

# Software Citation at CHEP

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



**23RD INTERNATIONAL CONFERENCE ON  
COMPUTING IN HIGH ENERGY AND NUCLEAR PHYSICS**

9-13 July 2018  
National Palace of Culture  
Sofia, Bulgaria



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>

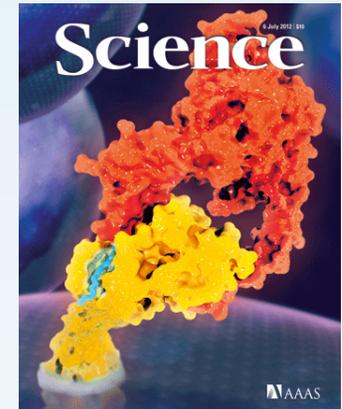
S.J. Hettrick, M. Antonioletti, L. Carr, N. Chue Hong, S. Crouch, D. De Roure, et al, “UK Research Software Survey 2014”, Zenodo, 2014. doi: 10.5281/zenodo.14809.

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



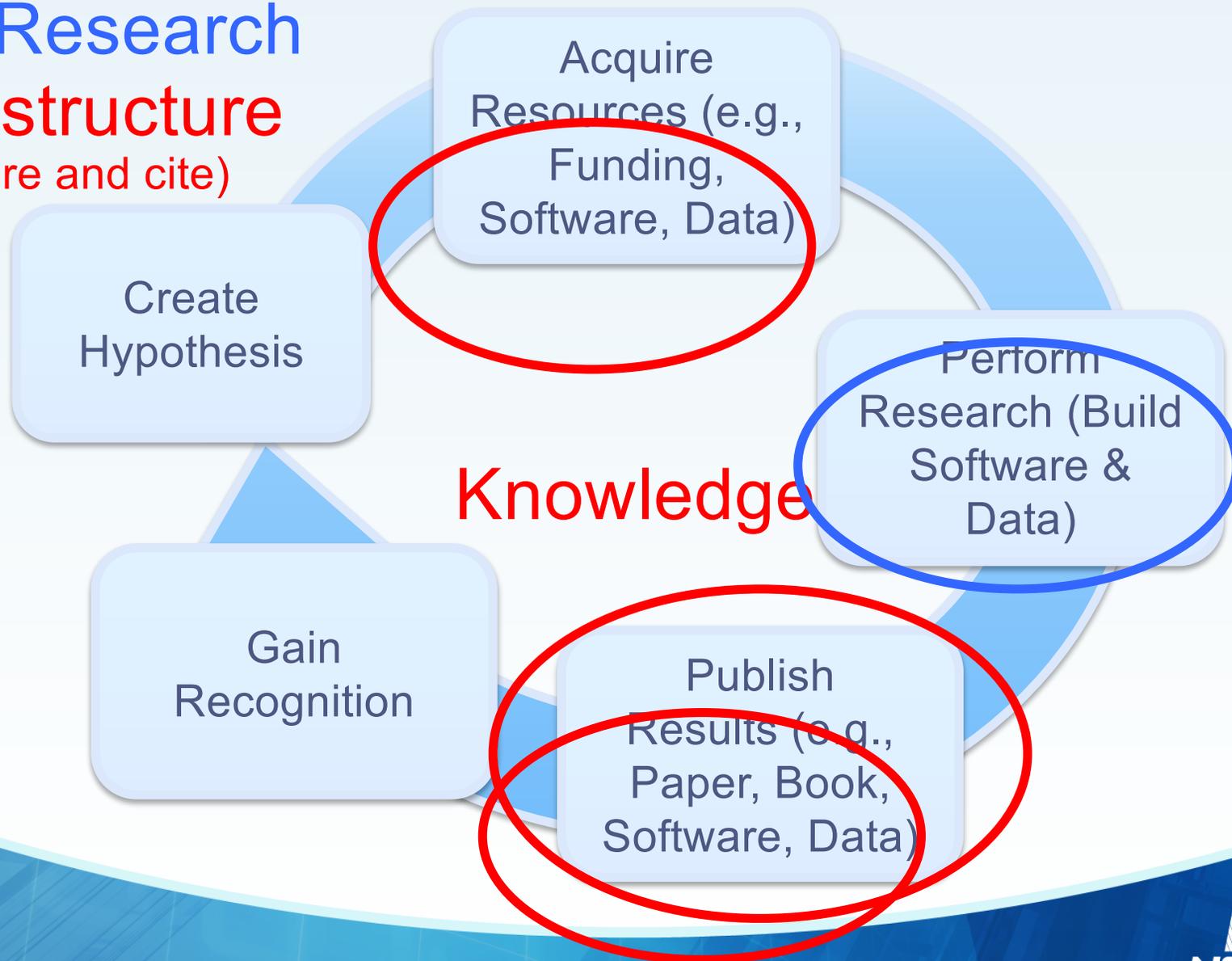
# Software in scholarship

- 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/peeri-cs.86](https://doi.org/10.7717/peeri-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



# CHEP examples (unpublished software)

- Geant4 project, “Geant” [software], version 10.4.2, 2018. Available from <https://github.com/Geant4/geant4/releases/tag/v10.4.2> [accessed 2018-07-09]
- ROOT project, “Root” [software], version 6.14.0, 2018. Available from <https://github.com/root-project/root/releases/tag/v6-14-00> [accessed 2018-07-09]
- Eigen project, “Eigen” [software], version 3.3.4, 2017. Available from <https://bitbucket.org/eigen/eigen/> [accessed 2018-07-09]
- Python project, “Python” [software], version 3.6.2, 2017. Available from <https://www.python.org/downloads/release/python-362/> [accessed 2018-07-09]
- LLVM project, “LLVM Core” [software], version 6.0.1. Available from <http://releases.lvm.org/download.html#6.0.1> [accessed 2018-07-09]
- R project, “R” [software], version 3.5.1. Available from <https://cran.r-project.org/src/base/R-3/> [accessed 2018-07-09]
- TensorFlow Project, “TensorFlow” [software], version 1.7.1, 2018. Available from <https://github.com/tensorflow/tensorflow/releases/tag/v1.7.1> [accessed 2018-07-09]
- 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 2018-07-09]



# CHEP examples (published software)

- Stefan Pfenninger, Bryn Pickering, Katrin Leinweber; Graeme Hawker, “calliope-project/calliope” [software], Release v0.6.2, Zenodo, 4 June 2018. <http://doi.org/10.5281/zenodo.1262406>
- Lukas Heinrich and Kyle Cranmer, “diana-hep/packtivity” [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>



# 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, 56 under review and pre-review
  - 6 July 2018: 320 accepted papers, 75 under review and 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 May 2017
- Want to join? Sign up on FORCE11 group page
  - <https://www.force11.org/group/software-citation-implementation-working-group>
- Now starting up task forces:
  - CodeMeta
  - Guidelines for communities (disciplines)
  - Publisher adoption
  - Repository best practices
  - Sign up via GitHub: <https://github.com/force11/force11-sciwg>



# CHEP software citation experiment

- Following semi-successful example of ACAT 2017...
- CHEP paper proceedings coming up
- This is something organizers and I want to do
  - When you write your paper, please be sure to cite the software you use
  - If you produce software that other people use, please make it citable and document how to cite it
  - If you review papers, please complain about software that isn't cited
- If you have examples of good software citations, please let me know so they can be added to the proceeding instructions

