Electronic Communications of the EASST
Volume 86 (1) Year 2026

deRSE26 - Selected Contributions of the 6th Conference for Research Software Engineering in Germany (Part 1)

Edited by Bernd Flemisch, René Caspart, Bernadette Fritzsch, Florian Goth, Jean-Noël Grad, Florian Thiery

bwRSE4HPC – A blueprint for better research software in Baden-Württemberg

Jasmin Hörter, Dominic Kempf, Marcel Koch, Martin Frank

DOI: 10.14279/eceasst.v86.1.2773

License: This article is licensed under a CC-BY 4.0 License.

Electronic Communications of the EASST (https://eceasst.org).
Published by **Berlin Universities Publishing**
(https://www.berlin-universities-publishing.de/)

bwRSE4HPC – A blueprint for better research software in Baden-Württemberg

Jasmin Hörter [1], **Dominic Kempf** [2], **Marcel Koch** [3] **and Martin Frank** [4]

[1] jasmin.hoerter@kit.edu, [3] marcel.koch@kit.edu, [4] martin.frank@kit.edu
Scientific Computing Center (SCC)
Karlsruhe Institute of Technology, Germany

[2] dominic.kempf@iwr.uni-heidelberg.de
Scientific Software Center (SSC)
University of Heidelberg, Germany

http://www.bwrse4hpc.de/

Abstract: As models and datasets grow, simulations become more complex and efficient software becomes increasingly important. Yet, researchers often lack the knowledge and resources to adapt their codes optimally for HPC clusters and implement best practices. *bwRSE4HPC* addresses this problem by providing expert support in research software engineering to scientists of all domains. RSEs join HPC users in collaboration projects of up to six months, ranging from usability to performance optimization and sustainability. Long-term maintenance or multi-year development projects are realized through joint funding schemes.

Keywords: State-level RSE initiative, software development, research software, sustainability, high-performance computing, energy efficiency

1 Introduction

The use of computer-aided modeling and simulation is no longer limited to technical research areas but is steadily increasing in all fields of science. At the same time, model architectures are constantly evolving, revealing a multitude of new possibilities. However, the development of modern, efficient, and user-friendly research codes, especially for HPC systems, requires expert skills in software engineering, which are not typically among the core competencies of many researchers. The dawn of the exascale era has accelerated this development, requiring profound changes that cannot be implemented without in-depth HPC expertise. Initiatives such as de-RSE e.V.[1] and its international counterparts[2], ReSA[3], or the DFG's guiding principles for the development of research software[4] attempt to raise awareness and present guidelines and possible solutions [CEF+25, Deu24, ABD+21, GAB+25, KCF+26]. On the state level, research software engineering (RSE) strategies are discussed in the High-Performance- and Data-Intensive-

[1] https://de-rse.org/en/positions.html
[2] https://researchsoftware.org/council.html
[3] https://www.researchsoft.org
[4] https://www.dfg.de/en/basics-topics/digital-topics/research-software/principles

Computing strategy of Baden-Württemberg [SHF+23]. The project bwRSE4HPC[5] aims to advance these endeavors by providing expert RSE support to HPC users in Baden-Württemberg. In the following, we present the governance structure and services of bwRSE4HPC and hope that these insights on how to establish and run a multi-institutional RSE support infrastructure can serve as a blueprint for other initiatives that form, e.g., in other states or within the DFG call for research software infrastructures[6].

2 Objectives

The development and maintenance of research software according to best practices is often a low priority for scientists, as it is time-consuming, requires expert knowledge, and does not directly translate into research output or, more precisely, is not considered legitimate research output in their respective field. Thus, software is viewed as a general-purpose tool and aspects like runtime or resource utilization are neglected. Moreover, whole research fields depend on dated legacy code whose maintenance is insufficient and which is increasingly incompatible with modern hardware architectures.

Another pressing issue is the availability of HPC resources as well as their environmental effects due to increased energy consumption. As a result of the growing influence of artificial intelligence across all scientific fields, an increasing demand for computing resources is foreseeable. Simultaneously, computing time is limited, and rising energy costs make computing jobs more expensive. For many years, researchers relied on the expansion of compute power to accelerate and extend their models. However, with the end of Moore's law [LTE+20] it is no longer sufficient to solely count on advances in hardware. Shifting incentives from *"more HPC cycles per kWh"* to *"more knowledge gained per kWh"* is therefore essential. This can be achieved through modern software design which is efficient, user-friendly, and sustainable. Existing algorithms and codes need to be adapted to realize their full potential. Raising awareness is an important first step, followed by supporting researchers in their endeavors to improve. For example, many HPC operators already report costs in terms of energy consumption and utilization rates of a compute job to users.[7]

Moreover, there are many domain- or institution-specific initiatives on research software engineering topics (among others SSC[8], HiRSE[9], Hifis[10], natESM[11], Carpentries[12]) that offer various services ranging from best practices to training programs and consulting. Complementing those, bwRSE4HPC focuses on collaboration projects of up to six months to help users within a clearly defined framework. Extensions for long-term maintenance or large-scale software development are also possible but require additional funding sources.

[5] https://www.bwrse4hpc.de
[6] https://www.dfg.de/de/foerderung/foerdermoeglichkeiten/programme/infrastruktur/lis/lis-foerderangebote/forschungssoftwareinfrastrukturen
[7] https://www.nhr.kit.edu/userdocs/horeka/jobmon/
[8] https://www.ssc.uni-heidelberg.de/de/node/361
[9] https://www.helmholtz-hirse.de
[10] https://www.hifis.net
[11] https://www.nat-esm.de
[12] https://carpentries.org

In addition to designing research software in a more sustainable and efficient way, we also aim to increase the appreciation of research software as a valuable part of the scientific process and legitimate research output. To achieve this, we approach each support request as a collaboration between equal partners. Input from the user group is required throughout the project, and we aim for co-authorship on any subsequent publication that was enabled by our contributions.

2.1 Project framework

bwRSE4HPC is a collaboration between the Scientific Computing Center[13] (SCC) at Karlsruhe Institute of Technology (KIT) and the Scientific Software Center[14] (SSC) at Heidelberg University [KKU24], funded by the Baden-Württemberg Ministry for Science, Research and Arts. Starting in 2025, five full-time equivalent (FTE) positions on the post-doc level are funded for a duration of three years, with 3.5 FTE located at SCC and 1.5 FTE located at SSC. All five positions are filled with fixed term Research Software Engineers (RSEs) who mainly work on time-limited collaboration projects (3.1) with a smaller portion for cultivating long-term cooperation (3.2) and training (3.3). Management and supervision of bwRSE4HPC lies with the group leaders at SCC and SSC. Decisions on collaboration requests are made according to the procedure described in section 3.1. The HPC user steering committee of Baden-Württemberg[15] (LNA-BW) serves as a sounding board for user needs and gives recommendations on the overall development of bwRSE4HPC.

While many classical RSE units are part of the research infrastructure within a dedicated institution, bwRSE4HPC employs personnel in two separate units (SCC and SSC), and offers services to all universities in Baden-Württemberg. The institutional as well as physical separation places some challenges, in particular reaching potential collaborators requires additional effort. However the project also profits from different perspectives and experiences. Within Baden-Württemberg, many IT services[16] are developed and operated by a few institutions but available to all universities (e.g. bwIDM for identity management [KLS+14]). Thus, users are familiar with the state wide system and regularly use services across institutions. Smaller universities profit from this systems, as they might not be able to employ their own RSE units. Contrary to many established units in the UK and US [KMRH19, Cos22], bwRSE4HPC focuses solely on research software development. Neighboring tasks like general IT support or research are not in our scope and are generally provided by other groups or initiatives. The operation as well as the general support for all Tier-3 HPC activities are consolidated in bwHPC[17], which consists of ten universities and five HPC clusters. Tier-2 HPC activities are nationalized within the NHR[18] consortium. We are in regular contact with both initiatives and coordinate training activities or referrals of user requests.

[13] https://www.scc.kit.edu
[14] https://www.ssc.uni-heidelberg.de/
[15] https://www.bwhpc.de/user_steering_committee.php
[16] https://www.bw-uni-it.de/services
[17] https://www.bwhpc.de
[18] https://www.nhr-verein.de

3 Services

To foster collaborations with research groups and work together to improve codes, two tracks of software development services are offered: A short-term collaboration with a fixed workload of up to 6 FTE months, or a long-term collaboration with a wider scope. While short-term collaborations are free of charge, long-term collaborations require additional funding, either provided directly by the users or through a joint funding proposal.

3.1 Collaboration projects ($<$ 6 months)

Researchers can apply for a short-term collaboration at any time, and each request goes through a transparent evaluation process. Valid requests enter the consultation stage, and a project plan, including the work items for the RSEs as well as the users contributions, is established. All projects are collaborations on equal terms between the RSE consultants and the HPC-users, including the proper attribution, which should be realized in the form of co-authorship on subsequent publications or software releases. Applications can cover any area, from optimizing and porting HPC codes to improving the usability and sustainability; or enabling HPC usage for non-HPC code. In the following, we outline the governance structure for each collaboration project, which is based in part on the experience from the SSC partner and their workflows [KKU24].

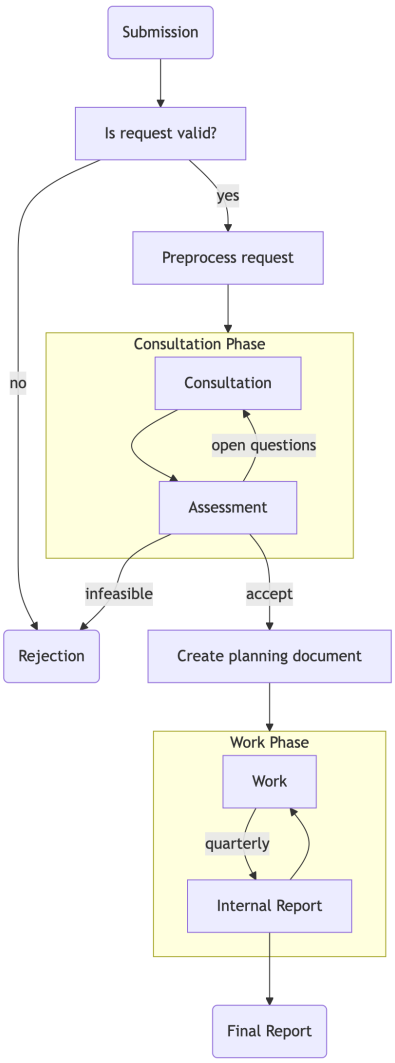


Figure 1: Workflow of a collaborative project

3.1.1 Initial contact and review of the request

The bwRSE4HPC team can be contacted via the bwSupportPortal, a Zammad instance at https://bw-support.scc.kit.edu, or via email to support@bwrse4hpc.de. Once a request is submitted, it is checked for validity. Potential reasons to consider a ticket invalid would be: no HPC context, too time-consuming (e.g. large-scale development from scratch, see 3.2) or no relation to Baden-Württemberg. Too obscure requests, or requests involving obsolete hard- or software, might also be rejected if the team does not possess the required knowledge and a thorough induction would take too much time.

After this first assessment, we categorize the request by applying the following labels:

- **Programming Language:** Python, C, C++, Fortran, Matlab, R, etc.
- **Project Type:** performance engineering, feature development, etc.
- **Technologies involved:** CUDA, OpenCL, MPI, OpenMP, CMake, PyTorch, TensorFlow, Apptainer, etc.
- **Scientific Domain** and responsible **bwHPC Competence Center**
- **Home Institution**

Doing this early on helps all other team members to prioritize whether to read the request in detail. Moreover, we collect data for statistical analysis and to improve our services.

We accept requests on both software with open-source and proprietary licenses. Users may have to choose a proprietary license due to restrictions, such as industry cooperation. While this isn't aligned with the best practices for RSE, we allow it currently to strengthen our portfolio. However, users may also be unaware of software licensing. In that case, we advise following best practices and adopting an open-source license instead.

3.1.2 Consultation

Based on the required skills, we identify two RSEs for a consultation. There are a number of benefits to having two RSEs: the team gets to know each other and their technical strengths, there is some redundancy in the consultation, and the RSEs can iterate between them on the next steps. In a response via the bwSupportPortal, an initial online meeting is offered. We aim for a ticket response time below a week, although the consultation might happen later. To have a more informed discussion, we ask for any relevant code samples or links to relevant software projects beforehand. Consultations are typically planned for one hour, starting with the users presenting their problem: What is the domain? What is the software? What do they want to achieve? Based on that, the RSEs inquire further to understand the context and specific aspects of the potential project. Sometimes domain researchers and RSEs describe the same issue differently, and bridging this gap is important. In most cases, after the first consultation meeting, more research is done by the RSEs and another consultation is scheduled to discuss potential solutions. In some cases, the solutions that the researchers want and what the RSEs would recommend might be fundamentally different[19]. This may be due to a lack of experience or understanding of best practices, and the consultation process will likely need to be extended to reach a common basis.

At the end of this iterative procedure, the bwRSE4HPC team and the researchers decide whether to pursue a collaboration project. In the case of oversubscription, we are transparent about the realization chances and potential timelines of the project. If all parties are in agreement, two RSEs are assigned to the project, with one taking the lead responsibility. The team should primarily be selected based on availability and matching skills.

[19] https://events.hifis.net/event/1741/contributions/13931/attachments/3522/7357/deRSE25.pdf

3.1.3 Project outline

Based on the researchers input, the responsible RSEs write a brief work plan including realistic and relevant technical information regarding the problem statement and suggested solution. It contains the following sections.

- **Project Partners:** List of partners and their affiliations.
- **Software Repositories:** Where is the software currently hosted (GitHub, GitLab etc.).
- **Software Licenses:** All licenses of codes that would be worked on.
- **HPC Cluster:** HoreKa, bwUniCluster, bwForCluster: BinAC, Helix, Justus, Nemo.
- **Problem Statement:** Short statement including the relevance of the software to the research being done, the current state and problems that arise, and what the suggested solution would achieve or which scientific progress would be enabled.
- **Suggested Solution:** Description of the solution in technical terms.
- **Milestones:** Ideally, there should be at least one milestone in the middle of the project that allows synchronization of the project progress and goals with the researchers.
- **Deliverables:** What will be delivered at the end of the project and in what form (including tests, documentation, continuous integration, etc.).
- **Expected Contributions:** This is of great importance for project success, as free services sometimes attract a lack of commitment to support the running project. Expected contributions are:
  - Providing access to compute resources and data required for the project.
  - Giving regular domain knowledge input to the project on both the PI level and on the working level.
  - Trying implemented solutions (User Acceptance Testing).
  - Integrate code that was written during the project into larger code bases (reviewing and merging created Pull Requests).
  - Commitment to publishing a new version of the software after completion and making it accessible with a persistent identifier (e.g. Zenodo) with additional guidance by us if necessary.
  - Inclusion of involved RSEs as co-authors on:
    * Papers using this software for *application code* developed by RSEs.
    * Papers describing this software for *library code* developed by RSEs.

The work plan is iterated upon until all parties arrive at an agreement regarding the above-mentioned points. This is important as it defines the responsibilities on both sides, and it prevents

unrealistic expectations. The agreed upon work plan is then later published on Zenodo under a Creative Commons license[20].

If we receive more requests than our resources can fulfill, we select based on the following criteria:

- Impact: Is there follow-up use? By wider audiences?
- Home institution (even distribution across BW universities).
- Domain (even distribution across disciplines).
- Preferential treatment of open-source codes.

The time frame for this step should be roughly a month from the initial contact.

3.1.4 Project work

While the concrete work is highly individual, certain aspects persist across every project. One aspect is that we try to improve the user's adherance to the current best practices and FAIR principles in RSE [BCK+22] where it relates to the project work. This can include setting up proper version control, licensing, documentation, or testing infrastructure. These improvements are presented, even if they were not part of the original work plan.

The entire bwRSE4HPC team conducts quarterly meetings where each ongoing collaboration is presented and discussed. Upon completion of each milestone or after fixed time intervals, the progress is shared with both the researchers and the team, and the next steps are re-evaluated and, if necessary, adapted. While one RSE will implement the majority of the work, the second RSE assumes a supporting role, for example, to discuss how to tackle technical challenges. Additionally, the second RSE reviews the code changes, which typically increases the quality and sustainability of the proposed solutions.

It should be noted that the RSEs rarely have any relevant domain knowledge. This is not an issue, since work is clearly defined in software engineering tasks, and the RSEs have excellent knowledge of best practices in that field. For example, RSEs with backgrounds in physics have worked successfully on projects in biology or engineering.

3.1.5 Final report

Once the project is completed, a report containing the following aspects is written.

- **Project Partners** and **Key Figures** (see 3.1.3)
- **Executive Summary:** A brief half page summary of the initial problem, the solution, and its expected impact.
- **Implemented Solution:** Detailed description of the implemented solution, including visualizations, performance plots, reproducibility, etc.

[20] https://zenodo.org/communities/bwrse4hpc/

- **Code Availability:** Link to the code that was written during the project (e.g. Pull Request in the Git Repository)
- **Possible Follow-ups:** These are suggestions to the researchers on what could be done to further improve the software. They also include potential time frames and serve as an invitation for joint funding proposals.

The report is published on Zenodo under the same conditions as the work plan. Additionally, it is linked to on the bwRSE4HPC website. The report fulfills two roles. It provides an understandable documentation of the software changes to the users while also enabling us to retain the gained knowledge from the collaboration. Moreover, we ask for feedback to evaluate the user experience and improve our services. About six months after completion, we do a follow-up on how the implemented solution is faring.

3.2 Collaboration projects (>6 months)

In addition to the short-term collaborations, we aim to establish bwRSE4HPC-as-a-Service for research groups throughout Baden-Württemberg. These long-term collaborations typically arise as follow-ups from short-term collaborations and could span several years to implement profound changes in specific software (e.g. development of new algorithms, refactoring, GPU-porting, translating programming languages), provide long-term maintenance, or consult on a long-term research project (e.g. CRC, RTG, ERC, ExStra). Since our resources do not cover these activities, we require additional funding to realize such projects. This can either be done through direct funding by a research group or by developing joint funding proposals.

3.3 Courses

There already exists a diverse training landscape with many courses and workshops from beginner to highly specialized expert level readily available online. Instead of replicating these offerings, we want to provide an in-person forum where we personally get in contact with researchers and address the most common questions or issues that we encounter in our collaborations and in consultations with HPC operators. We want to enable conversations, listen to concerns, and initiate future collaborations.

One pressing issue that emerged over the last couple of years is sustainability. Since the existing bwHPC course program did not address these issues, we developed a new workshop on ‘Sustainability in Research Software for HPC’. The half-day course explores two different aspects of sustainability: energy efficiency and sustainable software development. Aimed at both users of HPC systems and developers of HPC-capable research software, it raises participants’ awareness and provides various information sources, enabling participants to delve deeper into relevant topics.

4 State after One Year

Within the first year of operation, we have facilitated 12 collaboration projects, leaving us at full capacity. While our experiences do not yet warrant precise statistics, a more detailed experience

report is intended in the future. The current projects originated from six different institutions (five universities and one university for applied science) and four of the six bwHPC clusters. The application domains were wide-ranging, from energy systems to biochemistry and molecular physics, as well as engineering and even philosophy. The scope of the projects has been diverse as well. Smaller projects contained tasks such as extending tests, dependency management and installation setup, consulting on best practices, analyzing performance, or refactoring. Larger projects had more significant contributions to the original software, for example, by adding an MPI-based communication backend or a GPU-enabled library for linear algebra. Moreover, we helped first time HPC users to move their software from their Laptops to HPC clusters.

4.1 Example Projects

Two examples are highlighted in the following. During the *genomicrelatedness* project [Ise26] we worked with a research group from bioinformatics to modernize an existing Nextflow research pipeline used for genome sequencing. First the pipeline was refactored, modularized, and updated to Nextflow DSL2. This new version was then added to nf-core [EPF+20]. Now the modernized pipeline follows current best practices and enables reproducible, shareable workflows. While the previous version required the researchers to manually execute multiple scripts, the full pipeline with any user data can now be run in a single line via the Nextflow tool.

An example for a six-month collaboration with a mathematics group is the *M++ Integration of Ginkgo* [SKB26]. Here, the established finite element simulation toolbox M++ [BW21] was enhanced by integrating the Ginkgo library [ACF+22] to offload the time-intensive solving of linear equations to GPUs. Deep changes to M++ were required to enable zero-overhead data exchange with the Ginkgo library. With these updates, M++ is now enabled to use GPUs for their computationally most expensive parts, as well as profit from a variety of existing algorithms in Ginkgo.

5 Conclusion and Outlook

bwRSE4HPC closes a gap in the service portfolio for researchers in Baden-Württemberg. By entering into short- or long-term collaborations, scientists can improve the quality of software that is integral to their research in terms of usability, sustainability, or efficiency. Numerous groups have already requested this support and the successfully completed projects[21] show a significant impact on user software. Without bwRSE4HPC, it is highly unlikely that these changes would have been made.

For the continued success of bwRSE4HPC, the biggest obstacle is to reach audiences that need help the most. Using RSE- or HPC-specific channels often reaches groups that are already interested in the topic, aware of issues, and know where support is provided. The vast majority of HPC users, or prospective HPC users, however, are not aware that their code might profit from expert interventions. This includes especially graduate students as well as researchers in fields that are new to scientific computing. It is this group that holds the greatest potential for

[21] https://zenodo.org/communities/bwrse4hpc/records

optimization. We found that the most effective way to reach those users is through cluster-user mailing lists, graduate schools, and introductory courses offered by the computing centers.

We hope to continue our efforts in the coming years and establish bwRSE4HPC as a permanent service.

Acknowledgements: The authors acknowledge funding by the Ministry for Science, Research and Arts Baden-Württemberg for the period 2025–2027. Moreover, the authors thank Inga Ulusoy, Andreas Baer, René Caspart, Glen Hunter, Thomas Isensee, Tim Schrader and Kai Riedmiller for their contributions to the project.

Data availability: Project reports are available at https://zenodo.org/communities/bwrse4hpc/records.

Declaration on the use of AI: No artificial intelligence tools were used in the creation of this document.

Bibliography

[ABD+21] H. Anzt, F. Bach, S. Druskat, F. Löffler, A. Loewe, B. Renard, G. Seemann, A. Struck, E. Achhammer, P. Aggarwal, F. Appel, M. Bader, L. Brusch, C. Busse, G. Chourdakis, P. Dabrowski, P. Ebert, B. Flemisch, S. Friedl, B. Fritzsch, M. Funk, V. Gast, F. Goth, J. Grad, J. Hegewald, S. Hermann, F. Hohmann, S. Janosch, D. Kutra, J. Linxweiler, T. Muth, W. Peters-Kottig, F. Rack, F. Raters, S. Rave, G. Reina, M. Reiflig, T. Ropinski, J. Schaarschmidt, H. Seibold, J. Thiele, B. Uekermann, S. Unger, R. Weeber. An environment for sustainable research software in Germany and beyond: current state, open challenges, and call for action [version 2; peer review: 2 approved]. *F1000Research* 9(295), 2021. doi:10.12688/f1000research.23224.2

[ACF+22] H. Anzt, T. Cojean, G. Flegar, F. Göbel, T. Grützmacher, P. Nayak, T. Ribizel, Y. M. Tsai, E. S. Quintana-Ortí. Ginkgo: A Modern Linear Operator Algebra Framework for High Performance Computing. *ACM Trans. Math. Softw.* 48(1), Feb. 2022. doi:10.1145/3480935

[BCK+22] M. Barker, N. P. Chue Hong, D. S. Katz, A.-L. Lamprecht, C. Martinez-Ortiz, F. Psomopoulos, J. Harrow, L. J. Castro, M. Gruenpeter, P. A. Martinez, T. Honeyman. Introducing the FAIR Principles for research software. *Scientific Data* 9(622), 2022. doi:10.1038/s41597-022-01710-x

[BW21] N. Baumgarten, C. Wieners. The parallel finite element system M++ with integrated multilevel preconditioning and multilevel Monte Carlo methods. *Computers & Mathematics with Applications* 81:391–406, 2021. Development and Application

of Open-source Software for Problems with Numerical PDEs.
doi:10.1016/j.camwa.2020.03.004

[CEF+25] A. Czerniak, A. Ehrenhofer, B. Fritzsch, M. Funk, F. Goth, R. Hähnle, C. Haupt, M. Konersmann, J. Linxweiler, F. Löffler, A. Lüpges, S. Nielebock, B. Rumpe, I. Schieferdecker, T. Schlauch, R. Speck, A. Struck, J. P. Thiele, M. Tichy, I. Ulusoy. GI- und DE-RSE Muster-Leitlinie zur effizienten Entwicklung von Forschungssoftware. 2025.
doi:10.18420/2025-gi_de-rse

[Cos22] I. A. Cosden. An RSE Group Model: Operational and Organizational Approaches From Princeton University's Central Research Software Engineering Group. 2022.
doi:10.48550/arXiv.2210.16261

[Deu24] Deutsche Forschungsgemeinschaft. Umgang mit Forschungssoftware im Förderhandeln der DFG. Oct. 2024.
doi:10.5281/zenodo.13919790

[EPF+20] P. A. Ewels, A. Peltzer, S. Fillinger, H. Patel, J. Alneberg, A. Wilm, M. U. Garcia, P. Di Tommaso, S. Nahnsen. The nf-core framework for community-curated bioinformatics pipelines. *Nature Biotechnology* 38(3):276–278, 2020. QC 20210917.
doi:10.1038/s41587-020-0439-x

[GAB+25] F. Goth, R. Alves, M. Braun, L. Castro, G. Chourdakis, S. Christ, J. Cohen, S. Druskat, F. Erxleben, J. Grad, M. Hagdorn, T. Hodges, G. Juckeland, D. Kempf, A. Lamprecht, J. Linxweiler, F. Löffler, M. Martone, M. Schwarzmeier, H. Seibold, J. Thiele, H. von Waldow, S. Wittke. Foundational Competencies and Responsibilities of a Research Software Engineer: Current State and Suggestions for Future Directions [version 2; peer review: 2 approved]. *F1000Research* 13(1429), 2025.
doi:10.12688/f1000research.157778.2

[Ise26] T. Isensee. GENOMICRELATEDNESS - Project Report. Mar. 2026.
doi:10.5281/zenodo.19129952

[KCF+26] D. Kempf, R. Caspart, B. Flemisch, F. Goth, J. Linxweiler, F. Löffler, P. Schäfer, R. Speck, A. Struck, M. Ankenbrand, L. Castro, I. Ehlert, J. Grad, M. Hagdorn, A. Loewe, M. Schlottke-Lakemper, U. Schmitt, P. Sommer. Establishing central Research Software Engineering units in German research institutions [version 1; peer review: awaiting peer review]. *F1000Research* 15(719), 2026.
doi:10.12688/f1000research.171263.1

[KKU24] L. Keegan, D. Kempf, I. Ulusoy. Scientific Software Center at Heidelberg University: White Paper. Mar. 2024.
doi:10.5281/zenodo.10867903

[KLS+14] J. Köhler, S. Labitzke, M. Simon, T. Dussa, M. Nussbaumer, H. Hartenstein. bwIDM – Federated Access to IT-Based Services at the Universities of the State

of Baden-Württemberg. In *PIK - Praxis der Informationsverarbeitung und Kommunikation*. Volume 37, pp. 15–21. 2014.
doi:10.1515/pik-2013-0025

[KMRH19] D. S. Katz, K. McHenry, C. Reinking, R. Haines. Research Software Development & Management in Universities: Case Studies from Manchester's RSDS Group, Illinois' NCSA, and Notre Dame's CRC. In *2019 IEEE/ACM 14th International Workshop on Software Engineering for Science (SE4Science)*. Pp. 17–24. 2019.
doi:10.1109/SE4Science.2019.00009

[LTE+20] C. E. Leiserson, N. C. Thompson, J. S. Emer, B. C. Kuszmaul, B. W. Lampson, D. Sanchez, T. B. Schardl. There's plenty of room at the Top: What will drive computer performance after Moore's law? *Science* 368(6495):eaam9744, 2020.
doi:10.1126/science.aam9744

[SHF+23] D. von Suchodoletz, V. Heuveline, S. Farrenkopf, B. Neumair, O. Kohl-Frey, A. Pfister, J. Beutner, M. Resch, T. Walter, T. Nau, R. Dorn, M. Frank, C. Ruiner, G. Schneider, S. Wesner. Rahmenkonzept der Universitäten des Landes Baden-Württemberg für das High-Performance Computing (HPC) und Data-Intensive Computing (DIC) für den Zeitraum 2025 bis 2032. *Universitätstbibliothek Tübingen*, Dec. 2023.
doi:10.15496/publikation-90185

[SKB26] T. Schrader, M. Koch, N. Baumgarten. An Interface from M++ to Ginkgo for Accelerated Linear Algebra on HPCs - Project Report. Mar. 2026.
doi:10.5281/zenodo.19203502