Theme: Research Software Citation

Simon Worthington, editor, Generation R, April 2018, TIB. Theme to initially run over May 2018 and then periodically updated.

The submission of software as a research output is becoming more common. As a result a number of area need addressing and improving in research workflows, and in the research life cycle of a software projects.

Two areas are important for this theme of 'software citation' in terms of our editorial remit of taking a 'needs based approach to researchers', these are *the use of software* and *the development of software*. As examples:

- the benefits to the scientific systems can be that experiments *using software* can be replicated and built upon more easily, and
- in the area of *software development* can be helped by increased discovery and reuse.

The relationship of researchers to research software can quite often be characterized as transitory or 'on/off', and is compounded by accelerated development life cycles: a researcher might only briefly use a tool, research software R&D funding is quite often time-limited.

In this editorial theme of 'software citation' we want to look at what initiatives and project are working on these issues of software citation. We also want to look practical steps that a variety of users and public institutions can take to improve their systems for software citation. Our initial use cases are:

- a. the software maker or contributor,
- b. the researcher citing the software, and
- c. the researcher reading/using research literature/output and wanting to reuse the software.

Our themes are run over a flexible time period, but as default for four weeks, and then periodically we will revisit a theme. For this initial theme we will start early June and carry on over June and July. Our editorial approach is to support the Open Science community in its ongoing work in a given area. We do this by carrying short blog posts, and by maintaining a 'notebook' to engaging in discourse and carry resource documentation.

The future

Advanced software maintenance systems make use of systems for **version control** and **dependency management**. The result of these two types of technologies results in the ability to have any version in a software's release history being automatically available, and be able to run the software where applicable. With the recent addition of technologies such as **continuous integration** software can be validated to be in good working order and so ensure it is fit to be used.

Building on these three technologies, in the not too distant future, the working environment for software R&D will be able to being about a much greater sustainability. Example are a project like Binder for republish Jupyter notebooks https://mybinder.org

See the brief sketch on such future casting #Open SciFi story - what if?

Breaking down the topic

- A key issue is discovery and evaluation. Researchers need more information about software used in experiments, and to have access to the source code to be able to access and run the software cited.
- For software maintainers guidance is needed about what core metadata needs to be stored with the software in a similar way to how open licences and copyright notices are stored.
- Currently how to cite software is not clear or technically resolved.
- Look at projects going on to create systems for software citation: to record, to read, to collect, etc. Get input from different projects. What are their research questions/interim findings. So far these examples have been found: CodeMeta, CFF, and CiteAs.
- What information do journals and repositories want when submitting software
- How can citing be more useful for researchers. Is there enough information to be useful for research publication/software readers/users
- The Future: Can software cited be fully available in a – validated, CI, packet managed, dependency managed, and virtualised way – so that it can be retrieved or run live. e.g, Jupyter https://jupyter.org/ and Binder to republish https://mybinder.org
- Note: we need clear guidelines or pointers to this info, for different users: software makers, journal submissions of software, and academic writers of scholarly literature editorial groups wanting to cite software.
- Area survey. Top three info sources in each area: journals, papers, software citation software project, working groups and organisations
- Simple guides
- Look for working groups like Force11 working groups

Community

- #oscibar https://etherpad.wikimedia.org/p/oscibar2018_session10
- Zenodo
- CodeMeta https://github.com/codemeta/codemeta and Crosswalk
- CFF Citation File Format https://citation-file-format.github.io/
- CiteAs http://citeas.org/about by http://impactstory.org/
- Force11 working group running through https://github.com/force11/force11-sciwg/
    - FORCE11 Scholarly Communication Institute (FSCI) San Diego teaching july-aug 18 https://www.force11.org/fsci
- Wikidata
- Software Heritage
- Katrin Leinweber TIB - Software/Library Carpentry
- de-RSE (Research Software Engineers) Group (German group): http://www.de-rse.org/en/
- Sustainable Software for Science: Practice and Experiences (WSSSPE) working groups https://twitter.com/WSSSPE
- Jupyter https://jupyter.org Binder
- DOAP

- The Software Sustainability Institute https://www.software.ac.uk/
- CRediT (Contributor Roles Taxonomy) - casrai http://docs.casrai.org/CRediT

Community engagement

□ Feed out to an Open Science MOOC section in part 5. Module 5 of the Open Science MOOC, Open Research Software and Open Source https://github.com/OpenScienceMOOC/Module-5-Open-Research-Software-and-Open-Source/issues/1

□ Force11 https://www.force11.org/group/software-citation-working-group

□ Library / Software Carpentry (TIB)

□ Others (please make suggestions)

Literature and projects (Zotero)

A collaborative bibliography on software citation https://www.zotero.org/groups/1838445/o-s/items/tag/software-cite

Related journals

- PeerJ Computer Science

Key resources (provisional)

2016 Force11 working group running through Dec' 18. **Group** https://www.force11.org/software-citation-principles **Article** https://peerj.com/articles/cs-86/

2018 Nature Software Submission Guidelines’, 2018. https://doi.org/10.1038/d41586-018-02741-4

Content packages - wish list

Blog posts

☑ Theme announce: similar to this document here.

☑ Theme: editorial introduction

☑ Blog: Open Science Barcamp – report on software citation by moderator

☑ Blog: Software Citation 'How-to' article related to FAIR and follow on 'notebook' – Katrin Leinweber TIB

□ Forum/how-tos: The Software Citation community

□ Blog/ resource: top 3 lit, journals, projects, citation managers to use: **Literature and projects (provisional)** https://www.zotero.org/groups/1838445/o-s/items/tag/software-cite

☑ Force11 group report

☑ Project: CFF Citation File Format

□ Project: CodeMeta and crosswalk

□ Project: CiteAs

□ Software Herritage

□ Re-use and journals. Blog. How software is described and documented so that its citation can be of use to future practitioners and readers of research publications. e.g. 2018 Nature Software Submission Guidelines’, 2018. https://doi.org/10.1038/d41586-018-02741-4

□ Basics guidance for key users: a. software contributor, b. citation in research outputs, c. research literature/outputs reader/user.

□ Project: #openscifi Jupyter https://jupyter.org Binder. Note: You can get DOIs for repositories via tools like zenodo. Since a repo is the input format for binder this fits well so you can do doi -> repo -> binder (Binder Gitter)

□ Replication crisis and relationship to software citation

□ Datacite and updated schema

□ CRediT - casrai

Activity of theme time period (potential)

□ Provide concrete guidance for our three research users: a. software contributor, b. citation in research outputs, c. research literature/outputs reader/user. This will involve internal and external sources.

□ Expand list and map of users, use cases, and stakeholders involved and apply T-PINC analysis (SCP, Katz, 2016)

□ Who is involved in the SC area, list, categorize: working groups, journals, research groups, projects, funders, repositories, etc.

□ Literature and sources. list, categories and enrich metadata.

□ Learning resources. Contribute to Open Science MOOC - have community feed into this. Carry material along to FORCE11, Library /Software Carpentry and others.

□

- □ Open Science SKOS, taxonomy. Update and find related taxonomies like CRediT (Contributor Roles Taxonomy) http://docs.casrai.org/CRediT
- □ Convene an online-video workshop - topic?
- □ Core questions, issues. comments. Raise these issues in social media with partners.
- □ Social Media diffusion of blog post issues and related discourse with community. A plan needs to be developed for each article.
- □ Pre-announce the theme and make available for comment over period.
- □ Hold a software makers and contributors online citation sprint.
- □ Make an infographic
- □ Collaborative bibliography work and improve the sources for the topic.

Questions, issues and comments

- Make sure users a, b, and c have concrete and high quality instructions for making and using citations. This should mainly involve curating links and commenting.
- What are the benefits of software citation
  - Aid research replication
  - Aid the building of software: speed, cost, dissemination
  - Improve the reliability of research that uses software
  - Building better software and contemporary research infrastructure systems (CRIS)
- Who benefits from software citations
- What are the barriers to software citation creation and use
- What has to be considered when making software citation systems. SCP paper (Katz et al., 2016) credit and attribution, UID, persistence, accessibility (all parts available to run), specificity (which version).
- What are the milestones that have got us where we are now.
- Recent collaborative efforts are improving the situation: what and how to cite, improving metadata standards
- Credit and attribution. supporting researchers through recognition. Examples such as Transitive Credit (Katz, Smith)
- OpenAIRE to include software. Currently OpenAIRE policy is not to include it – forum issue
- Future casting - what should or could come into existence as an optimal software citation system
- The Software Paper
- Resolving how to cite software
- How to submit software citation to a journal
- Curation and indexing of software

The Future?

- Discover and run scientific code, Code Ocean https://codeocean.com/
- Simulations in the browser, Jupyter https://jupyter.org/ and Binder to republish https://mybinder.org
- Continuous Integration and validation, with dependencies and full contributor audit

Learning resources

FORCE11 Scholarly Communication Institute (FSCI) San Diego teaching july-aug 18 https://www.force11.org/fsci

Digital Preservation Coalition – webinar and resources, EPISODE 3: Software (Re)Use Cases, https://dpconline.org/events/past-events/webinars/spw-software-reuse-cases

See also in series: **“Episode 4: Software in Scholarly Communications”** featuring Special Guests: Veronica Ikeshoji-Orlati (Vanderbilt University), Neil Chue Hong (Software Sustainability Institute) and James Howison (University of Texas at Austin) with Research & Facilitation Lead Elizabeth Parke (McGill University). Organisers:

Working Group, Software Heritage https://wiki.softwareheritage.org/index.php?title=Working_groups

CalTech CodeMeta implementation https://www.library.caltech.edu/news/enhanced-software-preservation-now-available-caltechdata

Create a grid for the theme

To include: timeline, issues, keywords, actors, stakeholders, users and questions to ask.