“Software and source codes” College: Minutes from the work session, November 30th 2023

Roberto Di Cosmo, François Pellegrini, Mathieu Giraud, Morane Gruenpeter, Daniel Le Berre, Violaine Louvet, Sabrina Granger

Ouvrir la science !

“Software and source codes” College: Minutes from the work session, November 30th 2023

Roberto DI COSMO
(Coordination)
François PELLEGRINI
Mathieu GIRAUD
Morane GRUENPETER
Daniel LE BERRE
Violaine LOUVET

_____

November 30th 2023

“Software and source codes” College: Minutes from the work session, November 30th 2023

DOI : 10.52949/58 | Associated document: Highlights of the "Software Pillar of Open Science" (SPOS) workshop DOI : 10.52949/53

Roberto DI COSMO (coordinator)
Inria and Paris Cité University, director of Software Heritage
François PELLEGRINI
Inria, University of Bordeaux
Mathieu GIRAUD
CNRS, University of Lille
Morane GRUENPETER
Inria, Software Heritage
Daniel LE BERRE
University of Artois, CNRS
Violaine LOUVET
CNRS, Grenoble Alpes University

Minutes editors

Sabrina GRANGER
Inria, Software Heritage
François PELLEGRINI
Inria, University of Bordeaux

November 2023

Conception graphique : opixido

“Software and source codes” College: Minutes from the Work Session 2

Abstract

Academia has increasingly recognized software as an essential tool, product, and subject of study. In this regard, the "Software, Pillar of Open Science" (SPOS) workshop, held on November 29th, 2023, was a significant event organized by the French Committee for Open Science.

This workshop brought together a large array of diverse high-level stakeholders, including public and private funders and infrastructure representatives, along with researchers and software engineers. The event was structured to facilitate a deeper understanding of the complex interplay between software and open science. The SPOS 2023 workshop addressed several pivotal themes:

- Recognizing and Supporting Software Dissemination: This session, chaired by Mathieu Giraud (CNRS, University of Lille) and Violaine Louvet (CNRS, Grenoble Alpes University), delved into the importance of career recognition and making research software contributions visible. The panellists explored methodologies for cataloguing research software, monitoring its evolution, and recognizing software contributions in academic careers.
- Software's Role in Research Reproducibility: Chaired by Nicolas Rougier (Inria, University of Bordeaux, CNRS), this panel focused on the crucial contribution of scientific software to research reproducibility, discussing challenges, opportunities, and best practices in the field.
- Social Impact and Sustainability of Publicly Funded Research Software: Led by Daniel Le Berre (University of Artois, CNRS), this session explored the broader impact and sustainability of research software from financial, organisational, and technical perspectives.

All along the workshop, the distinguished speakers and panellists from various sectors contributed to rich insights, sharing experiences and viewpoints on the multiple dimensions of research software in open science.

Following the impactful discussions of the SPOS 2023 workshop, a private meeting was convened the day after, on November 30th, to tackle these issues more deeply. Held at the Ministry of Higher Education Research and Innovation in Paris, this meeting was structured around four breakout sessions focusing on key areas:

- Catalogues of research software
- Sustainability
- Software forges

- Recognition of software production in academic careers.

These sessions enabled an extensive exploration of the stakes raised during the workshop. The more informal format eased nuanced discussions and collaborative brainstorming among experts in the field.

This report, focusing on the private meeting, aims to capture the essence and outcomes of these breakout sessions, providing insights into the future direction of software in open science.

Table of contents

1 | The genesis of the “Software and source codes” College

The welcome address of the “Software Pillar of Open Science” workshop was given by Prof. Roberto Di Cosmo, co-chair of the “Software and source codes” College. This college is one of the five permanent bodies[1] of the French national Committee for Open Science. The presentation summarized the key milestones that led the Committee for Open Science to create the “Source Codes and Software” College from the “Free and open source software” working group which was initiated in 2018 as part of the first French national plan for Open Science. Three years later, this experts group became a full-fledged College, following the publication the second French national plan for Open Science (2021).

The expansion of the scope of action of the initial working group and its sustainability result from the place granted to software in the second national plan for open science that puts software, publications and data on the same footing. In this document, Open Source is considered as the best choice for Open Science. This framework document mentions that each research software produced in France should be archived and referenced in Software Heritage, and described in HAL, the national open access repository.

The Software College focuses on “research software” defined as follows:

“Research software is developed to meet specific scientific needs. It is designed, maintained, and used by scientists (researchers and engineers) and research institutions, possibly on an international scale. It can result from research work as well as support it, notably through publications before/on/around/with the software. It can be formalised in different ways (a platform, a middleware, a workflow or a library, a script, a module or a plugin for another software) and thus be in interaction in an ecosystem or on the contrary be more autonomous.”

While the process of recognizing software as an essential element of scientific production is well underway, it is important to remember that it builds on a long history of sharing and has strong ties with community of practices developed in the context of free software. The free and open-source approach is intrinsically intertwined with the need for openness in science and the associated digital commons, in order to ensure reproducibility but also to encourage contributions and the emergence of community software.

All the dimensions of the complexity of this object are taken into account in the work of the “Source codes and software” college, which includes members of the free and open-source movement who have been active for decades, and will benefit from the feedback of the whole scientific community.

[1] List of the colleges: “Publications” ; “Research data” ; “European and international relations” ; “Skills and training” ; “Software and source codes”.

To this end, the college is currently divided into five working groups, each of which focusing on the complexity of software practices in research from a different angle:

- Identifying and highlighting software production in higher education and research
- Tools and technical and social best practices
- Valorization and sustainability
- Liaison and national, European, and international outreach
- Recognition and careers

Since software is intrinsically international, as part of its mission, the Software College strives to make its productions available to a large international audience, by systematically translating them into English.

Key resources available online and shared with the group are detailed below.

Framework of the Software College in the National Plan for Open Science: key documents

Second French Plan for Open Science

- https://www.ouvrirlascience.fr/second-national-plan-for-open-science/

The "Software and source code" college

- https://www.ouvrirlascience.fr/software-and-source-codes-college/
- Blog post about the college mission and subgroups: https://www.ouvrirlascience.fr/research-software-as-a-pillar-of-open-science/

Productions of the "Software and source codes" College

- Software forges in higher education: https://www.ouvrirlascience.fr/higher-education-and-research-forges-in-france-definition-uses-limitations-encountered-and-needs-analysis/
- Opportunity notes on software in Higher Education and Research: https://www.ouvrirlascience.fr/opportunity-note-encouraging-a-wider-usage-of-software-derived-from-research/
- Introduction to source code and software for Higher Education: https://www.ouvrirlascience.fr/source-code-and-software/
- Educational resource from the Committee for Open Science: https://www.ouvrirlascience.fr/passport-for-open-science-a-practical-guide-for-phd-students/
- How to archive and reference source code: https://www.softwareheritage.org/howto-archive-and-reference-your-code/

More resources

- Data, source code and algorithms roadmap for the French Ministry of Higher Education and Research (in French only): https://www.ouvrirlascience.fr/politique-des-donnees-des-algorithmes-et-des-codes-sources-feuille-de-route-2021-2024/
- News from the Open Science Committee: https://www.ouvrirlascience.fr/category/news/blog-en/

Getting in touch with the "Software and source Codes" College

- Submit a proposal, ask a question: coso-logiciels-contact@groupes.renater.fr

2 | Report from the breakout sessions

2.1 | Breakout session 1: Catalogues of Research Software

Coordinators: Violaine Louvet (CNRS, Grenoble Alpes University), Morane Gruenpeter (Inria, Software Heritage)

The topic "Catalogues of research software" covers different aspects, depending on the instigator, the goal and the target audience of the catalogue. The first goal is to build a state-of-the-art review of what exists in this domain. In the breakout session, the group discussed preliminary work done in the Software College and how to go further in the context of the international ecosystem of open science.

The first issue is to identify what is expected from a research software catalogue from the points of view of the different stakeholders: researchers, institutions, funders, other audience. For instance, the researchers' main expectation is to be able to easily find software adapted to their needs in a centralised location. While for institutions and research units, a research software catalogue is primarily a way to promote and to make visible this scientific production. It is also a communication tool and somewhat like a marketing tool.

An important issue is then to identify the limit of what should be a software listed in the catalogue. In other terms, how to define the types of software that should be entered in the catalogue? The "Software and source code" college disseminated a survey to identify what should appear in the catalogue. Some information is particularly important, for example descriptive metadata, community users' feedback, but also development status and some statistics.

Two key issues can be identified:

- How to ensure catalogue data quality?
- How to manage the completeness stake while fixing the limit of the type of software to be listed?

A moderation process and a badge system can help dealing with catalogue data quality. Having a curated repository may provide accurate metrics for institutions and laboratories.

The completeness stake is a much more complex issue: how to manage the contributions and to have a clear idea of what must be listed in the catalogue? The question is open and is also linked to maturity and the use of the software.

The participants highlighted two other concerns: how to support researchers, to help them describing their software and how to train the different stakeholders to improve the quality of software. Implementing a tool, even based upon automated workflows, calls for some end-users' support at a national scale. For instance, the data management service support clusters that have been deployed in France, via the "Ateliers de la donnée", may provide an example of organization. These data management clusters are geographically close to research teams so they can provide researchers with initial expertise in rational research data management.

After establishing all these points, the group discussed the existing infrastructures which can be used in order to build the catalogue. The starting point was that updated information can be found in the forge used for the development. Then the natural process will be to use the Software Heritage infrastructure and the Codemeta metadata schema.

In France, there is a strong link between Software Heritage and the open archive HAL. HAL can be used to enrich the Software Heritage collections, via two important features: a complete description (based upon an automatic harvesting of metadata from the Codemeta files) and a manual moderation process, which answers one of the issues above. Further information about the software deposit in HAL is available.

This joint venture between HAL and Software Heritage raised the interoperability question at an international level: is such a workflow implementable with other platforms from other countries?

Finally, the group discussed the links with other databases and how to exploit existing data, for example from HCERES - which is the institution for the evaluation of research in France - or ANR - which is one of the funders of research in France. The French open science barometer is also part of this technical landscape.

Next steps

- 1. Publish and disseminate a state-of-the-art review, possibly after opening it to community review.
- 2. Provide recommendations for a Research Software catalogue. The finalisation of this work can be the opportunity to organize a workshop to gather international community input.
- Pre-requisite of step 2: To be able to write the recommendations, it is necessary to clarify the workflow for the catalogue.
- 3. In parallel, work on how to organize the support, how to train staff (often librarians, not only research software engineers) to help with software referencing (best practices, metadata, curation, ...).
- 4. Explore collaboration with the SciCodes consortium, to share what is done at an international level.

2.2 | Breakout session 2: Software Sustainability

Coordinator: Roberto Di Cosmo (Inria, Paris Cité University, Software Heritage)

The term "sustainability", when one speaks of software, covers a broad spectrum of issues that involve technical, organisational, legal and funding aspects, among others. In this breakout session, the group explicitly focused on the financial aspects of sustainability. We note here that we focus specifically on *sustainability of existing projects*, and not on funding of new ones, which is more straightforward to address in usual R&D funding schemas.

Even when focusing on financial aspects only, there are a plurality of issues to address, and we list here the key ones put forward by the participants:

1. How should one make sure that funding is available to support software?
2. Once funding is available, how to decide what software projects should be funded?
3. Once the funding is available, and the projects to support are chosen, how can the financial support be actually implemented?

To address the first question, one approach we see explored by funders, both private and public, is to reserve money in the R&D budget they manage for software maintenance and evolution, and not just for software creation. This can be done in a variety of ways, for example:

- Set up *granting schemas specifically dedicated* to support project maintenance and evolution
- Reserve a percentage (e.g. 5%) *of the overall budget available to the funder* to fund software project maintenance and evolution
- Reserve a percentage (e.g. 5%) *in the budget of each grant* for the grantee to fund software projects maintenance and evolution

Whatever the approach taken to provision funding, it will hardly be sufficient to fund all possible software projects, which means it is necessary to decide *which* open source software project to fund (or not, implicitely), and *how*.

The participants mentioned several approaches that they are aware of:

Fund maintenance of projects whose development has already been funded by the funder.	Netherlands eScience Center Call for Sustainable Software 2023: https://www.esciencecenter.nl/calls-for-proposals/call-for-sustainable-software-2023-ss-2023/ UKRI Software for research communities: https://www.ukri.org/opportunity/software-for-research-communities/

	DFG Call for proposals Research Software Sustainability: https://www.dfg.de/resource/blob/172676/861b6b46e50aef649ee6634c7a2d0b6a/161026-dfg-call-proposal-software-en-data.pdf DFG Call for Proposals to Increase the Usability of Existing Research Software: https://www.dfg.de/en/news/news-topics/announcements-proposals/2022/info-wissenschaft-22-85
Come up with some metrics to find "critical" open-source projects, and focus funding on some of those that qualify as such[2].	A typical approach to defining "critical" software is to estimate *the number of other projects that depend on it*, or *the size of its user community*, approximated by the number of mentions in the literature, or other measures of popularity for a software project (stars, forks, downloads, etc.) Scale is often related to importance, but not always. Something only used in one application may also be critical, depending on the application.

Finally, once the funding is secured and the project is chosen, one is left with the question of how to actually provide the funds in a way that they can be efficiently used by the project. Here too there are a variety of paths that can be followed, depending on the maturity of the project and the way it is structured:

- Fiscal sponsors
- Non-profit foundations
- Hosting institution
- For-profit companies

Following a question he was asked during the last panel of SPOS 2023, Di Cosmo stressed the importance of exploring *yet another way* of enabling software sustainability: *empowering users to directly fund open source software they depend on*, using a small fraction of their own funds to support them directly. The key issue at stake is that, *while buying a licence for a proprietary software is today an administrative no-brainer* in any laboratory, *sending funds to an open-source project is usually an administrative nightmare*. Spending 100$ or even 1000$ on a proprietary software licence is a trivial action: there is a dedicated budget line for this kind of expenditure, and the procedure is well known. On the contrary, sending money to an open-source project usually involves either going through a "donation", which is simply forbidden for research organisations or universities using public money, or becoming a member of a consortium, which involves paperwork, legal scrutiny and long delays. Nobody wants to go through all that hassle for such small amounts, so the end result is that what would be a very effective way for creating a stream of small amounts from large user bases, largely sufficient to maintain the software they depend on, is actually prevented from materialising.

[2] A typical example is the CII (Critical Internet Infrastructure) fund set up by the Linux Foundation after the HeartBleed bug revealed that a key cryptographic library was maintained by just one person, in his free time, shocking the world.

The importance of removing the barriers to such a flow of small amounts from large user bases cannot be underestimated: several popular software projects that need funding for their sustainability have user bases that can be counted in the hundreds of thousand, when not millions, so if even a fraction of their users could easily target a fraction of their research funds to their support, there would be no need to go searching for emergency funding from private or public funders, *which would then be free to focus their efforts on projects that may be less popular, and yet quite critica*l. The case of Zotero, reported by M. Greenberg, who had first-hand experience on it, is a tell-tale: a well "architectured" approach allowed it to propose a subscription schema that fits the usual budget regulations, unleashing the stream of funding needed to maintain it.

This is a significant "dead angle" in the current approaches to fund open-source research software, that the group believes deserves an in-depth discussion, in the framework of a dedicated workshop.

Next step

- Organise a workshop dedicated to funding open-source research software, bringing together a large panel of stakeholders: representatives of a broad range of research projects with hands-on experience of the issue, experts in procurement/budget spending, representatives of private and public funders, representatives of Open Source foundations, representatives of research performing organisations (universities).

2.3 | Breakout session 3: Software Forges

Coordinator: Daniel Le Berre (University of Artois, CNRS)

The report on the forges used in higher education and research in France revealed that there were numerous isolated public forges: 67 public forges are currently listed in the report, 58 of them being instances of GitLab. The reasons for the existence of those numerous institutional forges include sovereignty concerns in the early stages of software development, the ability to tailor the tool to its own needs and the desire to stamp the software production.

The main issue with those public forges is the lack of interaction with people outside their hosting institution. Indeed, an account on those forges is needed to fully interact with software projects: raising issues, providing feedback on features, getting or providing help, contributing code or documentation.

The institutional forges in France generally do not allow anybody to create a full developer account because they want to restrict the projects publicly available to their own: they do not want to host the software project of individuals unrelated to the institution. Two examples of the consequences of that situation were discussed.

First, software journals such as the *Journal of Open Source Software* (JOSS) explicitly exclude the submission of software hosted on a forge which is not open to anybody for opening issues and contributing code. It means that most open-source research software hosted on French institutional forges cannot be published by and promoted through JOSS.

Second, the Coriolis research software awarded during the first open science open-source research software prize in 2022 had to move from its institutional forge to GitHub during summer 2023 to accommodate its growing community.

Most visible international open-source research software is hosted on the commercial forge GitHub, which is perfectly fine for those projects. The need for an institutional forge is more important at the early stage of development, before the research results are published. Many open-source research software is not developed by software engineers, so an institutional forge for growing research software may not need all the professional features found in commercial forges. However, it is important that their ergonomics, the user experience, is close to the one of commercial forges.

A way to solve the above issues is to use a federation of forges, the ForgeFed protocol for instance. In principle, it allows all instances of the forges which implement that protocol to communicate, i.e. it allows any user of any forge to raise issues or contribute code to another federated forge. There are currently two forge software implementing the federation of forges: the reference implementation of ForgeFed is provided by Forgejo, while a community effort has started to implement ForgeFed in GitLab as well.

Next step

- Organize an event at the national level to bring together all the stakeholders

2.4 | Breakout session 4: Recognition of Software Production in Academic Careers

Coordinators: Mathieu Giraud (CNRS, University of Lille), Sophie Renaudin (Assistance Publique Hôpitaux de Paris)

The goal of this session was to think about how to recognize software production in researchers' and engineers' careers, and how policies at the institutional level can support such production.

Individual and group/team evaluation reports should contain dedicated fields for contributions to research software. These contributions can take various forms: design of algorithms, coding, documentation, community management, etc. Hence, if a dedicated section exists for evaluating these fields, there should be some form of normalization, so that evaluators are aided in their assessment work.

Metrics are double-faced. On the one hand, evaluation is based on the possibility to evaluate work intensity and contribution quality. On the other hand, purely quantitative evaluation can be misleading (as it is for research publications). For instance, the number of downloads is not a relevant metric, as many mature software benefit from indirect redistribution through third-party packaging (e.g. as Linux distribution packages, embedding into third-party software, etc.), which may underestimate downloads, while in some cases continuous integration processes may inflate them.

A way to evidence software production is to have a registry. However, this registry must be curated in some way, else it can be flooded with irrelevant/outdated data connected to pieces of software of low quality. Hence, registries must be somewhat curated, in order to be of relevant "quality" for researcher's evaluation. It is e.g. the case for Inria, with the BIL, which is curated by the innovation and valuation departments of each research centre. Registries should not be mandatory for evaluation: even when registries do not exist, researchers should be recognized. This is also the case when researchers contribute to an existing software, which is declared in a "foreign" registry. Also, recent software may not yet be included in registries that require some level of maturity for inclusion.

Resources

- Inria criteria for software self-assessment
- Policies in Research Organisations for Research Software (PRO4RS) Working Group, ReSA task force on research software authorship and contribution https://www.researchsoft.org/tf-authorship-contribution/
- Amsterdam Declaration on Funding Research Software Sustainability Toolkit - addressing recommendations including:
- 7. Funders should stimulate the training, hiring, and funding of both professional research and technical staff able to reuse, develop, and maintain sustainable research software.
- 8. Funders should facilitate appropriate reward and recognition measures that enable career progression for all people involved in the creation and maintenance of research software.
- 9. Funders should require citation practices for research software that recognise substantial contributors to all aspects of the software.
- Encouraging entry, retention, diversity, and inclusion in research software careers (Barker & Katz, 2022)
- FORCE11 software citation principles (Arfon, Katz, & Niemeyer, 2016)
- Senior level RSE career paths (with an s) (Katz et al., 2021)
- Foundational Competencies and Responsibilities of a Research Software Engineer (Goth et al.)
- US-RSE and the IEEE Computer Society, Research Software Engineers: Creating a Career Path—and a Career

Next Step

- Produce a short document drafting / outlining policies for organizational recognition of software contributions.

3 | Summary of finding and next steps

3.1 | Conclusion

- The meeting served as a trigger for more collaborative time and marked the beginning of a significant journey
- Emphasis on writing a comprehensive report to document the event and its outcomes
- Highlighted the interconnectedness of various subjects like sustainability, funding, recognition, tools, forges, and catalogues
- Recognition of the progress made by the college, with a general assessment that its work is ground-breaking and far ahead of most that is known to the external participants

3.2 | Feedback Around the Table

- Acknowledgment of the complex, yet constructive, challenges faced
- Appreciation for the work done by the college, especially in terms of openness
- Concerns about promoting contributions to open-source software not specifically created in research and its recognition in career advancements
- Notable that the prize was awarded to software development with significant female representation, a step towards diverse acknowledgment
- Discussions on the challenges of internationalising the French model, especially in the context of the EU and the ReSA funder forum
- Suggestions for better representation of France in international forums and improving communication channels
- Emphasis on fostering international collaborations and mutualizing efforts
- The need to produce tangible results to promote practical solutions at an international level, furthering the conversation on known problems

Appendix

Breakout sessions: composition

Catalog

- Boulet, Pierre
- Chuffart, Florian
- Gruenpeter, Morane (coordinator)
- Jeangirard, Eric
- Louvet, Violaine (coordinator)
- Marmo, Chiara

Sustainability

- Andres, Aurelia
- Claudio, Almudena
- Cruz, Maria
- Di Cosmo, Roberto (coordinator)
- Greenberg, Josh
- Katz, Daniel
- Taraborelli, Dario

Forges

- Chaput, Jean-Paul
- Gérard, Sébastien
- Giraud, Timothée
- Jeannas, Jean-Yves
- Le Berre, Daniel (coordinator)

- Pop, Sorina
- Roesch, Etienne

Career and Recognition

- Barker, Michelle
- Clément-Fontaine, Mélanie
- Giraud, Mathieu (coordinator)
- Monat, Raphaël
- Moreno, Juan-José
- Pellegrini, François
- Renaudin, Sophie
- Rougier, Nicolas

Detailed agenda of the Morning Work Session – November 30th 2023

Location: Ministry of Research, Paris

09:30 - 09:45: Welcome Address and Overview of the Software College in the CoSO

- Welcome address by the organizing committee.
- Overview of the Software College's role within the Committee for Open Science (CoSO)

09:45 - 10:10: Presentation of the Guests

- Introduction of international guests.
- Guests share insights on similar institutional structures in their home countries.

10:10 - 11:00: Thematic Working Groups

Participants will divide into thematic groups focusing on the following areas:

- Catalogs of research software
- Funding and sustainability of research software
- Strategic choice of software forges in research.
- Recognition of software production in academic careers

11:00 - 11:20: Coffee Break and Group Photo

- Networking and socializing over coffee
- Taking a group photo for the event's records

11:20 - 11:50: Restitution of the Working Groups

- Groups share outcomes of their discussions
- Exploration of future collaboration paths based on the group work

11:50 - 12:00: Closing of the Session

- Final remarks and summary of the morning's outcomes
- Transition to the optional lunch

12:00 onwards: Lunch at the Ministry’s Cafeteria

- Informal continuation of discussions
- Opportunity for further networking in a relaxed setting