing of Research Software



Chapter 5: Research Software Engineering Support

Technical Support

- Consulting on Research Software Development
- Development Support
- Long-term Maintainance
- Development Training
- Consulting Licensing/Exploitation

Technical Services

- Central Version Control System/Code
 Repositories
- FAIR4RS Principles with long-term software archiving
- Research Data Management

Appreciation of Research Software Engineering

- Research Software as Research Output
- Research Software Engineering Prices
- Shared Authorships

Financial Support

- Part of research project fundings
- Shared RSE
- Centralized vs. Decentralized RSEs

Steps to adoption

Variant-rich template

- Main text body plus
 - Elective sections [[Option A]]
 - Alternative sections (ie. research software classification, recommended licensing) [[Variant A1]] / [[Variant A2]]
 - Lists of services
 - Choice of dedicated RSE department or de-centralized support only

Every institution should select and adapt according to their preferences!

// Redaktionelle Anmerkungen [[Anmerkung für Leitlinienerstellende/Hochschullleitungen: ...]] // Auswählbare Alternativen [[Variante A1]] Text Passage 1 [[Variante A2]] Text Passage 2 [[Variante A Ende]]

// Optionale Ergänzung [[Option B]] Optional nutzbarer Text [[Option B Ende]]

Possible adoption workflow





Critical steps in the adoption workflow



Dissemination, incentives and feedback channels

Availability

• DOI 10.18420/2025-gi_de-rse



RSE-Software Entwicklungs Leitlinien

View Guidelines

A guidelines and policy template for the development of research software at German research institutions

A recommendation by GI, RSE working group and de-RSE

Research Software Engineering (abbreviated: RSE) is the application of software engineering practices to research software, i.e., software created for and primarily used in scientific research projects. RSE is not High-Performance Computing (HPC), Computational Science Engineering (CSE), Artificial Intelligence, or Data Science. It has different goals and a distinct professional profile for Research Software Engineers.

Due to the increased relevance and complexity of research software, efficient and high-quality software development for research software also requires a range of organizational, methodological, and legal measures. This results in consequences both for the leadership to provide organizational support and for the individuals involved in software development to adhere to certain methodological guidelines. These guidelines were created based on the distilled knowledge (as of 2024) about software development in general (Software Engineering) and research software in particular (Research Software Engineering). It is planned to regularly update these guidelines and incorporate new insights.

The authors propose these guidelines as a template to aid universities, colleges, and research institutions in the adaptation of fundamental policies and recommendations for research projects involving research software.

• GitHub

https://github.com/gi-ev/RSE-software-entwicklungs-leitlinien



Q Search

Dissemination and incentives



Funding agencies?

Earn badges for your institution / research software: Build trust

Research software and research visibility

Research process and output crucially dependent on research software

Breakout sessions

Breakout group A: Adoption at the institutional level

- How do you incentivize adoption? (maybe more precise question with options, ie badges, recognition on guidelines website, etc)
- How do you make the adaptation of the variant-rich template user-friendly?
- How do you understand what is missing in your institution and how to ramp up to fulfill the guidelines?
- What can support during adoption look like and who can give it? (i.e. community-driven)
- How can the right level of organisation be determined, at which the guidelines should be adopted (whole university? each institute separately? research cluster or consortium? ...)
- Who are the right stakeholders at the institution that need to be addressed for adopting the guidelines?
- Which other resources regarding research software quality are you aware of and how can this be connected?

Tooling? ie.

- website (click-and-select)
- decision trees
- carpentries-style short course

Breakout group B: Accessibility of guidelines for RSEs/SWCs

- How do you understand if your development process is compliant with guidelines?
- How do you understand if your code is compliant with guidelines?
- What steps do you need to take to align better with the guidelines?
- Tooling to determine strategies for your targeted level of development (AK/TRL)?
- Which level of expertise can be expected from readers? Are additional info materials like a glossary needed to make them understandable? Or is it okay to rely on people reading up on topics on their own? Will and how can the RSE center / de-centralized RSE section provide (communal) support for guidelines compliance?
- Is additional specific information needed for making the guidelines easily applicable?
- Which sections require more/less information? Please elaborate.

Tooling? ie.

- carpentries-style short course
- workflows that check for compliance (GH actions)
- Interactive website highlighting relevant strategies for selected application class/TRL of the software

Reporting back: Outcomes from the groups

Summary and call for actions

Time for a survey?

