Publishing Platform for Scientific Software – Lessons Learned –

EGU2015-13068, R316

Martin Hammitzsch (1), Bernadette Fritzsch (2), Dominik Reusser (3), Björn Brembs (4), Gernot Deinzer (4), Peter Loewe (5), Martin Fenner (6), Xenia van Edig (7), Roland Bertelmann (1,8), Heinz Pampel (1,8), Jens Klump (9), and Joachim Wächter (1)
(1) GFZ German Research Centre for Geosciences, Potsdam, Germany, (2) Alfred Wegener Institute Helmholtz Centre for Polar and Marine Research (AWI), (3) Potsdam Institute for Climate Impact Research (PIK) e.V., (4) University Regensburg,
(5) German National Library of Science and Technology / University Library Hannover (TIB/UB), (6) PLOS (Public Library of Science), (7) Copernicus Gesellschaft mbH, (8) Helmholtz Open Science Coordination Office, (9) CSIRO Mineral Resources Flagship

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. However, 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, e.g.:

- 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

For strengthening the recognition, for increasing its visibility and for promoting the reproducibility of research results, concepts for the publication of scientific software have to be developed, tested, evaluated, and then transferred into operations. For this, the publication and citability of scientific software have to fulfil scientific criteria by means of defined processes and the use of persistent identifiers, similar to data publications. For this, first approaches have been already established with:

- Software journals using individual policies for software related papers
- Zenodo and Figshare minting DOIs for source code copies and software release packages
- Institutes offering software repositories and digital repositories for research results

However, various challenges are not addressed yet and remain open for implementation (Fig.1)

Fig.1: Challenges not addressed (blue font) and fairly covered (green and yellow font)

Challenges

To address these challenges a blueprint for a scientific software publishing platform and a systematic implementation plan has been designed. In addition, the potential of journals, software repositories and persistent identifiers have been evaluated to improve the publication and dissemination of reusable software solutions.

It is important that procedures for publishing software as well as methods and tools for software engineering are reflected (Fig.2) in the architecture of the platform, in order to improve the quality of the software and the results of research.

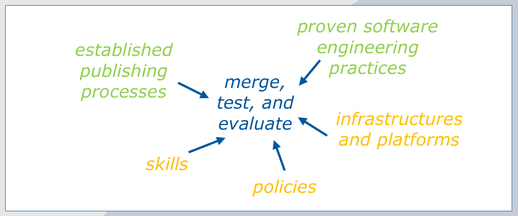

Fig.2: Integrate proven concepts and open issues

In addition, it is necessary to work continuously on improving specific conditions that promote the adoption and sustainable utilization of scientific software publications. Among others, this would include policies for the development and publication of scientific software in the institutions but also policies for establishing the necessary competencies and skills of scientists and IT personnel.

To implement the concepts developed in SciForge a combined bottom-up / top-down approach (Fig.3) is considered for implementation in different scientific domains, e.g. in earth sciences, climate research and the life sciences.

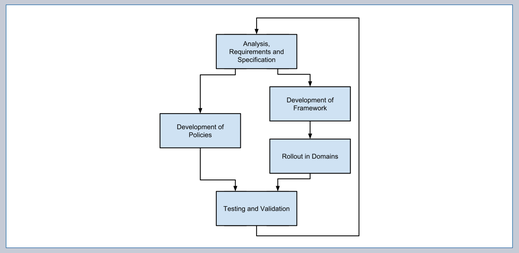

Fig.3: Implementation, testing, and validation cycle

Bottom-up Approach

A platform for scientific software publishing should be iteratively implemented, tested, and evaluated 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 platform for the publication of scientific software can be improved and stabilized incrementally as a tool with software, science, publishing, and user oriented features. Based on specific domain requirements, a reference platform for the publication of scientific software can be developed and proven to close the conceptual gap between scientific software journals and software repositories. The platform should encompass central and decentral components (Fig.4).

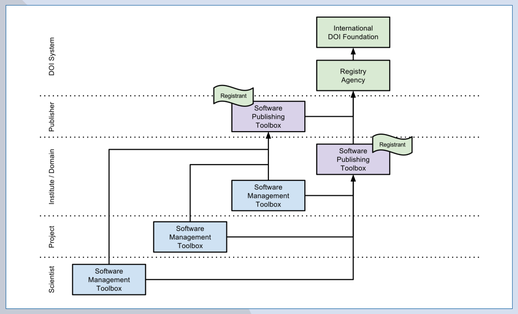

Fig.4: Central and decentral components

Central components support workflows for the publication of software, including a review process, quality assurance and the assignment of persistent identifiers (DOI). Decentral components at the researchers' workplace comprise 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. By applying these principles conditions will be promoted which support not only the publication but also the subsequent reuse of scientific software.

Top-down Approach

However, the success of the reference platform for the publication of scientific software not only depends on the quality of the technological developments.

Therefore 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 reuse of the reference platform. 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 official certification by publishers and other organizations.

Publishing Platform

Based on the concept of digital publications a reference platform should be implemented with relevant key functionalities for the scientific process (Fig.5). For the platform and their interfaces following primary components have been identified:

- Landing page: provides an open access to the software publication and related information
- Software repository: supports processes for software documentation and software engineering and is based on established tools such as a revision control system
- Interfaces to persistent identifier and metadata systems: realize gateway to the publishing sector and make use of relevant services such as for the registration of software publications by minting DOIs

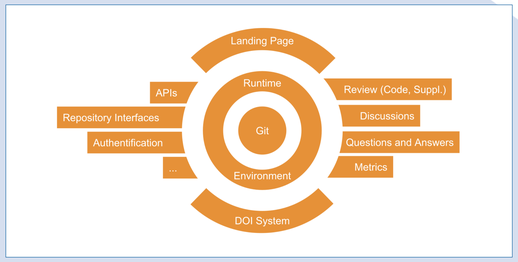

Fig.5: Concept for scientific software publishing platform

Central core of the platform is a distributed version control system that is based on established technologies in the software world and that provides functionalities relevant for software engineering, which can also be used to support the publication workflow.

Another key component is the landing page, which provides access to the software publication and required information and tools known from the publishing sector. In addition, the landing page acts as a project website with information and tools relevant for the use and development of software and for the communication as it is common in well-maintained software projects. The Landing Page is thus a special and very important component. It acts as a bridge between the software world and the publishing sector and blends all relevant information and tools from both worlds. This includes appropriate references to the corresponding papers and research data already published.

Furthermore, the connection to the DOI system as interface to the publishing sector is a major component. It ensures the minting of DOIs and guarantees agreed conformity to defined processes and exchanged information in the publishing sector. Relevant information must be stored and managed with standardized metadata and defined interfaces so that information is exchanged and propagated and can be accessed and processed by automated systems in the publishing sector and the software world.

www.sciforge-project.org
www.gfz-potsdam.de

sciforge

Acknowledgements
This work is funded by DFG through the project SciForge