FAIRsFAIR

Fostering Fair Data Practices in Europe

Tips and traps on the road to FAIR software principles

Patricia Herterich - DCC, University of Edinburgh
Morane Gruenpeter - Inria, Software Heritage

FAIRsFAIR "Fostering FAIR Data Practices In Europe" has received funding from the European Union's Horizon 2020 project call H2020-INFRAEOSC-2018-2020 Grant agreement 831558

FAIRSFAIR
Fostering Fair Data Practices in Europe

Findable Accessible Interoperable Reusable

Building bridge between communities

Software development communities
&
Research Software communities

Created by Juicy Fish from Noun Project

Created by Vectors Point from Noun Project

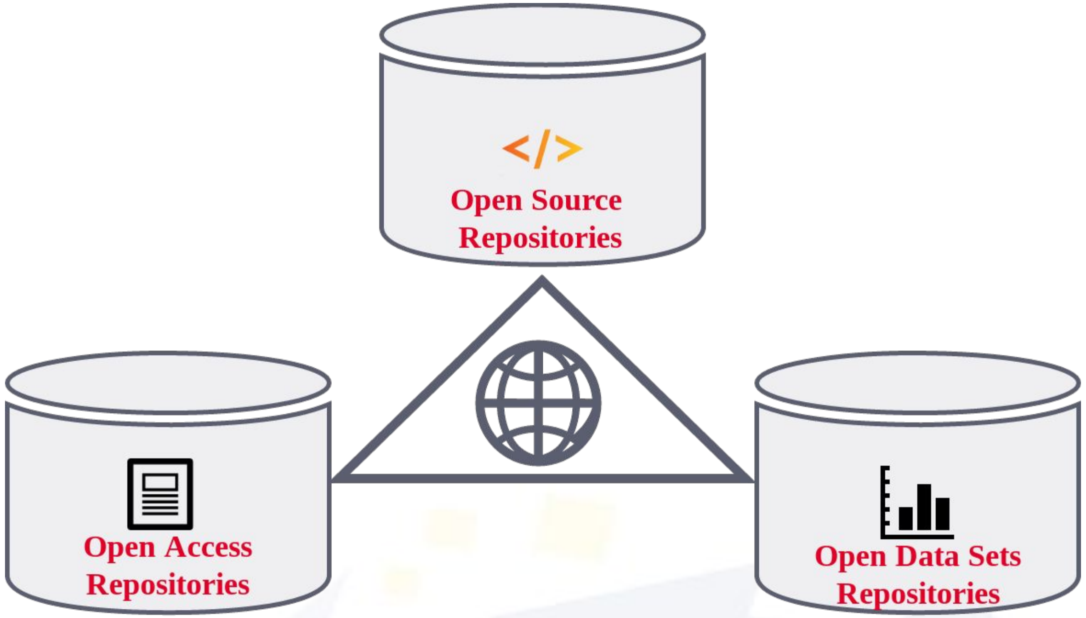
Goals for today

- Introduce the FAIRsFAIR report on FAIRness of software
- Provide our recommendations for developing FAIR principles for software
- Get your feedback on the recommendations and how they should be taken forward

Software in Research: A pillar of Open Science

Multiple facets, it can be seen as:

- a **tool**
- a research **outcome** or result
- **the object** of research



Three pillars of Open Science
Gruenpeter, Software Heritage CC-By 4.0 2019

Why are we here? A plurality of needs

Researchers

- **archive and reference** software used and created in articles
- **find** useful software
- **get credit** for developed software
- **verify/reproduce/improve** results

Laboratories/teams

- **track** software contributions
- **produce** reports
- **maintain** web page

Research Organization

know its **software assets** for:

- technology transfer,
- impact metrics,
- strategy

Software is not just another type of data

Recommendation n°5 :

Recognise that FAIR guidelines will require ***translation for other digital objects*** *and support such efforts.*

2019: ‘Six Recommendations for Implementation of FAIR Practice’

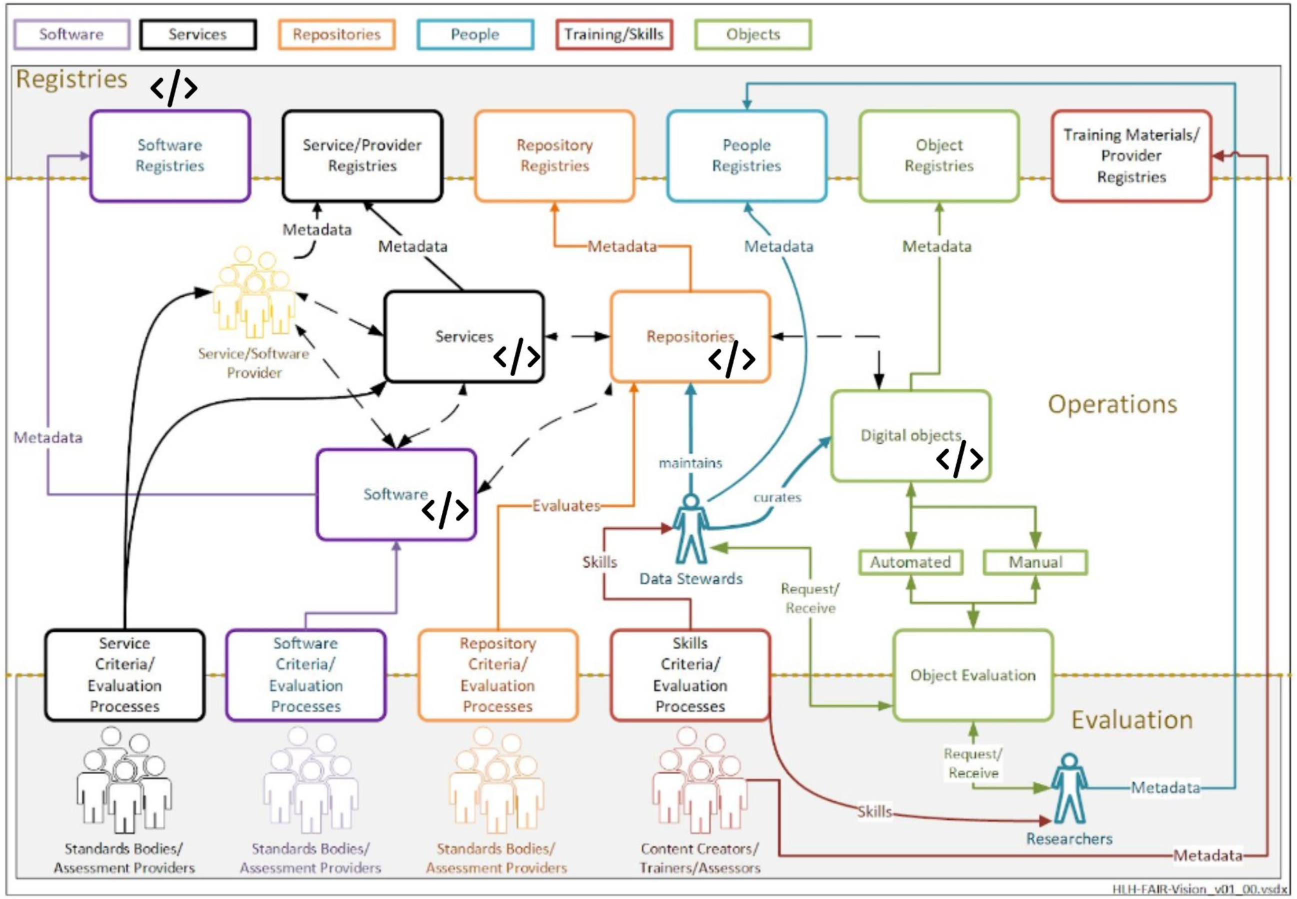
(FAIR Practice TF, 2020)

Recommendation n° 2 :

Make sure ***the specific nature of software*** *is recognized and not considered as “just data” particularly in the context of discussion about the notion of FAIR data.*

2019: the **Opportunity Note** by the French national Committee for Open Science's Free Software and Open Source Project Group (Clément-Fontaine, 2019)

Software in the FAIR ecosystem


Software
Services
Repositories
People
Training/Skills
Objects
Registries
</>
Software Registries
Service/Provider Registries
Repository Registries
People Registries
Object Registries
Training Materials/ Provider Registries
Metadata
Metadata
Metadata
Metadata
Metadata
Service/Software Provider
Services
Repositories
Operations
Digital objects
Metadata
maintains
curates
Software
Evaluates
Skills
Data Stewards
Request/ Receive
Automated
Manual
Service Criteria/ Evaluation Processes
Software Criteria/ Evaluation Processes
Repository Criteria/ Evaluation Processes
Skills Criteria/ Evaluation Processes
Object Evaluation
Evaluation
Request/ Receive
Metadata
Skills
Researchers
Metadata
Standards Bodies/ Assessment Providers
Standards Bodies/ Assessment Providers
Standards Bodies/ Assessment Providers
Content Creators/ Trainers/Assessors
HLH-FAIR-Vision_v01_00.vsdx

Ecosystem components, to highlight the software roles in the Ecosystem, the symbol </> was added (Original diagram 3 from L'Hours & Von Stein, 2020)

*"Central to the realisation of FAIR are **FAIR Digital Objects**, which may represent data, **software** or other research resources. These digital objects must be accompanied by persistent identifiers, metadata and contextual documentation to enable discovery, citation and reuse. Data should also be accompanied by the **code** used to process and analyse the data."*

Rec. 16: *Apply FAIR broadly: "FAIR should be applied broadly to all objects (including metadata, identifiers, **software** and DMPs) that are essential to the practice of research, and should inform metrics relating directly to these objects."*

Turning FAIR into reality (2018)

10.5281/zenodo.4095092

FAIRsFAIR Assessment report on 'FAIRness of software'

1. **Literature review** on the application of FAIR principles to research software

2. **State-of-the-art overview** of current solutions, challenges and practices in research software

3. **10 recommendations** *for the creation of* FAIR guiding principles for research software

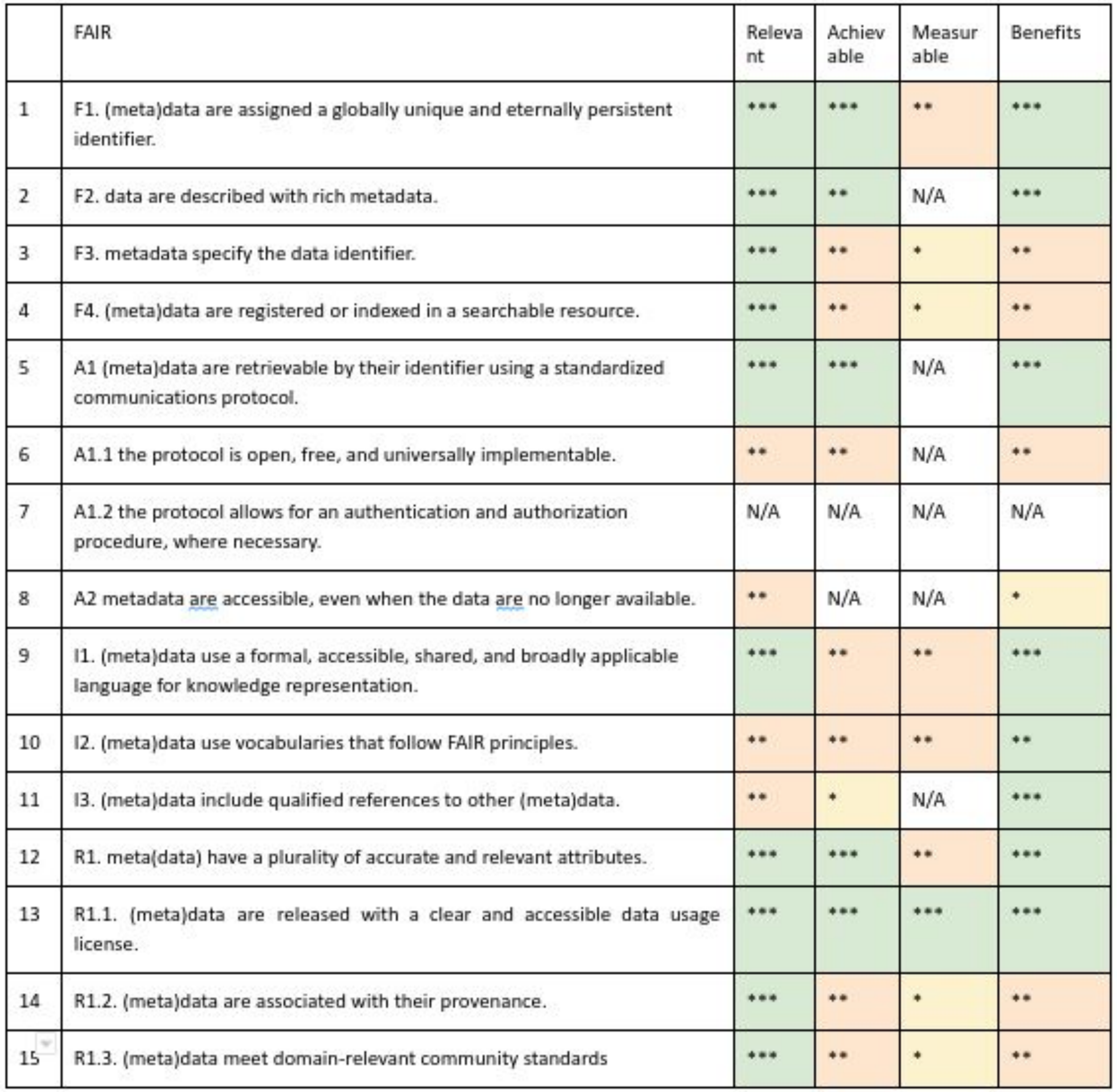
Gruenpeter, M., Di Cosmo, R., Koers, H., Herterich, P., Hooft, R., Parland-von Essen, J., Tana, J., Aalto, T., Jones, S. (2020). M2.15 Assessment report on ‘FAIRness of software’ (Version 1.1). Zenodo. https://doi.org/10.5281/zenodo.4095092

10.5281/zenodo.4095092

Literature review

- Surveyed **9 publications**
- Analyzed their view on
  - relevance,
  - achievability,
  - measurability
  - and benefits of each FAIR principle when applied to software

N/A doesn’t appear (white)
* observed in a small subset (one paper)
**medium subset (2-3)
*** large subset (3+ papers)
! disagreeing



	FAIR	Relevant	Achievable	Measurable	Benefits
1	F1. (meta)data are assigned a globally unique and eternally persistent identifier.	***	***	**	***
2	F2. data are described with rich metadata.	***	**	N/A	***
3	F3. metadata specify the data identifier.	***	**	*	**
4	F4. (meta)data are registered or indexed in a searchable resource.	***	**	*	**
5	A1 (meta)data are retrievable by their identifier using a standardized communications protocol.	***	***	N/A	***
6	A1.1 the protocol is open, free, and universally implementable.	**	**	N/A	**
7	A1.2 the protocol allows for an authentication and authorization procedure, where necessary.	N/A	N/A	N/A	N/A
8	A2 metadata are accessible, even when the data are no longer available.	**	N/A	N/A	*
9	I1. (meta)data use a formal, accessible, shared, and broadly applicable language for knowledge representation.	***	**	**	***
10	I2. (meta)data use vocabularies that follow FAIR principles.	**	**	**	**
11	I3. (meta)data include qualified references to other (meta)data.	**	*	N/A	***
12	R1. meta(data) have a plurality of accurate and relevant attributes.	***	***	**	***
13	R1.1. (meta)data are released with a clear and accessible data usage license.	***	***	***	***
14	R1.2. (meta)data are associated with their provenance.	***	**	*	**
15	R1.3. (meta)data meet domain-relevant community standards	***	**	*	**

10.5281/zenodo.4095092

Beyond the FAIR principles

- Interoperability: **dependencies and execution environment**
- Usage of **version control systems** to track changes
- **Credit** and attribution
- Testing & Software **quality**
- **Long-term** access

10.5281/zenodo.4095092

State-of-the-art overview

- **Existing infrastructures**
    - **Archives :**

      Software Heritage
      THE GREAT LIBRARY OF SOURCE CODE

      zenodo

    - **Publishers**

      DAGSTUHL ARTIFACTS SERIES · eLife · IPOL IMAGE PROCESSING ON LINE · JOSS The Journal of Open Source Software · software

    - **Registries / Aggregators,**

      scanR · swMATH · OpenAIRE

    - **Research Software Training**

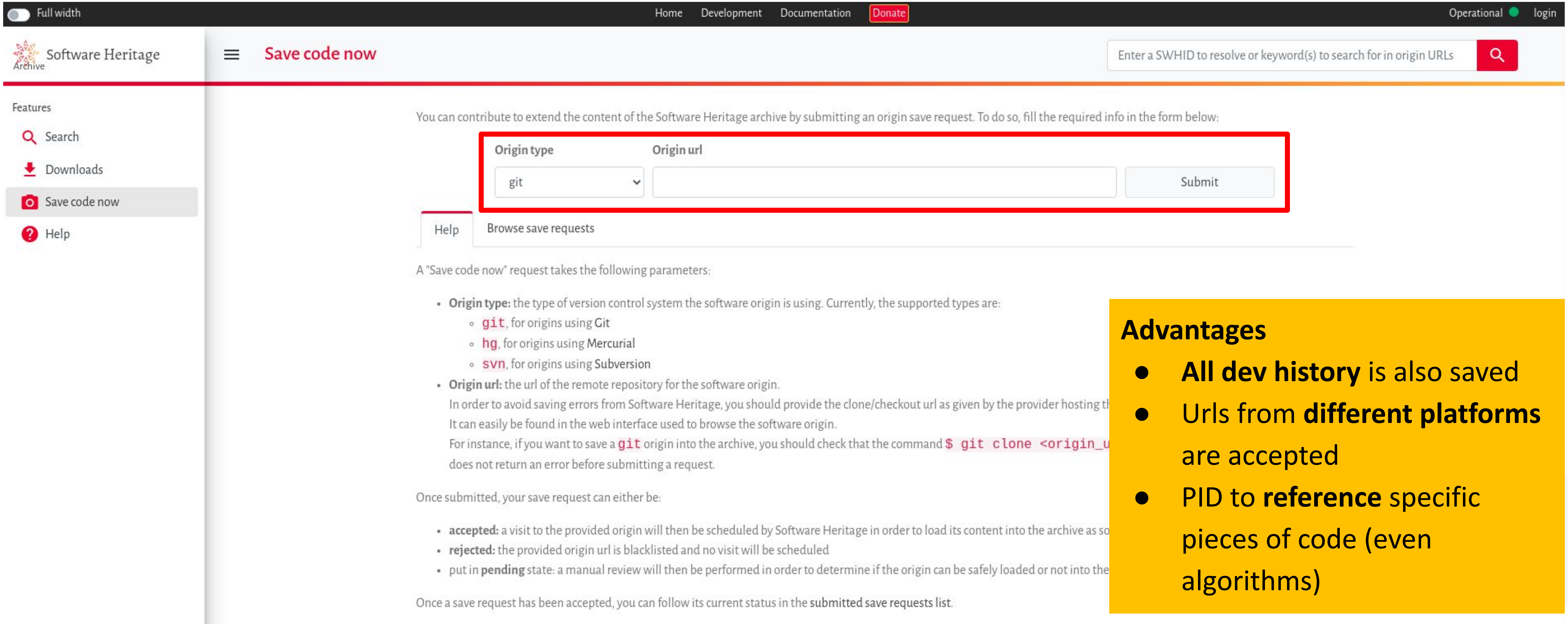
      THE CARPENTRIES

- **Existing components and mechanisms**
    - **Identification**
        - *extrinsic*: ASCL-ID, ARK, doi

          RRID · WIKIDATA · swMATH
        - *intrinsic*: SWHID
    - **Metadata**: CodeMeta, CFF
    - **Licenses**: Open Source, SPDX
    - **Artifact evaluation and badging**: AEC, ACM, NISO

[Follow the tutorial]

Save your whole repository in just one click!



You can contribute to extend the content of the Software Heritage archive by submitting an origin save request. To do so, fill the required info in the form below:

Origin type: git
Origin url:
Submit

Help | Browse save requests

A "Save code now" request takes the following parameters:

- **Origin type:** the type of version control system the software origin is using. Currently, the supported types are:
  - `git`, for origins using Git
  - `hg`, for origins using Mercurial
  - `svn`, for origins using Subversion
- **Origin url:** the url of the remote repository for the software origin.
  In order to avoid saving errors from Software Heritage, you should provide the clone/checkout url as given by the provider hosting t
  It can easily be found in the web interface used to browse the software origin.
  For instance, if you want to save a `git` origin into the archive, you should check that the command `$ git clone <origin_u`
  does not return an error before submitting a request.

Once submitted, your save request can either be:

- **accepted:** a visit to the provided origin will then be scheduled by Software Heritage in order to load its content into the archive as so
- **rejected:** the provided origin url is blacklisted and no visit will be scheduled
- put in **pending** state: a manual review will then be performed in order to determine if the origin can be safely loaded or not into the

Once a save request has been accepted, you can follow its current status in the **submitted save requests list**.

Advantages

- **All dev history** is also saved
- Urls from **different platforms** are accepted
- PID to **reference** specific pieces of code (even algorithms)

Challenges seen in the **FAIRsFAIR** survey*

Technical challenge:
Software dependencies and environment

Documentation

Accessibility & Licensing

Time & Skill

Quality control

Software sustainability & management plan

**Other challenges are identified in FAIR4RSWG -subgroup1 report arXiv:2101.10883

*the FAIRsFAIR survey presented in D2.1 and captured in https://doi.org/10.5281/zenodo.3518922

Recommendations and adoption

Each recommendations has a requirement level, as defined in RFC2119:

- **MUST** is an absolute requirement
- **SHOULD** is a needed requirement for which exceptions are possible
- **MAY** is an optional requirement

Keep in mind:

1. Any new principle may lead to **extra requirements enforced on researchers**,
2. Researchers are already facing significant challenges when developing or maintaining software, which is a complex and living object.
3. **Clear and immediate benefits** should be offered to the researcher.

Feedback exercise

Each team will answer the following questions on the team’s assigned recommendations

1. **Do you agree?**
   - add +1 or -1 next to the recommendation

2. **How to satisfy this recommendation?**
   - propose actions to satisfy the recommendation
   - propose ways to verify that the recommendation is satisfied

10.5281/zenodo.4095092

10 Recommendations

Team giraffe 🦒

Recommendation n°1

FAIR principles for research software outcomes **MUST** be produced by taking into account the *specific nature of software* and not as just a simple adaptation of the FAIR guiding principles for data.

Recommendation n°2

Applying principles and recommendations to software demands *effort, time and skill*. The **realistic** nature of these principles **MUST** be considered.

Recommendation n°3

A large community forum **MUST** be **consulted** when writing the principles. This community forum **MUST** *include stakeholders* from **different disciplines** and with **different roles**, looking at software in all its aspects: as a *tool*, as a *research outcome* and as *the object of research*.

Recommendation n°4

Existing infrastructures that already provide solutions for software artifacts **SHOULD** be asked to *review the FAIR principles for research software*.

10.5281/zenodo.4095092

10 Recommendations

Recommendation n°5

Each principle **MUST** be *relevant* for software source code.

Recommendation n°6

Each principle **MUST** be *achievable* for software source code.

Team elephant 🐘

Recommendation n°7

Each principle **SHOULD** be *measurable* for software source code; *detailed explanations of how* a measurable principle is measured **MUST** be available.

10.5281/zenodo.4095092

10 Recommendations

Recommendation n°8

Each principle **SHOULD** contribute to software **recognition** in *scholarly communication.*

Recommendation n°9

Each principle **SHOULD** contribute to the *curation quality* of the software resource.

Team zebra 🦓

Recommendation n°10

Each principle **MAY** solve one or more research software *challenges*
(e.g credit, reproducibility, sustainability & management, documentation, quality control, quality metadata, licensing and more).

Feedback exercise

Each team will answer the following questions on the team’s assigned recommendations

1. **Do you agree?**
   - add +1 or -1 next to the recommendation

2. **How to satisfy this recommendation?**
   - propose actions to satisfy the recommendation
   - propose ways to verify that the recommendation is satisfied

How to get involved after today?

Join the

FAIR for Research Software (FAIR4RS) Working Group

Defining FAIR Principles for Research Software

First subgroup 1 output January 2021:

A Fresh Look at FAIR for Research Software

<ReSA>
Research Software Alliance

Join the WG

RDA
RESEARCH DATA ALLIANCE

FORCE11
The Future of Research Communications and e-Scholarship

Steering committee:

Morane Gruenpeter, Paula A. Martinez, Carlos Martinez, Michelle Barker, Daniel S. Katz, Leyla Garcia, Neil Chue Hong, Fotis Psomopoulos and Jennifer Harrow

Thank you for joining us - keep in touch

FAIRsFAIR:
@FAIRsFAIR_EU

https://www.fairsfair.eu/fairsfair-newletters/

Morane:
morane@softwareheritage.org
@moraneottilia, @SWHeritage

https://www.softwareheritage.org/newsletter/

Patricia:
p.herterich@ed.ac.uk
@pherterich, @digitalcuration

https://www.dcc.ac.uk/news/pipeline-newsletter