Relevance and Challenges regarding Research Software Sustainability

M. Hammitzsch and J. Wächter, Helmholtz Centre Potsdam - GFZ German Research Centre for Geosciences

State of the Art

Scientific software has become an indispensable commodity for the production, processing and analysis of empirical data but also for modelling and simulation of complex processes. Software has a significant influence on the quality of research results. The current use of software and the recognition of software engineering does not reflect the importance of this work in the research process.

The reasons for this are manifold and result in a variety of peculiarities, such as:

- Lack of recognition for the academic performance of scientific software development;
- Missing anchoring in the scientific reputation system;
- Limited availability and usability of scientific software;
- Multiple and parallel development activities;
- Insufficient skills in software engineering;
- Lack of quality standards for the development and review of scientific software;
- Lack of reproducibility;
- Unclear rules for publication regarding to licenses and intellectual property.

To address these challenges several promising solutions already exist, such as:

- Software journals using individual policies for software related papers;
- Digital Repositories minting DOIs for source code copies and software release packages;
- Foundations and companies providing environments for Free and Open Source Software (FOSS) projects;
- Institutes offering software/code repositories and digital repositories for research results.

Challenge Software Sustainability

From our point of view software sustainability is a process including a balanced set of defined measures addressing, firstly, the overall software production process, secondly, the publication, preservation and use of software. Thus, measures fostering software sustainability include the:

- Implementation of software according to best practices in software engineering;
- Packaging and publication of releases according to specified rules and quality criteria;
- Transfer into software repositories allowing preservation, maintenance and accessibility;
- Promotion of software re-use within research infrastructures and virtual research environments.

As for research data, answers to a variety of related questions and a common understanding of handling scientific software with defined processes have to be developed jointly. Amongst others, these processes have to cover issues regarding quality assurance, versioning and documentation, traceability, reproducibility and reusability, archiving and the use of persistent identifiers, metrics for evaluation and validation, measuring of productivity and impact as well as the dissemination and recognition of scientific achievement. Furthermore, open access and the use and interplay of software publication, data publication and traditional paper publication have to be considered.

On the one hand, decentral components at the researcher's workplace should comprise methods and tools for the handling of software and the versioning and documentation of sources. The use of best practices in software engineering will help to prepare scientific software for publishing. On the other hand, central components support workflows for the publication of software, including a review process, quality assurance and the assignment of persistent identifiers (DOI). In addition, the potential of journals, software repositories and persistent identifiers should be utilized and combined to improve the publication and dissemination of sustainable software solutions.

Approach

To implement the concepts a combined bottom-up / top-down approach should be considered that will be implemented in parallel in different scientific domains, e.g. in earth sciences, climate research and the life sciences.

Therefore in a top-down approach policies and strategies for the handling of scientific software have to be developed to foster the adoption and acceptance as well as the operationalisation and re-use of platforms. This will include the conceptualisation and promotion of a software management plan. Furthermore, recommendations for the institutionalisation serve a basis to ensure sustainability, e.g., through acceptance, approval, or certification by communities, publishers and other organizations.

Bottom-up, concepts should be iteratively implemented, tested, and evaluated by scientists and research software engineers. Platforms to distribute, communicate, and publish scientific software should be developed continuously on the basis of gained experiences and results. The platform services should be extended one by one corresponding to the requirements of the communities. Thus, the implemented platforms can be improved and stabilized incrementally with tools supporting software engineering, re-use of software, scientific workflows, and publishing processes.

By their adoption both approaches, bottom-up and top-down, will result successively in a change of the scientific practices which is the key for sustainability. This assumes that scientist learn to write and release code and software as they learn to write and publish papers. But this requires the relevant skills that can be passed to colleagues and followers.