FAIRSFAIR
Fostering Fair Data Practices in Europe

FAIR + Software: decoding the principles

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

Outline

- Introduction
- Software source code a (forgotten) pillar of research
- Existing mechanisms, components and infrastructures
- FAIR ecosystem
- Literature analysis
- 10 meta-recommendations for the FAIR software definition

10.5281/zenodo.4095092

Milestone 2.15 `FAIRness of software`

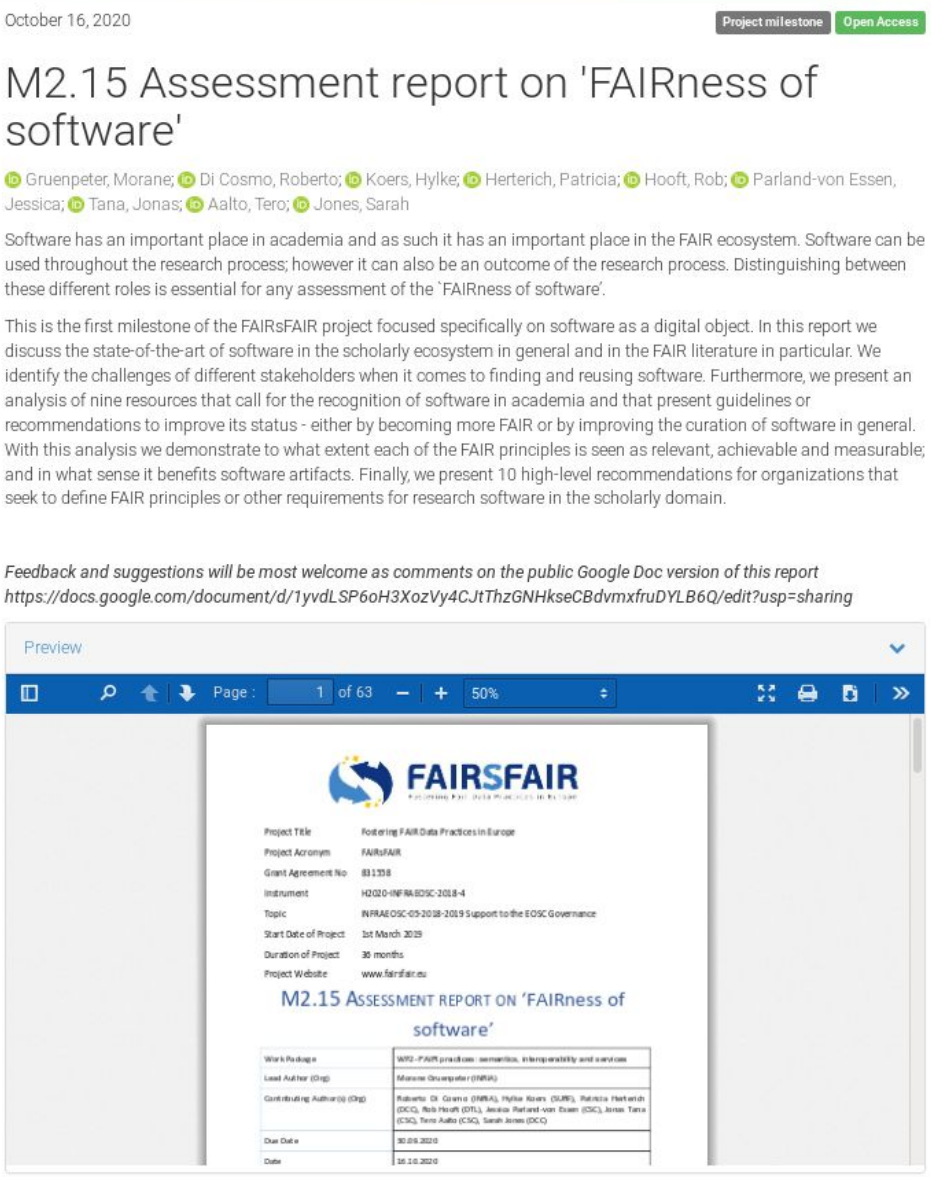
Version 1.1

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

- **Roberto Di Cosmo (INRIA),**
- **Hylke Koers (SURF),**
- **Patricia Herterich (DCC),**
- **Rob Hooft (DTL),**
- **Jessica Parland-von Essen (CSC),**
- **Jonas Tana (CSC),**
- **Tero Aalto (CSC),**
- **Sarah Jones (DCC)**

Reviewed by experts from the FAIR4RS Working Group

Open consultation document



October 16, 2020 Project milestone Open Access

M2.15 Assessment report on 'FAIRness of software'

Gruenpeter, Morane; Di Cosmo, Roberto; Koers, Hylke; Herterich, Patricia; Hooft, Rob; Parland-von Essen, Jessica; Tana, Jonas; Aalto, Tero; Jones, Sarah

Software has an important place in academia and as such it has an important place in the FAIR ecosystem. Software can be used throughout the research process; however it can also be an outcome of the research process. Distinguishing between these different roles is essential for any assessment of the `FAIRness of software'.

This is the first milestone of the FAIRsFAIR project focused specifically on software as a digital object. In this report we discuss the state-of-the-art of software in the scholarly ecosystem in general and in the FAIR literature in particular. We identify the challenges of different stakeholders when it comes to finding and reusing software. Furthermore, we present an analysis of nine resources that call for the recognition of software in academia and that present guidelines or recommendations to improve its status - either by becoming more FAIR or by improving the curation of software in general. With this analysis we demonstrate to what extent each of the FAIR principles is seen as relevant, achievable and measurable; and in what sense it benefits software artifacts. Finally, we present 10 high-level recommendations for organizations that seek to define FAIR principles or other requirements for research software in the scholarly domain.

Feedback and suggestions will be most welcome as comments on the public Google Doc version of this report
https://docs.google.com/document/d/1yvdLSP6oH3XozVy4CJtThzGNHkseCBdvmxfruDYLB6Q/edit?usp=sharing

Preview

Page : 1 of 63 50%

FAIRSFAIR

M2.15 Assessment report on 'FAIRness of software'

Goals for today

- Review different **mechanisms, components and infrastructures** that can improve the FAIRness of software in the scholarly ecosystem (section 5)
- Present the analysis if the FAIR principles are seen **relevant, achievable and measurable** when it comes to software in the literature (section 3)
- Introduce **10 high-level recommendations** for future work to define FAIR principles for research software (section 6)

Outline

- Introduction
- Software source code a (forgotten) pillar of research
- Existing mechanisms, components and infrastructures
- FAIR ecosystem
- Literature analysis
- 10 meta-recommendations for the FAIR software definition

What is software?

*"**Software**, instructions that tell a computer what to do. Software comprises the entire set of programs, procedures, and routines associated with the operation of a computer system. The term was coined to differentiate these instructions from hardware—i.e., the physical components of a computer system.*
"

Encyclopædia Britannica Access date: November 18, 2020
https://www.britannica.com/technology/software

Software as a concept

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

Software artifact

- each revision in source code form
- binaries produced for different environments

Software is all around us

Apollo 11 Guidance Computer (~60.000 lines), 1969

"When I first got into it, nobody knew what it was that we were doing. It was like the Wild West."

Margaret Hamilton

Tim Berners-Lee invented the **World Wide Web, 1989**, while working at CERN on a NeXT machine

“The Semantic Web is not a separate Web but an extension of the current one, in which information is given well-defined meaning, better enabling computers and people to work in cooperation."

Berners-Lee, T., Hendler, J., & Lassila, O. (2001, May 17).

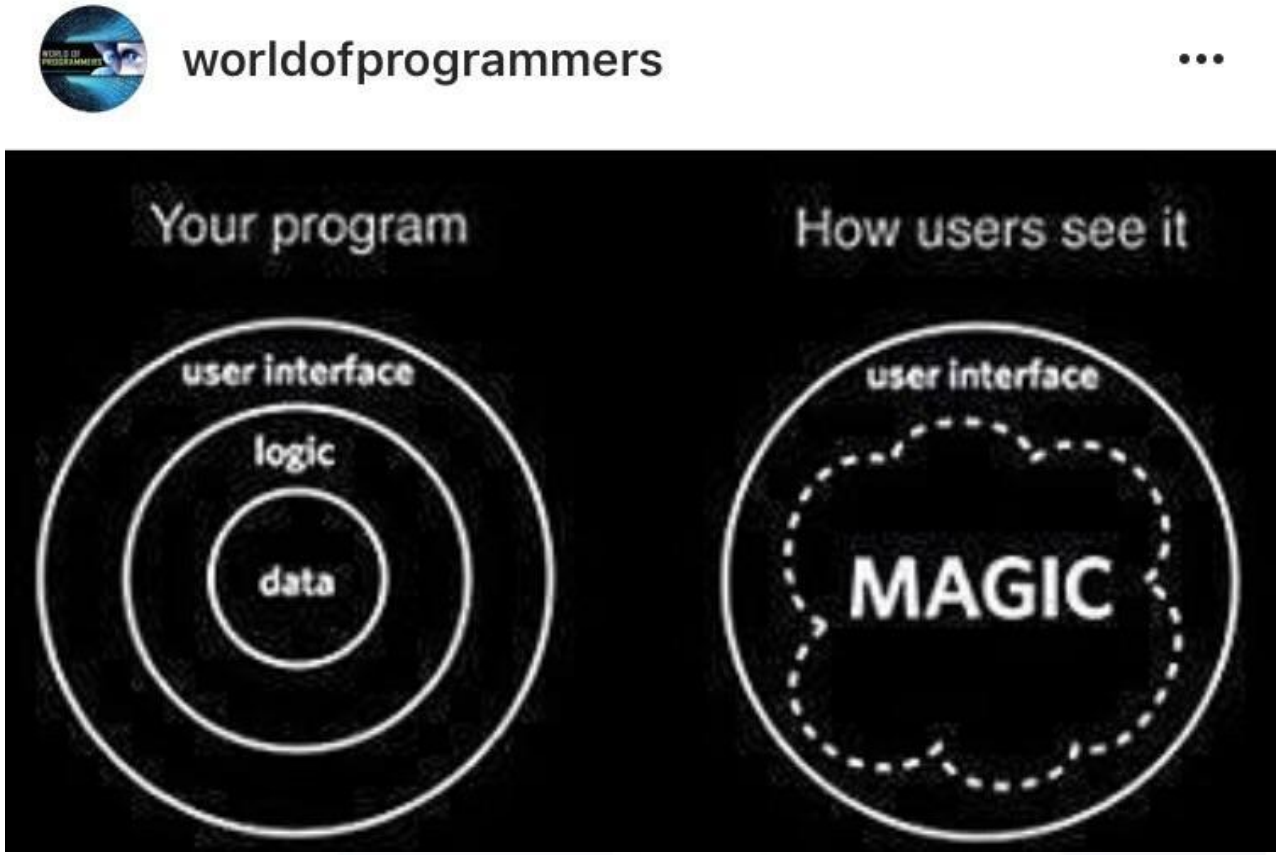
Software Source Code is special (not just data)

Software evolves over time

- projects may last decades
- the development history is key to its understanding

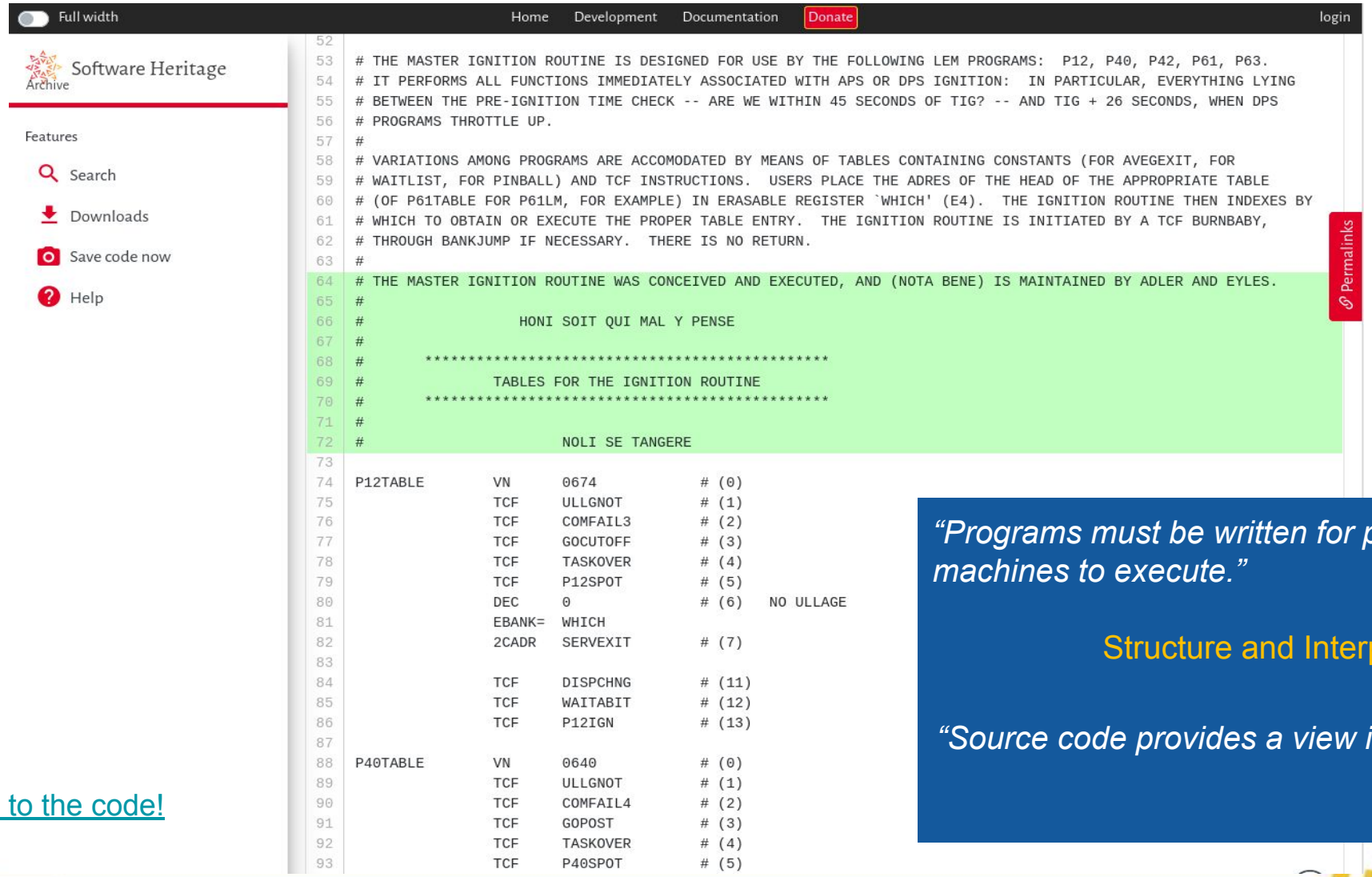
Complexity

- millions of lines of code
- large web of dependencies
  - easy to break, difficult to maintain
- sophisticated developer communities



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

Software Source Code human readable and executable knowledge




```
52
53  # THE MASTER IGNITION ROUTINE IS DESIGNED FOR USE BY THE FOLLOWING LEM PROGRAMS:  P12, P40, P42, P61, P63.
54  # IT PERFORMS ALL FUNCTIONS IMMEDIATELY ASSOCIATED WITH APS OR DPS IGNITION:  IN PARTICULAR, EVERYTHING LYING
55  # BETWEEN THE PRE-IGNITION TIME CHECK -- ARE WE WITHIN 45 SECONDS OF TIG? -- AND TIG + 26 SECONDS, WHEN DPS
56  # PROGRAMS THROTTLE UP.
57  #
58  # VARIATIONS AMONG PROGRAMS ARE ACCOMODATED BY MEANS OF TABLES CONTAINING CONSTANTS (FOR AVEGEXIT, FOR
59  # WAITLIST, FOR PINBALL) AND TCF INSTRUCTIONS.  USERS PLACE THE ADRES OF THE HEAD OF THE APPROPRIATE TABLE
60  # (OF P61TABLE FOR P61LM, FOR EXAMPLE) IN ERASABLE REGISTER `WHICH' (E4).  THE IGNITION ROUTINE THEN INDEXES BY
61  # WHICH TO OBTAIN OR EXECUTE THE PROPER TABLE ENTRY.  THE IGNITION ROUTINE IS INITIATED BY A TCF BURNBABY,
62  # THROUGH BANKJUMP IF NECESSARY.  THERE IS NO RETURN.
63  #
64  # THE MASTER IGNITION ROUTINE WAS CONCEIVED AND EXECUTED, AND (NOTA BENE) IS MAINTAINED BY ADLER AND EYLES.
65  #
66  #                 HONI SOIT QUI MAL Y PENSE
67  #
68  #       ***********************************************
69  #               TABLES FOR THE IGNITION ROUTINE
70  #       ***********************************************
71  #
72  #                     NOLI SE TANGERE
73
74  P12TABLE        VN      0674            # (0)
75                  TCF     ULLGNOT         # (1)
76                  TCF     COMFAIL3        # (2)
77                  TCF     GOCUTOFF        # (3)
78                  TCF     TASKOVER        # (4)
79                  TCF     P12SPOT         # (5)
80                  DEC     0               # (6)   NO ULLAGE
81                  EBANK=  WHICH
82                  2CADR   SERVEXIT        # (7)
83
84                  TCF     DISPCHNG        # (11)
85                  TCF     WAITABIT        # (12)
86                  TCF     P12IGN          # (13)
87
88  P40TABLE        VN      0640            # (0)
89                  TCF     ULLGNOT         # (1)
90                  TCF     COMFAIL4        # (2)
91                  TCF     GOPOST          # (3)
92                  TCF     TASKOVER        # (4)
93                  TCF     P40SPOT         # (5)
```

[Go to the code!](#)

"Programs must be written for people to read, and only incidentally for machines to execute."

Harold Abelson, 1985
Structure and Interpretation of Computer Programs (1st ed.),

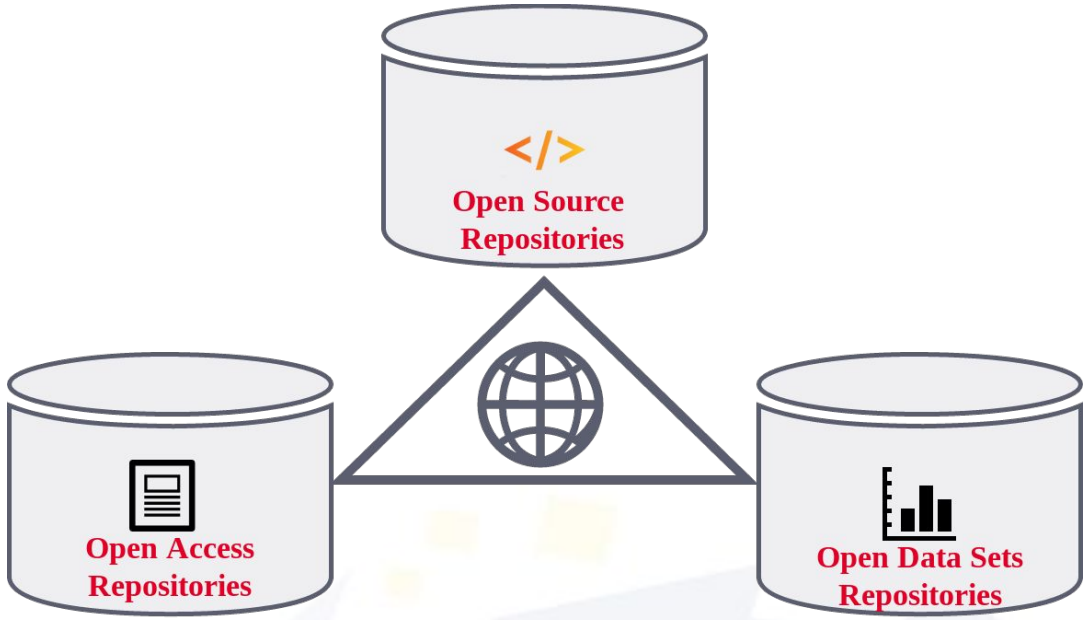
"Source code provides a view into the mind of the designer."

Len Shustek, 2006
Computer History Museum

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

Outline

- Introduction
- Software source code a (forgotten) pillar of research
- Existing mechanisms, components and infrastructures
- FAIR ecosystem
- Literature analysis
- 10 meta-recommendations for the FAIR software definition

The landscape of Existing Mechanisms and Components

- Software Identification
  - extrinsic: ASCL-ID, ARK, DOI, RRID, swMath-ID, Wikidata
  - intrinsic: SWHID
- Metadata: CodeMeta
- Software licenses and SPDX
- Software curation
- Software artifact evaluation and badging
  - AEC
  - ACM
  - NISO

Software Identification: what target to identify?

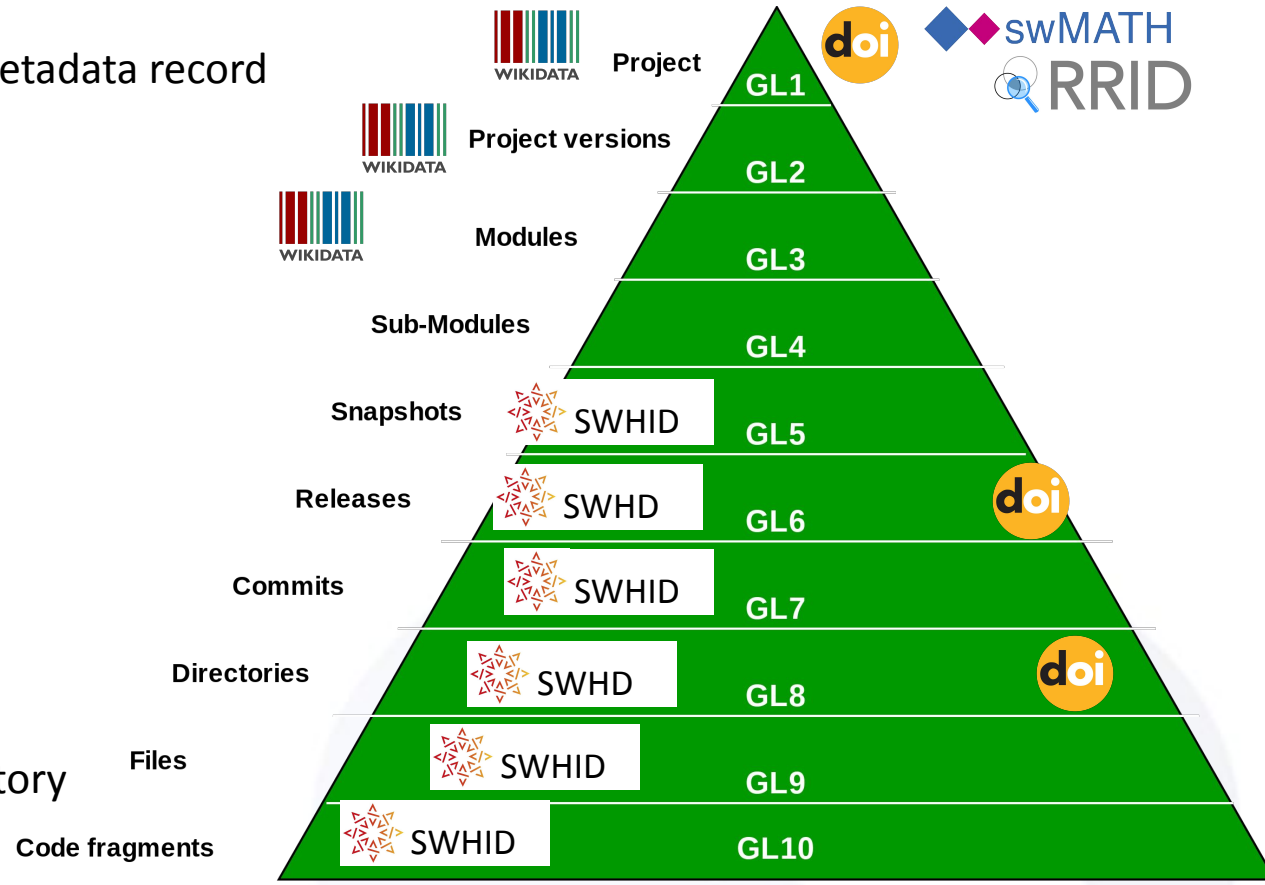
Software concept / project / collection

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



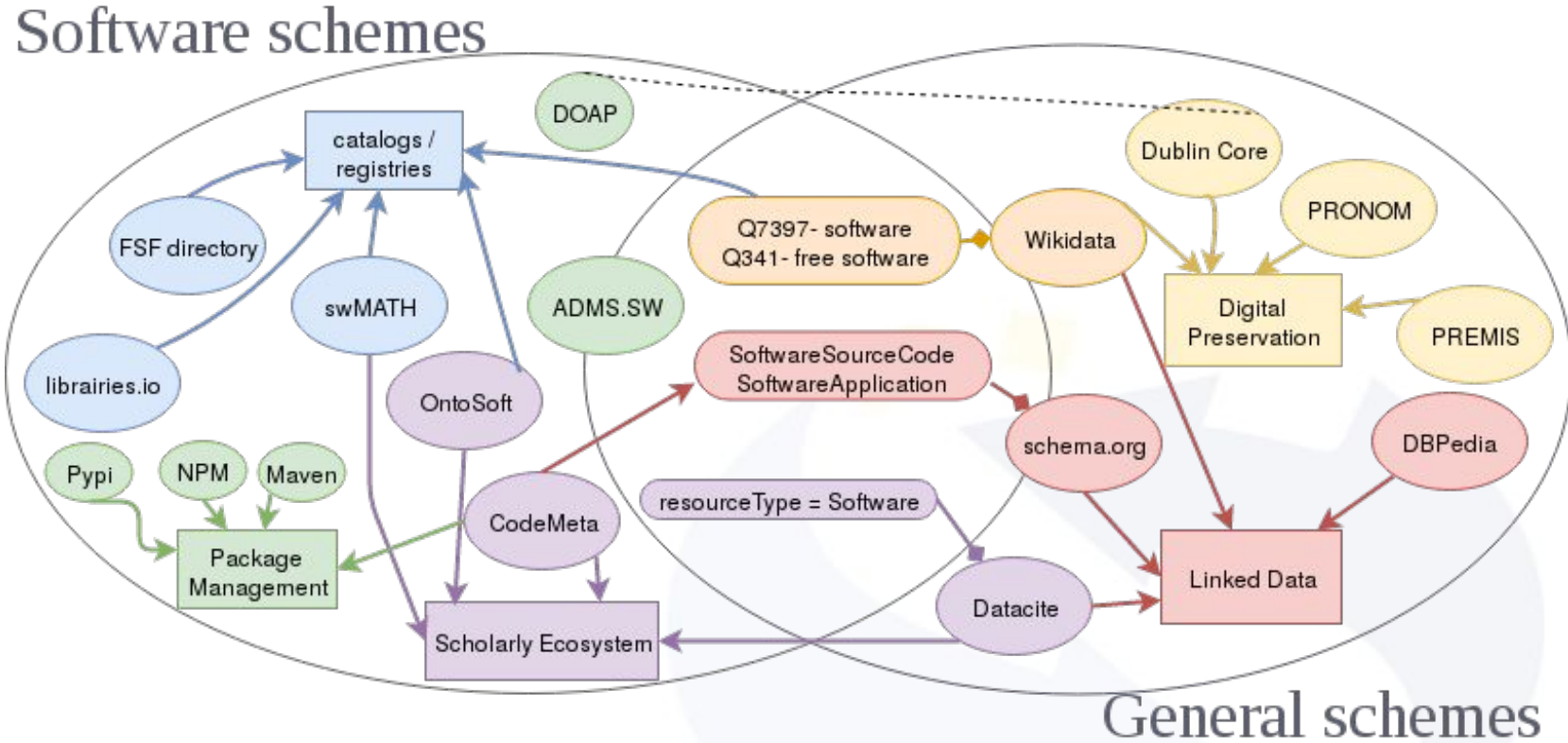
GL= Granularity Level

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

Metadata landscape

Each vocabulary is also linked to its ecosystem:

- digital preservation;
- linked data;
- catalogs / registries;
- scholarly ecosystem



Software ontologies landscape from Pathways for Discovery of Free Software (slide deck from LibrePlanet 2018). (Gruenpeter & Thornton, 2018) CC-by-4

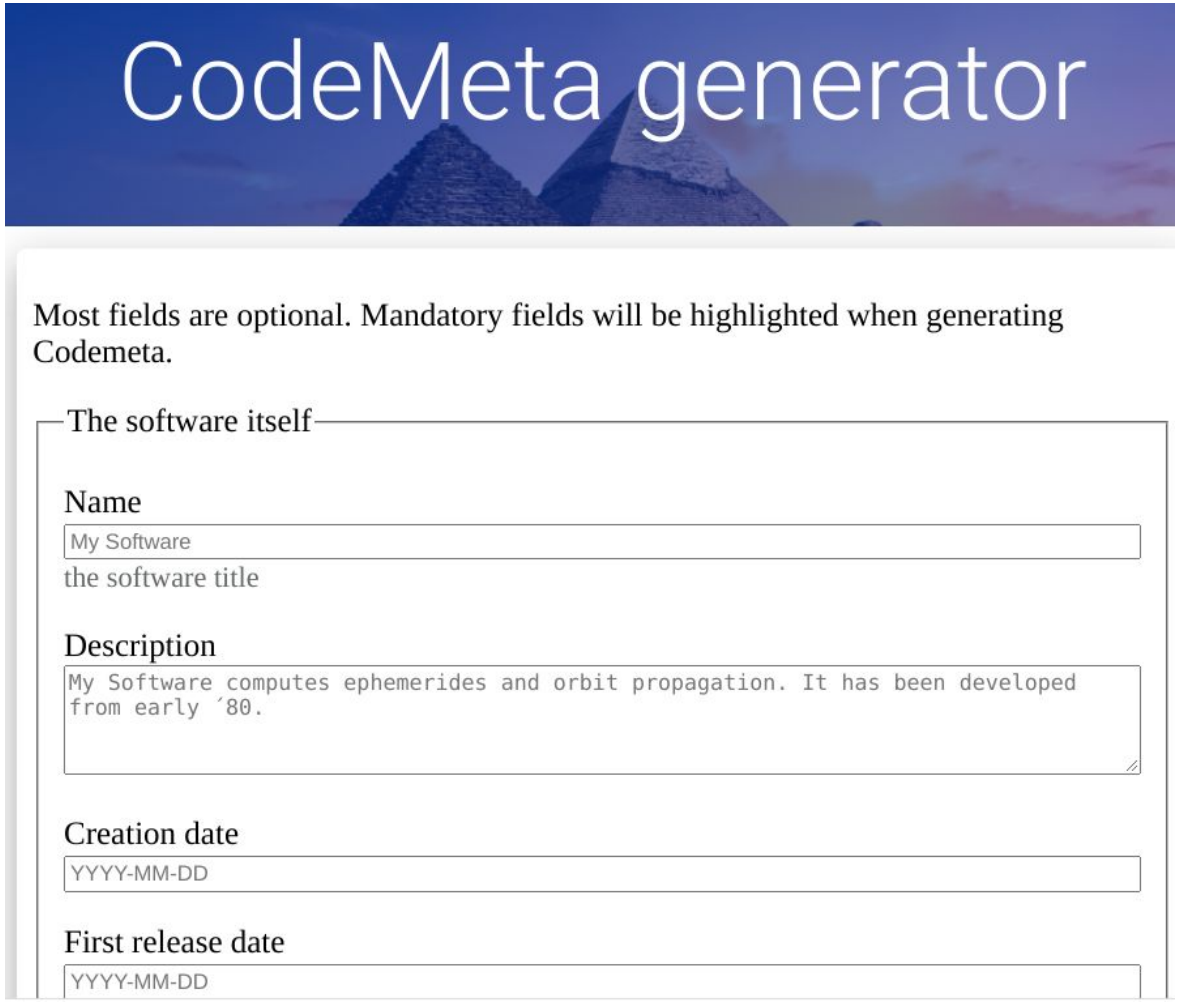
CodeMeta initiative

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

An open source tool to create codemeta.json files

Contributed to the community by

Software Heritage
THE GREAT LIBRARY OF SOURCE CODE



Use it directly on the CodeMeta hosted version

Contributions are welcome on the code repository

ACM take on Reproducibility

"Sometimes, if you don't have the software, you don't have the data"

Christine Borgman, Paris, 2018

ACM Terminology (no consensus yet!)

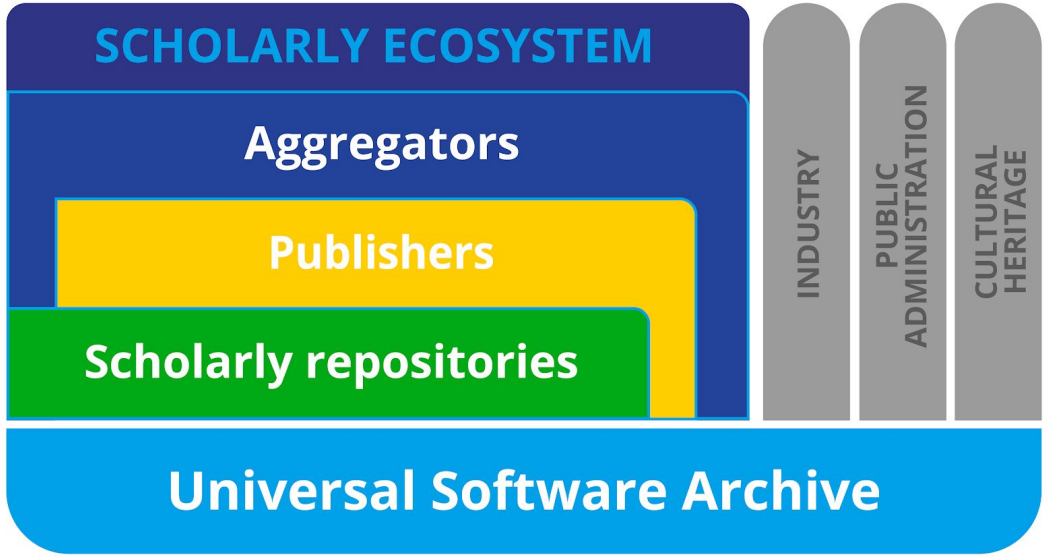
- **Repeatability** \ same team, same experimental setup
- **Reproducibility*** \ different team, same experimental setup
- **Replicability*** \ different team, diffe experimental setup

**"As a result of discussions with the National Information Standards Organization (NISO), it was recommended that ACM harmonize its terminology and definitions with those used in the broader scientific research community"*

NISO Taxonomy, Definitions, and Recognition Badging Scheme Working Group

The landscape of Existing Infrastructures

- **Archives** (HAL, Software Heritage, Zenodo)
- **Publishers** (SoftwareX, JOSS, Dagstuhl, eLife, IPOL)
- **Registries** / **Aggregators** (swMATH, scanR, OpenAIRE)
- **Research Software Training** (The carpentries)



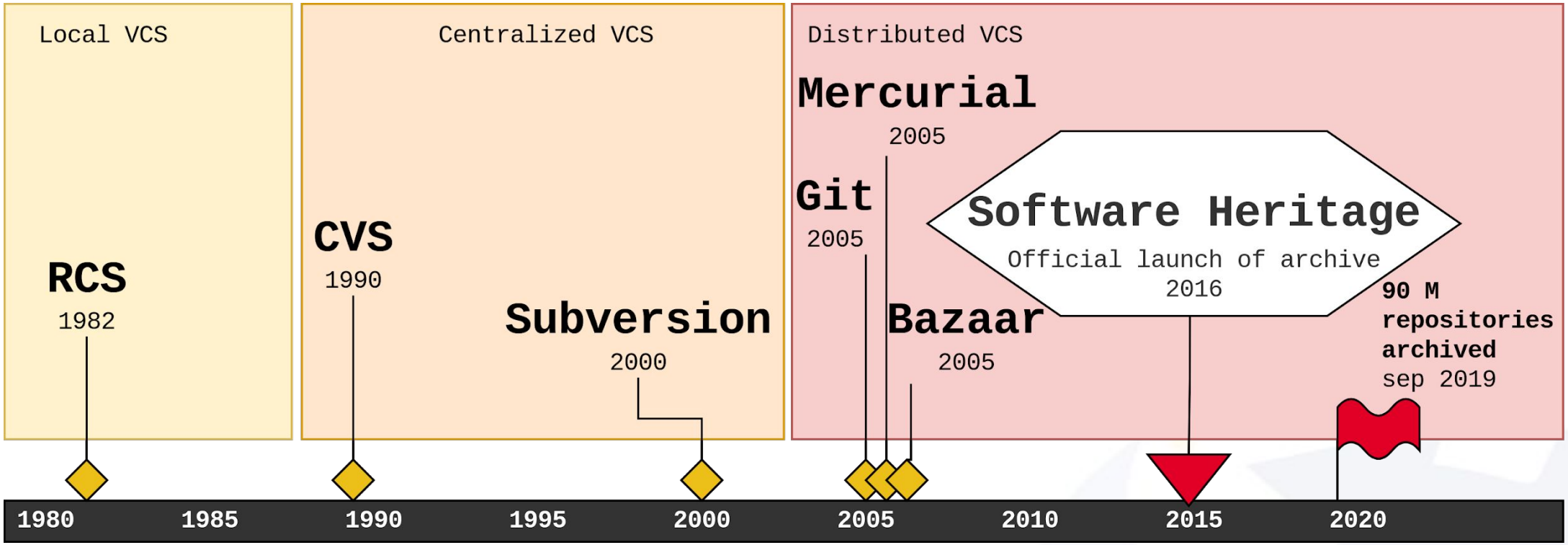
Four pillars: Archive, Reference, Describe, Credit

2020 - EOSC Scholarly Infrastructures for Research Software

Link to document waiting for the EU commission publication (community consultation ended on the 10.11.2020)

- Chairs
  - Roberto Di Cosmo, Software Heritage
  - José Benito Gonzalez Lopez, Zenodo

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

FAIRSFAIR
Fostering Fair Data Practices in Europe

Software Heritage

THE GREAT LIBRARY OF SOURCE CODE

Collect, **preserve** and **share** all software source code

Preserving our heritage, enabling better software and better research for all

How it works?

- automatic pull from different forges (GitHub, GitLab, BitBucket),
- intrinsic metadata is extracted from the content itself,
- deposited artifacts are accepted only from known sources where metadata was moderated and curated
- Save Code Now mechanism for git, svn and mercurial repositories
- SWHID persistent identifiers for all the source code artifacts

Visit the archive

Outline

- Introduction
- Software source code a (forgotten) pillar of research
- Existing mechanisms, components and infrastructures
- FAIR ecosystem
- Literature analysis
- 10 meta-recommendations for the FAIR software definition

Bridge between Software and FAIR

Four Foundational Principles

Findable

Accesible

Interoperable

Reusable

2016 - The FAIR guiding principles (Wilkinson et al. 2016)

Figure 2: Illustration of ANDS resources which reflect or crosscut the FAIR principles. Image: ANDS. CC: BY 4.0

Where we stand?

2018 - The *Turning FAIR into reality* report (European Commission, 2018)

Action 16.2: *The FAIR data principles and this Action Plan must be **tailored for specific contexts** - in particular to the relevant research field - and the precise application nuanced, while respecting the objective of maximising data accessibility and reuse. Stakeholders: Research communities; Data service providers; Policymakers*

2019 - `Six Recommendations for Implementation of FAIR Practice` (FAIR Practice TF, 2020)

***Recommendation n°5** : Recognise that FAIR guidelines will require **translation for other digital objects** and support such efforts.*

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

***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.*

FAIR Ecosystem

</> software icon

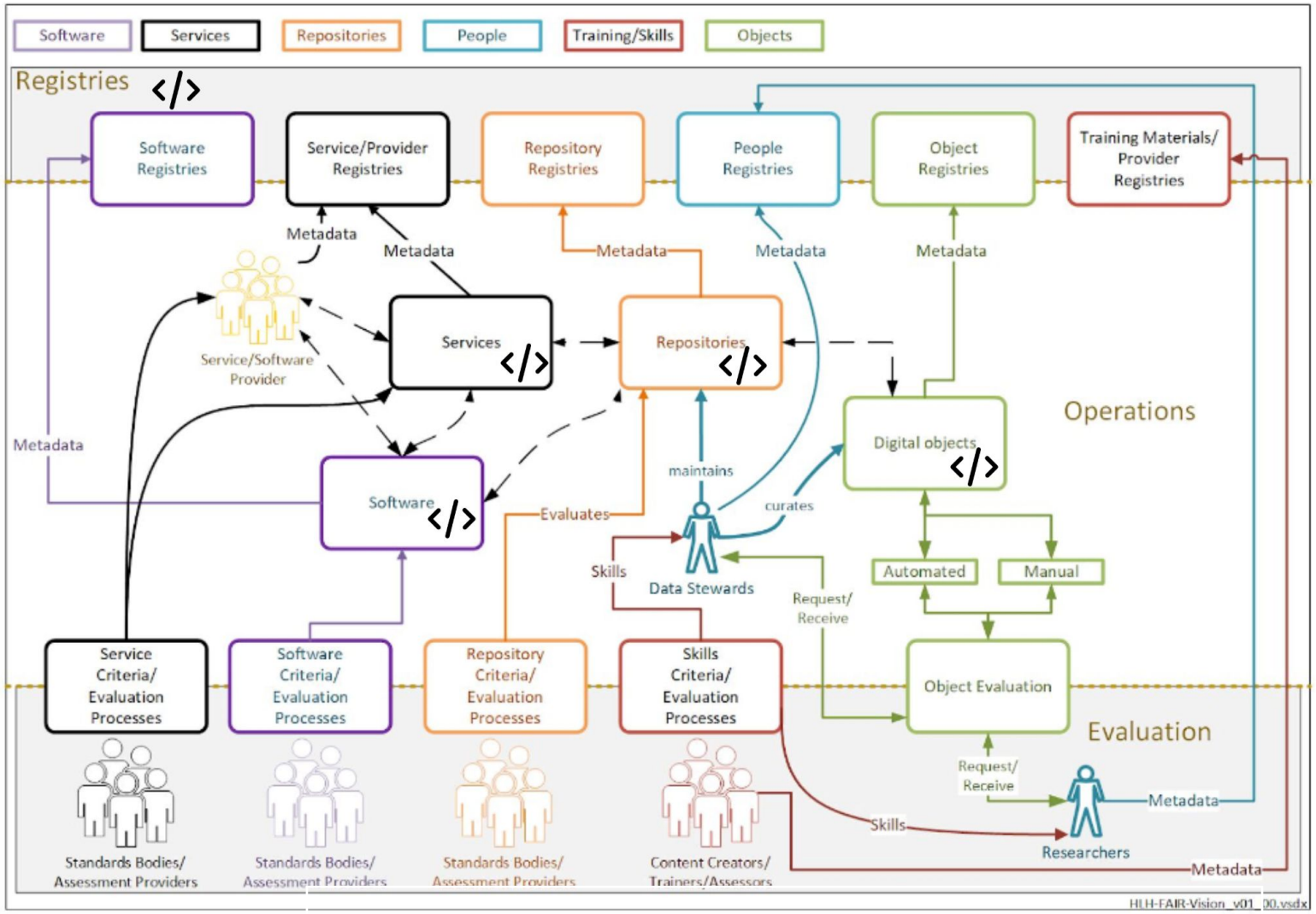


Figure 1: FAIR vision: Ecosystem components, to highlight the software roles in the Ecosystem, the symbol </> was added (Original diagram 3 from L'Hours & Von Stein, 2020)

Outline

- Introduction
- Software source code a (forgotten) pillar of research
- Existing mechanisms, components and infrastructures
- FAIR ecosystem
- Literature analysis
- 10 meta-recommendations for the FAIR software definition

Literature analysis

1. **Towards FAIR principles for research software (Lamprecht et al., 2019)** published in the Data Science journal, issue 'FAIR Data, Systems and Analysis' aiming on translating the FAIR principles to research software. Their effort is supported with two case studies, along with recommendations for rewriting the FAIR principles to make them more applicable to software.
2. **"5 recommendations for FAIR software" from the Netherlands eScience Center and DANS** straightforward guidelines for researchers on how to make software FAIR, which are available on a dedicated website to help researchers with their own software.(https://fair-software.nl/)
3. **Software citation principles (Smith et al., 2016)**
   published in PeerJ Computer Science is a result of the FORCE11 Software Citation Working Group, defining high level principles on software citation.
4. **RDA Software Source Code Interest Group (SSC IG) P13 activity translating FAIR principles to software**
   an ad-hoc activity conducted during the RDA P13 SSC IG session, where participants were asked to map the existing FAIR principles for data to possible principles for software. Participants were asked to add items that are not in the FAIR principles.
5. **From FAIR research data toward FAIR and open research software (Hasselbring, 2020)** published in the journal IT - Information Technology and aims at translating the FAIR principles to research software and producing a list of recommendations based on the FAIR principles and other resources.
6. **Attributing and Referencing (Research) Software: Best Practices and Outlook From Inria (Alliez et al., 2019)** published in IEEE Computing in Science & Engineering aiming to analyze the existing practices handling research software at the Inria research center and providing recommendations to the academic community.
7. **Software vs. data in the context of citation (Katz et al., 2016)** a PeerJ preprint, which details the differences between software and data, and providing simple recommendations for software citation.
8. **The science code manifesto (Barnes et al., 2011)** an online manifesto, published in 2011 by the Climate Code Foundation. It was endorsed by 1227 researchers and organizations. It proposes five principles to reform scientific software in institutions.
9. **CoSO Opportunity Note: Encouraging a wider usage of software derived from research by The Committee for Open Science's Free Software and Open Source Project Group (Clément-Fontaine, 2019)**
   a committee note from the French National Open Science committee declaring the importance of software in Open Science and formulating recommendations to encourage and promote better practices for handling software in institutions.

Methodology

The *exact text used in the original resource* is included, to preserve the original semantics

Key criteria

- **Relevant** - is this principle seen to be relevant to software by being frequently mentioned in the proposed resources?
- **Achievable** - seen to be achievable when it comes to software?
- **Measurable** - seen to be measurable on software artifacts?
- **Benefits** - seen to be useful and benefits the software resource?
  - **Quality curation** of the software resource
  - **Recognition** of software in scholarly communications

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

R1.1. (meta)data are released with a clear and accessible data usage license

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

*"Any creative work (including software) is automatically protected by copyright. Even when the software is available via code sharing platforms such as GitHub, no one can use it unless they are explicitly granted permission. This is done by adding a software license, which defines the set of rules and conditions for people who want to use the software." (***"5 recommendations for FAIR software")**

*"Copyright: The copyright ownership and license of any released source code must be clearly stated." (***The science code manifesto***)*

FAIR	Relevant	Achievable	Measurable	Benefits
R1.1. (meta)data are released with a clear and accessible data usage license.	***	***	***	***

*"Software is a creative work, scientific data are facts or observations In particular, software is generally subject to copyright protection as a creative work that can continue to evolve over time, while scientific data is frequently considered outside the domain of copyright as it is comprised of contextual facts about the world…"(***Software vs. data in the context of citation)**

"Software and its associated metadata have independent, clear and accessible usage licenses ***compatible with the software dependencies****. [Rephrased and extended] "(***Towards FAIR principles for research software)**

"Recommendation n° 9: Encourage and facilitate the creation of "legal toolboxes" to ensure the long-term preservation of free software resulting from research."

(CoSO opportunity note)

*"Ideally licenses should be in rights expression languages" (***SSC IG P13 activity)**

Software license is only mentioned in the use cases table and with an + sign which states: *indicate that the use case would benefit from that metadata if available.(***Software citation principles)**

I.3 (meta)data include qualified references to other (meta)data

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

(not explicitly discussed) **("5 recommendations for FAIR software")**

*"First, the frequent lack of availability of the software source code, and/or of **precise references** to the right version of it, is a major issue [7]. Solving this issue (Reproducibility) requires stable and perennial source code archives and specialized identifiers [9]."* **(Attributing and Referencing (Research) Software)**

"The software should be linked to a list of publications using the code, to other versions of the code, to relevant versions of tools and libraries used, and to derived code." **(The science code manifesto)**

FAIR	Relevant	Achievable	Measurable	Benefits
I3. (meta)data include qualified references to other (meta)data.	**	*	N/A	***

(not explicitly discussed) **(Software vs. data in the context of citation)**

"it is therefore necessary to define reference methodologies for technology transfer based on existing mechanisms (....), and to share them with the actors concerned (...)."

(CoSO opportunity note)

"Discarded"

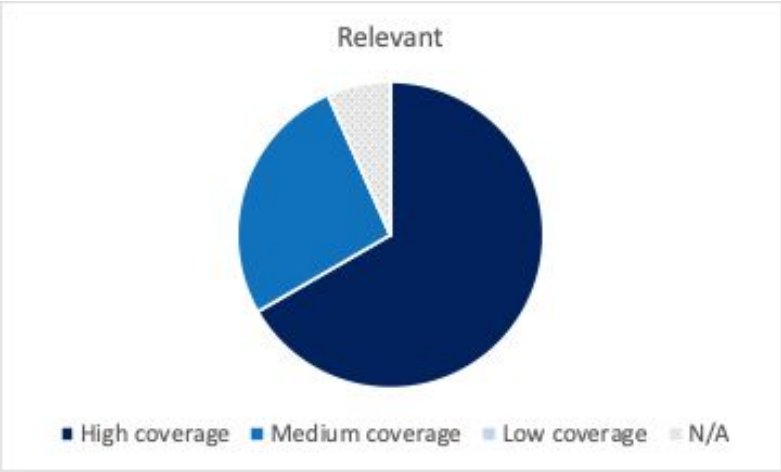
*"I3 aims to interconnect data sets by semantically meaningful relationships..... However, such relationships are difficult to translate to the case of research software. We found the closest resemblance of this principle to be in software dependencies." => I4S (***Towards FAIR principles for research software)**

(not explicitly discussed) **(SSC IG P13 activity)**

(not explicitly discussed) **(Software citation principles)**

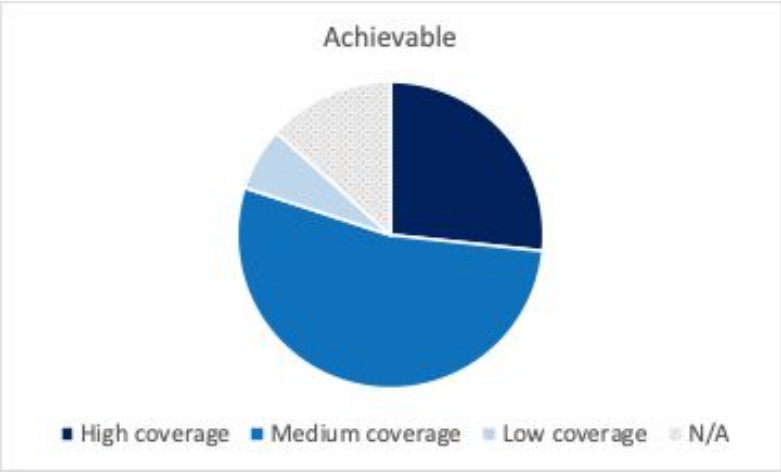
*I4S- Software **dependencies are documented** and mechanisms to access them exist. [Newly proposed]*

Compendium of FAIR software analysis



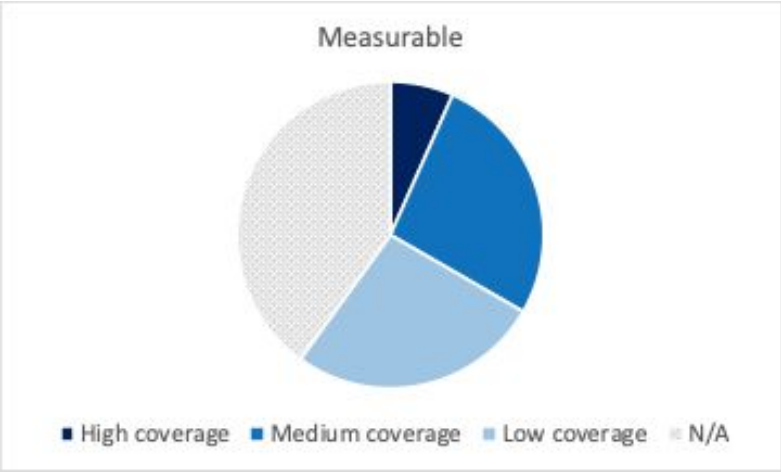
	FAIR	Relevant	Achievable	Measurable	Benefits
1	F1. (meta)data are assigned a globally unique and eternally persistent identifier.	***	***	**	***
2	F2. data are described with rich metadata.	***	**	N/A	***
3	F3. metadata specify the data identifier.	***	**	*	**
4	F4. (meta)data are registered or indexed 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	***	**	*	**

Compendium of FAIR software analysis



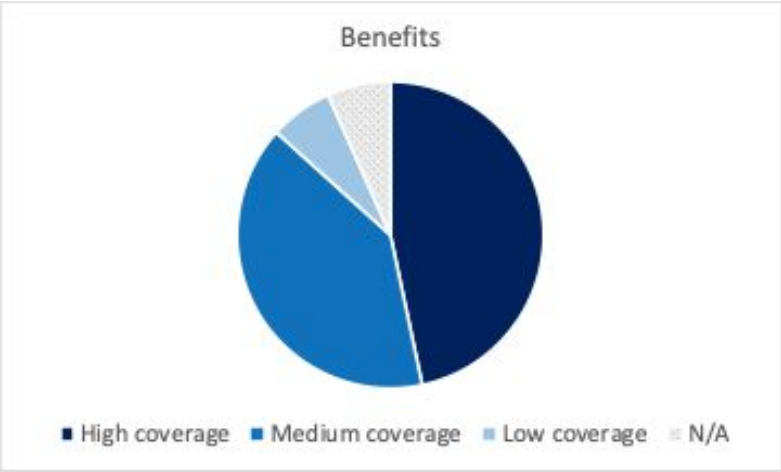
	FAIR	Relevant	Achievable	Measurable	Benefits
1	F1. (meta)data are assigned a globally unique and eternally persistent identifier.	***	***	**	***
2	F2. data are described with rich metadata.	***	**	N/A	***
3	F3. metadata specify the data identifier.	***	**	*	**
4	F4. (meta)data are registered or indexed 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	***	**	*	**

Compendium of FAIR software analysis



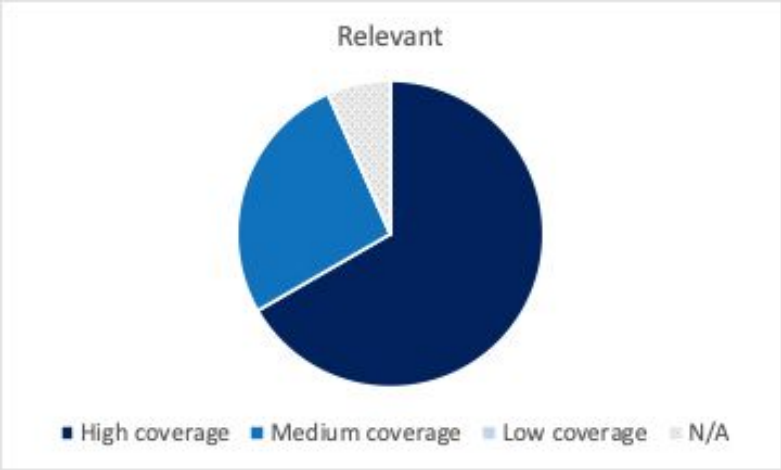
	FAIR	Relevant	Achievable	Measurable	Benefits
1	F1. (meta)data are assigned a globally unique and eternally persistent identifier.	***	***	**	***
2	F2. data are described with rich metadata.	***	**	N/A	***
3	F3. metadata specify the data identifier.	***	**	*	**
4	F4. (meta)data are registered or indexed 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	***	**	*	**

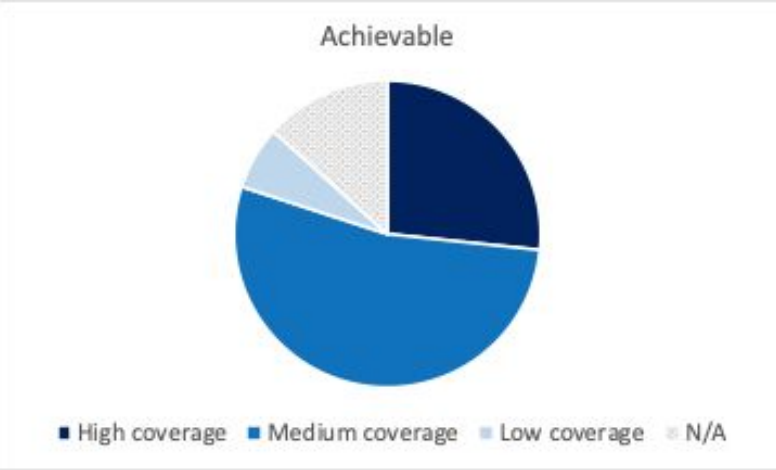
Compendium of FAIR software analysis

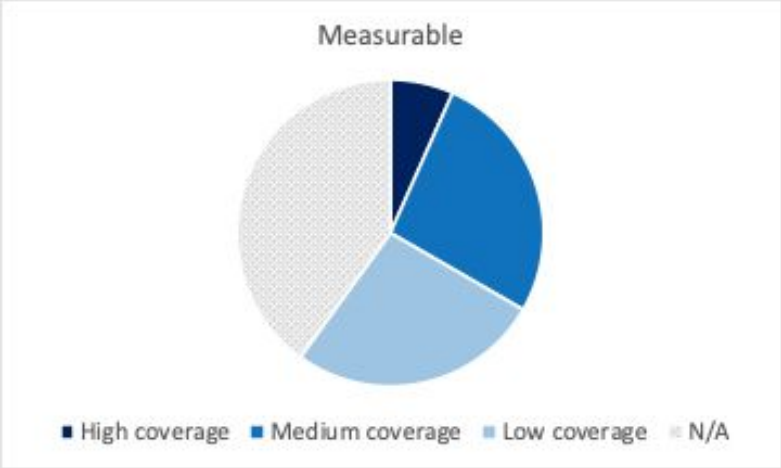


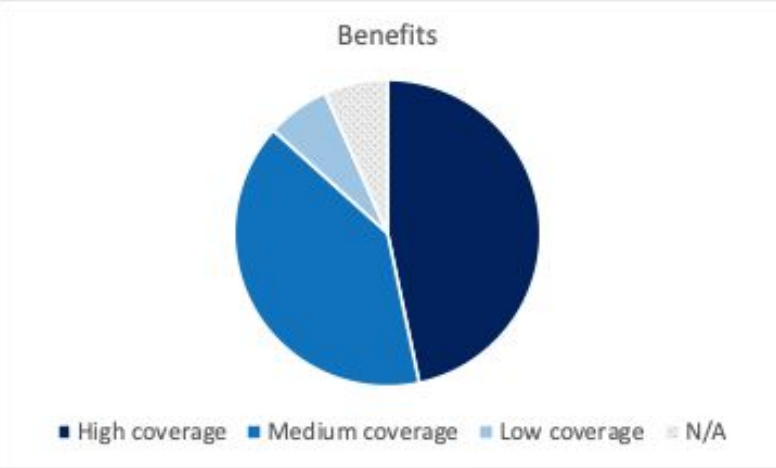
	FAIR	Relevant	Achievable	Measurable	Benefits
1	F1. (meta)data are assigned a globally unique and eternally persistent identifier.	***	***	**	***
2	F2. data are described with rich metadata.	***	**	N/A	***
3	F3. metadata specify the data identifier.	***	**	*	**
4	F4. (meta)data are registered or indexed 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	***	**	*	**

Coverage of facets within the literature corpus









Beyond the FAIR principles

- Interoperability: **dependencies and execution environment**
  - *I4S- Software **dependencies are documented** and mechanisms to access them exist. [Newly proposed]* **(Towards FAIR principles for research software)**
  - "...software dependencies need to be clearly documented in a formal, accessible, machine-readable, and shared way, and formally described following each programming language format." ***(*Towards FAIR principles for research software)**
- **Usage of version control systems** to track changes

  Using a version control system allows you to easily track changes in your software, both your own changes as well as those made by collaborators(**5 recommendations for FAIR software)**
- **Credit and attribution**
  - ***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.*
  - ***Credit**: "Software contributions must be included in systems of scientific assessment, credit, and recognition."(***The science code manifesto)**
  - Recommendation n° 4: Construct a consensual definition of a "contribution" to research software.**(CoSO)**
  - Recommendation n° 5: Build tools which integrate this notion of a contribution to be able to effectively credit authors/designers for their software contributions.**(CoSO)**
- **Testing & Software quality**
  - *"Adequate **documentation** is important, but so are **engineering practices** such as providing **testing frameworks** and **test data** for **continuous integration** to ensure that future adaptations can be tested to ensure that they work correctly."(***From FAIR research data toward FAIR and open research software)**

Challenges seen in the **FAIRsFAIR** survey*

What **challenges** do researchers in your community encounter when trying to:

A. **find** relevant research software on the web

B. **re-use** relevant research software on the web

- **Technical challenge: Software dependencies and environment**
- **Documentation**
- **Accessibility & Licensing**
- **Time & Skill**
- **Quality control**
- **Software sustainability & management plan**

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

Outline

- Introduction
- Software source code a (forgotten) pillar of research
- Existing mechanisms, components and infrastructures
- FAIR ecosystem
- Literature analysis
- 10 meta-recommendations for the FAIR software definition

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

It is to be acknowledged that any new principle may lead to **extra requirements enforced on researchers**, who are already facing significant challenges when developing or maintaining software, which is a complex and living object.

In order to maximize adoption, **clear and immediate benefits** should be offered to the researcher.

10 Recommendations

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 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.

Recommendation n°7

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

Recommendation n°8

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

10 Recommendations

Recommendation n°9

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

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).

FAIR for Research Software (FAIR4RS) Working Group

FORCE11
The Future of Research Communications and e-Scholarship

RDA
RESEARCH DATA ALLIANCE

<ReSA>
Research Software Alliance

Main objective

Defining **FAIR principles for research software**

Timeline

- **April 2020** - Formed after the RDA VP15
- **July 2020** - Launched with 4 subgroups
- **September 2020** - Endorsed by RDA

Steering committee:

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

Join the WG

How to get involved?

1. **Participate** in the community review of the FAIRsFAIR milestone

   2.15 assessment report on `FAIRness of software`

2. **Join** the FAIR4RS Working Group

   a. receive updates

   b. contribute to the subgroups work

   c. discuss the FAIR definition for research software

3. **Adopt** the existing infrastructures and mechanisms

4. **Spread the word** and let's start **recognizing software in academia**

Thank you for joining us

Keep in touch: morane@softwareheritage.org
@moraneottilia, @FAIRsFAIR_EU, @SWHeritage
https://www.fairsfair.eu/fairsfair-newletters/
https://www.softwareheritage.org/newsletter/