Scholarly infrastructures and quality requirements of software source code

Software Source Code IG

research data sharing without barriers
rd-alliance.org

10th November 2020 - RDA 16th Virtual Plenary Meeting

Housekeeping

- Collaborative notes https://tinyurl.com/6tj2tu8
- These slides https://tinyurl.com/3cmwzjhc

- Meeting etiquette
  - Add your name to the participants list
  - Add your questions in the chat
  - Raise your hand if you wish to speak
  - **Please be aware that the session is being recorded and will be made publicly available**

Agenda

time	presentations	lead
00.00	Short **introduction** to the group and its aims (5 minutes) Overview of other **ongoing working groups** related to software in RDA (5 mins)	Morane
00.10	Presentation of the EOSC SIRS report: Scholarly infrastructures for research software (25 mins)	Roberto
00.35	Update of the activity in the **FORCE11 Software Citation Implementation Working Group** including the ongoing task forces (CodeMeta, journals, repositories...)(15 mins)	Neil
00.50	A **mini-workshop** session on code **quality** assessment (30 mins) **Conclusion** of the workshop activity (5 mins)	Neil
01.25	**Next steps** (5 minutes)	Morane

The Software Source Code IG

Co-chairs:

- Neil Chue Hong
- Julia Collins
- Roberto Di Cosmo
- Stepping down: Mingfang Wu

TAB Liaison: Gretchen Greene

VP17 coordinator: Morane Gruenpeter

Objectives:

A forum for discussing **research software** inside RDA

- issues on **management**, **sharing**, **discovery**, **archival** and **provenance** of software source code.
- It will pay special attention to **source code** that generates research data and plays an **important role** in scientific publications.

https://www.rd-alliance.org/groups/software-source-code-ig

Why software source code?

Full width | Home | Development | Documentation | Donate | login

Software Heritage Archive

Features

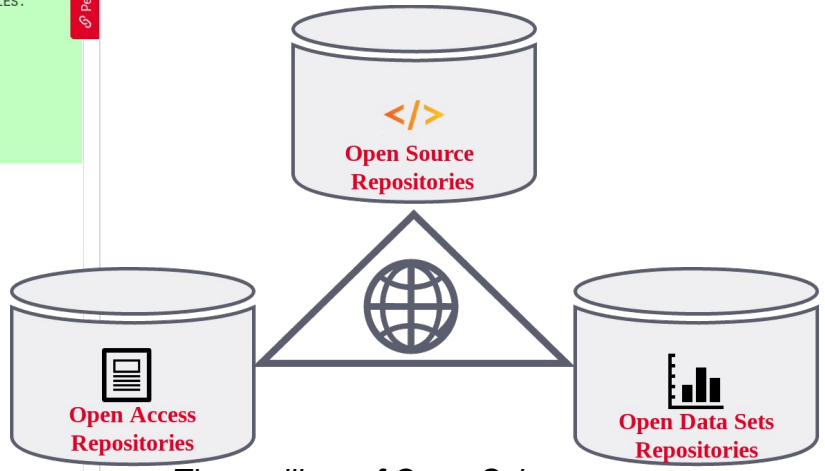
- Search
- Downloads
- Save code now
- Help

Permalinks

```
# THE MASTER IGNITION ROUTINE IS DESIGNED FOR USE BY THE FOLLOWING LEM PROGRAMS:  P12, P40, P42, P61, P63.
# IT PERFORMS ALL FUNCTIONS IMMEDIATELY ASSOCIATED WITH APS OR DPS IGNITION:  IN PARTICULAR, EVERYTHING LYING
# BETWEEN THE PRE-IGNITION TIME CHECK -- ARE WE WITHIN 45 SECONDS OF TIG? -- AND TIG + 26 SECONDS, WHEN DPS
# PROGRAMS THROTTLE UP.
#
# VARIATIONS AMONG PROGRAMS ARE ACCOMODATED BY MEANS OF TABLES CONTAINING CONSTANTS (FOR AVEGEXIT, FOR
# WAITLIST, FOR PINBALL) AND TCF INSTRUCTIONS.  USERS PLACE THE ADRES OF THE HEAD OF THE APPROPRIATE TABLE
# (OF P61TABLE FOR P61LM, FOR EXAMPLE) IN ERASABLE REGISTER `WHICH' (E4).  THE IGNITION ROUTINE THEN INDEXES BY
# WHICH TO OBTAIN OR EXECUTE THE PROPER TABLE ENTRY.  THE IGNITION ROUTINE IS INITIATED BY A TCF BURNBABY,
# THROUGH BANKJUMP IF NECESSARY.  THERE IS NO RETURN.
#
# THE MASTER IGNITION ROUTINE WAS CONCEIVED AND EXECUTED, AND (NOTA BENE) IS MAINTAINED BY ADLER AND EYLES.
#
#                 HONI SOIT QUI MAL Y PENSE
#
#       ********************************************
#             TABLES FOR THE IGNITION ROUTINE
#       ********************************************
#
#                     NOLI SE TANGERE

P12TABLE        VN      0674            # (0)
                TCF     ULLGNOT         # (1)
                TCF     COMFAIL3        # (2)
                TCF     GOCUTOFF        # (3)
                TCF     TASKOVER        # (4)
                TCF     P12SPOT         # (5)
                DEC     0               # (6)   NO ULLAGE
                EBANK=  WHICH
                2CADR   SERVEXIT        # (7)

                TCF     DISPCHNG        # (11)
                TCF     WAITABIT        # (12)
                TCF     P12IGN          # (13)

P40TABLE        VN      0640            # (0)
                TCF     ULLGNOT         # (1)
                TCF     COMFAIL4        # (2)
                TCF     GOPOST          # (3)
                TCF     TASKOVER        # (4)
                TCF     P40SPOT         # (5)
```

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

Len Shustek, Computer History Museum



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

[Go to the code!]

Chronology

BOF RDA P9, Barcelona April 2017
motivations => 60 participants

RDA P10, Montreal September 2017
motivations, survey of ontologies, metadata use cases

RDA P11, Berlin March 2018 started the idea for a dedicated identification WG

RDA P13, Philadelphia April 2019
FAIR for Software Source Code and launch of the **SCID WG**

FORCE2019, Edinburgh October 2019
full day hackathon on **Research Software** (motivated the CodeMeta generator)

RDA VP15, Australia March 2020
Open discussion about the creation of a new group, the **FAIR4RS WG** (which was launched in June 2020)

RDA VP16, Costa Rica November 2020
Existing efforts and practices in Academia

RDA VP17!!!

Software at RDA and in academia

- Related groups:
  - RDA, ReSA and FORCE11 FAIR for Research Software Working Group (FAIR4RS WG)
    - Welcome to join the work defining FAIR principles for research software
  - RDA & FORCE11 Software Source Code Identification WG(SCID IG)
    - Output published in September 2020
  - FORCE11 Software Citation Implementation Working Group (SCIWG)
    - Ongoing WG about software citation
- Related software sessions during VP17:
  - IG - Software Source Code: Scholarly Infrastructures and Quality Requirements of Software Source Code
  - WG - FAIR for Research Software (FAIR4RS): Early Outcomes of the FAIR 4 Research Software: Definitions, Principles and Road Ahead
  - WG - CURE-FAIR: Progress Update
  - IG - Research Data Management in Engineering - Engineering RDM Initiatives
  - BoF - Defining FAIR for Machine Learning (ML)

FAIR4RS WG

Endorsed by RDA Sep 2020

- A joint **RDA** Working Group, **FORCE11** Working Group, and Research Software Alliance (**ReSA**) Taskforce.
- Coordinating of a range of existing community-led discussions on:
  - How to define and effectively apply FAIR principles to research software,
  - How to achieve adoption of these principles.

https://www.rd-alliance.org/group/fair-4-research-software-fair4rs-wg/

Ice-breaker question 1

Why are you interested in Software Source Code?

https://tinyurl.com/6tj2tu8

SIRS report presentation
By Roberto Di Cosmo

https://annex.softwareheritage.org/public/talks/2021/2021-04-22-SIRS4RDA.pdf

Ice-breaker question 2

Are you using a scholarly infrastructure for Source Code?

https://tinyurl.com/6tj2tu8

FORCE11
Software Citation Implementation WG update
By Neil Chue Hong

FORCE11 SCI WG

FORCE11 Software Citation Implementation Working Group
(co-chairs: N. Chue Hong, M. Fenner, D. S. Katz)

Following-on from FORCE11 Software Citation Working Group and the Software Citation Principles it developed

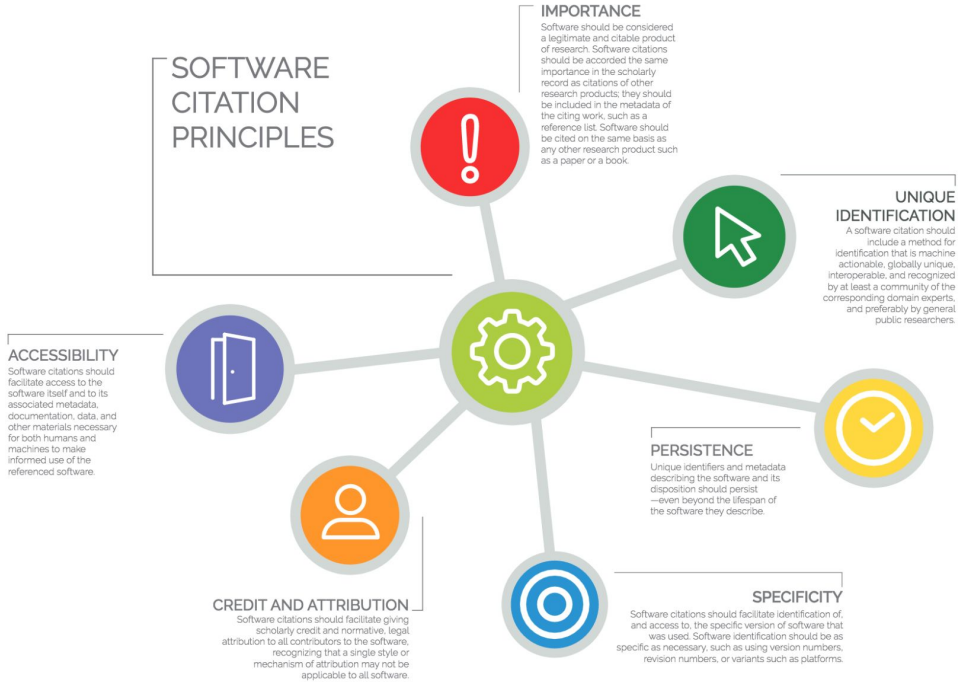
Objective: Produce concrete guidelines for software citation, and implement them within the scholarly research community (software developers, repositories and registries, journals and conference and publishers, indexers, institutions)

A community with monthly calls to discuss challenges and progress in implementing software citation, with task forces for

- **Guidance** - developing documents for developers, authors, and reviewer
- **Journals** - coordinating editors and publishers to simplify and implement guidance
- **Repositories** - developing best practices document for handling software
- **CodeMeta** - standardizing metadata for software, moving towards merging into schema.org

Software Citation Principles

Software is a critical part of modern research...



1. Importance
2. Credit and Attribution
3. Unique Identification
4. Persistence
5. Accessibility
6. Specificity

Smith AM, Katz DS, Niemeyer KE, FORCE11 Software Citation Working Group.(2016) Software Citation Principles. PeerJ Computer Science 2:e86. DOI: 10.7717/peerj-cs.86 and https://www.force11.org/software-citation-principles

Image courtesy of DataCite

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

The journey so far...

2015 — Software Citation WG started

~55 members (researchers, developers, publishers, repositories, librarians)

Reviewed existing community practices & developed use cases

2016 — Software Citation Principles published

Started with data citation principles, updated based on software use cases and related work, working group discussions, community feedback

Software citation principles published after community review: 10.7717/peerj-cs.86

2017/18 — Software Citation Implementation WG started

Group set up to:
1. endorse the principles
2. develop sets of guidelines for implementing the principles
3. help implement the principles
4. test specific implementations of the principles.

Wrote document to identify software citation implementation challenges: arXiv 1905.08674

2019/20 — Task forces publish resources

Guidance Task Force develops checklists for authors: 10.5281/zenodo.3479198 and developers: 10.5281/zenodo.3482768

Codemeta Task Force provides recommendations for schema changes

Repositories Task Force runs workshop to identify best practices

Journals Task Force starts adoption process with journals and publishers to promote The Importance of Software Citation: 10.12688/f1000research.26932.1

2021 — Adoption of software citation increases?

How does software citation fit with other related work:

FAIR for Research Software
Open Research / Open Science
Reproducibility
Software catalogs

Community review of "Best Practices for Research Software Registries and Repositories: A Concise Guide" from Repositories Task Force

Guidance Task Force

(Lead: Neil P. Chue Hong)

Objective:
Develop guidance for different stakeholders to help implement software citation, principally authors of research articles seeking to cite software correctly and developers of software looking to make their software easier to cite

Using checklists as an effective way of ensuring consistency and completeness

- See The Checklist Manifesto: How to Get Things Right by Atul Gawande for examples

Milestones and activity:

- Developed and published guidance for paper authors and software developers
  - Software Citation Checklist for Authors
  - Software Citation Checklist for Developers
- Developed Software Citation Primer used as basis for paper by Journals Task Force
- Started Software Citation Checklist for paper reviewers, on hold pending work from Journals Task Force and Code Review WG

Guidance Task Force on hiatus while Journals Task Force provides feedback on guidance

GitHub repository

Repositories ~~Task Force~~ Consortium

(Lead: Alice Allen)

Objective:
Bring together representatives of Research Software Registries and Repositories to discuss and improve practices

Milestones and activity:

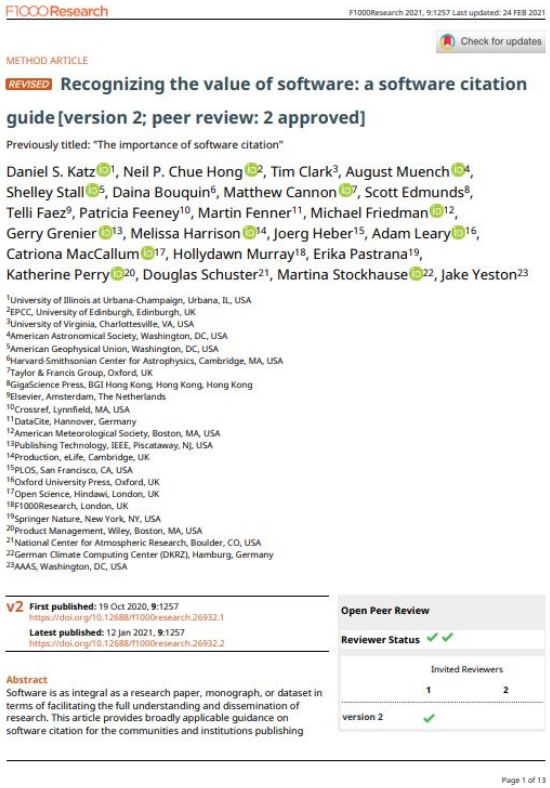
- Held two-day workshop (funded by the Sloan Foundation) bringing together managers and editors of scientific software registries and repositories
- Developed and published “Nine Best Practices for Research Software Registries and Repositories: A Concise Guide” https://arxiv.org/abs/2012.13117
- Task force completed work and spun out new coalition of registries and repositories to build a community of practice (consortium meets monthly): https://github.com/scicodes

Journals Task Force

(Lead: Daniel S. Katz / Shelley Stall)

Objective:
Work with organizations that publish journals, proceedings, monographs to improve how software is cited in their works and the scholarly processing ecosystem

Journals (and conferences) need guidance on what authors should submit regarding the software they use



F1000Research F1000Research 2021, 9:1257 Last updated: 24 FEB 2021

Check for updates

METHOD ARTICLE

REVISED **Recognizing the value of software: a software citation guide [version 2; peer review: 2 approved]**

Previously titled: "The importance of software citation"

Daniel S. Katz[1], Neil P. Chue Hong[2], Tim Clark[3], August Muench[4], Shelley Stall[5], Daina Bouquin[6], Matthew Cannon[7], Scott Edmunds[8], Telli Faez[9], Patricia Feeney[10], Martin Fenner[11], Michael Friedman[12], Gerry Grenier[13], Melissa Harrison[14], Joerg Heber[15], Adam Leary[16], Catriona MacCallum[17], Hollydawn Murray[18], Erika Pastrana[19], Katherine Perry[20], Douglas Schuster[21], Martina Stockhause[22], Jake Yeston[23]

[1]University of Illinois at Urbana-Champaign, Urbana, IL, USA
[2]EPCC, University of Edinburgh, Edinburgh, UK
[3]University of Virginia, Charlottesville, VA, USA
[4]American Astronomical Society, Washington, DC, USA
[5]American Geophysical Union, Washington, DC, USA
[6]Harvard-Smithsonian Center for Astrophysics, Cambridge, MA, USA
[7]Taylor & Francis Group, Oxford, UK
[8]GigaScience Press, BGI Hong Kong, Hong Kong, Hong Kong
[9]Elsevier, Amsterdam, The Netherlands
[10]Crossref, Lynnfield, MA, USA
[11]DataCite, Hannover, Germany
[12]American Meteorological Society, Boston, MA, USA
[13]Publishing Technology, IEEE, Piscataway, NJ, USA
[14]Production, eLife, Cambridge, UK
[15]PLOS, San Francisco, CA, USA
[16]Oxford University Press, Oxford, UK
[17]Open Science, Hindawi, London, UK
[18]F1000Research, London, UK
[19]Springer Nature, New York, NY, USA
[20]Product Management, Wiley, Boston, MA, USA
[21]National Center for Atmospheric Research, Boulder, CO, USA
[22]German Climate Computing Center (DKRZ), Hamburg, Germany
[23]AAAS, Washington, DC, USA

v2 **First published:** 19 Oct 2020, **9**:1257
https://doi.org/10.12688/f1000research.26932.1
Latest published: 12 Jan 2021, **9**:1257
https://doi.org/10.12688/f1000research.26932.2

Abstract
Software is as integral as a research paper, monograph, or dataset in terms of facilitating the full understanding and dissemination of research. This article provides broadly applicable guidance on software citation for the communities and institutions publishing

Open Peer Review
Reviewer Status ✓ ✓
Invited Reviewers: 1, 2
version 2 ✓

Page 1 of 13

Milestones and activity:

- Paper (co-written with ~20 publishers) providing guidance to journals & conferences published
- Publicized in Scholarly Kitchen and many talks

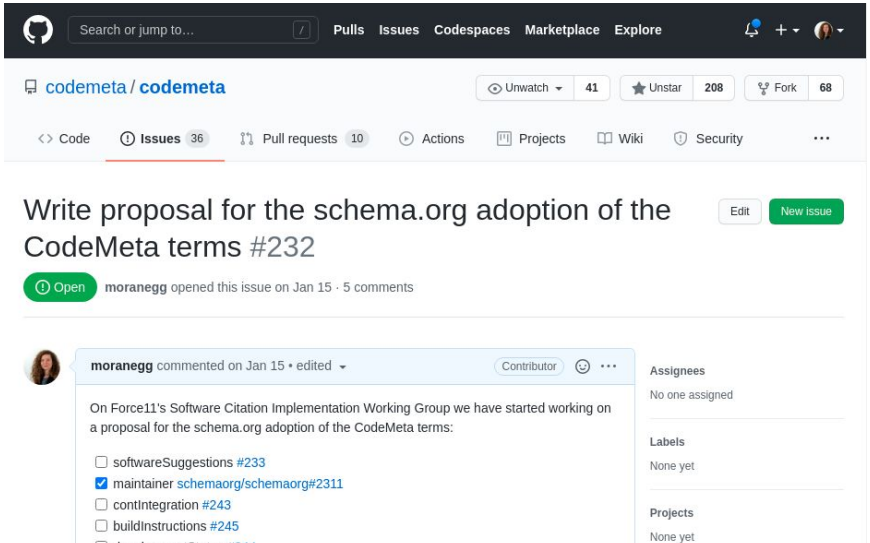
Next steps:

- Help produce versions with examples and citation styles appropriate for intended audiences
- Work on what happens after article is submitted – how citations are processed and indexed, working with JATS4R WG

CodeMeta Task Force

(Leads: Martin Fenner / Morane Gruenpeter)

Objective:
Understand metadata needed to describe software. Update CodeMeta and express all codemeta properties using schema.org



Milestones and activity:

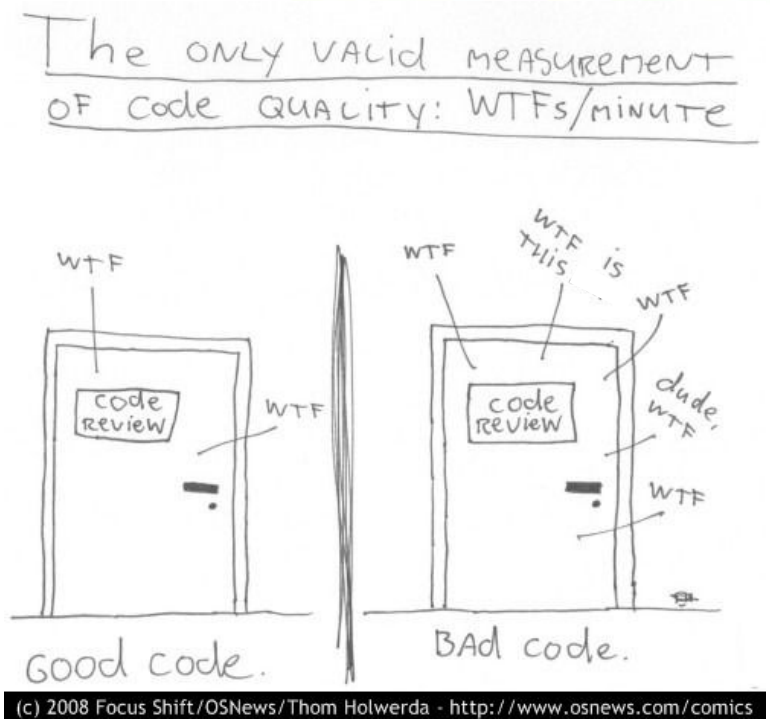
- Reviewed CodeMeta schema and opened issues in the CodeMeta repository with proposals for updating schema
- Validate & integrate proposals in the next CodeMeta release (v3)
- Preparing formal proposal to schema.org
- Planning a webinar in September to showcase existing tooling for generating and using codemeta files.
  - E.g. CodeMeta generator from Software Heritage
    - Hosted version
    - Github repository

Next steps for Software Citation Implementation

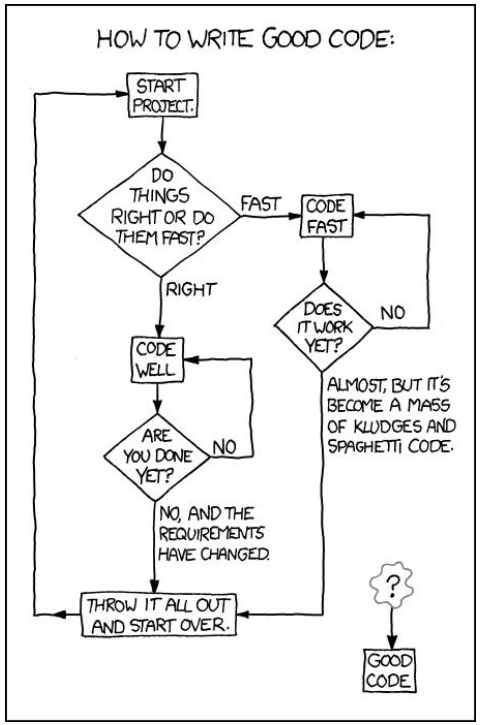
- Raised the profile of software citation with many stakeholder groups
  - Lots of good work done, starting to have effects
- Communities need to produce their own versions of the generic guidance and promote and incentivize the use of software citation
- Tracking effects is a research challenge
  - Can we determine if software citation increases over time?
- Other technical and social challenges remain
  - Swimming in a sea of identifier types
  - Where and how metadata should be stored
- And citation is just a part of a larger FAIR, open, and reproducible goal

Mini-workshop: code quality assessment

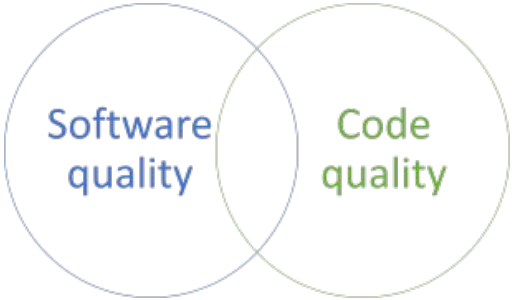
What is code quality?



Source: https://www.osnews.com/story/19266/wtfsm/



Source: https://xkcd.com/844/



Software quality: does the software serve its purpose well?
Code quality: does the code serve its purpose well?

- Readable
- Modifiable

Based on: https://blog.ploeh.dk/2019/03/04/code-quality-is-not-software-quality/

How do we assess code quality?

https://tinyurl.com/6tj2tu8

We'll answer a series of questions in parallel, via the collaborative notes document, and reflect on our answers.

1. What is your **definition** of code quality?
2. What do you look for when assessing the **readability** of a piece of software source code?
3. What do you look for when assessing whether software source code is **modifiable** / **reusable**?

Collaborative notes:
https://tinyurl.com/6tj2tu8

This will help the Code Review Community WG develop guidance for code review during development, and at the time of publication - contact Holly Meunier to get involved in the group

Software / Code / Curation quality: who can assess?

4. Who should/can ensure the **software quality (functionality)**, **code quality (readability)** or the **curation quality (FAIRness?)** of the software? Which roles are responsible?

5. Which type of quality can be ensured by **repository managers**? What should repository managers look for when source code is **deposited** to ensure it is well **curated**?

6. Do we need to **keep humans in the loop**? Can quality be assessed automatically? If so, which type of quality and why?

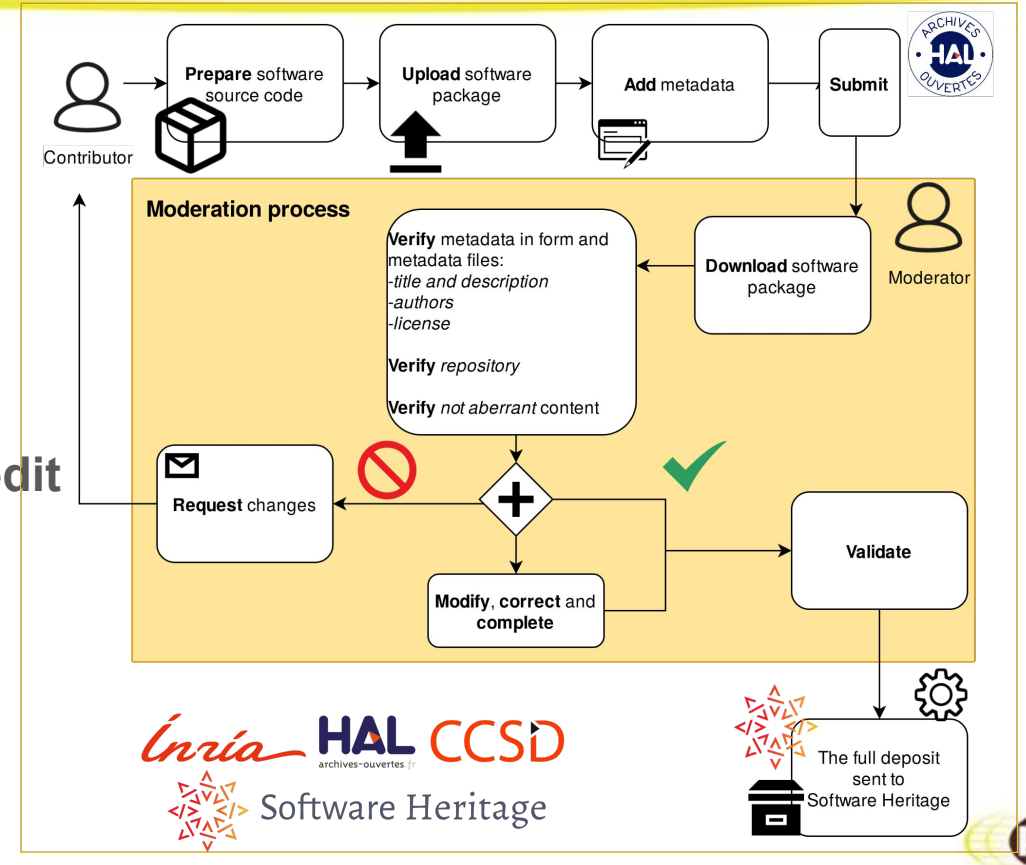
Collaborative notes:
https://tinyurl.com/6tj2tu8

Software curation on hal.archives-ouvertes.fr

Detecting extraneous or **abusive content** (illegal or harassing),

Verifying **consistency** between the **metadata** and the software source code itself, and verifying **due credit** to authors

Completing or **correcting** the deposit metadata if needed.



Curated Archiving of Research Software Artifacts: Lessons Learned from the French Open Archive (HAL) on IJDC https://doi.org/10.2218/ijdc.v15i1.698

Next steps for the SSC IG

https://tinyurl.com/6tj2tu8

Open questions (to answer on the notes):

- What **themes** would you like to discuss during the next plenaries?
- We are looking for **contributions** to the next session? (call for short presentations)

Join us on the mailing list:
https://www.rd-alliance.org/groups/software-source-code-ig

Thanks for joining