Leibniz Institute of
Agricultural Development
in Transition Economies
(IAMO)
www.iamo.de/en

IAMO
Policy Brief

Nº 42

Issue No. 42
August 2021

Research Software – Sustainable Development and Support

Franziska Appel
Axel Loewe

Research software has become a major asset in academic research. It often is the backbone of existing research methods, enables new research methods, implements and embeds research knowledge, and constitutes an essential research product in itself. Research software must be sustainable in order to understand, replicate, reproduce, and build upon existing research or conduct new research effectively. A change is needed in the way research software development and maintenance are incentivized, funded, structurally and infrastructurally supported, and legally treated. Failing to do so will threaten the quality and validity of research. In this context, more than 50 scientists from various institutions gathered at the Robert Koch Institute in November 2019 for a DFG roundtable discussion on sustainability aspects of research software and to draw attention to the need for funding instruments for the sustainable development and provision of research software. The subsequently published position paper (Anzt et al., 2021) seeks to increase the awareness of political and academic decision-makers on the importance and needs of sustainable research software practices. In particular, it recommends strategies and measures to create an environment for sustainable research software.

The Backbone of 21st Century Research: current state and open challenges

Computational analysis of large data sets, computer-based simulations, and software technology in general play a central role in virtually all scientific breakthroughs of the 21st century. Researchers often base their research on software that was developed in-house or as a community effort. Many of these software stacks cannot be sustained – often because the research software is not a first-class deliverable in a research project and it remains in a prototype stage, or because of missing incentives and resources to maintain the software after the end of the project funding. This results in a highly inefficient system where millions of lines of code are generated every year that will not be re-used after the end of a project or the developer's position.

In Germany, funding bodies are increasingly acknowledging the importance and value of sustainable research software and related infrastructure. For example, the German Research Foundation (DFG), the largest basic research funding body in Germany, opened calls for sustainable research software development and quality management in research software. In addition, the DFG's 2019 "Guidelines for Safeguarding Good Research Practice" Code of Conduct now explicitly lists software side-by-side with other research results and data.

These positive developments notwithstanding, guidelines and policies for sustainable research software development in Germany are unfortunately still lacking, and long-term funding strategies are missing. The conclusions and recommendations of the DFG Roundtable are summarized below based on Anzt et al. 2021.

How to decide which software to sustain?

The sustained funding of all existing software is not only impossible but would risk splintering the community and make the research community inefficient. Therefore, it is important to agree on a

list of transparent criteria that qualify a software product for sustained funding. Such a list must be general enough to be applied to research software from various research disciplines while also respecting differences between fields.

Here we focus on the mandatory criteria that should be fulfilled across domains. Further additional desirable criteria that can be implemented to different degrees depending on the use case and domain-specific software development requirements are mentioned in Anzt et al. 2021.

1) Usage and impact

a) Use beyond a single research group
b) At least one peer-reviewed scientific publication based on the software
c) Market analysis (software is important to a user base of relevant size and either unique or one of the main players in a field with several existing solutions)
d) Appropriate training material is provided

2) Transparency and quality

a) Public availability of the source code
b) Version control system
c) Licensed
d) Dependencies on libraries and technologies are defined

3) Maturity

a) Software management plan
b) Website with a clearly defined point of contact
c) Group of developers

Who is responsible for sustaining research software?

Both funding agencies and research institutions have roles in sustaining research software. Funding agencies should request that applicants include considerations about how software developed in a project can be sustained beyond the end of the funded project. This also calls for longer term funding modules for maintenance and support. A follow up on these plans during and after the project lifetime, i.e., a dedicated software management plan, is crucial.

Research institutions should establish attractive long-term career options in the academic environment to leverage the benefit of dedicated research software engineers (RSE) for research software practices. Decision makers should broaden the definition of research impact beyond traditional scientific publications to also include other impactful results, for example when recruiting for academic positions. Broadening the basis for assessment of research and researchers is explicitly mentioned in the "San Francisco Declaration on Research Assessment" (SF DORA), which many German and international funders and research institutions signed recently.

How can research software be sustainably funded?

Funding is a crucial factor for sustaining research software. Existing project-focused funding instruments on local, national, and international levels need to be complemented with funding instruments specifically designed for sustained research software maintenance and provision. For example, software projects fulfilling the above-mentioned sustainability criteria should have access to sustained funding as long as they live up to the standards and continue to be a central component of the research landscape.

Computing centers and supercomputing facilities for research need to receive earmarked resources for the support of sustainable software development. This funding is necessary to provide continuous integration services, a hardware portfolio for development, testing and benchmarking software, as well as personnel for training domain researchers in software design and the proper usage of the services.

The creation and maintenance of training materials for general research software engineering education as well as software-specific documentation and tutorial creation need to be reflected in funding opportunities. This can either happen by dedicating modules of research or software grants to providing support and the generation of training material or by creating funding schemes focusing on interdisciplinary software development education.

Which infrastructure is needed to sustain research software?

Project management tools

The deployment of tools for distributed software development and software management is recommended as central research infrastructure to enable cross-institutional collaboration. An important aspect of this is the cataloging of research software to reduce the duplication of development efforts. This can efficiently be realized by promoting the registration of all research software with a unique identifier and developing a tool that allows to explore the research software landscape. Research software contributors should use their ORCID iD to be uniquely identifiable and referable.

Developer training, motivation, and knowledge exchange

Workshops and seminars that provide easy access to hands-on training on software-related questions should be promoted and supported. A recent initiative in this direction are the Helmholtz HIFIS Events. However, transferring this knowledge as part of a formalized education (i.e., in the curricula of university degrees) is crucial for software development skills.

Networking and community building can lead to a permanent establishment of sustainable research software. Software Carpentry and similar efforts like the creation of the Data Science Academy HIDA in the Helmholtz Association help to exchange and distribute knowledge. Local chapters of RSE groups and (inter-)national conferences further foster community building. Research institutions as well as funding agencies should actively promote and support the creation of such groups.

The creation of a national Software Sustainability Institute (involving funded positions to establish services and create training material) similar to the U.K.'s Software Sustainability Institute (SSI), which serves as a national contact for all aspects related to research software, is strongly recommended.

Research software discovery, publication and archiving

Proper software publication and the ability to find existing software solutions for a given problem are necessary for the research community to exploit synergies and avoid redundant development. Retrievability requires publication in a globally accessible location with appropriate metadata, e.g. Citation File Format (CFF). Comprehensive metadata (contributors, contact, keywords, linked publications, etc.) and publishing platforms enable proper citations, which in turn benefits research evaluation. Especially in interdisciplinary environments, the creation of a meta index covering important (disciplinary) software indexes is recommended in order to ease discovery of software locations.

GitLab or GitHub (common collaborative working environments) linked to repositories like Zenodo are examples of suitable publication platforms. Repositories like Zenodo mint DOIs, allow versioning and are publicly funded for long-term access. Metadata and citations are key for tools like PIDgraph, DataCite.org, CrossRef, which utilize Persistent Identifiers (PIDs) like DOIs. In Germany, it is expected that the Nationale Forschungsdateninfrastruktur (NFDI) will create or support discovery platforms at a central point that will ease information retrieval. Unfortunately, research software does not yet play the central role that it should in the NFDI.

The publication of a certain software version for reference in research articles requires simple ways to archive research software on a long-term basis. Publishing software, their dependencies, and environment in containers can ease evaluation and further reuse. Software preservation aims to extend the lifetime of software that is no longer actively maintained. One solution to keep the software in an executable state by preserving its context and runtime environment is to use containers such as Docker. Another threat is losing project repositories on global platforms like GitHub or BitBucket. Here, global platforms like Software Heritage harvest those repositories and prevent loss by long-term archiving.

Legal aspects

An essential question regarding the sustainability of research concerns the free and open distribution of research software and other research output. It should be discussed whether research funding organizations such as the DFG should expect publishing all funded software under such licenses, following the paradigm of "public money, public code". The use of an FSF- or OSI-approved FLOSS license for example would enable a truly free model and also reduce legal issues. If licenses such as Apache or MIT are applied, the research institutions may later still commercialize the software if appropriate. Such open source licensing is also beneficial for start-ups that intend to provide professional services for the software.

Especially in light of the new DFG Code of Conduct, the installation of a research software task force is important for all (German) research organizations/associations. As part of their missions, such a group should organize a local legal help desk, organize educational offers and (if not implemented yet) develop the software policy of the research performing organization/association. As an example, with the help of on-boarding processes performed by the research software task force, RSEs should be able to keep the clearance of rights as simple as possible right from the start. This helps to avoid that – out of uncertainty and fear to make a legal mistake – some research groups end up not choosing any license at all, which will severely hinder reuse of the software. We suggest that the local task forces build a network with the other research organizations. Networks enable exchange of ideas and can generate a bottom-up strategy to organize RSE standards and possibly build a basis to form the aforementioned software sustainability institute.

Conclusions

- Implementation of funding schemes and incentives for sustainably supporting the development and maintenance of research software based on clear and transparent criteria.
- Building attractive career paths for research software engineers (RSEs).
- Installation of infrastructure that enables sustainable software development including platforms for discovery, collaboration, continuous integration, testing, and long-term preservation.
- Establishment of a nation-wide institution similar to the Software Sustainability Institute (SSI) to provide project consulting and code review services as well as sustainable software development training.
- Sustainable software development practices as an integral component of university teaching curricula.
- Decision of research funding bodies whether the "public money, public code" paradigm should be applied, which requires that all publicly funded software has to be publicly available under a Free/Libre Open Source Software (FLOSS) license.

Further Information

Literature

Anzt, H.; Bach, F.; Druskat, S.; Löffler, F.; Loewe, A.; Renard, B. Y.; Seemann, G.; Struck, A.; Achhammer, E.; Aggarwal, P.; Appel, F.; Bader, M.; Brusch, L.; Busse, C.; Chourdakis, G.; Dabrowski, P. W.; Ebert, P.; Flemisch, B.; Friedl, S.; Fritzsch, B.; Funk, M. D.; Gast, V.; Goth, F.; Grad, J.-N.; Hegewald, J.; Hermann, S.; Hohmann, F.; Janosch, S.; Kutra, D.; Linxweiler, J.; Muth, T.; Peters-Kottig, W.; Rack, F.; Raters, F. H. C.; Rave, S.; Reina, G.; Reißig, M.; Ropinski, T.; Schaarschmidt, J.; Seibold, H.; Thiele, J. P.; Uekermann, B.; Unger, S.; Weeber, R. (2021): An environment for sustainable research software in Germany and beyond: Current state, open challenges, and call for action. F1000Research 9;295. URL: https://doi.org/10.12688/f1000research.23224.2 (status: 30 July 2021).

de-RSE e.V. – Gesellschaft für Forschungssoftware (Society for Research Software): Research Software Engineers (RSEs) – verantwortlich für wissenschaftliche Software. URL: https://de-rse.org/de/ (status: 30 July 2021).

Contact

Dr.-Ing. Axel Loewe
axel.loewe@kit.edu
Tel.: +49 721 608-42790

Karlsruhe Institute
of Technology (KIT)
Kaiserstraße 12
76131 Karlsruhe
Germany
www.ibt.kit.edu

Dr. Franziska Appel
appel@iamo.de
Tel.: +49 345 2928-117

Leibniz Institute of
Agricultural Development
in Transition Economies
(IAMO)
Theodor-Lieser-Str. 2
06120 Halle (Saale)
Germany
www.iamo.de/en

Printed edition: ISSN 23 63-5800
ISBN 978-3-95992-125-1

Online edition: ISSN 23 63-5797
ISBN 978-3-95992-126-8

The DFG call for sustainable research software development provided funding to IAMO with the project "AgriPoliS2020 – Sustainability of the Research Software AgriPoliS".

IAMO

Leibniz Institute of Agricultural Development in Transition Economies (IAMO)

The Leibniz Institute of Agricultural Development in Transition Economies (IAMO) analyses economic, social and political processes of change in the agricultural and food sector, and in rural areas. The geographic focus covers the enlarging EU, transition regions of Central, Eastern and South Eastern Europe, as well as Central and Eastern Asia. IAMO is making a contribution towards enhancing understanding of institutional, structural and technological changes. Moreover, IAMO is studying the resulting impacts on the agricultural and food sector as well as the living conditions of rural populations. The outcomes of our work are used to derive and analyse strategies and options for enterprises, agricultural markets and politics. Since its foundation in 1994, IAMO has been part of the Leibniz Association, a German community of independent research institutes.

Leibniz Association