Lessons Learned

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 availability of scientific software have to fulfill 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).

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.

• Published software processes and proven software engineering practices
• Skills, policies and infrastructures and platforms

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).

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 representation of sources. The use of best practices in software engineering will help to prepare scientific software for publishing. By applying these principles components 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 includes the conception and implementation 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.

State of the Art

Scientific software is a major part of the publication of scientific papers. Its development is part of the workflow of scientific research and its publication is an important part of research dissemination. While software engineering is used in industrial and enterprise environments, it is not commonly used in research environments. In the scientific community, the publication of software is seen as a separate and additional task to the publication of research papers. This has led to a lack of recognition and reproducibility of software engineering in research. The publication of software is often overlooked and not properly documented, leading to a lack of visibility and reproducibility.

The State of the Art in scientific software publication is characterized by the following challenges:

• 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 availability of scientific software have to fulfill 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).

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.

• Published software processes and proven software engineering practices
• Skills, policies and infrastructures and platforms

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).

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 representation of sources. The use of best practices in software engineering will help to prepare scientific software for publishing. By applying these principles components 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 includes the conception and implementation 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.