CODE beyond FAIR: towards sustainable research software

Roberto Di Cosmo, Sabrina Granger, Nicolas Jullien, Konrad Hinsen, Daniel Le Berre, Violaine Louvet, Camille Maumet, Clémentine Maurice, Raphaël Monat & Nicolas P. Rougier

`rmonat.fr`

SWH Ambassadors Meeting
2025-12-16

Research software

What is research software (RS)?

"Research Software includes source code files [...] created during the research process [...]. Software components [...] that [...] were not created [...] with a clear research intent should be considered software in research [...]."

— Gruenpeter et al. [Gru+21]

Research software

What is research software (RS)?

"Research Software includes source code files [...] created during the research process [...]. Software components [...] that [...] were not created [...] with a clear research intent should be considered software in research [...]."

— Gruenpeter et al. [Gru+21]

Research software as a pillar of research

- RS mentions: 33% in 2013 to 48% in 2021 Bassinet et al. [Bas+23]
  large scale study of 908,000 articles with French authors

Research software

What is research software (RS)?

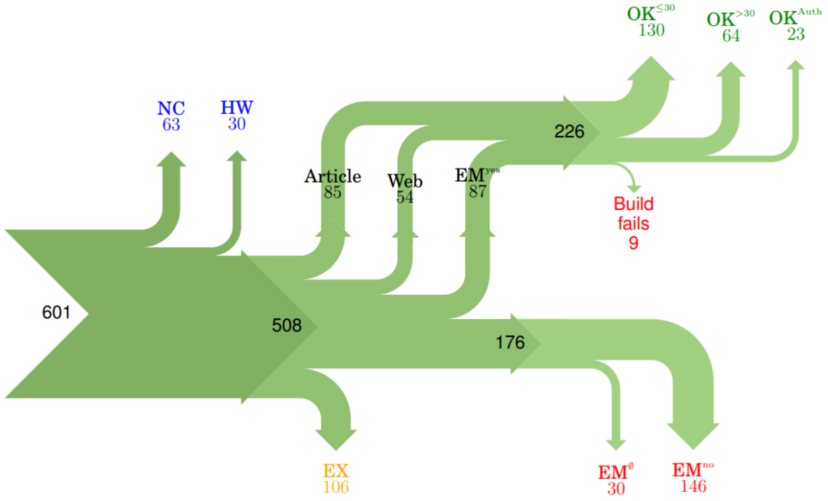
“Research Software includes source code files [...] created during the research process [...]. Software components [...] that [...] were not created [...] with a clear research intent should be considered software in research [...].”

— Gruenpeter et al. [Gru+21]

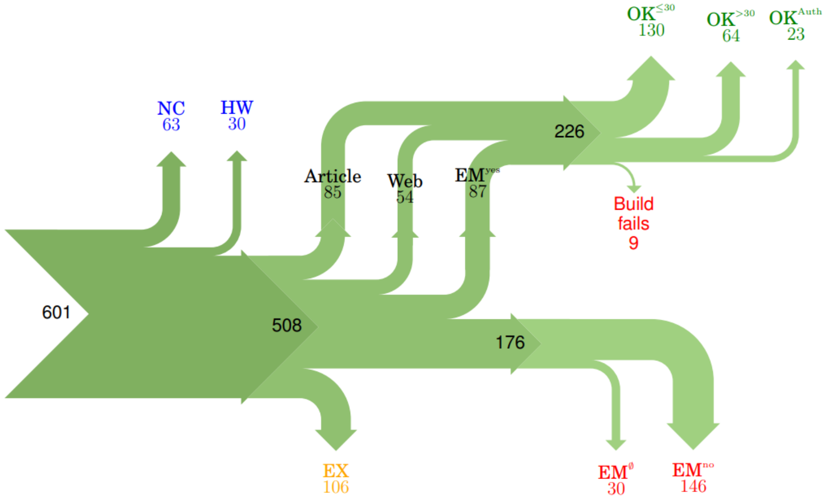
Research software as a pillar of research

- RS mentions: 33% in 2013 to 48% in 2021 Bassinet et al. [Bas+23]
  large scale study of 908,000 articles with French authors
- 10% of French RS uses proprietary licensing [Cat+24]

Reproducibility crisis – Collberg and Proebsting [CP16] – CS publications in 2013



Reproducibility crisis – Collberg and Proebsting [CP16] – CS publications in 2013

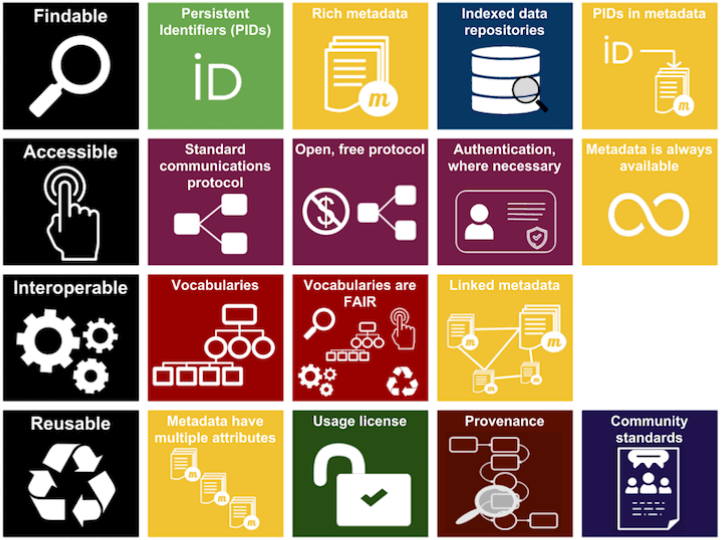


$\Longrightarrow$ Guidelines & principles to improve the state of research software

FLOSS: Free/Libre Open-Source Software

- FLOSS exists since the 80s
- Academia is a small part of it since the beginning
- Successes: Linux, Firefox, VLC, ...

FAIR principles for data [Wil+16]



FAIR principles for research software (FAIR4RS) – Barker et al. [Bar+22]

- **F** — Software, and its associated metadata, is easy for both humans and machines to find. (F1, F1.1, F1.2, F2, F3, F4)

FAIR principles for research software (FAIR4RS) – Barker et al. [Bar+22]

- **F** — Software, and its associated metadata, is easy for both humans and machines to find. (F1, F1.1, F1.2, F2, F3, F4)
- **A** — Software, and its metadata, is retrievable via standardised protocols. (A1, A1.1, A1.2, A2)

FAIR principles for research software (FAIR4RS) – Barker et al. [Bar+22]

- **F** — Software, and its associated metadata, is easy for both humans and machines to find. (F1, F1.1, F1.2, F2, F3, F4)
- **A** — Software, and its metadata, is retrievable via standardised protocols. (A1, A1.1, A1.2, A2)
- **I** — Software interoperates with other software by exchanging data and/or metadata, and/or through interaction via application programming interfaces (APIs), described through standards. (I1, I2)

FAIR principles for research software (FAIR4RS) – Barker et al. [Bar+22]

- **F** — Software, and its associated metadata, is easy for both humans and machines to find. (F1, F1.1, F1.2, F2, F3, F4)
- **A** — Software, and its metadata, is retrievable via standardised protocols. (A1, A1.1, A1.2, A2)
- **I** — Software interoperates with other software by exchanging data and/or metadata, and/or through interaction via application programming interfaces (APIs), described through standards. (I1, I2)
- **R** — Software is both usable (can be executed) and reusable (can be understood, modified, built upon, or incorporated into other software). (R1, R1.1, R1.2, R2, R3)

CODE beyond FAIR

Beyond current principles

- Burden should not be handled by RS developers only

CODE beyond FAIR

Beyond current principles

- Burden should not be handled by RS developers only
- Build upon the experience of FLOSS

CODE beyond FAIR

Beyond current principles

- Burden should not be handled by RS developers only
- Build upon the experience of FLOSS
- Take into account specifics of software (life, deps., …)

CODE beyond FAIR

Beyond current principles

- Burden should not be handled by RS developers only
- Build upon the experience of FLOSS
- Take into account specifics of software (life, deps., …)

A gradual roadmap for research software developers

CODE beyond FAIR

Beyond current principles

- Burden should not be handled by RS developers only
- Build upon the experience of FLOSS
- Take into account specifics of software (life, deps., ...)

A gradual roadmap for research software developers

- Do not deter engagement: mandatory, recommended, optional guidelines

CODE beyond FAIR

Beyond current principles

- Burden should not be handled by RS developers only
- Build upon the experience of FLOSS
- Take into account specifics of software (life, deps., ...)

A gradual roadmap for research software developers

- Do not deter engagement: mandatory, recommended, optional guidelines
- Acknowledge diverse backgrounds of RS developers

CODE beyond FAIR

Beyond current principles

- Burden should not be handled by RS developers only
- Build upon the experience of FLOSS
- Take into account specifics of software (life, deps., …)

A gradual roadmap for research software developers

- Do not deter engagement: mandatory, recommended, optional guidelines
- Acknowledge diverse backgrounds of RS developers
- Accommodate disparities among disciplines

CODE beyond FAIR

Beyond current principles

- Burden should not be handled by RS developers only
- Build upon the experience of FLOSS
- Take into account specifics of software (life, deps., ...)

A gradual roadmap for research software developers

- Do not deter engagement: mandatory, recommended, optional guidelines
- Acknowledge diverse backgrounds of RS developers
- Accommodate disparities among disciplines

A call to action for all stakeholders

- Ensure means to set these new practices in stone

A roadmap for research software developers

Open

Publish your source code on a public forge **Mandatory**

- ~~Code available upon request~~ does not work empirically [CP16]
- Forges provide tailored development and collaboration tools

Open

Publish your source code on a public forge — **Mandatory**

Save your repository on dedicated archive — **Mandatory**

⚠ Repositories and forges can close (Google Code, 2016)

▶ UNESCO-backed Software Heritage as a long-term, dedicated software archive

Open

Publish your source code on a public forge — **Mandatory**

Save your repository on dedicated archive — **Mandatory**

License your code with an open license — **Strongly recommended**

- No license = no rights distributed
- See Choose a License and institutional policies

Open

Publish your source code on a public forge	Mandatory
Save your repository on dedicated archive	Mandatory
License your code with an open license	Strongly recommended
Declare authorship and rightholders	Recommended

- Legal aspects
- For citing software

Open

Publish your source code on a public forge	Mandatory
Save your repository on dedicated archive	Mandatory
License your code with an open license	Strongly recommended
Declare authorship and rightholders	Recommended

Document

Especially important in RS, defining state-of-the-art, specialized methods

Choose meaningful names **Recommended**

Document

Especially important in RS, defining state-of-the-art, specialized methods

Choose meaningful names	Recommended
Comment code	Recommended

Document

Especially important in RS, defining state-of-the-art, specialized methods

Choose meaningful names	Recommended
Comment code	Recommended
Provide examples, notebooks and/or tutorials	Recommended

Document

Especially important in RS, defining state-of-the-art, specialized methods

Choose meaningful names	Recommended
Comment code	Recommended
Provide examples, notebooks and/or tutorials	Recommended
Document the API	Optional

Document

Especially important in RS, defining state-of-the-art, specialized methods

Choose meaningful names	Recommended
Comment code	Recommended
Provide examples, notebooks and/or tutorials	Recommended
Document the API	Optional

Execute

Goal: avoid dependency hell and ensure computational reproducibility.

List software and hardware dependencies **Recommended**

- Version of dependencies (including operating system)
- Improper use can create crashes or even incorrect results [Bha+19]

Execute

Goal: avoid dependency hell and ensure computational reproducibility.

List software and hardware dependencies — **Recommended**

Provide a computational environment — **Optional**

- Guix, Nix, or at least containers (Docker, VMs)

Execute

Goal: avoid dependency hell and ensure computational reproducibility.

List software and hardware dependencies — **Recommended**

Provide a computational environment — **Optional**

Implement a test suite — **Optional**

- Can also help you as a developer to detect regressions!

Execute

Goal: avoid dependency hell and ensure computational reproducibility.

List software and hardware dependencies	Recommended
Provide a computational environment	Optional
Implement a test suite	Optional
Show real-life usage example with expected results	Optional

Execute

Goal: avoid dependency hell and ensure computational reproducibility.

List software and hardware dependencies	Recommended
Provide a computational environment	Optional
Implement a test suite	Optional
Show real-life usage example with expected results	Optional

Collaborate

Respond to issues	Recommended

Collaborate

Respond to issues	Recommended
Provide contribution guidelines	Recommended

Collaborate

Respond to issues	Recommended
Provide contribution guidelines	Recommended
Describe maintenance, features and support limits	Recommended

Collaborate

Respond to issues	Recommended
Provide contribution guidelines	Recommended
Describe maintenance, features and support limits	Recommended
Build and animate a community	Optional

Collaborate

Respond to issues	Recommended
Provide contribution guidelines	Recommended
Describe maintenance, features and support limits	Recommended
Build and animate a community	Optional

A call to action for all stakeholders

Institutions

Support development **Mandatory**

- Hiring research software engineers
- Actionable legal frameworks for distributing FLOSS research software

Institutions

Support development — **Mandatory**

Promote software — **Recommended**

- Software as scientific contributions for hiring/promotion [Can+21; Ver+25]
- Awards to create visibility and raise awareness [Cat+23]

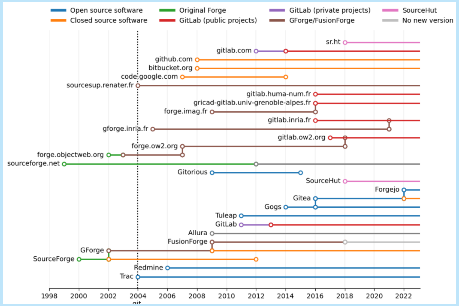
Institutions

Support development Mandatory

Promote software Recommended

Build and maintain institutional forges Optional

- 81 different forges in French higher education [Le +23]



- Key issue: international interoperability

Institutions

Support development	Mandatory
Promote software	Recommended
Build and maintain institutional forges	Optional

Funders

Provide grants for long-term support — **Mandatory**

- Long-term sustainability cannot be the sole responsibility of scientists.
- Recent initiatives by the Software Sustainability Institute or the German Research Foundation.

Funders

Provide grants for long-term support — **Mandatory**

Promote reproducibility — **Recommended**

Natural follow-up of important funders (ERC) promoting open science.

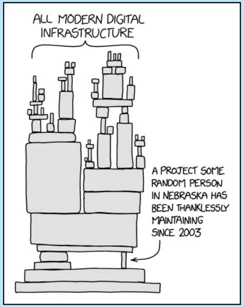
Funders

Provide grants for long-term support — **Mandatory**

Promote reproducibility — **Recommended**

Facilitate collaborations — **Optional**

- subset of a grant could be dedicated to the support of a core library
- RS equivalent of the publicly-funded German Sovereign Tech Fund?


ALL MODERN DIGITAL INFRASTRUCTURE
A PROJECT SOME RANDOM PERSON IN NEBRASKA HAS BEEN THANKLESSLY MAINTAINING SINCE 2003

XKCD #2347

Funders

Provide grants for long-term support	Mandatory
Promote reproducibility	Recommended
Facilitate collaborations	Optional

Libraries

Libraries can accompany the turn into research software and bring their expertise to ensure proper metadata definition and archival.

Prepare software archival plans	**Mandatory**

Libraries

Libraries can accompany the turn into research software and bring their expertise to ensure proper metadata definition and archival.

Prepare software archival plans — **Mandatory**

Create and curate software metadata — **Mandatory**

Examples: SWHID, CodeMeta, Bioschemas ComputationalTool or Automated Software Metadata Publication.

Libraries

Libraries can accompany the turn into research software and bring their expertise to ensure proper metadata definition and archival.

Prepare software archival plans **Mandatory**

Create and curate software metadata **Mandatory**

Catalog software **Recommended**

- Institutional: NASA's Software Catalog, French Catalog for Research Software
- Specific communities: swMATH, bio.tools

Libraries

Libraries can accompany the turn into research software and bring their expertise to ensure proper metadata definition and archival.

Prepare software archival plans	Mandatory
Create and curate software metadata	Mandatory
Catalog software	Recommended

Publishers

Enforce open source **Mandatory**

- Code should be published alongside papers
- ~~Availability upon request~~

Publishers

Enforce open source **Mandatory**

Link publications and codes **Recommended**

Examples: Dagstuhl Artifact Series (DARTS) [Dag] and IACR’s Artifact Archive [IAC]

Publishers

Enforce open source — **Mandatory**

Link publications and codes — **Recommended**

Review software — **Optional**

artefact evaluation processes since 2011 [Di +20; Inf+25]

- ▶ Artifact = computational environment + software
- ▶ Goal: reproducibility of software-related experimental claims

Publishers

Enforce open source	Mandatory
Link publications and codes	Recommended
Review software	Optional

Conclusion

Conclusion

- Guidelines to go beyond FAIR principles

Conclusion

- Guidelines to go beyond FAIR principles
- CODE: accessible, gradual roadmap for research software developers

Conclusion

- Guidelines to go beyond FAIR principles
- CODE: <u>accessible, gradual</u> roadmap for research software developers
- Call to action to ensure research software is treated as a first-class citizen

Conclusion

- Guidelines to go beyond FAIR principles
- CODE: accessible, gradual roadmap for research software developers
- Call to action to ensure research software is treated as a first-class citizen
- Published versions:

Conclusion

- Guidelines to go beyond FAIR principles
- CODE: accessible, gradual roadmap for research software developers
- Call to action to ensure research software is treated as a first-class citizen
- Published versions:
  - Preprint (under revision) `hal.science/hal-04930405` [Di +25a]

Conclusion

- Guidelines to go beyond FAIR principles
- CODE: accessible, gradual roadmap for research software developers
- Call to action to ensure research software is treated as a first-class citizen
- Published versions:
  - Preprint (under revision) `hal.science/hal-04930405` [Di +25a]
  - *Nature* Commentary
    "Stop treating code like an afterthought: record, share and value it" [Di +25b]

Conclusion

- Guidelines to go beyond FAIR principles
- CODE: accessible, gradual roadmap for research software developers
- Call to action to ensure research software is treated as a first-class citizen
- Published versions:
  - Preprint (under revision) `hal.science/hal-04930405` [Di +25a]
  - *Nature* Commentary
    "Stop treating code like an afterthought: record, share and value it" [Di +25b]
- Comments? Other examples in mind?

References – I

[Bar+22] Michelle Barker et al. "Introducing the FAIR Principles for research software". In: Scientific Data 1 (2022), page 622.

[Bas+23] Aricia Bassinet et al. "Large-scale Machine-Learning analysis of scientific PDF for monitoring the production and the openness of research data and software in France". working paper or preprint. 2023.

[Bha+19] Jayanti Bhandari Neupane et al. "Characterization of Leptazolines A-D, Polar Oxazolines from the Cyanobacterium Leptolyngbya sp., Reveals a Glitch with the "Willoughby-Hoye" Scripts for Calculating NMR Chemical Shifts". In: Organic Letters 20 (2019), pages 8449–8453. DOI: `10.1021/acs.orglett.9b03216`.

References – II

[Can+21] Anne Canteaut et al. Software Evaluation. Research Report. Inria, Jan. 2021.

[Cat+23] Isabelle Catala et al. “Establishing a national research software award”. working paper or preprint. 2023. DOI: 10.12688/openreseurope.16069.1.

[Cat+24] Isabelle Catala et al. Production et valorisation des logiciels issus de la recherche publique franç Technical report. Comité pour la science ouverte, Nov. 2024. DOI: 10.52949/47.

[CP16] Christian Collberg and Todd A Proebsting. “Repeatability in computer systems research”. In: Communications of the ACM 3 (2016), pages 62–69.

References – III

[Dag] Dagstuhl. Dagstuhl Artifact Series. URL: `https://drops.dagstuhl.de/entities/journal/DARTS` (visited on 06/21/2024).

[Di +20] Roberto Di Cosmo et al. Scholarly Infrastructures for Research Software. Report from the EOSC Execu European Commission. Directorate General for Research and Innovation., 2020. DOI: `10.2777/28598`.

[Di +25a] Roberto Di Cosmo et al. “CODE beyond FAIR”. working paper or preprint. Feb. 2025.

References – IV

[Di +25b] Roberto Di Cosmo et al. "Stop treating code like an afterthought: record, share and value it". In: Nature 8084 (Oct. 2025), pages 284–286. DOI: 10.1038/d41586-025-03196-0.

[Gru+21] Morane Gruenpeter et al. Defining Research Software: a controversial discussion. Version 1. Sept. 2021. DOI: 10.5281/zenodo.5504016.

[IAC] IACR. IACR Artifact Archive. URL: https://artifacts.iacr.org/ (visited on 06/21/2024).

[Inf+25] Informatics Europe et al. Informatics Research Evaluation, 2025 Revised Report. Mar. 2025. DOI: 10.5281/zenodo.15834583.

References – V

[Le +23] Daniel Le Berre et al.
Higher Education and Research Forges in France - Definition, uses, limitatior
Technical report. Comité pour la science ouverte, Nov. 2023. DOI:
10.52949/37.

[Ver+25] Thanasis Vergoulis et al.
D5.1 Landscape analysis of existing rewards and mechanisms for research sc
Mar. 2025. DOI: 10.5281/zenodo.14978474.

[Wil+16] Mark D. Wilkinson et al. "The FAIR Guiding Principles for scientific data management and stewardship". In: Scientific Data 1 (Mar. 2016). DOI: 10.1038/sdata.2016.18.