Perceptions on the adoption of Free/Open Source Software policies by a Scientific Institution

Teresa Gomez-Diaz, Tomas Recio

Maple Transactions

Perceptions on the adoption of Free/Open Source Software policies by a Scientific Institution

The case study of the NIH

TERESA GOMEZ-DIAZ[*], University Gustave Eiffel-LIGM-CNRS, France
TOMAS RECIO[*], University Antonio de Nebrija, Spain

Abstract.

Background. As the Open Science context evolves broadly and Scientific Institutions implement Open Access and Open Science policies, it is of interest to observe which are the issues that come into play when these Institutions attempt to contribute to make their research outputs visible, accessible and reusable. The present work reflects on the problematic behind the establishment and adoption of policies regarding the dissemination of Research Software (RS) as Free/Open Source Software (FOSS) in a Scientific Institution. It takes advantage, as an initial motivation, of the *Request for Information (RFI): Best Practices for Sharing NIH Supported Research Software*. This request was issued by the USA National Institutes of Health (NIH) organization in October 2023, seeking responses to a set of questions regarding Research Software sharing and dissemination best practices.

Method. In this work we begin by establishing an initial, general framework regarding RS and FOSS fundamental concepts and terminology. Then we include the initial list of questions for the NIH-RFI, followed by our detailed answers, argued in the context of the provided framework.

Results. Besides presenting this large collection of answers, we also include diverse reflections on what we have observed and learned in the context of the analysis of the NIH-RFI, arguing that it also provides us with the opportunity to explore how the previously published work on RS and Open Science can be envisioned in the lens of the definition and adoption of RS sharing and disseminating policies by any Scientific Institution or research community.

Conclusions. We have detailed here a foundational framework, showing how it could be considered as the key tool for providing a systematic reply to the RS and FOSS issues arising in the NIH-RFI. This framework can be easily adapted to more general contexts, and, thus, could be of potential interest for other scientific organizations, as well as for scientific communities with an important RS production, like the Computer Algebra and the Symbolic Computation ones, aiming to clarify and upgrade their policies and practices regarding this kind of production.

Keywords. Open Science, research on research, research software, free/open source software, scientific policies, dissemination, evaluation.

Recommended Reference Format:
Teresa Gomez-Diaz and Tomas Recio. 2025. Perceptions on the adoption of Free/Open Source Software policies by a Scientific Institution: The case study of the NIH. *Maple Trans.* 5, 3, Article 22904 (July 2025), 26 pages. https://doi.org/10.5206/mt.v5i3.22904

[*]Both authors contributed equally to this research.

Authors' addresses: Teresa Gomez-Diaz, University Gustave Eiffel-LIGM-CNRS, Est of Paris, France, Teresa.Gomez-Diaz@univ-eiffel.fr; Tomas Recio, University Antonio de Nebrija, Madrid, Spain, trecio@nebrija.es.

https://doi.org/10.5206/mt.v5i3.22904

1 Introduction

We are now in times when the importance of software to the scientific enterprise is widely recognized, and, for example, USA science-funding agencies like the National Science Foundation (NSF), the National Institutes of Health (NIH), and others are increasingly funding software work in their research programs [8, 48, 50, 55, 61]. Much of this software complies with the Free/Open Source Software (FOSS) concept, a kind of software that has become a critical part of the modern economy [1, 18, 59], and thus, has also become a natural part of the scientific endeavor, as highlighted in the *NASA Funds Open-Source Software Underpinning Scientific Innovation* web page:[1]

> *The NASA science community [...] demonstrates the need for sustained support and maintenance of open-source software. These projects are integral to our missions, critical to our data infrastructure, underpin machine learning and data science tools, and are used by our researchers, every day, to advance science that protects our planet and broadens our understanding of the universe.*

More generally, currently, we are also in Open Science times, where scientific policies are progressively aiming to make scientific knowledge openly available, accessible and reusable for everyone, as well as to increase scientific collaborations and sharing of information for the benefits of science and society [76, 30].

For example, the European Union, has implemented Open Access and Open Science policies with their Horizon 2020 and Horizon Europe programs [17], and, likewise, the White House Office of Science and Technology Policy (OSTP) has recently upgraded their policy guidance to federal agencies with research and development expenditures on updating their public access policies [65, 61]. Other examples we would like to mention here correspond to the French Open Science policy [57], the Spanish Open Science policy [58], the UK Open Research policy [75] or the Open Access policies for scientific production in Latin America and the Caribbean and in the European Union [68]. Thus, we are in a context where research outputs, including Research Software (*RS* from now on) [29], are increasingly shared and disseminated in order to be rendered visible, accessible and reusable [30], and where organizations like the G7 Open Science Working Group (OSWG) do recommend *to deepen research-on-research on open science at an international level, and increase coordination and knowledge-sharing* [20].

As a consequence, research institutions are adopting Open Access policies, mainly related to the access and dissemination of publications and data, or sometimes embracing more broader Open Science policies. The next table (see Table 1) mentions several of these institutions and their websites dedicated to Open Science, without any claim for exhaustivity.

Table 1. Several scientific institutions have adopted Open Science policies.

Institution	Link
CERN (Switzerland)	https://openscience.cern/
CNRS (France)	https://www.science-ouverte.cnrs.fr/en/
CSIC (Spain)	https://www.csic.es/en/open-science
Helmholtz (Germany)	https://os.helmholtz.de/en/
NASA (USA)	https://science.nasa.gov/open-science/
NIH (USA)	https://osp.od.nih.gov/policies/public-access/

[1] https://www.nasa.gov/news-release/nasa-funds-open-source-software-underpinning-scientific-innovation/

Some of these organizations' Open Science policies do include recommendations regarding RS like NASA [60], CERN [10] or Helmhotz [56], or the already mentioned [57, 75]. Other initiatives may come from the organization of thematic sections, for example, Open Science in Hydrology [14].

The benefits of Open Access and Open Science policies extend beyond immediate cost savings. Among others, free/open access to scientific literature enables broader dissemination of knowledge, which can accelerate innovation across disciplines. For instance, healthcare professionals can access the latest medical research without financial or institutional constraints, improving patient care and efficiency [9, 67, 76], see also [30] in its Motivation section, and the references mentioned there, regarding this subject of the benefits of Open Science (and possible negative impacts) and [73, 69].

In this quickly evolving context, we would like to highlight that the USA biomedical research agency, the NIH, has been a driving force behind many of the recent innovations in science and technology that are improving the health of all humankind [63, 64]. Now, this organization has released a Request for Information (RFI) for Best Practices for Sharing NIH Supported Research Software (NIH, 2023) [62, 61] to study the adoption of policies to promote broad dissemination of NIH-funded and/or NIH-developed RS. The initial list of the NIH RFI questions is included in the Section 3. They provide a complete and thorough vision of the issues that appear in RS development, sharing, and dissemination landscape as understood by this institution. At the time of writing the present work, the NIH refers to the Frequently Asked Questions (FAQ) for *Best Practices for Sharing Research Software*[2] in their Data Science website section, and to the high level policy document [64], which includes, among other goals, a (third) goal to *Provide new opportunities in software, computational methods, and Artificial Intelligence*, with the objective 3-3 to *Support FAIR Software Sustainability*, mentioning, again, the best practices for software sharing FAQ that align with the FAIR software principles (see endnote 82), and that it is also included in Section 3.

The present work aims to reflect on the general problem behind the adoption of FOSS policies for RS [29] produced in Scientific Institutions. To this end, we consider and extend the study of the case of the NIH. Let us mention that, in order to timely contribute to the NIH RFI request, we did provide answers to a partial set of selected questions in the form of a preprint [42] that was sent to the NIH in January 2024, as it was, for us, the starting opportunity to observe how our previous work on RS and Open Science could be suitable to provide answers to the raised questions.

We acknowledge that there is a certain high self-citation rate in this paper, since our work on these issues has been developed during a long period of time (starting in 2006 for the first author of this work). Let us also add that both authors have long and wide experience in using and developing software, or in collaborating with software projects in our lengthy and diverse scientific careers, which has broadly influenced our vision in these issues.

Let us also remark that the NIH questions that appear while reflecting about Open Science policies, and, in particular, while regarding the sharing and dissemination practices of RS as FOSS, do happen also in a similar manner for every scientific organization wishing to clarify and upgrade their policies and best practices and, thus, the setting and the answers we propose in here can be adapted to more general or diverse frameworks, and to other scientific organizations. In particular, we consider they could also be of interest for scientific communities with an important RS production, like the Computer Algebra and the Symbolic Computation ones, where both authors have developed part of their scientific careers.

The present work is structured as follows: first, we begin by establishing an initial, general framework regarding RS and FOSS fundamental concepts and terminology in Section 2. Then, in Section 3, we include the initial list of questions of the NIH RFI, followed, in Section 4, by the

[2] https://datascience.nih.gov/tools-and-analytics/best-practices-for-sharing-research-software-faq

proposed answers, which will be completed at the Discussion and Conclusions Section, where we reflect on what we have observed and learned in the context of the NIH RFI.

2 Setting the context for the adoption of RS and FOSS policies

This Section is devoted to the setting of the initial framework that seems to us necessary when considering the establishment and adoption of policies regarding the sharing and dissemination of RS as FOSS. Notice that this initial context has a double fold, as we should refer to a scientific side and to the FOSS side. The scientific part refers to two of the concepts that we consider most important, that is, the ones of *Research Software (RS)* and *Research Team (RT)*. On the other side, we also revisit here briefly the concepts of *Free Software*, and *Open Source Software*. Both sides do raise legal issues, mainly regarding copyright and authorship rights in software as well as software licensing.

Let's start first with the scientific side of this context setting. The RS definition we provided in Section 2.1 of [29] is the following one:

> **Research Software** *is a well identified set of code that has been written by a (again, well identified) research team. It is software that has been built and used to produce a result published or disseminated in some article or scientific contribution. Each research software encloses a set (of files) that contains the source code and the compiled code. It can also include other elements as the documentation, specifications, use cases, a test suite, examples of input data and corresponding output data, and even preparatory material.*

Let us remark that in [29] it is included a thorough discussion and analysis that has driven us to propose this definition. As a conclusion of this RS definition, we have the following characteristics for this particular kind off research output:

- what is done: code, software as a well identified set of files,
- who does it: author(s), but also contributors and/or scientific expert(s),
- to make what: research, science, that is, there are associated scholar publications,
- and the most important point is the quality and correctness of the produced scientific results [53].

Other discussions and works related to the RS concept can be found, for example, at [4, 25, 35, 41, 46, 53, 54, 72] and references mentioned there.

We have also provided a definition of what Research Team means in [39]:

> **Research Team** *is a well identified set of persons that are involved in whatever ways to produce a result published or disseminated in some article or scientific contribution in the academic context.*

In the context of the present work, the academic contribution we refer to in this definition is limited to RS.

These two definitions are important in order to help Scientific Institutions to determine the perimeter of the software produced inside their limits, with at least a member of the RT having participated in the writing of a substantial part of the software. To better illustrate this issue we recall here the definition of *Commission software* [16]:

> *'Commission software' means*
> *(i) software for which the intellectual property rights are held by the Commission, on behalf of the Union, and*
> *(ii) software owned by a third party, available under an open source licence, and which has been modified by the Commission or by a third party at the Commission's request.*

Now, the next part of this Section deals with the FOSS side of this context. First of all let us recall here the *Free software* and the *Open Source Software* definitions. We think this is required because we still detect, regularly, some misunderstandings regarding these two concepts, mainly in the Academic context.

The concept of *Free Software* was given by R. M. Stallman and the Free Software Foundation (FSF) in 1985.

> *A program is free software*[3] *if the program's users have the four essential freedoms:*
>
> **Freedom 0** *The freedom to run the program as you wish, for any purpose.*
>
> **Freedom 1** *The freedom to study how the program works, and change it so it does your computing as you wish.*
>
> **Freedom 2** *The freedom to redistribute copies so you can help others.*
>
> **Freedom 3** *The freedom to distribute copies of your modified versions to others. By doing this you can give the whole community a chance to benefit from your changes.*
>
> Access to the source code is a precondition for Freedoms 1 and 3.

There are two characteristics related to this definition that we consider relevant on the context of the present work. The first one is that it is our understanding that it reflects well the scientific activities of study, modification and dissemination of software. The second one is of legal nature, as it is necessary to accompany the software with a license, in order to allow users and collaborators to verify the rights of use, copy, modification and redistribution.

In order to stress this last point, we recall here the definition of *reuse* from [16]:

> *'reuse' means the use of software by natural or legal persons, for commercial or non-commercial purposes, including the right for those users to use, study, copy, share and modify the software;*

Notice that licenses do not replace legal issues of copyright and authorship, rather they have a complementary role, as if there is no license, nor other legal mention, the default legal context that applies is *All rights reserved*, and means that no one can run, load, etc, the software under consideration (except the rightholders) [1, 18, 25, 26, 40]. We have studied the important role of licenses in Open Science in [30].

On the other hand, the concept of *Open Source Software* was defined by the Open Source Initiative (OSI) in 1998. As the OSI mentions,[4] Open source doesn't just mean access to the source code. We include here a selected part of the definition, please consult the OSI website for the comprehensive concept:

> *The distribution terms of open-source software must comply with the following criteria:*
>
> (1) *Free Redistribution*
> *The license shall not restrict any party from selling or giving away the software as a component of an aggregate software distribution containing programs from several different sources. The license shall not require a royalty or other fee for such sale.*
> (2) *Source Code*
> (3) *Derived Works*
> (4) *Integrity of The Author's Source Code*
> (5) *No Discrimination Against Persons or Groups*
> (6) *No Discrimination Against Fields of Endeavor*
> (7) *Distribution of License*
> (8) *License Must Not Be Specific to a Product*

[3] https://www.gnu.org/philosophy/free-sw.html
[4] http://www.opensource.org/docs/osd

(9) *License Must Not Restrict Other Software*
(10) *License Must Be Technology-Neutral*

Open Source Software respects *all* these conditions, and, similarly to the case of Free Software, a license is needed to verify these conditions. The source code should be available in Open Source Software, otherwise the software does not comply with these conditions.

As one can understand from the above two definitions, these two software movements (around the FSF and the OSI) correspond to very different philosophies, but are sometimes confluent in common goals as, for example, the quality of the produced software.

A computer program falls within the category of Free and/or Open Source Software if the license complies with the conditions established in these two definitions. Most of the licenses used in this kind of software make it both "Free" and "Open Source" so there is no real legal distinction between these two concepts, despite the philosophical aspects. However, many members of the software development community prefer not to adopt one or the other philosophy, this is the reason why the term *Free/Open Source Software* (or FOSS) is widely used. This is the terminology we adopt here, while respecting the original formulation of the NIH questions, usually referring to *open source software*.

Notice that software licenses and licensing information can be found at the FSF, [5] the OSI,[6] and the Software Package Data Exchange (SPDX).[7] In particular, the SPDX License List includes a list of commonly found licenses used in free and open or collaborative software, among others, with a standardized short identifier, the full name, the license text, and a canonical permanent URL for each license, in order to enable and facilitate efficient and reliable identification of such licenses.

For further information on FOSS related issues, please consult [1, 18, 26, 40, 59], for example.

Please note that the authors of the present document are not legal experts, in spite of the fact that they have acquired some basic legal notions in copyright and licensing issues [2, 25, 26, 40], mainly in the French, Spanish and European legal context. It is not our intention, nor the intention of the present document, to provide answers to these kind of legal questions. If you have some, please refer to the legal experts in your Head Institutions.

To end this Section, we would like to call attention to the word *open*. Yes, open is a tricky word that can hide complex issues. Does open correspond to Open Access as defined by the Budapest Open Access Initiative,[8] to an Open Science policy (which one?), to the concept of Open Source Software as defined by the OSI, or does it indicate just that the source code of a RS is available? In our opinion, the word open is largely used in many different contexts, with different meanings, and, every time, it should trigger, at least, the three following questions: which are the definitions, the policies and the licenses that may be hiding behind it.

3 The NIH list of questions

This Section includes, first, the initial list of questions by the NIH regarding RS sharing best practices, as they were displayed at the NIH RFI in 2023 [62, 42]. The detailed statement of these questions is included here for two reasons. First, because they provide a complete and thorough vision of the issues that appear in the RS development, sharing, and dissemination landscape. Secondly, to highlight that the context of this vision is very ample, and to provide answers to all the raised questions is too difficult and out of the scope of this reaction to the RFI. Nevertheless, we consider

[5] https://www.fsf.org/licensing/
[6] https://opensource.org/licenses/
[7] https://spdx.org/licenses/
[8] https://www.budapestopenaccessinitiative.org/

it to be of interest to show here how a research funding institution like the NIH has approached the matter of the establishment of a RS sharing policy as FOSS.

Thus, our contribution will select a partial set of these questions and propose some answers in Section 4, answers that are based on previously published work, referenced at the end of this document.

Item1 Comment on the current NIH Best Practices for Sharing Research Software.
- **I1.1** Why should I share software and code as "open source" software?
- **I1.2** How do I make software source code "open"?
- **I1.3** Why should I use a license when distributing code?
- **I1.4** How do I choose a license under which to release software developed as part of an NIH award?
- **I1.5** How can I make my software citable?
- **I1.6** How should I acknowledge NIH as the funder?
- **I1.7** Are there any restrictions I should consider in deciding whether to share the research software I develop?
- **I1.8** Can research software I have developed be allowed for use in medical practice or clinical settings?
- **I1.9** Do I have to check software developed for security vulnerabilities prior to sharing it?
- **I1.10** What metadata should be considered when sharing research software?
- **I1.11** To what extent should I include documentation for the software?
- **I1.12** Does NIH have any requirements or benchmarks for research software quality before releasing it?

Item2 Describe how, when, and where you share your research software. What, if any, resources for best practices do you rely upon to make your shared software open and reusable?

Item3 What existing standards or criteria do you use to evaluate the openness, FAIRness, quality, and/or security of the software you share or reuse?

Item4 Describe the collaborative settings in which you develop and share research software. Name communities or organizations, if any, you participate in that are actively promoting or developing software sharing best practices.

Item5 What factors influence your decision to share or reuse your research software (or not)? What technical, policy, financial, institutional, and/or social barriers to sharing or reuse of research software have you encountered?

Item6 Comment on your ability to reuse open-source research software developed by others. Describe factors used to determine whether to reuse existing research software or develop anew.

Item7 How can NIH support research software communities of practice to better aid development of best practices for sharing and reuse of high-quality research software?

Item8 Comment on any other topic which may be relevant for NIH to consider in enhancing the sharing of research software.

The following part of this Section is related to the Frequently Asked Questions for Best Practices for NIH Sharing Research Software[9] as displayed at the time of writing the present work. This FAQ includes the questions **I1.1** to **I1.12** related to **Item1** and the NIH provided answers.

Frequently Asked Questions

Introduction

[9] https://datascience.nih.gov/tools-and-analytics/best-practices-for-sharing-research-software-faq

The National Institutes of Health (NIH) promotes broad dissemination of research products including NIH-funded and/or NIH-developed research software. These FAQs offer best practices for sharing research software and source code, developed under research grants in any stage of development, in a free and open format. Releasing the software source code in an "open" manner means that you permit users to use, modify, and/or redistribute the code. The FAIR principles (findable, accessible, interoperable, and reusable) also provide a useful framework for better software management and sharing.

I1.1 ***Why should I share software and code as "open source" software?***
Openly sharing research software, as well as source and object code, provides transparency. It also serves the key objectives of rigor and reproducibility as part of responsible conduct of research. These objectives are consistent with existing NIH data sharing policies and guidance.
Sharing software and associated code allows for software sustainability as well as contributes to the advancement of science and fosters collaboration among researchers across institutions.

I1.2 ***How do I make software source code "open"?***
Making software and code "open" means sharing software and code in a way that allows researchers to use code, modify and redistribute it. This can be done by releasing software and code in an open-access and open-source format in an appropriate, unrestricted, publicly accessible repository with version control. Repositories that allow the provisioning of additional metadata and provide search tools and finding aids can enhance the FAIR-ness of software.
Software may be shared in multiple ways:

- *as source code or the executable version,*
- *as code libraries published to general or specific package distribution channels,*
- *as workflows or containers,*
- *as services and APIs.*

In many cases both the code and the packages or runtime may be shared via different repositories. Researchers should use funder specified/preferred repositories. In the absence of any guidance, researchers are encouraged to select a repository that is appropriate for the software and code generated from the research project that has long-term sustainability via social coding repositories that are open, revision-control, source-code management systems such as GitHub,[10] GitLab,[11] or Bitbucket.[12]
Code may also be registered in a community repository to enhance discoverability. Specialized software packages may also be distributed via package management and library utilities such as Conda[13] or Bioconductor.[14] Containers and workflows may be shared through repositories such as Dockstore.[15]

I1.3 ***Why should I use a license when distributing code?***
Open licenses facilitate and encourage the reuse of the code by clarifying and documenting the terms of how the software can be used, modified, and redistributed by others, and for what purposes. Open licenses also disclaim any liability for problems associated with

[10] https://github.com/
[11] https://about.gitlab.com/
[12] https://bitbucket.org/product/
[13] https://docs.conda.io/
[14] https://www.bioconductor.org/
[15] https://dockstore.org/

the software after the code has been released to the public. For those reusing and building on the work of others, it is important to review the license and usage requirements.

I1.4 ***How do I choose a license under which to release software developed as part of an NIH award?***
Follow guidance provided by your funder. If no specific guidance is provided, NIH suggests using one of the licenses approved by the Open Source Initiative (OSI).[16] *An OSI license makes using and contributing to research software easier because these licenses are well known and understood.*

I1.5 ***How can I make my software citable?***
Providing a reference citation to the released software will allow others to easily cite your work. Minting persistent identifiers or easily accessible URLs in stable locations will ensure credit to you as a developer and make it easier for the research community at large to discover the software.
Consider strategies that allow your software to be citable such as Zenodo's ability to assign unique Digital Object Identifiers (DOIs)[17] *(one type of persistent identifiers) for source code released in GitHub or adding a citation.cff file*[18] *to your GitHub repository. This may be especially important when specific software version references are required to reproduce a research result.*

I1.6 ***How should I acknowledge NIH as the funder?***
When acknowledging NIH support, provide the same information as you would in a publication and refer to the NIH guidance on Communicating and Acknowledging Federal Funding.[19] *Be sure to include this information in the software documentation, the license agreement, and the repository website or the disk/drive from which people download the software. Make sure to identify all contributors to the research software.*

I1.7 ***Are there any restrictions I should consider in deciding whether to share the research software I develop?***
NIH encourages inclusion of plans to share software developed under NIH grants as part of resource sharing plans. Release of research software, however may be restricted when:
- *Patent protection for the software is under consideration.*
- *The software code contains or may be used to obtain personally identifiable information.*
- *The software is associated with a medical device.*

NIH recommends consulting your institution's Technology Development and Transfer Office or Sponsored Projects Office as appropriate to determine whether you should restrict release of your research software.

I1.8 ***Can research software I have developed be allowed for use in medical practice or clinical settings?***
In most instances, software developed for research may not be permitted for clinical use or medical purposes. If the software is intended for use in a clinical or medical setting, the software may have to undergo appropriate scientific and regulatory reviews by the U.S. Food and Drug Administration (FDA). NIH also recommends consulting your institution's Technology Development and Transfer Office to seek appropriate guidance for sharing such software.

[16] https://opensource.org/licenses
[17] https://guides.github.com/activities/citable-code/
[18] https://docs.github.com/en/github/creating-cloning-and-archiving-repositories/creating-a-repository-on-github/about-citation-files
[19] https://grants.nih.gov/policy/federal-funding.htm

I1.9 ***Do I have to check software developed for security vulnerabilities prior to sharing it?***
It is encouraged to check software for vulnerabilities prior to sharing it. The SANS institute has published a list of the top 25 software errors. These lists could help guide you in review of the software source code. See http://www.sans.org/top25-software-errors/.

I1.10 ***What metadata should be considered when sharing research software?***
Providing rich metadata allows research software to be discoverable and reusable. Open metadata allows for exchanges between systems, and reusable metadata eliminates duplication of efforts. Metadata to include with software vary by the use cases—for example keywords and descriptions allows for discoverability of software, credit for academic software requires citation metadata, and research replicability requires software versions—often one or more of these are required. Consider metadata that allow software and code to be linked to publications, data sources, funding support, and other digital objects related to research. NIH also suggests considering metadata recommendation provided by repositories specialized to manage and share software when developing and releasing software.
Metadata to consider include, but not limited to are:

- *Title: [Name of software]*
- *Description: [Describe the purpose of the software]*
- *Persistent Identifier: [A unique persistent identified (PID) such as a digital object identifier (DOI) or accession number supports data discovery, reporting and assessment.]*
- *Software Language and Version/Standard: [For example, C++ ISO/IEC 14882:2020]*
- *Author(s): [Names of software developers and contributors]*
- *Grant Number: [In this format R01GM987654]*
- *Publications: [Persistent identifiers and citations for publications by your team related to this software or code.]*

I1.11 ***To what extent should I include documentation for the software?***
Documenting your code with sufficient detail allows other programmers the opportunity to update, extend, or execute your software application. Documenting the design and purpose of your code provides others with a better understanding of your code. Providing compute platform-specific details for how your code can be installed and used would allow other programmers to test and debug the code appropriately for reproducible usage.
Better Scientific Software[20] *is one of many information sources that provides guidance on code documentation practices.*

I1.12 ***Does NIH have any requirements or benchmarks for research software quality before releasing it?***
There are no NIH-wide standard requirements related to quality of research software. However, NIH encourages researchers to adopt best practices of research software engineering while developing software funded by NIH with the aim of enhancing the sustainability of the software. Consider checklists such as those developed by:

- *The Linux Foundation's Core Infrastructure Initiative*[21]
- *The Software Sustainability Institute*[22]

[20] https://bssw.io/items?topic=documentation
[21] https://www.bestpractices.dev/en
[22] https://www.software.ac.uk/

- *The US Research Software Sustainability Institute*[23] *is one of many information sources for software sustainability practices.*

Conclusion
Several resources are available for developing and sharing research grade software. Working groups such as Force 11,[24] *WSSPE,*[25] *The Society for Research Software Engineering,*[26] *RDA*[27] *and projects such as CodeMeta*[28] *and Metadata2020.*[29]
All provide guidance, tools, and access to practicing communities. NIH recommends considering these when developing and releasing software.
For specific questions on sharing research software developed under your NIH award, contact your assigned Program Officer or your institution's Sponsored Projects or Technology Development and Transfer Offices.
This page last reviewed on April 3, 2023.

4 Selected questions related to the NIH list, and proposed answers

Once we have settled the main concepts of RS, RT, FOSS in Section 2, we can rely on [21, 22, 31] sustained by [25, 29, 35, 43] and many of the other publications and works referenced here, in order to provide answers related to a partial set of the NIH list of questions presented in the previous Section 3. Thus, we revisit and complete here the provided answers given in [42]. We will refer to questions & answers from the NIH Frequently Asked Questions (also included in the second part of Section 3) whenever it is appropriate.

Please note that in this section we use **I** for Item, **Q** for Question, **A** for Answer.

Question1. What does sharing a RS mean? This question is related to item **I1.1**.

Answer1. Sharing RS means that the producer RT gives the RS to another person/team external to the project or that the team makes the RS available in some web page or a repository, cloud, etc.

Q2. Why should I share software and code as "open source" software? This question is related to item **I1.1**.

A2. At the very moment the RT gives the RS away, that is, the *sharing and dissemination step*, the legal conditions to use, study, copy, modify, and redistribute the software should be ensured, that is, it is necessary to establish the RS *sharing conditions*.

Notice that this *sharing and dissemination step* is an important one in the life cycle of a RS, or of any software [40]. As far as the software remains in the hands of the development team, there are usually no legal issues related to the rights of use, copy, modification or redistribution. But at the sharing step, these rights should be ensured with a license, as usually the goal is that the RS is used (studied, modified…) by other teams [48, 51]. This also means that the RS rightholders are clearly identified and have agreed on the license that goes with the RS.

We can observe the different contexts where the answer to this question can be settled. We did choose to respond in a legal and licensing context, on the other hand, the NIH has chosen a scientific context to answer this question, enhancing thus the key objectives of rigor and reproducibility

[23] http://urssi.us/
[24] https://www.force11.org/
[25] http://wssspe.researchcomputing.org.uk/
[26] https://society-rse.org/
[27] https://rd-alliance.org/
[28] https://github.com/codemeta/codemeta
[29] https://metadata2020.org/

as part of responsible conduct of research, see question & answer **I1.1** of the FAQ in Section 3. Nevertheless, as explained in [43]:

> *Over these legally protected actions stand the usual scientific actions, as researchers, as part of their daily activities, do use, contribute to, write, share and disseminate, modify, include and re-distribute RS components.*

We remark then that it is almost impossible to allow for reproducibility of the research results obtained with the RS or for fostering collaboration among researchers across institutions, without legally ensuring the mentioned rights with the corresponding license.

Q3. How do I make software source code "open"? Why should I use a license when distributing a RS?. This question is related to items **I1.2**, **I1.3**.

A3. To make the software source code open, it can just mean that you share the source code, but, usually, this is not enough, without, for example, a documentation, the scientific context, or many other elements that can be part of a RS, as defined in Section 2.

In order to ensure the legal conditions for use, study, copy, modify, and redistribute the RS, our work has been centered to accompany the RS with a FOSS license [2, 23, 24, 25, 31, 40], that is, to disseminate the RS as FOSS [1, 18, 59].

Other legal means do exists, like establishing collaboration contracts, which should be discussed with the legal experts of the RT Head Institutions and/or with their Technology Development and Transfer Offices.

Please note that sharing only the executable part of the RS provided with black boxes can hardly allow for the study or the reproducibility of the scientific results obtained with the RS, as no access to the source code hinders, and most of the times severely obstructs, the study of the functioning of the RS, see question & answer **I1.2** of the FAQ in Section 3. Moreover, the RS can not be qualified anymore neither as Free nor Open Source Software if the set of files that constitute the RS does not include the source code.

The license corresponds to the legal means to protect potential users, collaborators and authors of the RS. It can include *as is* clauses to deal with warranties (or rather the absence of warranties) about quality or features of the RS, as well as *reciprocity clauses* that should be respected. Users of FOSS, or of any other kind of software, should be well aware of the license that goes with the software, as well as to check the possible "as is" clauses.

We would like also to remark that, once the use of the RS is legally ensured with a FOSS license, it might be difficult, if not impossible, to restrict its use for determined purposes; see question & answer **I1.3** of the FAQ in Section 3.

Another question is *where to share the RS?*, and for its answer we refer to those provided by the NIH in **I1.2** of the FAQ in Section 3 and to the *Where?* RS conundrum question of Section 4 in [43].

Q4. How do I choose a license under which to release a RS?. This question is related to item **I1.4**.

A4. There are several kinds of arguments to choose a license, for example related to a specific legal context (USA law, French law, European law…). In France, the list of licenses that can be used to share your RS is given in a Décret.[30] The use of other licenses is possible, but it is necessary to file an application for approval with the French administration. The European Commission has decided in 2021 [16] the use of the European Union Public Licence (EUPL) for the *Commission software*, and lists the exceptions to this rule, and, as another example, the Spanish law mentions that the EUPL license should be procured for transferring and reusing computer applications, associated

[30] https://www.data.gouv.fr/fr/licences

documentation and any other information object whose intellectual property rights are owned by a Public Administration, without prejudice of other licenses that can guarantee the same rights [71].

Other reasons to choose some kind of license can be related to the reciprocity clauses that appear in the licenses of software (or other RS) components that have been included in your own RS. These clauses should be respected, see for example [1, 2, 18, 40] for more information.

The choice of licenses other than FOSS licenses can involve legal issues that should be considered with the help of the Technology Development and Transfer Offices of the Head Institutions, see also question & answer **I1.7** of the FAQ in Section 3.

Q5. How can I make my software citable? This question is related to item **I1.5**.

A5. Among many other works on software citation issues, we have proposed in Section 2.5 of [29] three possible ways to cite your RS that take into account practices that are already well established in some research communities (see for example [51]):

- *the reference to a research software paper or other kind of scientific publication that includes, and relies on, a software peer review procedure, or*
- *the reference to a standard research article that includes a description of the RS and the implemented algorithms, explaining motivations, goals and results, or*
- *a typical label, associated to the RS itself, and that identifies it as a research output, specifying its title, authors, version, date, and the place the software can be recovered from.*

See also Section 2.5 of [29] and Section 5 of [39] for further discussion on citation and referencing issues.

As mentioned in question & answer **I1.5** of the FAQ in Section 3, the use of CITATION.cff files[31] is becoming a popular tool to deal with citation issues, the Citation File Format schema is maintained[32] by S. Druskat at the German Aerospace Center (DLR) and J. H. Spaaks at the Netherlands eScience Center.

Q6. Can RS I have developed be allowed for use in medical practice or in clinical settings? This question is related to item **I1.8**.

A6. The conditions in which your RS is to be used in medical practice or clinical settings should be carefully considered by the RT and clearly stated in its documentation, the website of the project... It should also be carefully studied by the user team, and in particular, users should be well aware of clauses *as is* specified in the RS license.

In particular, and as the question & answer **I1.8** of the FAQ in Section 3 emphasizes, the software may have to undergo appropriate scientific and regulatory reviews.

Q7. What metadata should be considered when sharing RS?. This question is related to item **I1.10**.

A7. As presented in Section 2.5 of [29] (references have been revisited and updated, and they do refer to the list included at the end of the present document):

> *A more complex way for RS identification than a citation form is the use of metadata sets. The Software Citation Implementation Working Group has worked over several possibilities for software metadata sets*[33] *(see also* [51] *and the references mentioned there). The PRESOFT (Preservation for REsearch SOFTware) metadata set proposed for RS in* [27]

[31] https://citation-file-format.github.io/
[32] https://github.com/citation-file-format/citation-file-format/blob/main/README.md
[33] https://force11.org/groups/software-citation-implementation-working-group/

is built over the skeleton of the RS description cards that where published between 2008 and 2013 by the PLUME project [28]*. This metadata set benefits from the PLUME experience, which validates the proposed model, and sets a reasonable level of standards to manage RS identification.*

The metadata set proposed in Section 1 of [27] is the following:

(1) Software name. If you need to choose a name, avoid the name of a brand and other software names.
(2) Short software description. A short sentence describing your software.
(3) Software web page or website.
(4) Link to source code or package. Consider Persistent Identifies (PIDs).
(5) Contact. Email address.
(6) Research unit in charge of the software.
(7) Main developers and their affiliations.
(8) Software version.
(9) Date of the software version.
(10) License.
(11) Scientific discipline. Classifications may follow ACM Computing Classification System[34] or more general classifications.[35]
(12) Main functionalities. Give for example some Keywords.
(13) Main technical characteristics. Give for example some Keywords.
(14) Other keywords.

Question & answer **I1.10** of the FAQ in Section 3 shows the Metadata set proposed by the NIH.

Q8. To what extent should I include documentation for the software? This question is related to item **I1.11**. The answer given here includes the question of the RS maintenance and support, and the possible lack of resources for these tasks are related to items **I1.7** and **I5** (see also *A12*).

A8. Our vision on this important issue of documentation is that it can have several levels. The more basic level is for the RT to explain summarily the main functionalities of the RS, and to provide basic examples of use, with input and output files. The use examples give hints about how the software is to be launched, how to write an input file, and which is the kind of output that users are to expect. The idea is to give basic information about what the RS is able to do, and the examples can be easily modified by potential users to understand how to launch and use the software, and how to write their own use examples.

Similarly, the RT can indicate the basic support and maintenance that could be provided, for example to explain the scientific context of the RS, or to correct some bugs. This can be done with the intention to detect possible collaborators.

But tasks related to documentation, maintenance and support ask for time and resources [48, 55]. Large development RTs may have the necessary resources to build and give sound documentation for users and for the potential future development teams, as well as to ensure further maintenance, support and other developments.

One of the benefits of releasing RS as FOSS is that other teams can contribute to improve the RS, in order to take in charge the sharing of some responsibilities like documentation or maintenance, to check for vulnerabilities or to write bug corrections. They can even fork the initial development in order to provide for further work or to ensure sustainability [18, 40].

[34] https://dl.acm.org/ccs
[35] https://erc.europa.eu/sites/default/files/2023-03/ERC_panel_structure_2024_calls.pdf, https://en.wikipedia.org/wiki/Outline_of_academic_disciplines

But the users, the RT and potential future development teams should be well aware that the main goal of a RT providing a RS is, usually, the research work, and maybe not to provide sound maintenance, support or documentation during months or years after the RS has been released: the RT may not have the needed support or resources (human, financial…) to deal correctly with these tasks at short or long term, with members of the team leaving for other works or projects, or maybe the RT will prefer to engage in new, different scientific projects. The RT should carefully consider the level of documentation, support and/or maintenance that is to be provided and clearly state the sharing and dissemination conditions of the RS in the related documentation, websites, README files…

Finally, the lack of RT resources for providing further documentation, maintenance and/or support can influence the RT decision to share and disseminate a RS as well as influence potential users' decisions.

Question & answer **I1.11** of the FAQ in Section 3 shows the NIH recommendations on the documentation issue.

Q9. Describe how, when, and where you share your RS. This question is related to item **I2**.

A9. These three questions: How? When? and Where? are part of what it is known as the Kipling method[36] (or the 5Ws+1H method in management practices that can be applied in different contexts [47]).

We have studied these kind of questions while providing answers to the Borgman's conundrum challenges [7] that were initially formulated concerning the difficulties to share Research Data (RD). They are well known in the context of Open Science: *which RD might be shared*, *by whom*, *with whom*, *under what conditions*, *why*, and *to what effects*. We have provided answers to the Borgman's conundrum challenges in [35] and we have also studied these questions as part of the *conundrum challenges* for RS in [43]. We include here a short summary as presented in [44], but see [43] for the whole discussion.

Which RS is to be shared? This is a RT decision about the RS that is to be shared, which version, in which form, and when. This may be a complex decision, that determines which components or which versions are to be shared, to share early versions or not, or the experimental branches in the development…

By whom? The RS production RT who is involved in the development, its documentation, its maintenance… and that has decided to share and disseminate the RS.

How? Following a dissemination procedure like the one proposed here in *A10*.

Where? RS can be shared in repositories like Zenodo, in forges like GitHub, or in institutional repositories, in web pages (personal, project…)

With whom? Each scientific communication act has its own target public, initially it can be one of the associated publications, but maybe there is some interdisciplinary value, so [7] (see also [48])…

> *intended users may vary from researchers within a narrow specialty to the general public.*

Under what conditions? The license gives the RS sharing conditions.

Why, and to what effects? To answer funding demands (Institution, project funding…), to follow Open Science policies and/or best practices, for the validation and the reproduction of published results or in relation with some research evaluation context…

[36] Or the *Six Honest Serving Men* that appear in the *The Elephant's Child*, by Rudyard Kipling (1902), https://www.kiplingsociety.co.uk/poem/poems_serving.htm is part of *Just so stories for little children*, available at https://www.studyassistant.org/wp-content/uploads/2021/05/Just-So-Stories-by-Rudyard-Kipling.pdf

Q10. Which best practices do you rely upon to share and disseminate your software? This question is related to item **I2**.

A10. A RS sharing and dissemination procedure has been proposed in [23, 24, 36]. The last and revised version has been presented in [36] in order to study how to propose a similar dissemination procedure for RD. We include here a revised version of the RS dissemination procedure that can be found in [36]. Steps marked with (*) are to be revisited regularly for each main version release.

- Choose a name or title to identify the RS, avoid trademarks and other proprietary names, you can associate date, version number, target platform… Consider best practices in file names [15].
  Note the importance of the choice of a name using standard naming conventions, as reflected in the recent Census III of Free and Open Source Software [59]:
  > *There is a critical need for a standardized software component naming schema that is in widespread use. Until one is widely used, strategies for software security, transparency, and more will have limited effect.*

  See also the advise of [18] to choose a good name for your RS project.
  PURLs[37] or package URLs propose an attempt to standardize existing approaches to reliably identify and locate software packages by using strings to identify and locate the components in an uniform way, across programming languages, package managers, packaging conventions, tools, APIs and databases.
- (*) Establish the list of authors and affiliations (this is the so called *research team step*). This list can include an associated percentage of participation, completed with minor contributors. Such information can be of importance in order to deal with rightholders and rights sharing. If the list is too long, keep updated information in a web page or another document like a Software Management Plan (SMP) [27, 45], for example, where you can mention the different contributor roles. This question about authors and contributors has been deepened in [39], where we propose a homogeneous view for RS, RD and research publications. This is the step in which the intellectual property producer's rights are to be established. Producers include the RS authors and rightholders. This is then the step in which RS legal issues related to copyright information are dealt with.
  Copyright issues can be a difficult matter to handle in large projects, see for example [19, 74]. Note that SMPs for RS are still not widely required by research funders, although new recommendations are emerging in the landscape [14, 60, 70]. See [45] for further information and references on this subject. We also remark that *plans* are mentioned in question & answer **I1.7** of the FAQ in Section 3.
- (*) Establish the list of included software and data components, indicate their licenses (or other documents like the component's documentation…) giving the rights to access, copying, modification and redistribution for each component. Take into consideration best citation practices.
- Choose a software license, with the agreement of all the rightholders and authors, and establish a signed agreement if possible. The licenses of the software components that have been included and/or modified to produce the RS can have impact in your license decision. As mentioned in Section 2, software licenses and licensing information can be found at the Free Software Foundation, the Open Source Initiative, and the Software Package Data Exchange. This is then the step in which legal issues related to the RS sharing conditions are to be taken into consideration. Indicate the license in the RS files, its documentation, the project web

[37] https://github.com/package-url/purl-spec

pages… Give licenses like GNU FDL,[38] Creative Commons (CC),[39] LAL[40]… to documentation and to websites related to the RS.

- Choose a website, forge, or deposit to distribute your product; licensing and/or conditions of use, copy, modification, and/or redistribution should be clearly stated, as well as the best way to cite your work. Good metadata (see *A7*) and respect of open standards are always important when giving away new components to a large community: it helps others to reuse your work and increases its longevity. Use Persistent Identifiers (PIDs)[41] if possible.
  See also *A9* and the *Where?* conundrum question for RS [43].
  Note that security issues should be considered before releasing the RS, see questions & answers **I1.8** and **I1.9** of the FAQ in Section 3. For example, the Procedure for the licensing of Commission software also includes a security verification step [16].
- (*) This step deals with the utility of the RS and how it has been used for your research (this is the *research work step*). Establish the list of main functionalities, and archive a tar.gz or similar for the main RS versions in safe place. Keep a list of the associated research work, including published articles. Update your documentation, SMP, website… with the new information in each main RS version.
- Inform your laboratories and Head Institutions about this RS dissemination (if this has not be done in the license step).
- Create and indicate clearly an address of contact.
- Release the RS.
- Inform the community (mailing lists…), consider the publication of a software paper [13], see for example the list of Journals where you can publish articles focusing on software in the Software Sustainability Institute web page [49].

This proposed procedure is flexible and can be adapted to many different situations, it can also be used for RD [36].

As already mentioned, this dissemination procedure has been proposed in 2010 in France [23], where the the Research Ministry in charge of the Open Science policy [57] has conducted a RS national survey in 2023. The survey listed 1331 RS, and have found that around 80% are disseminated with a license.

Q11. What existing standards or criteria do you use to evaluate the RS?. This question is related to item **I3**.

A11. The CDUR evaluation protocol has been initially formulated for RS in [29] and has been extended to RD in [36], see also [37, 38]. It has been designed to help evaluators, members of evaluation committees, and evaluated researchers, members of the RT responsible of the RS or RD. We include here a short presentation as given in [37] (with minor modifications), but, please, do refer to [29] for a thorough presentation and discussion of this protocol.

There are four steps in the proposed CDUR evaluation protocol, which is flexible enough to be applied in different evaluation contexts. They are to be applied in the following chronological order:

(C) Citation. This step measures if the RS or RD are well identified as research outputs, i.e. if there is a good citation form, or good metadata. We look here to best citation practices: the

[38] http://www.gnu.org/copyleft/fdl.html
[39] https://creativecommons.org/choose/
[40] http://artlibre.org/licence/lal/en/
[41] http://en.wikipedia.org/wiki/Persistent_identifier

citation form that is proposed for your RS and/or RD, and how you cite RS and RD done by other RTs.
This is a legal related point where we ask for authors (if any) to be well identified, which are their affiliations, and, for example, the percentage of their participation in software writing.

(D) Dissemination. In this point we look to best dissemination practices, in agreement with the scientific policy of the evaluation context. The dissemination of RS and RD needs a license to set the sharing conditions and may follow procedures like, for example, the one presented in *A10* (see also [23, 24, 36]). For RD there are maybe further legal issues to look at (personal data, *sui generis* database rights…).
This is a policy point in which we look at Open Science requirements [30, 32, 65, 76].

(U) Use. This point examines "software" or "data" aspects, in particular the correct results that have been obtained, and we can also look if their reuse has been facilitated, the output quality, best software/data practices such as documentation, testing, maintenance, installation or reuse protocols, up to read the code, launch the RS, use examples…
This is the reproducibility point that looks at the validation of the scientific results obtained with the RS and/or the RD. This is also the step that can take into consideration reviews or security vulnerabilities, see questions & answers **I1.8**, **I1.9**, **I1.12** of the FAQ in Section 3.

(R) Research. This point examines the research aspects associated to the RS and/or RD production: the quality of the scientific work, the proposed and coded algorithms and data structures, which are the related publications, the collaborations, the funded projects…
This point measures the impact of the RD and/or RS related research work. It is to be adapted to the practices that may appear in different scientific communities.

We observe that this dissemination question labelled **I3** in Section 3 does not appear in the list of questions of the FAQ in Section 3, but the FAIR principles [3, 77] are mentioned as a goal in the introduction:

> *The FAIR principles (findable, accessible, interoperable, and reusable) also provide a useful framework for better software management and sharing*

and in the question & answer **I1.2** of the FAQ in Section 3.

Notice that we have studied the relationships between the FAIR Principles [77, 3] and the CDUR evaluation protocols in [36, 38], where we explain how CDUR can be used to improve the FAIRness of RS (and RD).

Q12. What technical, policy, financial, institutional, and/or social barriers to sharing or reuse of RS have you encountered? This question is related to item **I5**.

A12. Among all the possible barriers that RTs may usually encounter when sharing RS, we would like to mention the following ones.

To share RS does take time and resources, and many times it involves tasks (technical, administrative…) that may be far away of the research interests of a RT. It also needs to take into account legal issues (intellectual property rights, licensing…), and usually RTs are not very much aware of legal matters, nor do find easily the help needed to deal with these questions. It may happen also that the research work is done in the framework of international collaborations, which does not help to deal with these issues easily.

On the other hand, and even if this is changing a lot nowadays, the scientific work is evaluated mainly regarding the publications, and does not take into account the production of RS or other outputs [11, 12, 29, 36, 48, 51, 55]. Until now, to do the effort of RS dissemination has not much scientific value, and it is usually done in order to facilitate the reproducibility of the scientific results, to increase citations or to look for collaborations with other RTs that may have common scientific interests.

Notice also the *Needs and gaps* mentioned in Section 2.5 of the UNESCO report on Open Hydrology [14].

Regarding the issue of reuse, in our opinion, the first barrier is that it is still difficult to make RS visible and accessible, and among other questions, it is important to choose correctly *Where?* to share the RS [43] (see *A9*). But once a RT has access to some RS, it should be verified the legal conditions that allow the reuse (licenses), as well as the technical software context that facilitates the actual reuse: documentation, availability of versions for your own computer environment, installing procedures, used components, programming language…

Finally, the Scientific Institutions are currently adopting new Open Science policies [30, 32, 65, 76] (see also Table 1 and other references given in the Introduction section), and it will take time until these policies are well installed, adopted and followed. In our view, it is necessary that these policies are installed and followed in a sound manner, which, among other consequences, will broadly facilitate the sharing and reuse RS and other research outputs.

Q13. Comment on your ability to reuse RS developed by others. This question is related to item **I6** (see also *A12*).

A13. Our intention with the proposition of the dissemination procedures and the CDUR evaluation protocols is to help the scientific community at large to promote and adopt RS (and RD) best practices on sharing, dissemination and evaluation of these research outputs. As a consequence, RTs will be more aware of procedures that should be followed to better share their work, and to facilitate their research output's reuse conditions, leading, in this manner, towards better recognition and evaluation.

If these proposed procedures and protocols are largely adopted, they will also be helpful to potential users to better and quickly recognize research outputs shared and disseminated in good conditions, that have taken into account, in particular, the facilitation of their reuse, as the Use step of the CDUR protocols deals with this issue in a precise and transparent way.

As studied in [30, 33, 34, 43], there is a dissemination/evaluation loop: if you improve the evaluation context, the RTs will adapt to this new context and improve the dissemination conditions of their research outputs, facilitating thus their reusability, as well as their positive evaluation.

As it is easy to see, we have stressed many times the legal side of the important issues, as to make RS (and RD) visible, and accessible means that their reuse is legally ensured and technically facilitated. But the reuse of research outputs is, among all, a scientific point: each RT, as a potential RS user, must consider how any output is fit for their research work, if it is necessary to modify it, and if the output comes from another scientific discipline, how it can be adapted to a new context. These are the kind of issues that can be taken into account in a transparent way in the Research step of CDUR.

Q14. How can NIH support RS communities of practice to better aid development of best practices for sharing and reuse of high-quality RS?. This question is related to item **I7**.

A14. We would like to propose three points that could be considered by NIH:

(1) Establish clear Open Science policies that include the sharing and dissemination conditions in which NIH RS is to be rendered visible, accessible and reusable [30, 32, 76].
(2) Establish clear dissemination procedures to widespread best dissemination practices, like the ones proposed in *A10* ([23, 24, 36]).
(3) Establish clear RS evaluation protocols like the CDUR ones proposed in *A11* ([29, 36]).

Notice also the *Opportunities and recommendations* mentioned in Section 2.6 of the UNESCO report on Open Hydrology [14].

5 Discussion and Conclusions

The National Institutes of Health (NIH) [63, 64] was seeking responses to a fairly complete list of questions regarding RS sharing and dissemination best practices [62, 61], a list that is included in Section 3.

In [42] we have selected a subset of these questions and provided answers that reflect the acquired expertise in RS and Open Science, answers that we revisit, complete and update in the present document. For us, it is also the opportunity to explore how the previously published work on RS and Open Science can be envisioned in the lens of the definition and adoption of RS sharing and disseminating policies by such kind of institution or by any other Scientific Institution or research community, work that is the product of the first-hand, complementary experience acquired by the authors of the present work during many years of use and development of software in several scientific communities and other contexts, as shown in the included list of publications.

Besides proposing and justifying answers to the NIH RFI, in the process of the careful consideration of each question, and in the analysis of the FAQ reproduced in Section 3, we have observed the following issues.

First, we consider it relevant to mention that we prefer the terminology of free/open source software (or FOSS), as the *free software concept* is older than the *open source software concept*, and, in our opinion, it is better adapted to the scientific work where researchers do study, use, copy, modify and redistribute RS as part of their everyday activities, actions that do not seem to us to be reflected in the open source software definition.

Second, while *plans* are specifically mentioned in the question & answer **I1.7** of the FAQ in Section 3, there is no mention of Software Management Plans [14, 27, 45, 60, 70] nor is there a mention of software papers [13, 49], which seem to us useful tools to accompany a RS project.

Third, let us remark that in [42], we have not provided an answer to **I1.6** in Section 3: *How should I acknowledge NIH as the funder?* as this is to be decided by the NIH, or the corresponding Scientific Institution.

Fourth: our work in dissemination and evaluation protocols is placed in a fairly general scientific context, where we rarely face security vulnerabilities or medical regulations, and thus, it does not stress what can be very important issues in medical practices.

These last two points can easily be taken into account in the dissemination procedures (as seen in *A10*) and in the CDUR evaluation protocols (*A11*) that can be adapted by every institution or to different evaluation contexts, and so, we will include, from now on, references like the SANS Institute TOP 25 Most Dangerous Software Errors[42] (and how to avoid them), while stressing the importance of the choice of a RS name [18, 59]. Moreover, the need to undergo appropriate scientific and regulatory reviews should always be explored in medical (and maybe other scientific) contexts. The Use step of CDUR is particularly adapted to consider the acknowledgment of the security issues, as well as many other software quality requirements. Additionally, the Research step of CDUR is the one to be appropriated by the scientific communities in order to take into consideration their specific research quality standards.

Fifth. The item **I3** of Section 3 mentions the standards or criteria do you use to evaluate the FAIRness of RS, and we would like to complete here the answer given in *A12*. The FAIR principles have been first stated for data management, but with the intention to be applied to all scholarly digital research objects [77]. In [3], these principles have been specifically studied in the context of

[42] https://www.sans.org/top25-software-errors/

RS. In *A12* we have explained the CDUR protocol that we have proposed in [29] to evaluate RS, work that have been completed in [36, 38] in order to better understand the relationships between the CDUR protocols and the FAIR principles for both RS and RD. As addressed in the conclusions of [36]:

> *We consider that our dissemination and evaluation (CDUR) proposals, if followed correctly, may clearly contribute towards a more sound implementation of FAIR principles for RS and RD, as they provide robust instructions for their producers to make them more findable and accessible, as well as arguments to choose suitable dissemination platforms to complete the FAIR framework. Moreover, interoperability and reusability could be also fostered with best documentation practices, such as it is proposed in our dissemination procedure; practices that can be evaluated with our CDUR protocol.*

But notice that important issues like definition, sharing & dissemination and evaluation protocols have disappear in the final FAQ list, reducing thus the initial query setting.

Sixth. The item **I7** of Section 3 mentions *How can NIH support research software communities of practice to better aid development of best practices for sharing and reuse of high-quality research software?* This question is of manifold nature, and we have proposed an answer in *A15* that we would like to complete here, as it is related to software sustainability issues mentioned in question & answer **I1.1** of the FAQ in Section 3.

As observed in [40], FOSS communities and scientific communities may have common ways of functioning, but they may have also differences as, in the research context, there is usually a RT leader, researchers with a permanent position in an academic institution, and also PhD related students, lab engineers, other technical staff, and the scientific teams may evolve in complex contexts, as studied in [39].

To support RS communities could require, in specific cases where the RS becomes a fairly important component, to be involved as an institution, and it is to be decided the resources (human, funding…) that the institution should allow and dedicate to this RS development in order to sustain the RS and its community in the long term. To this end, we would like to mention the example of the European Commission [6, 66] in the adoption of the Content management system[43] Drupal,[44] see also The OpenEuropa Initiative[45] aiming at strengthening the adoption of open source tools and practices in consolidating the European Institutions' web presence. This is an example of the implication of an institution like the European Commission in the adoption and strong support of the Drupal FOSS and its community of practice, and similar initiatives will maybe become more widespread.

For further information on FOSS communities, management, governance, communication and other issues see, for example, [18].

In summary, in this work we have profited from the opportunity provided by the NIH RFI questions regarding the issues behind the adoption of adequate policies and practices to improve RS sharing and dissemination practices in the context of Open Science.

Thus, we have first described a general framework, based in previously published work on the Open Science subject, that has been the basic tool for addressing such questions from a general perspective, providing answers that we think could be of potential interest for diverse scientific organizations and scientific communities with an important Research Software production. Indeed, tomorrow's research is built with today's building blocks (publications, RS, RD…), these building

[43] https://en.wikipedia.org/wiki/Content_management_system
[44] https://new.drupal.org/
[45] https://github.com/openeuropa/documentation

blocks should be solid from a scientific, technical, and legal point of view,[46] it is also necessary to construct a solid Open Science building process.

Acknowledgments

We would like to thank the USA NIH for such an inspiring work. We also thank Guillaume Garcia (Sciences Po, Paris) for fruitful discussions.

[46] *La recherche de demain se construit avec les briques faites aujourd'hui, il est nécessaire que ces briques soient solides du point de vue technique, scientifique et légal.* [25].

References

[1] T. Aimé (2010). A Practical Guide to Using Free Software in the Public Sector, DGI - Ministry for the Budget, Public Accounts and the Civil Service, https://doi.org/10.5281/zenodo.7096100

[2] J.-L. Archimbaud, T. Gomez-Diaz (2009). FAQ : licence & copyright pour les développements de logiciels libres de laboratoires de recherche, Publication for the French PLUME project (2007-2013), https://doi.org/10.5281/zenodo.7063146

[3] M. Barker, N.P. Chue Hong, D.S. Katz, et al. (2022). Introducing the FAIR Principles for research software. Sci Data 9, 622, https://doi.org/10.1038/s41597-022-01710-x

[4] I. Blanc Catala, R. di Cosmo, M. Giraud, et al. (2023). Establishing a national research software award [version 1; peer review: 2 approved]. Open Research Europe 2023, 3:185 https://doi.org/10.12688/openreseurope.16069.1

[5] I. Blanc, P. Boulet, G. Courbebaisse, et al. (2024). Production et valorisation des logiciels issus de la recherche publique française. Comité pour la science ouverte, 2024, https://cnrs.hal.science/hal-04844037

[6] H. Borges de Freitas (2024). Drupal History in the European Commission, DrupalCon, Barcelona 2024, https://www.youtube.com/watch?v=n00RCvU-1HI

[7] C.L. Borgman (2012). The conundrum of sharing research data. J. Am. Soc. Inf. Sci. Technol. 2012, 63, 1059–1078, https://doi.org/10.1002/asi.22634

[8] A. Carleton, M. Klein, J. Robert, et al. (2021). Architecting the future of Software Engineering. A National Agenda for Software Engineering Research & Development. Carnegie Mellon University, https://insights.sei.cmu.edu/documents/1308/2021_014_001_741195.pdf

[9] G. Catalano, E. Delugas, S. Vignetti (2025). Costs and Benefits of Open Science: Contributing to the Development of a Rigorous Assessment Framework. In: J. Gutleber, P. Charitos (eds). The Economics of Big Science 2.0. Science Policy Reports. Springer, Cham. https://doi.org/10.1007/978-3-031-60931-2_10

[10] CERN Director General (2022). CERN Open Science policy, https://cds.cern.ch/record/2835057/files/CERN-OPEN-2022-013.pdf

[11] Coalition for Advancing Research Assessment (COARA) (2022). Agreement on reforming research assessment, https://coara.eu/app/uploads/2022/09/2022_07_19_rra_agreement_final.pdf

[12] Declaration on Research Assessment (DORA) (2012). San Francisco Declaration on Research Assessment, https://sfdora.org/read/, http://www.ascb.org/dora/

[13] L. Dedieu (2014). Revues publiant des Data papers et des Software papers, CIRAD preprint, https://collaboratif.cirad.fr/alfresco/s/d/workspace/SpacesStore/91230085-7c0c-4283-b2c4-25bd792121a9

[14] N. Dogulu, K. Verbist, A. Mertens (2024). Open Hydrology, UNESCO, https://unesdoc.unesco.org/ark:/48223/pf0000390401

[15] DoRANum (2018). Données de la recherche : apprentissage numérique. Comment bien nommer ses fichiers ? https://doranum.fr/stockage-archivage/comment-nommer_fichiers_10_13143_wgqw-aa59/.
See also Harvard, File Naming Conventions, https://datamanagement.hms.harvard.edu/plan-design/file-naming-conventions

[16] European Commission (2021). Commission Decision of 8 December 2021 on the open source licensing and reuse of Commission software 2021/C 495 I/01, https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32021D1209(01)

[17] European Commission, Directorate-General for Research and Innovation (2024). Guidelines to the Rules on Open Access to Scientific Publications and Open Access to Research Data in Horizon 2020. Version 4.1. 2024, https://ec.europa.eu/info/funding-tenders/opportunities/docs/2021-2027/horizon/guidance/programme-guide_horizon_en.pdf

[18] K. Fogel (2005-2022). Open Source Software. How to Run a Successful Free Software Project, https://producingoss.com/

[19] Free Software Foundation (2021). FSF copyright handling: A basis for distribution, licensing and enforcement, https://www.fsf.org/blogs/licensing/FSF-copyright-handling

[20] G7 Science and Technology Ministers (2023). G7 Science and Technology Ministers' Meeting in Sendai Communiqué, https://www8.cao.go.jp/cstp/kokusaiteki/g7_2023/230513_g7_communique.pdf.
And Annex1: G7 Open Science Working Group (OSWG), https://www8.cao.go.jp/cstp/kokusaiteki/g7_2023/annex1_os.pdf

[21] T. Gomez-Diaz (2007-2013). Le thème PLUME Patrimoine logiciel d'un laboratoire, Zenodo Community, https://zenodo.org/communities/plume-patrimoine-logiciel-laboratoire/

[22] T. Gomez-Diaz (2007-2025). Page web with presentations, articles, posters and other productions since 2007, Laboratoire d'informatique Gaspard-Monge (LIGM), http://igm.univ-mlv.fr/~teresa/logicielsLIGM/

[23] T. Gomez-Diaz (2010). Diffuser un logiciel de laboratoire : recommandations juridiques et administratives, Publication for the French PLUME project (2007-2013), https://doi.org/10.5281/zenodo.7096216

[24] T. Gomez-Diaz (2014). Free software, Open source software, licenses. A short presentation including a procedure for research software and data dissemination, Zenodo Preprint, Septembre 2014, https://doi.org/10.5281/zenodo.11709.
Presented at the Workshop on open licenses: Data licencing and policies, EGI Conference 2015, Lisbon, May 2015, https://indico.egi.eu/indico/event/2452/session/75/.

Spanish version: Software libre, software de código abierto, licencias. Donde se propone un procedimiento de distribución de software y datos de investigación, Zenodo Preprint, Septembre 2015, https://doi.org/10.5281/zenodo.31547

[25] T. Gomez-Diaz (2015). Article vs. Logiciel : questions juridiques et de politique scientifique dans la production de logiciels. 1024 - Bulletin de la société informatique de France, N. 5, 2015, http://www.societe-informatique-de-france.fr/wp-content/uploads/2015/04/1024-5-gomez-diaz.pdf. First version published for the French PLUME project (2007-2013) in 2011, https://doi.org/10.5281/zenodo.7063154

[26] T. Gomez-Diaz (2017). Software legal protection, European IPR Helpdesk Bulletin Issue (26), http://igm.univ-mlv.fr/~teresa/logicielsLIGM/documents/Internacional/European_IPR_Helpdesk_Bulletin_Issue_26.pdf

[27] T. Gomez-Diaz, G. Romier (2018). Research Software management Plan Template V3.2. Projet PRESOFT, Bilingual document (FR/EN), Zenodo Preprint, https://zenodo.org/records/1405614

[28] T. Gomez-Diaz (2019). Le Projet PLUME et le paysage actuel des logiciels de la recherche dans la science ouverte. Zenodo Preprint, https://doi.org/10.5281/zenodo.2591474

[29] T. Gomez-Diaz, T. Recio (2019). On the evaluation of research software: the CDUR procedure [version 2; peer review: 2 approved], F1000Research 2019, 8:1353, https://doi.org/10.12688/f1000research.19994.2. See also https://pubmed.ncbi.nlm.nih.gov/31814965/

[30] T. Gomez-Diaz, T. Recio (2020-21). Towards an Open Science definition as a political and legal framework: on the sharing and dissemination of research outputs, Version 2 published in POLIS N. 19 (2020), https://doi.org/10.58944/yuro5734. Version 3 (2021) available in Zenodo https://doi.org/10.5281/zenodo.4577066

[31] T. Gomez-Diaz (2021). Free/Open source Research Software production at the Gaspard-Monge Computer Science laboratory (LIGM). Lessons learnt, FOSDEM 2021, Open Research Tools and Technologies devroom, https://fosdem.org/2021/schedule/event/open_research_gaspard_monge/

[32] T. Gomez-Diaz, T. Recio (2021). The future of Open Science asks for a common understanding, Poster, EGI Virtual Conference 2021, https://doi.org/10.5281/zenodo.6433533

[33] T. Gomez-Diaz, T. Recio (2022). On the dissemination/evaluation loop for Research Software, FOSDEM 2022, Open Research Tools and Technologies devroom, https://archive.fosdem.org/2021/schedule/event/open_research_gaspard_monge/

[34] T. Gomez-Diaz, T. Recio (2022). Research Software and Research Data: dissemination, evaluation and reusability in the Open Science context, Poster, 17th International Digital Curation Conference IDCC22, https://doi.org/10.5281/zenodo.6778872

[35] T. Gomez-Diaz, T. Recio (2022). Research Software vs. Research Data I: Towards a Research Data definition in the Open Science context [version 2; peer review: 3 approved], F1000Research 2022, 11:118, https://doi.org/10.12688/f1000research.78195.2. See also https://pubmed.ncbi.nlm.nih.gov/36415208/

[36] T. Gomez-Diaz, T. Recio (2022). Research Software vs. Research Data II: Protocols for Research Data dissemination and evaluation in the Open Science context [version 2; peer review: 2 approved], F1000Research 2022, 11:117, https://doi.org/10.12688/f1000research.78459.2. See also https://pubmed.ncbi.nlm.nih.gov/36483317/

[37] T. Gomez-Diaz, T. Recio (2023). A Code for Thought podcast: Research software and Research Data in Open Science, July 14th 2023, Zenodo Preprint, https://doi.org/10.5281/zenodo.8159906

[38] T. Gomez-Diaz, T. Recio (2023). How to achieve FAIRER research data by studying evaluation assessment protocols, Poster, Open Science FAIR Conference, Madrid 2023, https://doi.org/10.5281/zenodo.8398431

[39] T. Gomez-Diaz, T. Recio (2023). Articles, software, data: An Open Science ethological study, Maple Transactions, 3, 4, Article 17132 (December 2023), https://doi.org/10.5206/mt.v3i4.17132

[40] T. Gomez-Diaz (2024). Les logiciels de la recherche et leurs licences : trois visions sur un objet, Version 3, Training support, University Gustave Eiffel, Marne-la-Vallée, France, 2023, 62 pp., https://hal.science/hal-02434287

[41] T. Gomez-Diaz, T. Recio (2024). Open comments on the "Open Letter: Establishing a national research software award", Open Research Europe, https://doi.org/10.12688/openreseurope.16069.1

[42] T. Gomez-Diaz, T. Recio (2024). Open comments for the NIH Request for Information (RFI): Best Practices for Sharing NIH Supported Research Software (NIH, 2023), Zenodo Preprint, https://doi.org/10.5281/zenodo.10617585

[43] T. Gomez-Diaz, T. Recio (2024). The Conundrum Challenges for Research Software in Open Science, Computers 2024, 13(11), 302, https://doi.org/10.3390/computers13110302

[44] T. Gomez-Diaz, T. Recio (2025). The conundrum challenges for Research Software and Research Data in Open Science, FOSDEM 2025, Open Research devroom, https://fosdem.org/2025/schedule/event/fosdem-2025-4607-the-conundrum-challenges-for-research-software-and-research-data-in-open-science/

[45] Y.V. Grossmann, G. Lanza, K. Biernacka, T. Hasler, K. Helbig (2024). Software Management Plans–Current Concepts, Tools, and Application. Data Sci. J. 2024, 23, 43, https://doi.org/10.5334/dsj-2024-043

[46] M. Gruenpeter, D. Katz, A.-L. Lamprecht, et al. (2021). Defining Research Software: a controversial discussion (Version 1), Zenodo Preprint, https://doi.org/10.5281/zenodo.5504016

[47] G. Hart, (1996). The five W's: An old tool for the new task of task analysis. Tech. Commun. 1996, 43, 139-145, https://www.ingentaconnect.com/content/stc/tc/1996/00000043/00000002/art00004
[48] A. Hocquet, F. Wieber, G. Gramelsberger, et al. (2024). Software in science is ubiquitous yet overlooked. Nat Comput Sci 4, 465-468 (2024), https://doi.org/10.1038/s43588-024-00651-2
[49] N.C. Hong (2014). In which journals should I publish my software? Software Sustainability Institute Preprint, https://www.software.ac.uk/top-tip/which-journals-should-i-publish-my-software
[50] J. Howison, E. Deelman, M.J. McLennan, et al. (2015). Understanding the scientific software ecosystem and its impact: Current and future measures. Research Evaluation, 24(4): 454-470, https://doi.org/10.1093/reseval/rvv014
[51] L.J. Hwang, A. Fish, L. Soito, et al. (2017). Software and the scientist: Coding and citation practices in geodynamics. Earth and Space Science, 4, 670-680, https://doi.org/10.1002/2016EA000225
[52] R.C. Jiménez, M. Kuzak, M. Alhamdoosh, et al. (2017). Four simple recommendations to encourage best practices in research software [version 1; peer review: 3 approved]. F1000Research 2017, 6:876, https://doi.org/10.12688/f1000research.11407.1
[53] D. Kelly (2011). An Analysis of Process Characteristics for Developing Scientific Software. J Organ End User Com. 2011; 23(4): 64.79, https://doi.org/10.4018/joeuc.2011100105
[54] S. Hettrick (2016). Research Software Sustainability. Report on a Knowledge Exchange Workshop, https://www.knowledge-exchange.info/event/software-sustainability
[55] C. Merow, B. Boyle, B.J. Enquist, et al. (2023). Better incentives are needed to reward academic software development. Nat Ecol Evol 7, 626-627 (2023), https://doi.org/10.1038/s41559-023-02008-w
[56] R. Messerschmidt, H. Pampel, F. Bach, et al. (2021). Checklist to Support the Helmholtz Centers in Implementing Policies on Sustainable Research Software, Potsdam, Helmholtz Open Science Office, https://doi.org/10.48440/os.helmholtz.038. See also https://os.helmholtz.de/en/open-research-software/
[57] Ministry of Higher Education, Research and Innovation (2021-24). Second National Plan for Open Science, https://www.enseignementsup-recherche.gouv.fr/sites/default/files/2021-10/second-french-plan-for-open-science-13715.pdf
[58] Ministerio de Ciencia e Inovación (2023). Estrategia Nacional de Ciencia Abierta 2023-27, https://www.ciencia.gob.es/InfoGeneralPortal/documento/c30b29d7-abac-4b31-9156-809927b5ee49
[59] F. Nagle, K. Powell, R. Zitomer, and D. A. Wheeler (2024). Census III of Free and Open Source Software: Application Libraries, The Linux Foundation, December 2024, https://www.linuxfoundation.org/hubfs/LF%20Research/lfr_censusiii_120424a.pdf
[60] NASA Science Mission Directorate (2022-25). Open-Source Science Guidance, Version 2.2 February 5, 2025, https://assets.science.nasa.gov/content/dam/science/cds/open-science/SMD%20Open-Source%20Science%20Guidance%20v2.2%2020250205.pdf
[61] Nature Computational Science Editorial (2023). Code sharing in the spotlight, https://doi.org/10.1038/s43588-023-00566-4
[62] NIH (2023). Request for Information (RFI): Best Practices for Sharing NIH Supported Research Software (2023-24), https://grants.nih.gov/grants/guide/notice-files/NOT-OD-24-005.html
[63] NIH-Wide Strategic Plan 2021–2025, https://www.nih.gov/sites/default/files/about-nih/strategic-plan-fy2021-2025-508.pdf
[64] NIH Strategic Plan For Data Science 2025–2030, https://datascience.nih.gov/sites/default/files/users/user536/NIHDataScienceStrat_Plan_508.pdf
[65] Office of Science and Technology Policy (OSTP) (2022). Memorandum for the heads of executive departments and agencies for Ensuring Free, Immediate, and Equitable Access to Federally Funded Research, August 25, 2022, https://www.whitehouse.gov/wp-content/uploads/2022/08/08-2022-OSTP-Public-access-Memo.pdf
[66] R. Ordinana (2024). Promouvoir l'Open Source à la Commission Européenne : Le Parcours d'OpenEuropa, Open Source Experience, Paris 2024, https://www.youtube.com/watch?v=CKtEuEwsh6E
[67] A. Paic (2021). Open Science - Enabling Discovery in the Digital Age, Going Digital Toolkit Note, No. 13, https://goingdigital.oecd.org/data/notes/No13_ToolkitNote_OpenScience.pdf
[68] P. Rico-Castro, L. Bonora (2023). European Commission, Directorate-General for Research and Innovation, Open Access Policies in Latin America, the Caribbean and the European Union: Progress Towards a Political Dialogue, https://data.europa.eu/doi/10.2777/162
[69] T. Scassa (2019). Intellectual Property Rights in Ethically Open Science. Version 1.0.0. DARIAH Campus [Video], https://hdl.handle.net/21.11159/019595c5-1b36-7550-960c-b1f3478970f1
[70] Science Europe (2024). Developing and Aligning Policies on Research Software: Recommendations for RFOs and RPOs, https://scienceeurope.org/our-resources/recommendations-research-software/
[71] Secretariat-General for Digital Administration, Ministry of Economic Affairs and Digital Transformation (2022). Royal Decree 4/2010, of 8 January 2010, which regulates the National Interoperability Framework within the e-Government scope, https://administracionelectronica.gob.es/dam/jcr:48e2472a-d453-4023-a4e2-b2586ae06b60/RD_4-

2010_of-8-January-consolidated-acc1.pdf.
Spanish legal text available at https://www.boe.es/eli/es/rd/2010/01/08/4/con/
[72] V. Sochat, N. May, I. Cosden, C. Martinez-Ortiz, S. Bartholomew (2022). The Research Software Encyclopedia: a community framework to define research software. Journal of Open Research Software, Volume: 10, Issue: 1, Article: 2, https://doi.org/10.5334/jors.359
[73] J. Tennant (2019). Open Science is Just Good Science. Version 1.0.0. DARIAH Campus [Video], https://hdl.handle.net/21.11159/019595c5-2081-71b3-a983-b81d7e45709c.
Transcription version: J. Tennant, N. Breznau (2022). Legacy of Jon Tennant, "Open science is just good science," https://doi.org/10.31235/osf.io/hfns2
[74] C. Topham, Free Software Foundation (2022). Copyright assignment with the FSF, https://www.fsf.org/bulletin/2022/fall/copyright-assignment-with-the-fsf
[75] UK Research and Innovation (2024). Open research, https://www.layDogulu,KoenVerbistandAnneliesMertensukri.org/manage-your-award/good-research-resource-hub/open-research/
[76] UNESCO (2021). Recommendation on Open Science, https://unesdoc.unesco.org/ark:/48223/pf0000379949.locale=en
[77] M. Wilkinson, M. Dumontier, I. Aalbersberg, et al. (2016). The FAIR Guiding Principles for scientific data management and stewardship. Sci Data 3, 160018, https://doi.org/10.1038/sdata.2016.18