Open Science Days - Berlin 2019

The Experience of the Software Sustainability Institute

and an overview of

Mechanisms and Practices for the Publication of Research Software

Giacomo Peru (g.peru@epcc.ed.ac.uk) @SoftwareSaved

University of Edinburgh | Software Sustainability Institute
software & research
UK software survey 2014

public investment software use

Average 35%
The Software Sustainability Institute

cultivating better, more sustainable, research software
to enable world-class research

“better software, better research”
International collaboration on citation

How to cite and describe software

By Mike Jackson.

Researchers face significant challenges when trying to understand, reproduce or reuse research in which software has played an integral part. In this green paper, I give examples of the problems that can arise when reproducing someone else’s research, and propose some practical approaches to resolving, or at least reducing, them. I also look at the important distinction between describing the software that was used, and citing it.

1 Can I get a copy of the software that was used?

For many years I worked in management, and I met researchers who would not have been used in a paper.

The authors had cited software between CSS specific to DSGA-DA toolkit, which was only detailed knowledge of was used.

Software Credit Workshop

Blog post about the workshop and the topic of Software Credit

The workshop (15 October 2016 at the National History Museum, London) explored what contribution software can and should make for academic reputational credit. I.e., how the production of software tools and applications contributes to career advancement in the academic research setting for both researchers who builds software as part of their research and developers who build tools and support research.

Encouraging citation of software – introducing CITATION files

By Rebin Wilson, Fellow and postgraduate at the University of Southampton.

Put a plaintext file named CITATION in the root directory of your code, and put information in it about how to cite your software. Do on, do it now! It’s only take two minutes!

Software is very important in science – but good software takes time and effort that could be used to do other work instead. I believe that it is important to do this work, but to make it worthwhile, people need to get credit for their work, and in academia that means citations. However, it is often very difficult to find out how to cite a piece of software – sometimes it is hidden away somewhere in the manual or on the website, but often it requires sending an email to the author asking them how they want it cited. The effort that this requires means that many people don’t bother to cite the software they use, and thus the authors don’t get the credit that they need. We need to change this, so that software – which underlies a huge amount of important scientific work – gets the recognition it deserves.

Citation File Format (CFF)

CFF is a human- and machine-readable file format in YAML 1.2 which provides citation metadata for software. It is maintained openly on GitHub: https://github.com/citation-

The Future of Research Communications and e-Scholarship

www.software.ac.uk
Publication of Research Software:
Mechanisms and Practices
benefits of publication

scholarly record

credit

reproducibility

improving software quality
benefits of publication

- scholarly record
- credit
- reproducibility
- improving software quality
benefits of publication

- scholarly record
- credit
- reproducibility
- improving software quality
benefits of publication

- scholarly record
- credit
- reproducibility
- software quality
## options for publications

<table>
<thead>
<tr>
<th>REPOSITORY</th>
<th>REPOSITORY &amp; ARCHIVE</th>
<th>TRADITIONAL JOURNAL</th>
<th>SOFTWARE JOURNAL</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>pros?</strong></td>
<td><strong>pros?</strong></td>
<td><strong>pros?</strong></td>
<td><strong>pros?</strong></td>
</tr>
<tr>
<td><strong>cons?</strong></td>
<td><strong>cons?</strong></td>
<td><strong>cons?</strong></td>
<td><strong>cons?</strong></td>
</tr>
</tbody>
</table>
Publishing on a public platform for software source code (e.g. GitHub)

Deposit + basic documentation (README) + licence : minimum acceptable level of publishing software

Pros: discoverable with chances of reuse

Cons: 1) not archived; 2) not citable; 3) no quality control
GitHub + Archiving

Deposit + archive in e.g. Zenodo, figshare

**Pros:** 1) referenceable (DOI); 2) archived (long term)

**Cons:** citation to DOIs are not always supported by indexers
Software papers in traditional domain-specific journals

Many journals now accept papers *about* software (= where the primary goal of the paper is to describe a research software package and not to publish novel research results.

**Pros:** 1) reach the intended audience; 2) citable

**Cons:** no review of the software

Software papers in software-specific journals

These journals offer a venue for submitting papers describing their research software and may be domain specific or not.

Pros: 1) dedicated venue; 2) citable; 3) review of the software

Cons: software-focused, loss in domain knowledge prestige

## options for publications

<table>
<thead>
<tr>
<th>REPOSITORY</th>
<th>REPOSITORY + ARCHIVE</th>
<th>TRADITIONAL JOURNAL</th>
<th>SOFTWARE JOURNAL</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>discoverable</strong></td>
<td>discoverable</td>
<td>reach target audience</td>
<td>dedicated venue</td>
</tr>
<tr>
<td></td>
<td>referenceable</td>
<td>archived</td>
<td>citable</td>
</tr>
<tr>
<td></td>
<td>archived</td>
<td></td>
<td>review of software</td>
</tr>
<tr>
<td><strong>not archived</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>not citable</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>no quality control</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
why software papers?

‘native’ software citation not currently possible = not supported by indexers (work is ongoing to make this possible)

citation of papers is the only currency therefore software papers are the most reliable mechanism for credit
<table>
<thead>
<tr>
<th>软件特定的期刊</th>
<th>JORS</th>
<th>JOSS</th>
<th>SoftwareX</th>
</tr>
</thead>
<tbody>
<tr>
<td>重点：元论文</td>
<td>重点：软件</td>
<td>重点：高影响力</td>
<td>重点：软件和论文的全面审查</td>
</tr>
<tr>
<td>目标：再利用</td>
<td>深入软件审查</td>
<td>无纸化</td>
<td>无纸化</td>
</tr>
<tr>
<td>基于检查表的软件审查</td>
<td>深入软件审查</td>
<td>无纸化</td>
<td>无纸化</td>
</tr>
<tr>
<td>无纸化</td>
<td>无纸化</td>
<td>无纸化</td>
<td>无纸化</td>
</tr>
</tbody>
</table>
journal of open research software (JORS)
openresearchsoftware.metajnl.com

Ubiquity Press
peer reviewed software meta-papers describing research software with high reuse potential
focus on the metadata with basic checklist-based review of the software
journal of open source software (JOSS)
joss.theoj.org

NumFOCUS

“an open source developer-friendly journal for research software packages, designed to make it as easy as possible to create a software paper for a developer’s work”

focus on the software -- papers are very short
software specific journals - c

softwareX
www.journals.elsevier.com/softwarex

software with high impact track-record or potential in a field

review of the paper and of the software
call for reviewers!


handle 1-2 submissions per month
Conclusions

- growth in open source software
- call for transparency
- research ever more computational

are **all drivers** for the increase of research software publication

**bright future** for the publication of research software

groups and communities are addressing current **challenges**
thank you!

*Mahlzeit!*