

Software Citations at ACAT

Daniel S. Katz

Assistant Director for Scientific Software & Applications, NCSA

Research Associate Professor, CS

Research Associate Professor, ECE

Research Associate Professor, iSchool

dskatz@illinois.edu, d.katz@ieee.org, @danielskatz



ACAT 2017

21-25 August 2017

University of Washington, Seattle

National Center for Supercomputing Applications
University of Illinois at Urbana-Champaign

Software in research

- Claim: software (including services) essential for the bulk of research
- Evidence from surveys
 - UK academics at Russell Group Universities (2014)
 - Members of (US) National Postdoctoral Research Association (2017)
 - My research would not be possible without software: 67% / 63% (UK/US)
 - My research would be possible but harder: 21% / 31%
 - It would make no difference: 10% / 6%

S. Hettrick, "It's impossible to conduct research without software, say 7 out of 10 UK researchers," Software Sustainability Institute, 2014. Available at: <https://www.software.ac.uk/blog/2016-09-12-its-impossible-conduct-research-without-software-say-7-out-10-uk-researchers>

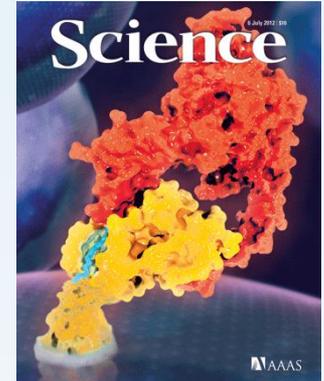
S.J. Hettrick, M. Antonioletti, L. Carr, N. Chue Hong, S. Crouch, D. De Roure, et al, "UK Research Software Survey 2014", Zenodo, 2014. doi: 10.5281/zenodo.14809.

U. Nangia and D. S. Katz, "Track 1 Paper: Surveying the U.S. National Postdoctoral Association Regarding Software Use and Training in Research," WSSPE5.1, 2017. doi: 10.6084/m9.figshare.5328442.v1



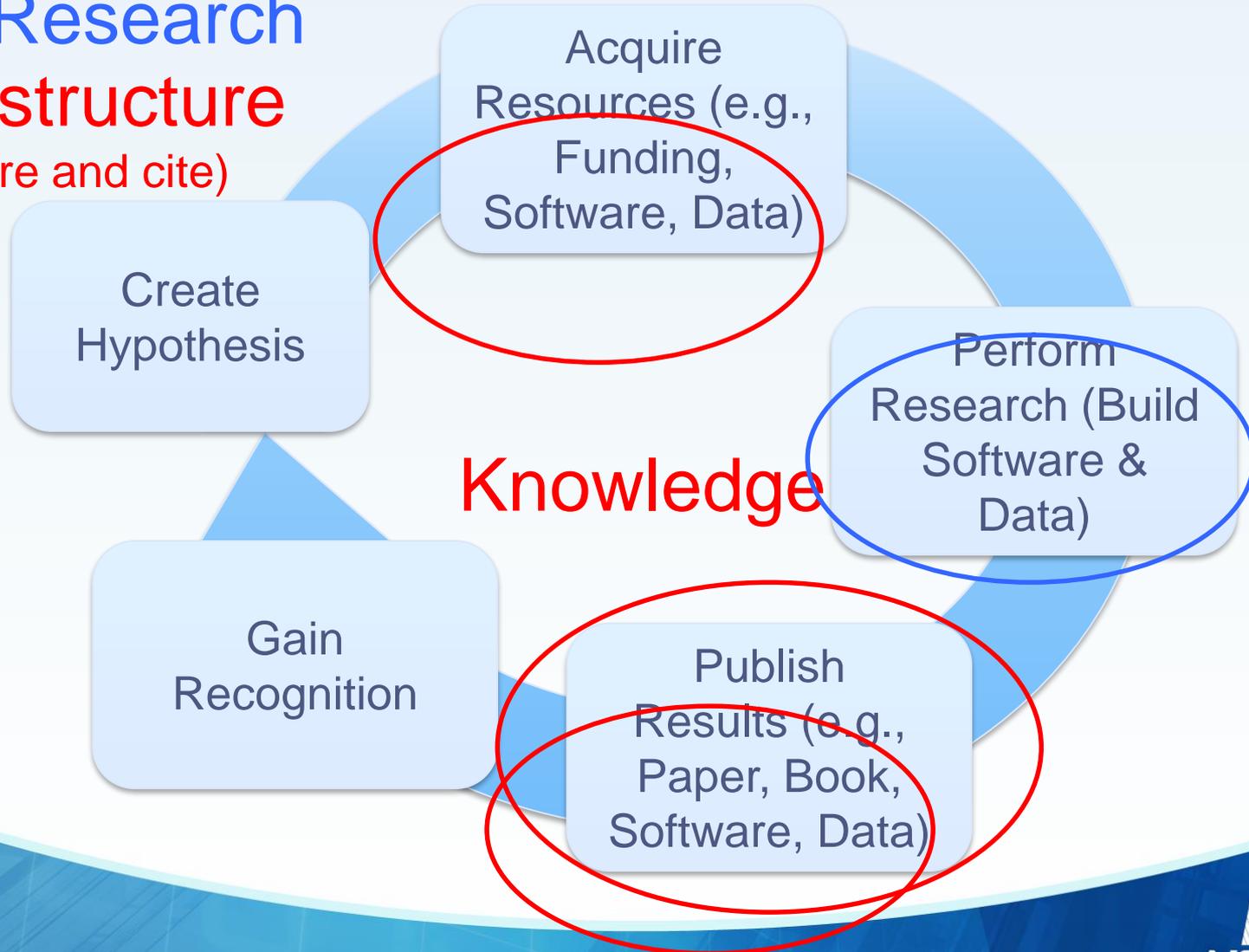
Software in scholarship

- Claim: software (including services) essential for the bulk of research
- Evidence from journals:
 - About half the papers in recent issues of Science were software-intensive projects
 - In Nature Jan–Mar 2017, software mentioned in 32 of 40 research articles
 - Average of 6.5 software packages mentioned per article



Software in research cycle

Research
Infrastructure
(share and cite)



How to better measure software contributions

- Citation system was created for papers/books
- We need to either/both
 1. Jam software into current citation system
 2. Rework citation system
 - Focus on 1 as possible; 2 is very hard.
- Overall challenge: not just to identify software in a paper
 - **To identify software used within research process**



Software citation principles: People & Process

- FORCE11 Software Citation group started July 2015 (co-leads Smith & Katz)
- WSSSPE3 Credit & Citation working group joined September 2015 (Niemeyer joined as co-lead)
- ~60 members (researchers, developers, publishers, repositories, librarians)
- Work on GitHub <https://github.com/force11/force11-scwg> & FORCE11 <https://www.force11.org/group/software-citation-working-group>
- Reviewed existing community practices & developed use cases
- Drafted software citation principles document
 - Started with data citation principles, updated based on software use cases and related work, updated based working group discussions, community feedback and review of draft, workshop at FORCE2016 in April
 - Discussion via GitHub issues, changes tracked
- Submitted, reviewed and modified (many times), now published (with reviews)
 - 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](https://doi.org/10.7717/peerj-cs.86) and <https://www.force11.org/software-citation-principles>
- Contains: principles (general statements), use cases (where the principles should apply), discussion (suggestions on how to apply principles)



Principle 1. 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**, such as publications and data; they should be included in the metadata of the citing work, for example in the reference list of a journal article, and should not be omitted or separated. Software should be cited on the same basis as any other research product such as a paper or a book, that is, authors should cite the appropriate set of software products just as they cite the appropriate set of papers.



Principle 2. 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.

Principle 3. 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.



Principle 4. Persistence

- **Unique identifiers and metadata describing the software and its disposition should persist** – even beyond the lifespan of the software they describe.

Principle 5. 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.**

Principle 6. 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.



Example 1: Make your software citable

- Publish it – if it's on GitHub, follow steps in <https://guides.github.com/activities/citable-code/>
- Otherwise, submit it to zenodo or figshare, with appropriate metadata (including authors, title, ..., citations of ... & software that you use)
- Get a DOI
- Create a CITATION file, update your README, tell people how to cite
- Also, can write a software paper and ask people to cite that (but this is secondary, just since our current system doesn't work well)



Example 2: Cite someone else's software in a paper

- Check for a CITATION file or README; if this says how to cite the software itself, do that
- If not, do your best following the principles
 - Try to include all contributors to the software (maybe by just naming the project)
 - Try to include a method for identification that is machine actionable, globally unique, interoperable – perhaps a URL to a release, a company product number
 - If there's a landing page that includes metadata, point to that, not directly to the software (e.g. the GitHub repo URL)
 - Include specific version/release information
- If there's a software paper, can cite this too, but not in place of citing the software



ACAT examples (unpublished software)

- Geant4 project, “Geant” [software], version 10.3.2, 2017. Available from <https://github.com/Geant4/geant4/releases/tag/v10.3.2> [accessed 2017-08-17]
- ROOT project, “Root” [software], version 6.10.4, 2017. Available from <https://github.com/root-project/root/releases/tag/v6-10-04> [accessed 2017-08-17]
- Eigen project, “Eigen” [software], version 3.3.4, 2017. Available from <https://bitbucket.org/eigen/eigen/> [accessed 2017-08-17]
- Python project, “Python” [software], version 3.6.2, 2017. Available from <https://www.python.org/downloads/release/python-362/> [accessed 2017-08-17]
- LLVM project, “LLVM Core” [software], version 4.0.1, Available from <http://releases.lvm.org/download.html#4.0.1> [accessed 2017-08-17]
- R project, “R” [software], version 3.4.1, Available from <https://cran.r-project.org/src/base/R-3/> [accessed 2017-08-17]
- TensorFlow Project, “TensorFlow” [software], version 1.3.0, Available from <https://github.com/tensorflow/tensorflow/releases/tag/v1.3.0> [accessed 2017-08-17]
- Ronan Collobert, Clement Farabet, Koray Kavukcuoglu, Soumith Chintala, Nicholas Leonard, Jonathan Tompson, Sergey Zagoruyko, Francisco Massa, Aysegul Dundar, Jonghoon Jin, Alfredo Canziani, Alban Desmaison, Cedric Deltheil, Hugh Perkins, “Torch” [software], commit a0bf77ff070ca27eb2de31c6465f8ffa4e399be2, available from <https://github.com/torch/torch7> [accessed 2017-08-17]



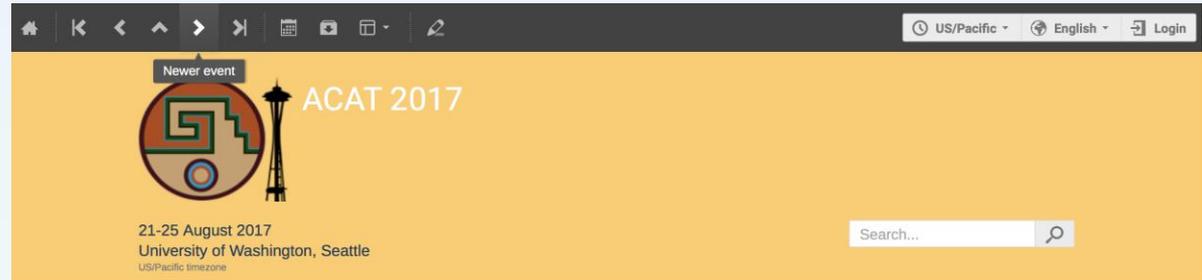
ACAT examples (published software)

- Stefan Pfenninger, Bryn Pickering, “calliope-project/calliope” [software], Release v0.5.2, Zenodo, 16 June 2017. <http://doi.org/10.5281/zenodo.810012>
- Lukas Heinrich and Kyle Cranmer, “diana-hep/packativity” [software], Initial Zenodo Release. Zenodo, 20 February 2017. <http://doi.org/10.5281/zenodo.309302>
- Anna Stasto, Bowen Xiao, David Zaslavsky, “SOLO” [software], version 1, figshare, 2014. <https://doi.org/10.6084/m9.figshare.1033996.v1>
- Edmund Noel Dawe, Piti Ongmongkolkul, Giordon Stark, “root_numpy: The interface between ROOT and NumPy,” Journal of Open Source Software, v2.16, August 2017. <https://doi.org/10.21105/joss.00307>



ACAT software citation experiment

- ACAT paper proceedings coming up
- This is something organizers (and I) want to do



- Overview
- Scientific Programme
- Call for Abstracts
- Timetable
- Contribution List
- Author List
- Speaker List
- Book of Abstracts
- Registration
- Participant List
- Videoconference Rooms
- Contact
- Presenter's Information & Proceedings**
- ACAT History
- The ACAT Series Blog
- ACAT Organization
- Accommodations
- Satellite Meetings
- Travel
- Seattle - Transportation, Things to Do, The Eclipse, etc.



- Proceedings link

Presenter's Information & Proceedings

If you have any question about the information or need something special, please do not hesitate to get in touch at acat2017@uw.edu! This page talks about plenary and parallel talks, posters, and paper submissions. We strongly encourage paper submissions!

Oral Presentations

There are two types of oral presentations:

- Plenary: 25 minutes + 5 minutes of questions
- Parallel: 15 minutes + 5 minutes of questions
- Parallel (Track 3 ONLY): 20 minutes + 5 minutes of questions

In all cases we will be running our talks from a central computer. Your talks must be uploaded to this Indico agenda, attached to your talk. After you log in, you should have modification access (contact us if you have trouble).

Any format is acceptable. But we will be running the presentation computers with the following software which will be used in order of preference for talk display:

- PowerPoint: we have a fully updated version of PowerPoint2016 from Office365 for both plenary and parallel
- OpenOffice (Libre Office, 5.4.3 or a newer stable version)
- PDF: The current version of the free Acrobat Reader software will be installed on each presentation computer.

Please do not link to your presentation stored elsewhere: we'd like this Indico to become an archive of the conference! Any file format can be uploaded though only the above two will be directly displayed during the presentation. If possible, upload the source (PowerPoint, KeyNote, LaTeX tar file, etc.) as then others can re-use easily parts of your presentation (with proper attribution, of course!).

Each room has a nice projector in it - they are all by default in a 16:9 aspect ratio.

The presentation computers are laptops, a few years old, with an up-to-date version of Win10 (Creators Update installed). These are not ultra powerful machines. If you have huge scatter plots or the like, there will be some lag displaying them!

Posters



ACAT software citation experiment

- On that page, at the bottom (as of Wed 23 Aug):

Citing Software

ACAT is trying an experiment this year: software citation of software. For a general background on this topic, see the talk given at the conference.

More info to come.

- “More info to come”
- Including guidelines from the organizers
- And these slides



Journal of Open Source Software (JOSS)

- In the meantime, there's JOSS
- A developer friendly journal for research software packages
- “If you've already licensed your code and have good documentation then we expect that it should take **less than an hour** to prepare and submit your paper to JOSS”
- Everything is open:
 - Submitted/published paper: <http://joss.theoj.org>
 - Code itself: where is up to the author(s)
 - Reviews & process: <https://github.com/openjournals/joss-reviews>
 - Code for the journal itself: <https://github.com/openjournals/joss>
- Zenodo archives JOSS papers and issues DOIs
- First paper submitted 4 May 2016
 - 31 May 2017: 111 accepted papers, 41 under review, ~15 submitted (pre-review)
 - 31 July 2017: 125 accepted papers, 35 under review, ~30 submitted (pre-review)



Working group status & next steps

- Software Citation Working Group (co-chairs Smith, Katz, Niemeyer) ended May 2017
- Software Citation Implementation group (co-chairs Katz, Fenner, Chue Hong) started (slowly) May 2017
- Now planning...
 - Work with institutions, publishers, funders, researchers, etc.,
 - Considering endorsement period for both individuals and organizations
 - Want to endorse? Email/talk to me
 - Write full implementation examples paper?
- Want to join? Sign up on new FORCE11 group page
 - <https://www.force11.org/group/software-citation-implementation-working-group>

