eosc | FAIR-IMPACT
Expanding FAIR solutions across EOSC

# Developing Guidelines for Metadata Collection and Curation for Research Software

**Morane Gruenpeter**
Software Heritage
Inria, France

eosc | FAIRCORE4EOSC

Funded by the European Union

# Housekeeping

👇

https://rb.gy/u5e04

Or
https://tinyurl.com/2023-05-23-RSMD-webinar

- Please be aware that the session is being recorded and will be made publicly available
- You can add yourself to the list of participants

**Version 1.0**
Lead Author (Org) : **Morane Gruenpeter** (INRIA)
Contributing Author(s) (Org):

- **Sabrina Granger (INRIA),**
- **Alain Monteil (INRIA)**
- **Neil Chue Hong (UEDIN-SSI),**
- **Elena Breitmoser (UEDIN-SSI),**
- **Mario Antonioletti (UEDIN-SSI),**
- **Daniel Garijo (UPM),**
- **Esteban González Guardia (UPM),**
- **Alejandra Gonzalez Beltran (UKRI-STFC),**
- **Carole Goble (UNIMAN),**
- **Stian Soiland-Reyes (UNIMAN),**
- **Gabriela Mejias (DataCite)**

**Contributions during the RDA-P20 Research Software workshop**

# Agenda

- Research Software in the FAIR-IMPACT project
- Preparing the guidelines - a large community effort
- The Research Software MetaData (RSMD) guidelines proposal
- How to contribute after this webinar?

**Goal: Review and discuss the proposed RSMD guidelines**

# FAIR-IMPACT in a nutshell

**Coordination & Support Action**

Budget: 10 million EUR
Time plan: 36 months
Starting date: June 2022
6 Core Partners

CSC

Data Archiving and Networked Services
DANS

DCC

DeiC

INRAE

Trust-IT Services
Communicating ICT to markets

*Strategic cooperation with the EOSC Partnership, ESFRI Clusters, the FAIRCORE4EOSC project*

*…and many others*

# Consortium Partners

Universität Bremen

Trust-IT Services
*Communicating ICT to markets*

cnrs

cessda

DCC

Software Sustainability Institute

Software Heritage
THE GREAT LIBRARY OF SOURCE CODE

**Task 4.3**

Inria

UKRI Science and Technology Facilities Council

DataCite

MANCHESTER 1824
The University of Manchester

POLITÉCNICA

Ontology Engineering Group

SURF

INRAE

EMBL-EBI

*Data Archiving and Networked Services*
DANS

Science & Technology Facilities Council

CSC

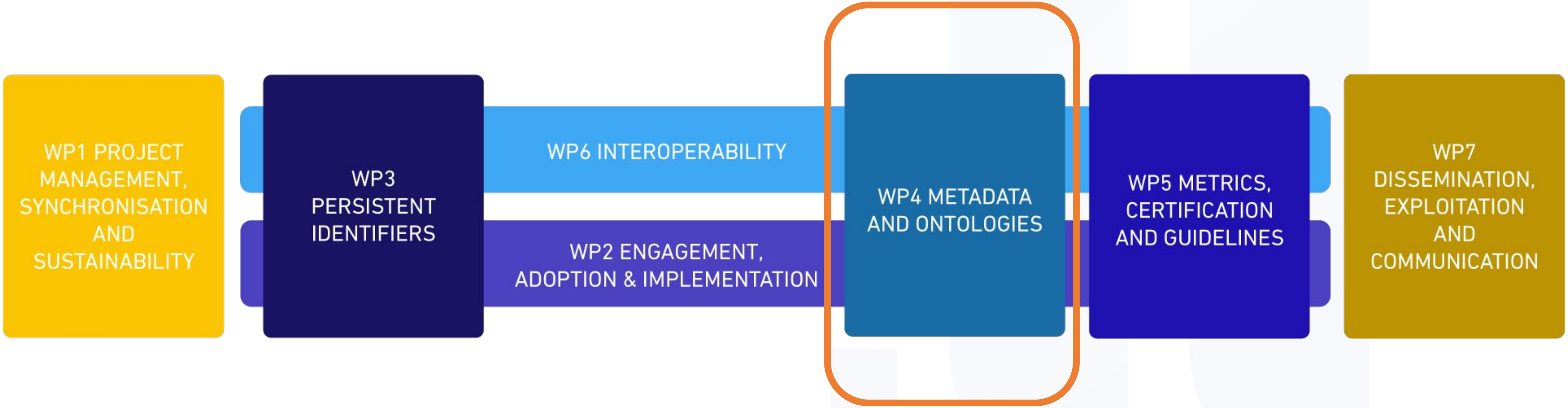
escience data FACTORY

LifeWatch ERIC

DeiC

# FAIR-IMPACT: Expanding FAIR solutions across EOSC

FAIR-IMPACT will identify *practices*, *policies*, *tools* and *technical specifications* to guide researchers, repository managers, research performing organisations, policy makers and citizen scientists towards a FAIR data management cycle. The focus will be on **persistent identifiers (PIDs)**, **metadata**, **ontologies**, **metrics**, **certification** and **interoperability**,


WP1 PROJECT MANAGEMENT, SYNCHRONISATION AND SUSTAINABILITY

WP3 PERSISTENT IDENTIFIERS

WP6 INTEROPERABILITY

WP2 ENGAGEMENT, ADOPTION & IMPLEMENTATION

WP4 METADATA AND ONTOLOGIES

WP5 METRICS, CERTIFICATION AND GUIDELINES

WP7 DISSEMINATION, EXPLOITATION AND COMMUNICATION

# #RSMD_guidelines timeline

June 2022
FAIR-IMPACT
Launch

March 2023
#RS_Workshop
Co-located
RDA P20

RESEARCH DATA ALLIANCE

Research Software
Metadata Guidelines

**May 2023**
**#RSMD_guidelines**
**Community**
**consultation**

May 2023
CodeMeta
V3.0 release

**Due**: June 2023
D4.4 submission

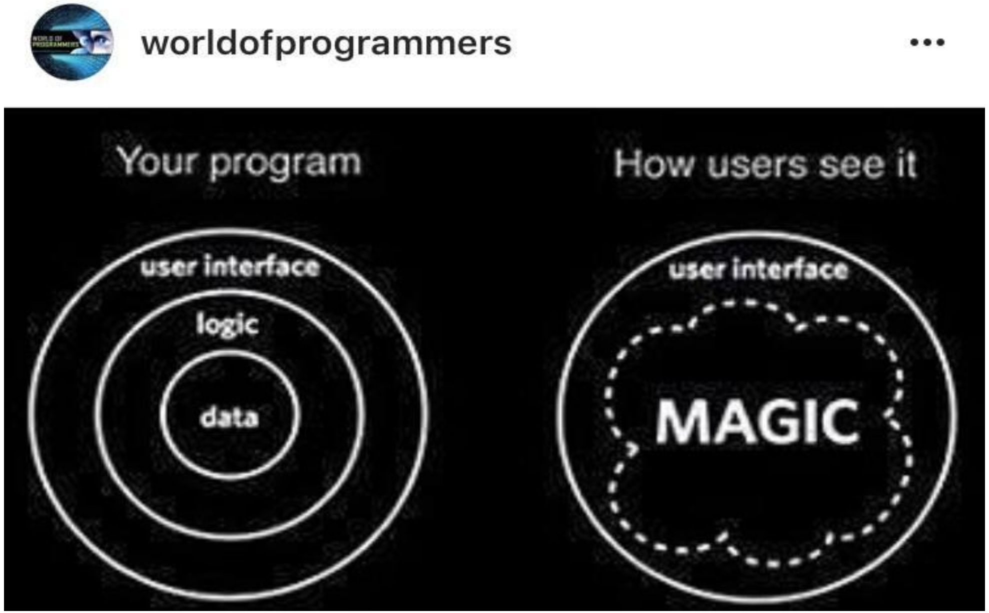
Sept. 2023
CodeMeta
webinar

October 2023
SSC IG session
RDA P21

**2024-2025**
Dissemination
& adoption

eosc | FAIRCORE4EOSC

# Clarifying the magic



https://www.reddit.com/r/ProgrammerHumor/comments/70fuamp/programming_is_magic/

**Software as a concept**

- **project** or entity
- the **community** around the project
- the software **idea** / algorithms / solutions

*Not a digital artifact*

**Software artifacts**

- Executables
- Source code

*A very large collection of digital artifacts*

# Defining Research Software

***Research Software***

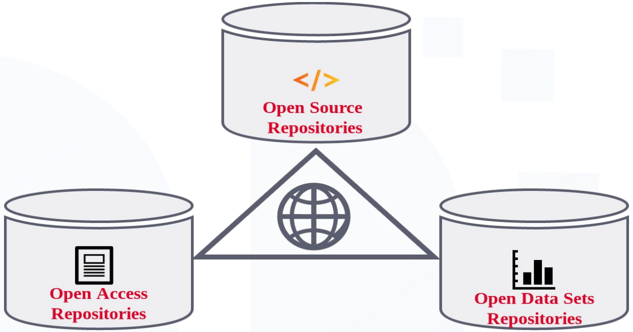
➔ created

- ○ **<u>during the research process</u>**
- ○ **<u>for a research purpose</u>**

***Software in research***

➔ used for research

FAIR4RS output: Gruenpeter et al. Defining Research Software: a controversial discussion (Version 1). Zenodo. https://doi.org/10.5281/zenodo.5504016



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

**Software has multiple facets:**

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

# 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:

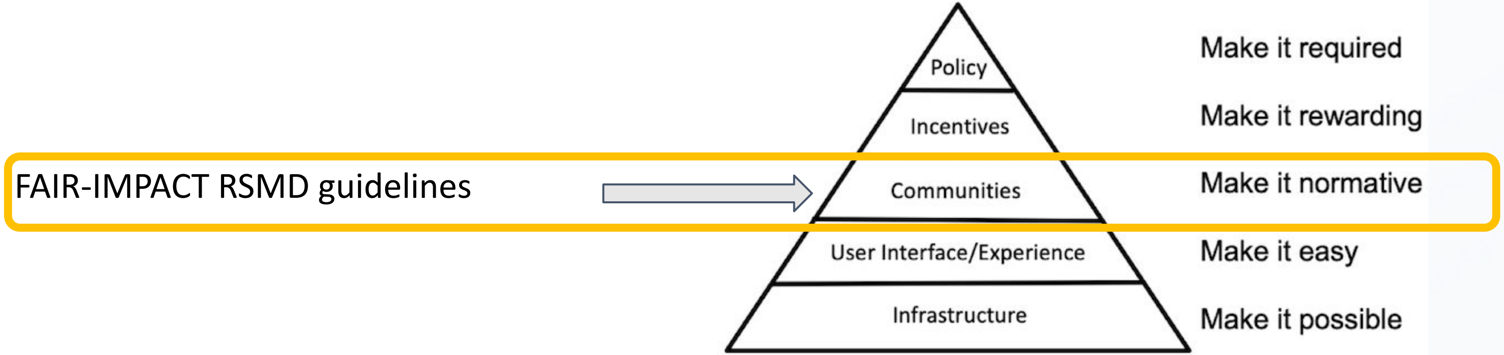
- technology transfer,
- impact metrics,
- Strategy

## Curators

- **verify**  and **curate** software metadata
- **provide** documentation on software curation
- **monitor** research teams' production

# The RSMD guidelines:

## Make it normative



Pyramid from Strategy for Culture Change: **Brian Nosek** (2019) https://www.cos.io/blog/strategy-for-culture-change

# The Research Software MetaData guidelines (RSMD)

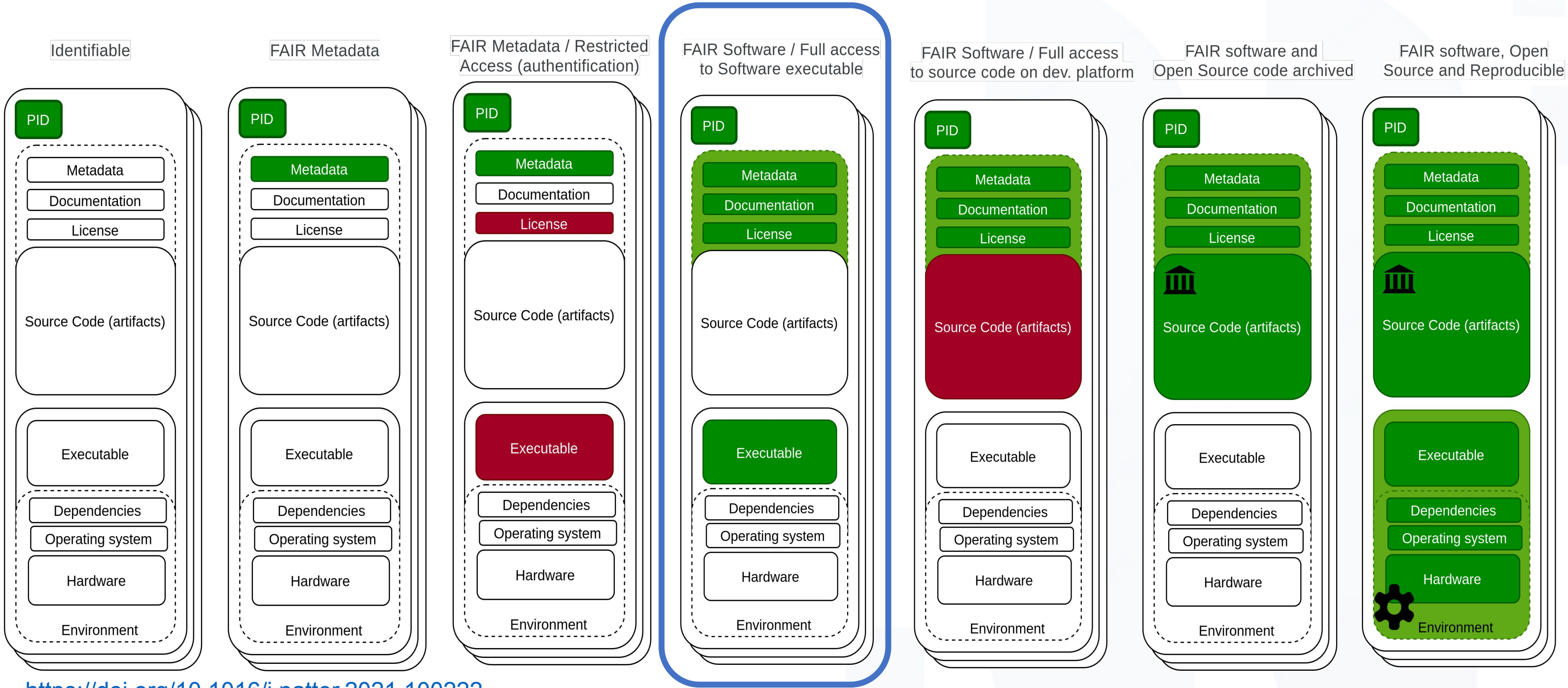
- Introduction
  - Scope & Goals
    - Why are we Here?
    - Who is the intended audience of this document?
  - Methodology
  - Use cases overview
- State of the art
  - Existing practices & guidelines
  - Metadata analysis
  - Curation workflows
- The **RSMD Guidelines** proposal for end-users
- Limitations & challenges
- Conclusion & next Steps
- Appendices
  - RSMD Checklist (only after the recommendations are stable)
  - Use cases collection
  - Infrastructure Using Codemeta

**Link to full deliverable will be shared at the end of the webinar…**

→

Zotero Library for software guidelines
https://www.zotero.org/groups/5018631/fair-impact_t4.3/library

# FAIR4RS principles published in 2021



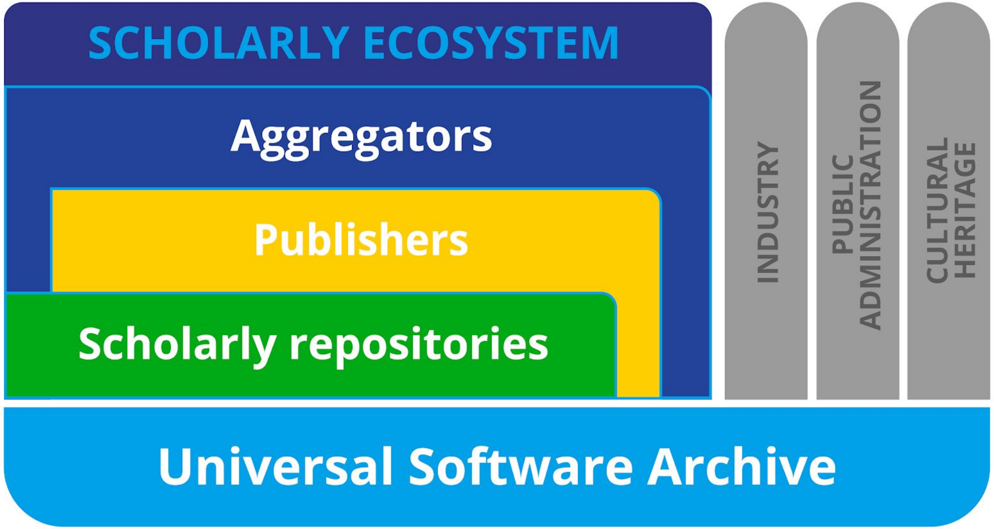
| Identifiable | FAIR Metadata | FAIR Metadata / Restricted Access (authentification) | FAIR Software / Full access to Software executable | FAIR Software / Full access to source code on dev. platform | FAIR software and Open Source code archived | FAIR software, Open Source and Reproducible |
|---|---|---|---|---|---|---|
| 🟩 PID | 🟩 PID | 🟩 PID | 🟩 PID | 🟩 PID | 🟩 PID | 🟩 PID |
| Metadata | 🟩 Metadata | 🟩 Metadata | 🟩 Metadata | 🟩 Metadata | 🟩 Metadata | 🟩 Metadata |
| Documentation | Documentation | Documentation | 🟩 Documentation | 🟩 Documentation | 🟩 Documentation | 🟩 Documentation |
| License | License | 🟥 License | 🟩 License | 🟩 License | 🟩 License | 🟩 License |
| Source Code (artifacts) | Source Code (artifacts) | Source Code (artifacts) | Source Code (artifacts) | 🟥 Source Code (artifacts) | 🟩 🏛 Source Code (artifacts) | 🟩 🏛 Source Code (artifacts) |
| Executable | Executable | 🟥 Executable | 🟩 Executable | Executable | Executable | 🟩 Executable |
| Dependencies | Dependencies | Dependencies | Dependencies | Dependencies | Dependencies | 🟩 Dependencies |
| Operating system | Operating system | Operating system | Operating system | Operating system | Operating system | 🟩 Operating system |
| Hardware | Hardware | Hardware | Hardware | Hardware | Hardware | 🟩 Hardware |
| Environment | Environment | Environment | Environment | Environment | Environment | 🟩 ⚙ Environment |

https://doi.org/10.1016/j.patter.2021.100222

# EOSC SIRS report published in 2020

## Scholarly Infrastructures for Research Software

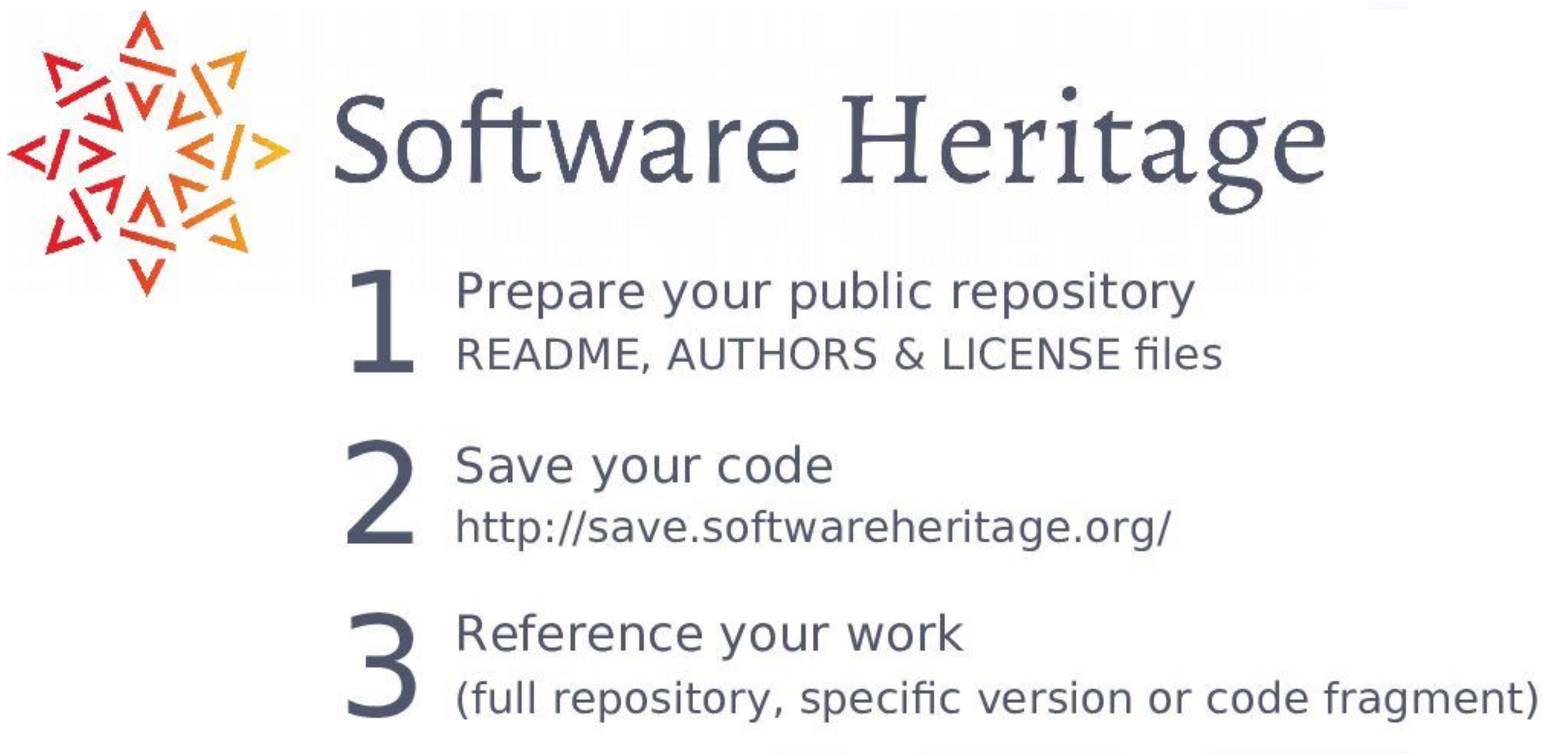
- Four Pillars
  - **Archive, Reference, Describe, Credit**
- State of the Art
  - Best Practices & Open Problems
  - Cross Cutting Concerns
- Participants
  - Representatives from 9 infrastructures:
  - Archives
    - HAL, **Software Heritage, Zenodo**
  - Publishers
    - **Dagstuhl**, eLife, IPOL
  - Aggregators
    - **OpenAIRE**, scanR, **swMATH**



**SIRS report:** European Commission, Directorate-General for Research and Innovation, *Scholarly infrastructures for research software : report from the EOSC Executive Board Working Group (WG) Architecture Task Force (TF) SIRS*, Publications Office, 2020, **https://data.europa.eu/doi/10.2777/28598**
Video: EOSC Software Infrastructures for Research Software: J. B. Gonzalez Lopez (CERN)

**FAIRCORE4EOSC is turning the SIRS report into a reality**
**WG6 creating the component called RSAC -** EOSC Research Software APIs and Connectors

# The guidelines for software archival



Software Heritage

1 Prepare your public repository
README, AUTHORS & LICENSE files

2 Save your code
http://save.softwareheritage.org/

3 Reference your work
(full repository, specific version or code fragment)

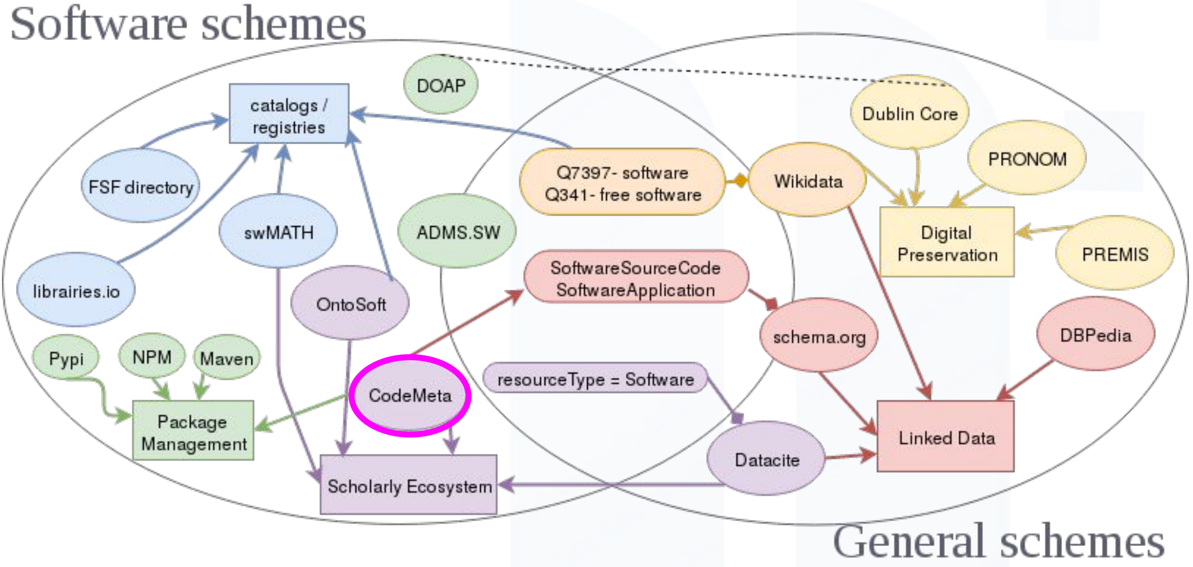
https://www.softwareheritage.org/save-and-reference-research-software/

# CodeMeta initiative - V1.0 released in 2017

- A subset of schema.org
- An academic community discussing software metadata
- A crosswalk table - mapping the metadata landscape

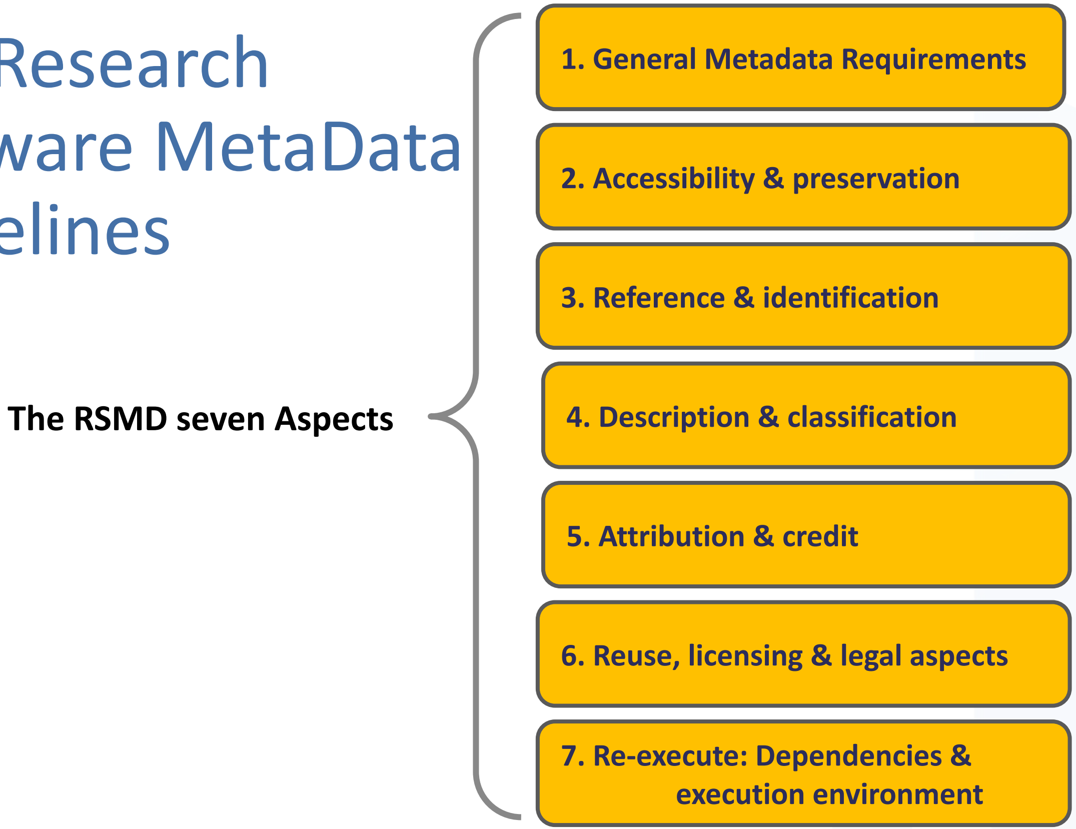
V2.0 released in 2020,

**V3.0 is expected soon!!**



Gruenpeter M. and Thornton K. (2018) Pathways for Discovery of Free Software (slide deck from LibrePlanet 2018). https://en.wikipedia.org/wiki/File:Pathways-discovery-free.pdf

# The Research Software MetaData guidelines



**The RSMD seven Aspects**

1. General Metadata Requirements
2. Accessibility & preservation
3. Reference & identification
4. Description & classification
5. Attribution & credit
6. Reuse, licensing & legal aspects
7. Re-execute: Dependencies & execution environment

**SIRS report**

A = Archive

R = Reference

D = Describe

C = Cite

**FAIR4RS**

A = Accessible

F = Findable

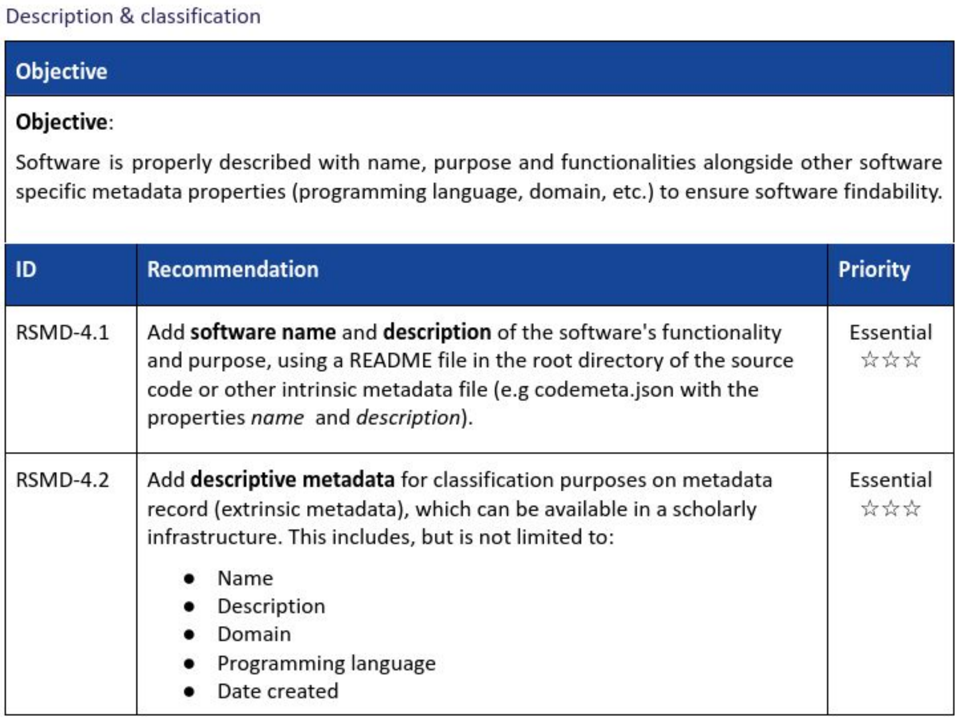
I = Interoperable

R = Reusable

# Each aspect has a high-level objective with a series of recommendations

High-level objective

Actionable, detailed recommendations



Description & classification

| Objective |
|---|
| **Objective**:<br>Software is properly described with name, purpose and functionalities alongside other software specific metadata properties (programming language, domain, etc.) to ensure software findability. |

| ID | Recommendation | Priority |
|---|---|---|
| RSMD-4.1 | Add **software name** and **description** of the software's functionality and purpose, using a README file in the root directory of the source code or other intrinsic metadata file (e.g codemeta.json with the properties *name* and *description*). | Essential<br>☆☆☆ |
| RSMD-4.2 | Add **descriptive metadata** for classification purposes on metadata record (extrinsic metadata), which can be available in a scholarly infrastructure. This includes, but is not limited to:<br>● Name<br>● Description<br>● Domain<br>● Programming language<br>● Date created | Essential<br>☆☆☆ |

# Validating the guidelines

We’ll answer these questions by writing in parallel in the document. **Please comment on others’ answers by using the Google Doc commenting function.**

For each aspect we will review its objective and each recommendation by answering the following questions in the table:

- Is this recommendation **clear**.
- Is this objective/recommendation **relevant** for research software?
  - 👍 / 👎 or +1 / -1

1. General Metadata Requirements
2. Accessibility & preservation
3. Reference & identification
4. Description & classification
5. Attribution & credit
6. Reuse, licensing & legal aspects
7. Re-execute: Dependencies & execution environment

### Objective:

To ensure the collection, curation, and maintenance of research software metadata, the following general requirements are recommended for end users, including researchers, software engineers, curators, and institution staff.

- Is this objective/recommendation **clear**.
- Is this objective/recommendation **relevant** for research software?

1. **General Metadata Requirements**
2. **Accessibility & preservation**
3. **Reference & identification**
4. **Description & classification**
5. **Attribution & credit**
6. **Reuse, licensing & legal aspects**
7. **Re-execute: Dependencies & execution environment**

# Where is the metadata available ? Extrinsic

**Catalogs and registries**

- ASCL
- swMath
- OpenAire
- libraries.io
- Research Software Directory - escience center
- …

**Scholarly repositories**

- Zenodo (InvenioRDM)
- HAL
- …

**Software development platforms (on platform page)**

- GitHub
- Bitbucket
- SourceForge
- …

**Package manager platform (not intrinsic file)**

- NPM
- PyPI
- …

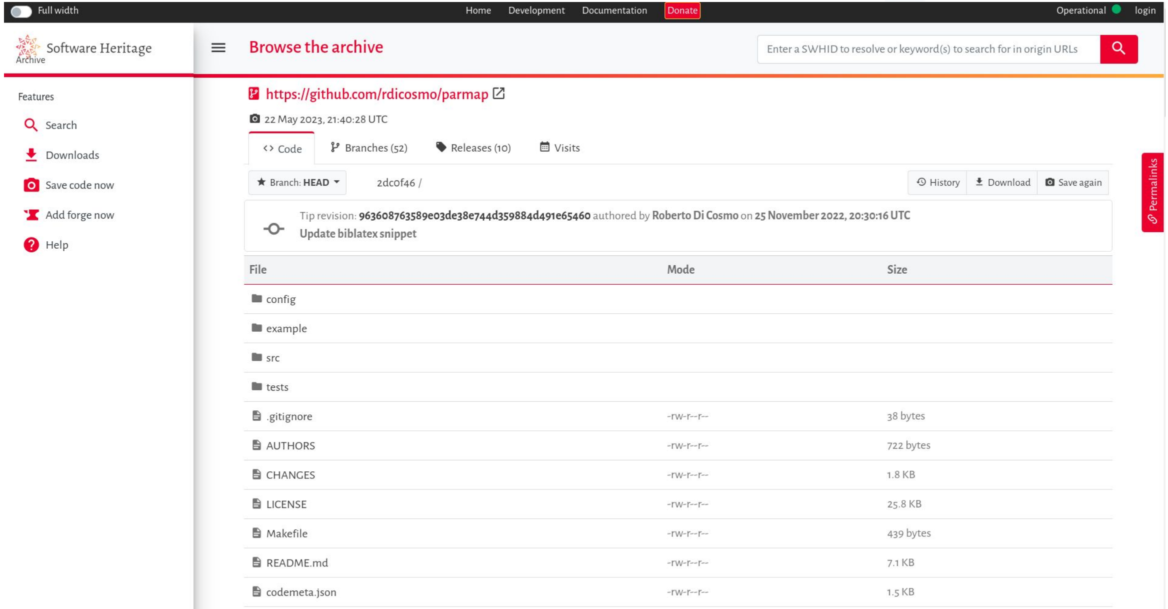
**Scholarly publishers**

- IPOL
- eLife
- Dagstuhl
- Episciences
- …

# The case of intrinsic metadata

## In the *software source code* itself

- README
- LICENSE
- AUTHORS
- **codemeta.json**
- package management
  - pom.xml
  - package.json
  - …
- CITATION.cff
- .About
- …



https://github.com/rdicosmo/parmap

22 May 2023, 21:40:28 UTC

2dc0f46 /

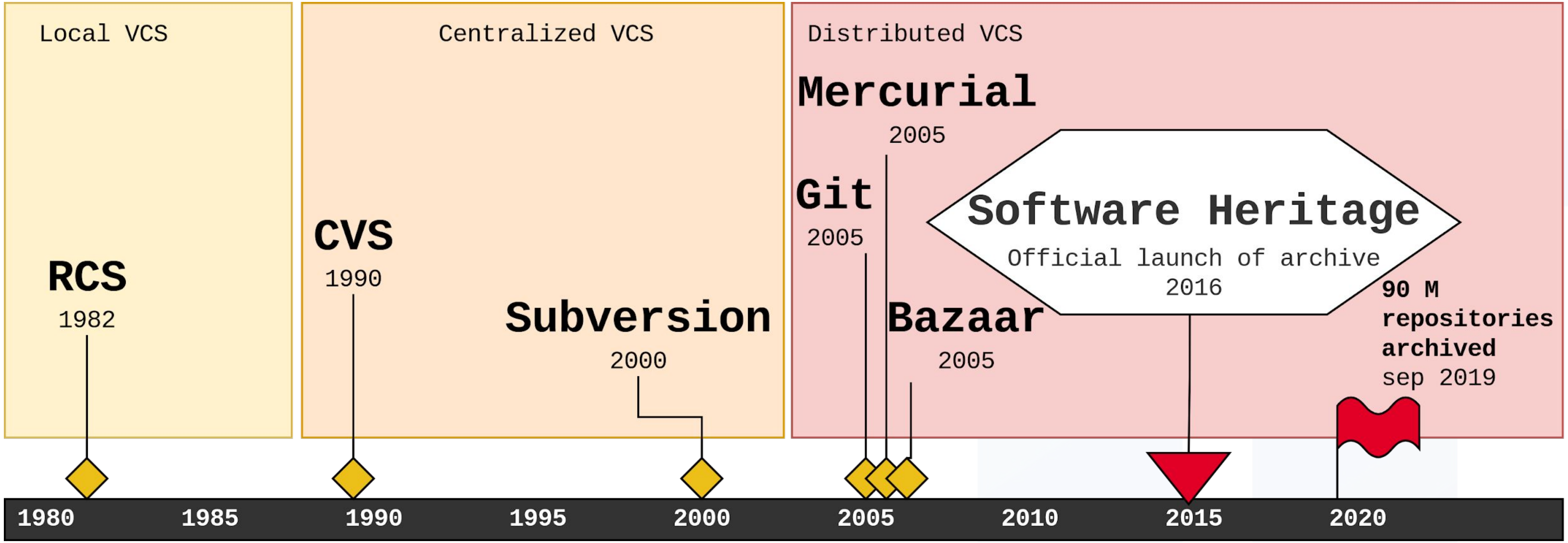
Tip revision: **963608763589e03de38e744d359884d491e65460** authored by **Roberto Di Cosmo** on **25 November 2022, 20:30:16 UTC**
**Update biblatex snippet**

| File | Mode | Size |
|---|---|---|
| config | | |
| example | | |
| src | | |
| tests | | |
| .gitignore | -rw-r--r-- | 38 bytes |
| AUTHORS | -rw-r--r-- | 722 bytes |
| CHANGES | -rw-r--r-- | 1.8 KB |
| LICENSE | -rw-r--r-- | 25.8 KB |
| Makefile | -rw-r--r-- | 439 bytes |
| README.md | -rw-r--r-- | 7.1 KB |
| codemeta.json | -rw-r--r-- | 1.5 KB |

**Human readable (e.g README)**

**Machine actionable (e.g codemeta.json)**

# Version control system (VCS) history



- records changes made to a (set of) source code file (s)
- allows to operate on versions: diff/merge/fork/recover etc.
- essential tool for software development

## Objective:

To ensure accessibility and preservation, researchers and software engineers are strongly recommended to follow the archival and sharing recommendations below.

- Is this objective/recommendation **clear**.
- Is this objective/recommendation **relevant** for research software?

**1. General Metadata Requirements**

**2. Accessibility & preservation**

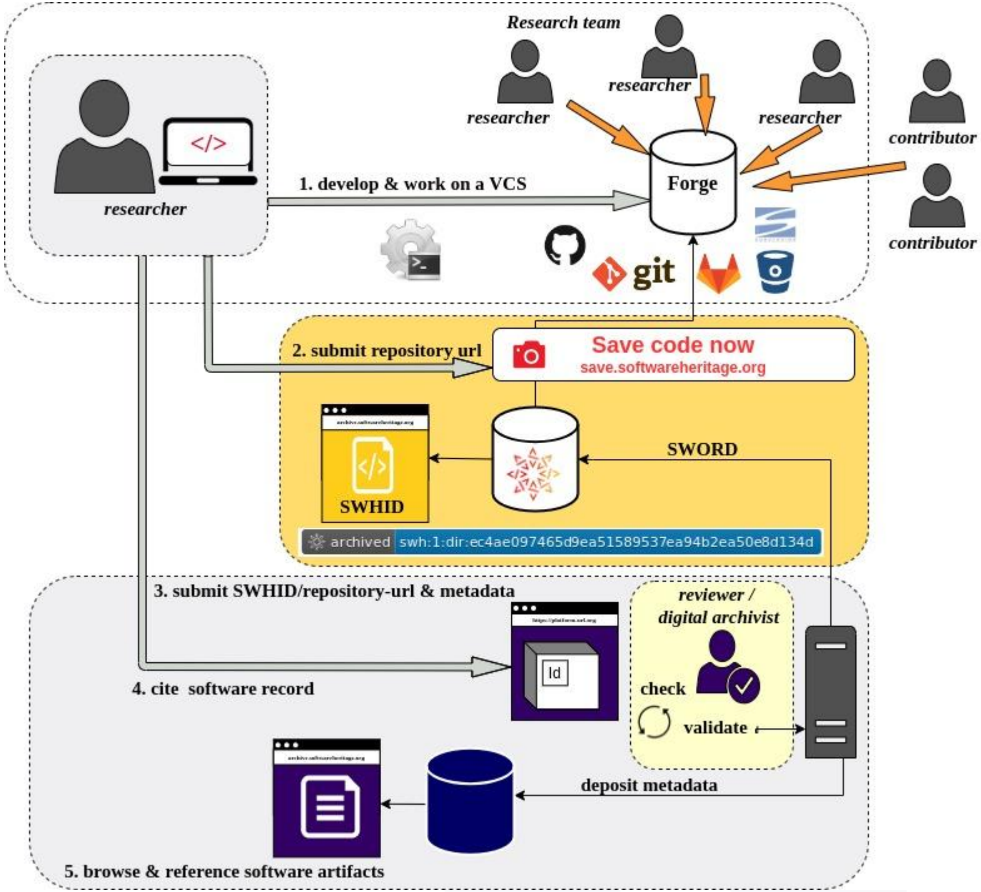
**3. Reference & identification**

**4. Description & classification**

**5. Attribution & credit**

**6. Reuse, licensing & legal aspects**

**7. Re-execute: Dependencies & execution environment**

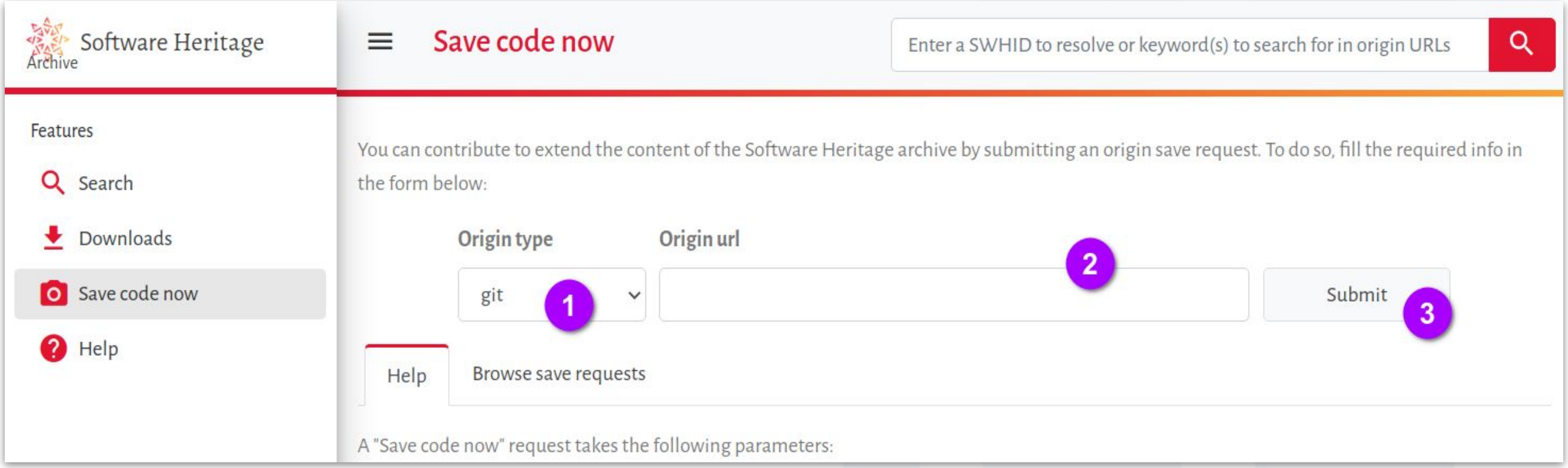

Research team
researcher
researcher
researcher
contributor
contributor
researcher
Forge
1. develop & work on a VCS
git
2. submit repository url
Save code now
save.softwareheritage.org
SWORD
SWHID
archived swh:1:dir:ec4ae097465d9ea51589537ea94b2ea50e8d134d
3. submit SWHID/repository-url & metadata
4. cite software record
Id
reviewer / digital archivist
check
validate
deposit metadata
5. browse & reference software artifacts

# Save any ~~your~~ code now!

Software Heritage
THE GREAT LIBRARY OF SOURCE CODE

https://save.softwareheritage.org/

Save code now



Software Heritage
Archive

Save code now

Enter a SWHID to resolve or keyword(s) to search for in origin URLs

Features

- Search
- Downloads
- Save code now
- Help

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

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

1. General Metadata Requirements
2. Accessibility & preservation
3. Reference & identification
4. Description & classification
5. Attribution & credit
6. Reuse, licensing & legal aspects
7. Re-execute: Dependencies & execution environment

## Objective:

To ensure that research software projects, modules, versions and source code artifacts can be precisely identified and referenced.

- Is this objective/recommendation **clear**.
- Is this objective/recommendation **relevant** for research software?

# Software source code identification

## Software concept / project / collection

Description in registry, a homepage or any other form of metadata record

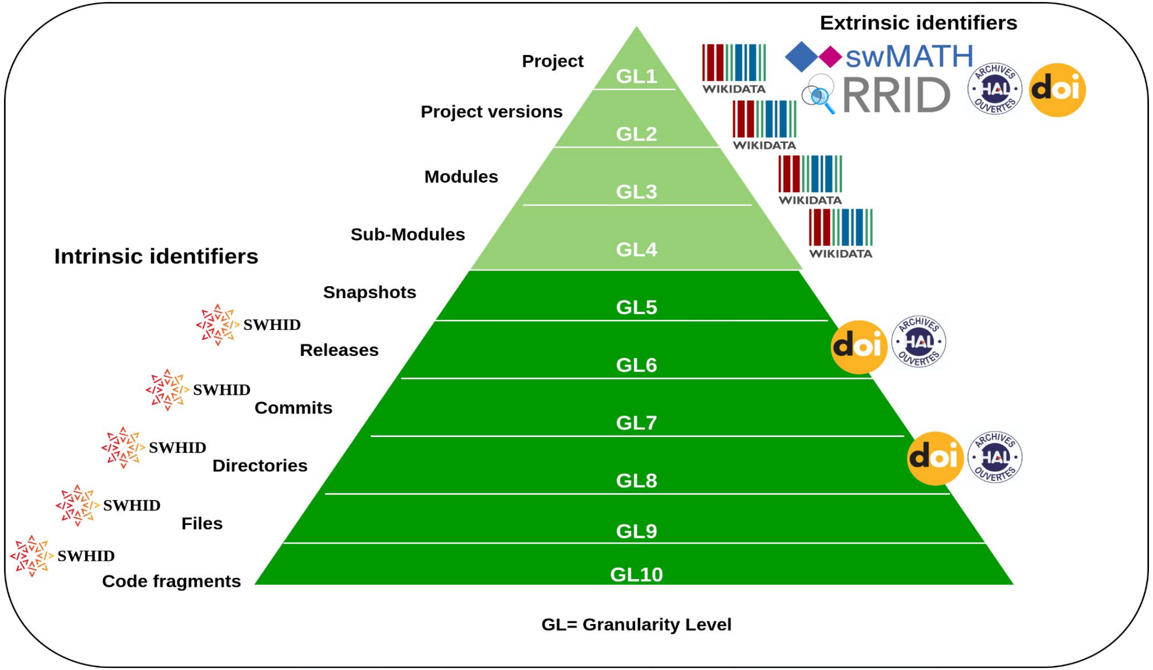
- Project versions (for example Python2 and Python3)
- Modules
- Sub-modules

## Software artifact

- Executable (download link)
- Software source code
  - Dynamic artifact - current development code
  - Archived copy
    - Snapshot (all branches, all dev history)
    - Release / Package
    - Commit- a specific point in development history
    - Directory
    - File
    - Algorithm

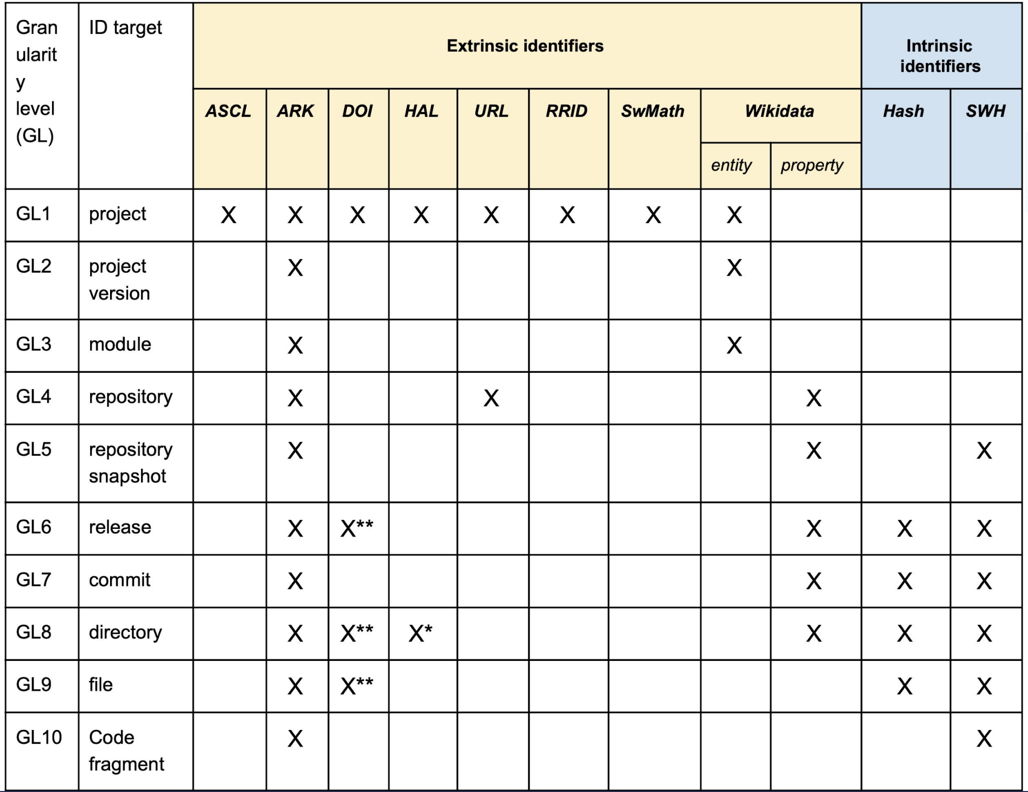
### Software context

- Complementary artifacts - Software artifacts that are external to the source code
  - the software environment, tutorial (Jupyter notebook), Data (input/output data), etc.
- Articles
- Documentation



Research Data Alliance/FORCE11 Software Source Code Identification WG et al. (2020). Use cases and identifier schemes for persistent software source code identification (V1.1). *Research Data Alliance*. https://doi.org/10.15497/RDA00053

# Granularity level summary (SCID output, 2020)



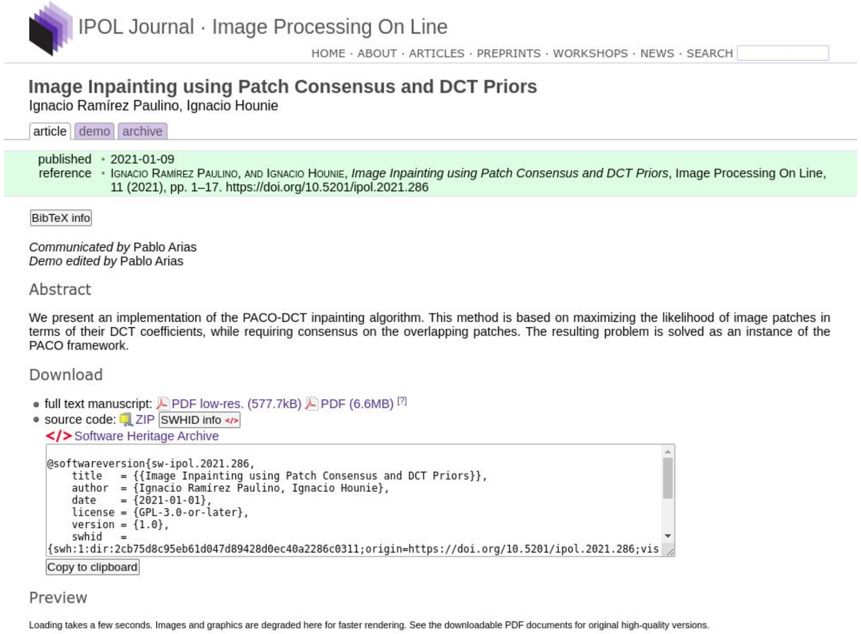
| Granularity level (GL) | ID target | Extrinsic identifiers | | | | | | | | | Intrinsic identifiers | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | *ASCL* | *ARK* | *DOI* | *HAL* | *URL* | *RRID* | *SwMath* | *Wikidata* | | *Hash* | *SWH* |
| | | | | | | | | | *entity* | *property* | | |
| GL1 | project | X | X | X | X | X | X | X | X | | | |
| GL2 | project version | | X | | | | | | X | | | |
| GL3 | module | | X | | | | | | X | | | |
| GL4 | repository | | X | | | X | | | | X | | |
| GL5 | repository snapshot | | X | | | | | | | X | | X |
| GL6 | release | | X | X** | | | | | | X | X | X |
| GL7 | commit | | X | | | | | | | X | X | X |
| GL8 | directory | | X | X** | X* | | | | | X | X | X |
| GL9 | file | | X | X** | | | | | | | X | X |
| GL10 | Code fragment | | X | | | | | | | | | X |

# DOI and SWHID on IPOL articles

- Journal Image Processing On Line (IPOL, https://www.ipol.im/)

- Research software packages are identified with:
  - The article DOI: (https://doi.org/10.5201/ipol.2021.286)
  - The software SWHID: The publisher deposits the software in Software Heritage with the DOI as an origin (https://archive.softwareheritage.org/swh:1:dir:2cb75d8c95eb61d047d89428d0ec40a2286c0311;origin=https://doi.org/10.5201/ipol.2021.286;visit=swh:1:snp:23a5f7ee209b593e9b3e60ebe2bc42f1e6b76ff3;anchor=swh:1:rel:2de235c8fc3dd527cfaaba5cbf1d8144fee14f40)

- Links from the paper and metadata DOI to:
  - the software deposit and its SWHID,
  - the live demo of the software (in the demo tab)



IPOL Journal · Image Processing On Line
HOME · ABOUT · ARTICLES · PREPRINTS · WORKSHOPS · NEWS · SEARCH

**Image Inpainting using Patch Consensus and DCT Priors**
Ignacio Ramírez Paulino, Ignacio Hounie

published • 2021-01-09
reference • Ignacio Ramírez Paulino, and Ignacio Hounie, *Image Inpainting using Patch Consensus and DCT Priors*, Image Processing On Line, 11 (2021), pp. 1–17. https://doi.org/10.5201/ipol.2021.286

*Communicated by* Pablo Arias
*Demo edited by* Pablo Arias

Abstract

We present an implementation of the PACO-DCT inpainting algorithm. This method is based on maximizing the likelihood of image patches in terms of their DCT coefficients, while requiring consensus on the overlapping patches. The resulting problem is solved as an instance of the PACO framework.

Download

- full text manuscript: PDF low-res. (577.7kB) PDF (6.6MB) [?]
- source code: ZIP SWHID info </>

</> Software Heritage Archive

```
@softwareversion{sw-ipol.2021.286,
    title   = {{Image Inpainting using Patch Consensus and DCT Priors}},
    author  = {Ignacio Ramírez Paulino, Ignacio Hounie},
    date    = {2021-01-01},
    license = {GPL-3.0-or-later},
    version = {1.0},
    swhid   =
{swh:1:dir:2cb75d8c95eb61d047d89428d0ec40a2286c0311;origin=https://doi.org/10.5201/ipol.2021.286;vis
```

Preview

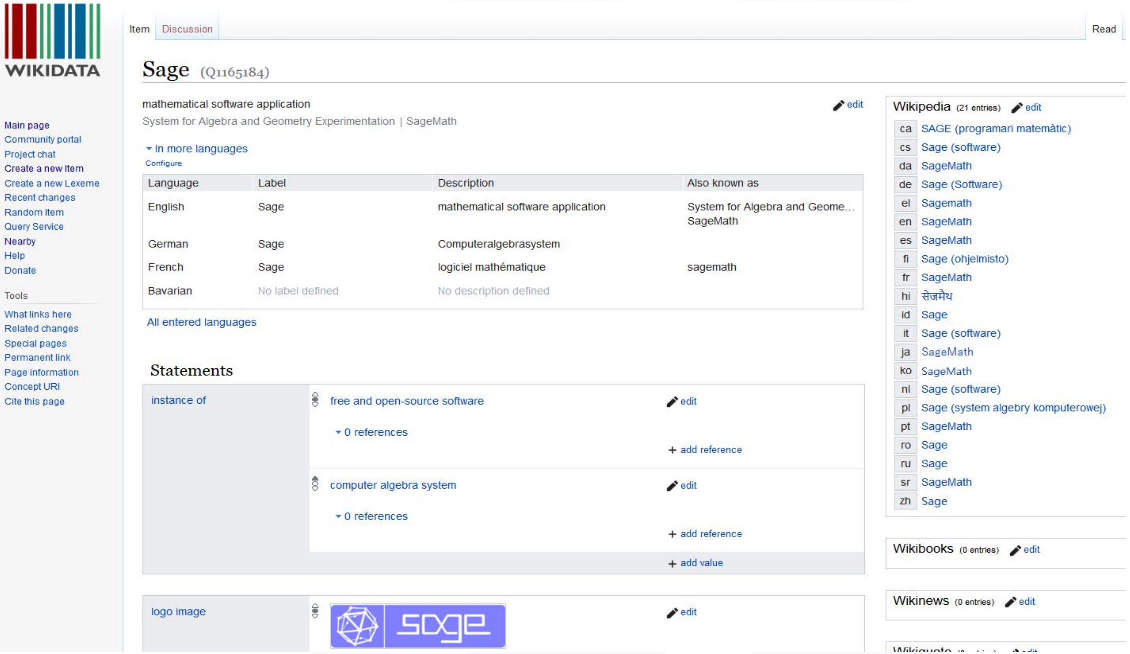
Loading takes a few seconds. Images and graphics are degraded here for faster rendering. See the downloadable PDF documents for original high-quality versions.

# Wikidata entities (Qxxx) - an extrinsic identifier

Q1165184=SageMath.

A few examples of external identifiers properties of used on software entities:

- Arch package sagemath
- Debian stable package sagemath
- Fedora package
- Free Software Directory entry
- Freebase
- Gentoo package
- Open Hub ID sage
- Quora topic
- Ubuntu Package
- swMATH work ID 825
- SWHID snapshot (15.11.2020)
- and many more



Sage (Q1165184)

mathematical software application ✎ edit

System for Algebra and Geometry Experimentation | SageMath

▾ In more languages

Configure

| Language | Label | Description | Also known as |
|---|---|---|---|
| English | Sage | mathematical software application | System for Algebra and Geome…<br>SageMath |
| German | Sage | Computeralgebrasystem | |
| French | Sage | logiciel mathématique | sagemath |
| Bavarian | No label defined | No description defined | |

All entered languages

Statements

| | | |
|---|---|---|
| instance of | free and open-source software<br>▾ 0 references | ✎ edit<br>+ add reference |
| | computer algebra system<br>▾ 0 references | ✎ edit<br>+ add reference |
| | | + add value |
| logo image | sage | ✎ edit |

Wikipedia (21 entries) ✎ edit

| | |
|---|---|
| ca | SAGE (programari matemàtic) |
| cs | Sage (software) |
| da | SageMath |
| de | Sage (Software) |
| el | Sagemath |
| en | SageMath |
| es | SageMath |
| fi | Sage (ohjelmisto) |
| fr | SageMath |
| hi | सेजमैथ |
| id | Sage |
| it | Sage (software) |
| ja | SageMath |
| ko | SageMath |
| nl | Sage (software) |
| pl | Sage (system algebry komputerowej) |
| pt | SageMath |
| ro | Sage |
| ru | Sage |
| sr | SageMath |
| zh | Sage |

Wikibooks (0 entries) ✎ edit

Wikinews (0 entries) ✎ edit

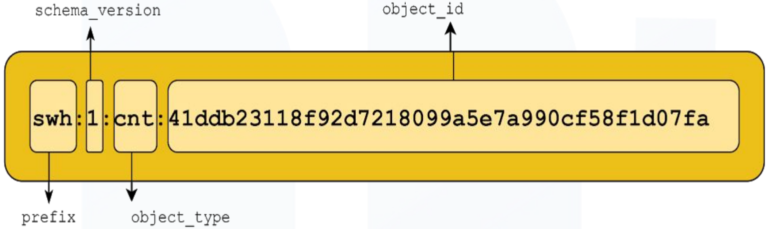
# The SoftWare Heritage ID - a.k.a SWHID

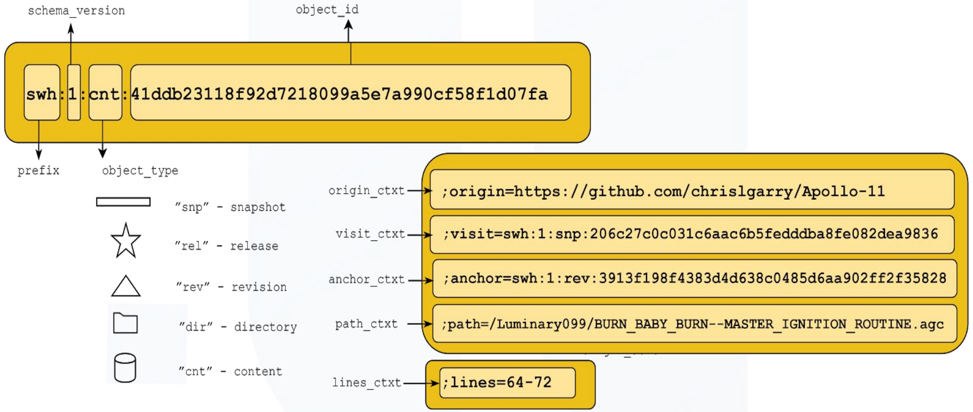
SWH provides a Persistent IDentifier (PID) that can identify each and every source code artifact with integrity, called a SWHID.

SWHIDs are **intrinsic identifiers** which are intimately bound to the designated object, they do not need a register, only agreement on a standard.

[Intrinsic vs. extrinsic blog post]()

Go to [API endpoint]()




schema_version
object_id
swh:1:cnt:41ddb23118f92d7218099a5e7a990cf58f1d07fa
prefix
object_type
"snp" - snapshot
"rel" - release
"rev" - revision
"dir" - directory
"cnt" - content
origin_ctxt
;origin=https://github.com/chrislgarry/Apollo-11
visit_ctxt
;visit=swh:1:snp:206c27c0c031c6aac6b5fedddba8fe082dea9836
anchor_ctxt
;anchor=swh:1:rev:3913f198f4383d4d638c0485d6aa902ff2f35828
path_ctxt
;path=/Luminary099/BURN_BABY_BURN--MASTER_IGNITION_ROUTINE.agc
lines_ctxt
;lines=64-72

Software Heritage
THE GREAT LIBRARY OF SOURCE CODE

## Objective:

To ensure software findability and comprehensibility, provide descriptive metadata (software's name, purpose, functionalities, programming language, domain, etc.). These metadata facilitate accurate representation of the software and enable users to easily discover and understand its capabilities.

- Is this objective/recommendation **clear**.
- Is this objective/recommendation **relevant** for research software?

**1. General Metadata Requirements**

**2. Accessibility & preservation**

**3. Reference & identification**

**4. Description & classification**

**5. Attribution & credit**

**6. Reuse, licensing & legal aspects**

**7. Re-execute: Dependencies & execution environment**

# Describe: What’s a good README

- ★ **MUST** include:
  - ○ Name and a description of the software.
- ★ **SHOULD** include:
  - ○ how to run and use the source code
  - ○ build environment, installation, requirements
- ★ **CAN** include:
  - ○ project website or documentation pointer and recent news
  - ○ visuals

extracted from Eric Steven Raymond and Make a README

# CodeMeta properties

## Identify

- identifier
- name
- author
- version, softwareVersion

## Execute

- codeRepository
- operatingSystem
- softwareRequirements
- buildInstructions
- **Not in CodeMeta:**
  a. Examples
  b. Compiler
  c. Executable link
  d. Other documentation

## Classify

- description
- releaseNotes
- keywords
- supportingData (in/out data)
- fileFormat
- programmingLanguage
- **Not in CodeMeta:**
  a. references
  b. algorithms

## Administrate

- maintainer
- copyrightHolder
- funder
- license
- editor
- publisher
- dateCreated
- dateModified
- datePublished
- developmentStatus

## Objective:

To ensure proper crediting and acknowledgment of software creators, authors, and contributors, it is important to follow citation recommendations.

- Is this objective/recommendation **clear**.
- Is this objective/recommendation **relevant** for research software?

**1. General Metadata Requirements**

**2. Accessibility & preservation**

**3. Reference & identification**

**4. Description & classification**

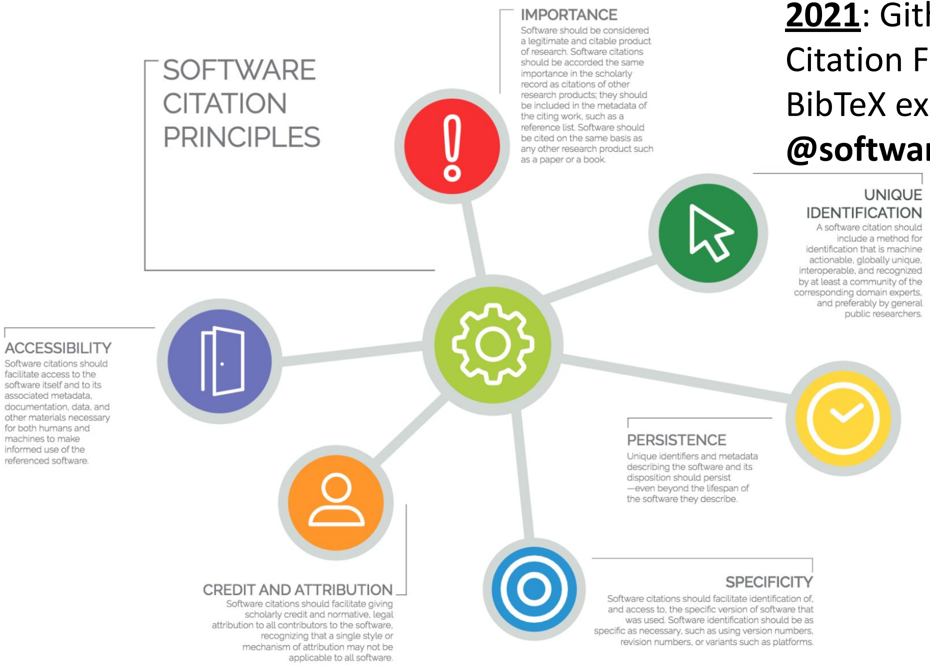
**5. Attribution & credit**

**6. Reuse, licensing & legal aspects**

**7. Re-execute: Dependencies & execution environment**

# Software citation principles - published in 2016

Software is a critical part of modern research...



SOFTWARE CITATION PRINCIPLES

IMPORTANCE
Software should be considered a legitimate and citable product of research. Software citations should be accorded the same importance in the scholarly record as citations of other research products; they should be included in the metadata of the citing work, such as a reference list. Software should be cited on the same basis as any other research product such as a paper or a book.

UNIQUE IDENTIFICATION
A software citation should include a method for identification that is machine actionable, globally unique, interoperable, and recognized by at least a community of the corresponding domain experts, and preferably by general public researchers.

PERSISTENCE
Unique identifiers and metadata describing the software and its disposition should persist —even beyond the lifespan of the software they describe.

SPECIFICITY
Software citations should facilitate identification of, and access to, the specific version of software that was used. Software identification should be as specific as necessary, such as using version numbers, revision numbers, or variants such as platforms.

CREDIT AND ATTRIBUTION
Software citations should facilitate giving scholarly credit and normative, legal attribution to all contributors to the software, recognizing that a single style or mechanism of attribution may not be applicable to all software.

ACCESSIBILITY
Software citations should facilitate access to the software itself and to its associated metadata, documentation, data, and other materials necessary for both humans and machines to make informed use of the referenced software.

... yet there is little support for its acknowledgement and citation

**<u>2021</u>**: Github & CFF - Citation File Format: BibTeX export with type **@software**

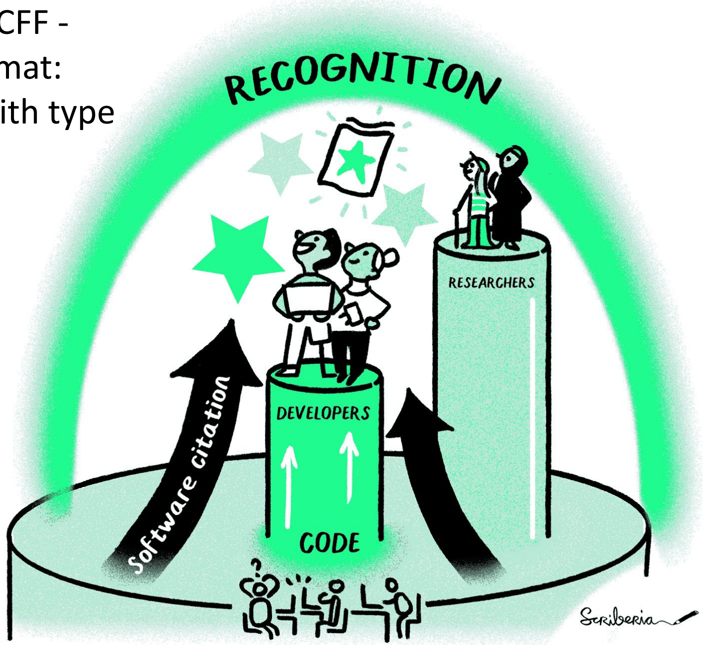


*Fig. 69* Research software developers get recognition by making software citable. *The Turing Way* project illustration by Scriberia. Zenodo. http://doi.org/10.5281/zenodo.3332807

## Objective:

To ensure proper software reuse and license compliance, it is essential to accurately describe software licensing and legal aspects. This includes providing clear guidance on proper usage and distribution rights, clarifying the terms and conditions under which the software can be used and shared.

- Is this objective/recommendation **clear**.
- Is this objective/recommendation **relevant** for research software?

**1. General Metadata Requirements**

**2. Accessibility & preservation**

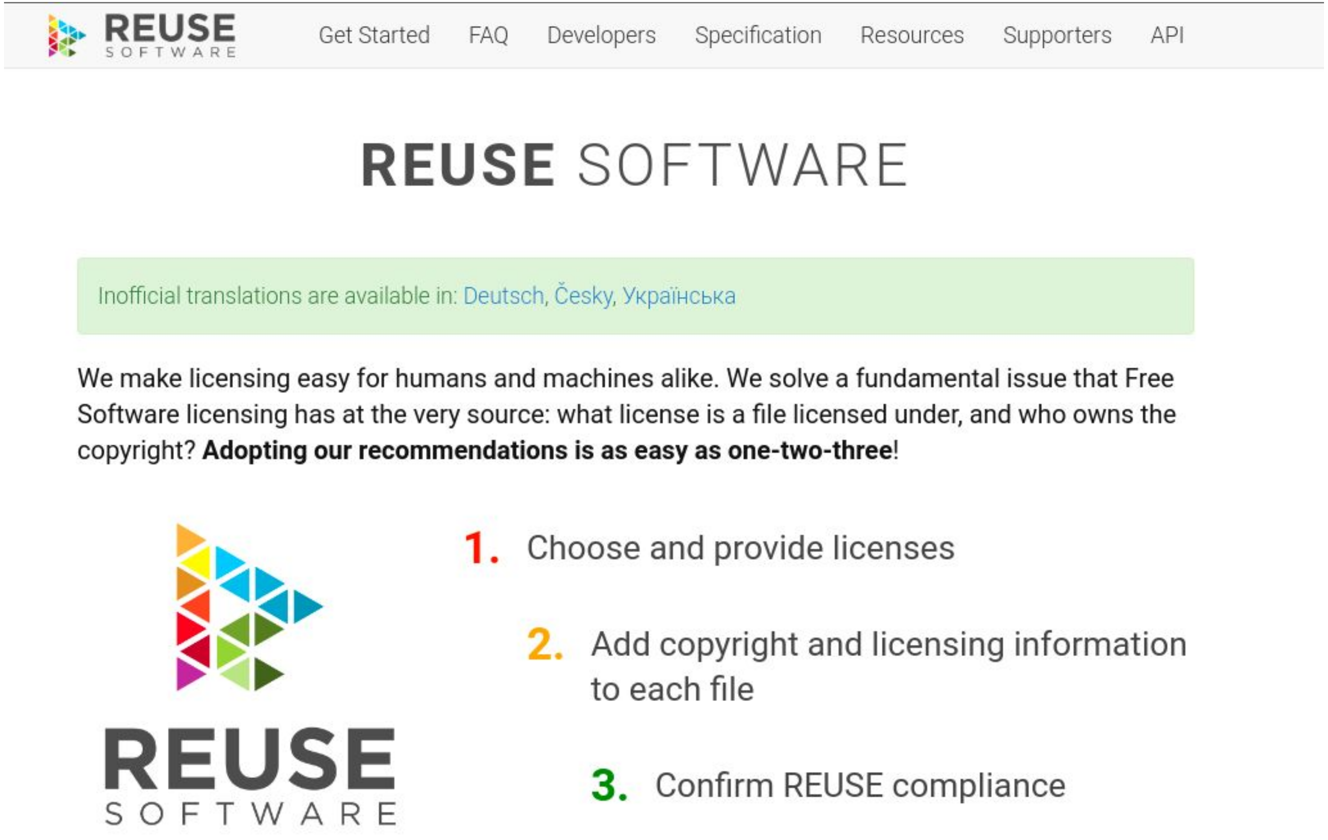
**3. Reference & identification**

**4. Description & classification**

**5. Attribution & credit**

**6. Reuse, licensing & legal aspects**

**7. Re-execute: Dependencies & execution environment**



# REUSE SOFTWARE

Inofficial translations are available in: Deutsch, Česky, Українська

We make licensing easy for humans and machines alike. We solve a fundamental issue that Free Software licensing has at the very source: what license is a file licensed under, and who owns the copyright? **Adopting our recommendations is as easy as one-two-three**!

1. Choose and provide licenses
2. Add copyright and licensing information to each file
3. Confirm REUSE compliance

SPDX

## Objective:

To ensure the usability of software and the ability to reproduce the same results in experiments, it is important that the software can be easily rebuilt and executed. This ensures that others can use the software effectively and achieve consistent outcomes.

- Is this objective/recommendation **clear**.
- Is this objective/recommendation **relevant** for research software?

**1. General Metadata Requirements**

**2. Accessibility & preservation**

**3. Reference & identification**

**4. Description & classification**

**5. Attribution & credit**

**6. Reuse, licensing & legal aspects**

**7. Re-execute: Dependencies & execution environment**

# Conclusion and next steps

D4.4 Research Software Metadata guidelines - deadline for the review (**May 29th at 12.00 UTC**)

- How to review and comment the deliverables?
  - Comment in the live notes
    - open until **May 25th at 12.00 UTC**
    - https://tinyurl.com/2023-05-23-RSMD-webinar
  - Review first draft of the deliverable V1.0 (which includes state of the art):
    - open until **May 29th at 12.00 UTC**
    - https://tinyurl.com/RSMD-guidelines-v1
- Contribute to CodeMeta! (participate in the community discussion)
  - V3.0 of the vocabulary is expected at the end of the month
  - A dedicated webinar will be scheduled for September 2023

# Making the RSMD guidelines useful

How can we make the RSMD guidelines format of most value to you?

Keep in touch: morane@softwareheritage.org
@moraneottilia, @SWHeritage

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

eosc | FAIR-IMPACT
Expanding FAIR solutions across EOSC

@fairimpact_eu /company/fair-impact-eu-projec
t

Funded by
the European Union