Towards sustainable research software

Daniel S. Katz (d.katz@ieee.org, @danielskatz)
Chief Scientist, NCSA
Associate Research Professor, CS, ECE, iSchool
University of Illinois at Urbana Champaign

McGill Centre for the Convergence of Health and Economics
Innovate the Way We Innovate in Agri-Food Sector and in Society
Carnegie Mellon University
PSC
University of Pittsburgh
DESAUTELS | McGill
Faculty of Management
Faculté de gestion

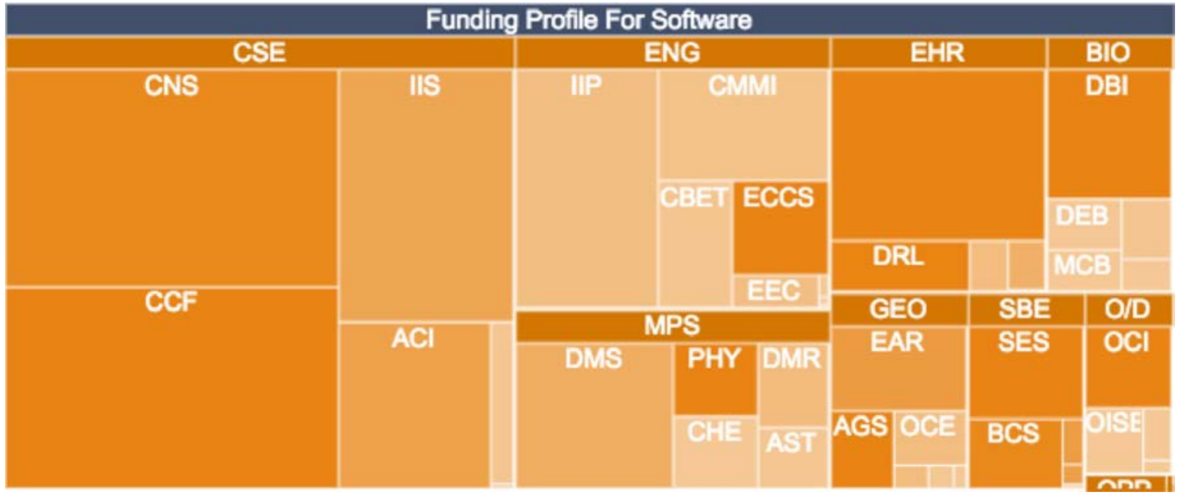
Precision Convergence Webinar Series
2 December 2021

ILLINOIS
NCSA | National Center for Supercomputing Applications

https://doi.org/10.5281/zenodo.5748175

Why do we care about research software?

- US NSF
  - 1995-2016: 18,592 awards totalling $9.6 billion with project abstracts that topically include “software”
  - ~20% of the overall NSF research budget
- US DOE
  - Of three ECP areas, most of two (application development & software) technology are research software
  - According to Paul Messina in 2017, “ECP is a 7-year project with a cost range of $3.5B–$5.7B”
- Digital Research Alliance of Canada
  - Three areas: advanced research computing, data management, and research software



Collected from http://www.dia2.org in 2017

Why do we care about research software?

- Surveys of UK academics at Russell Group Universities (2014) and members of (US) National Postdoctoral Research Association (2017):

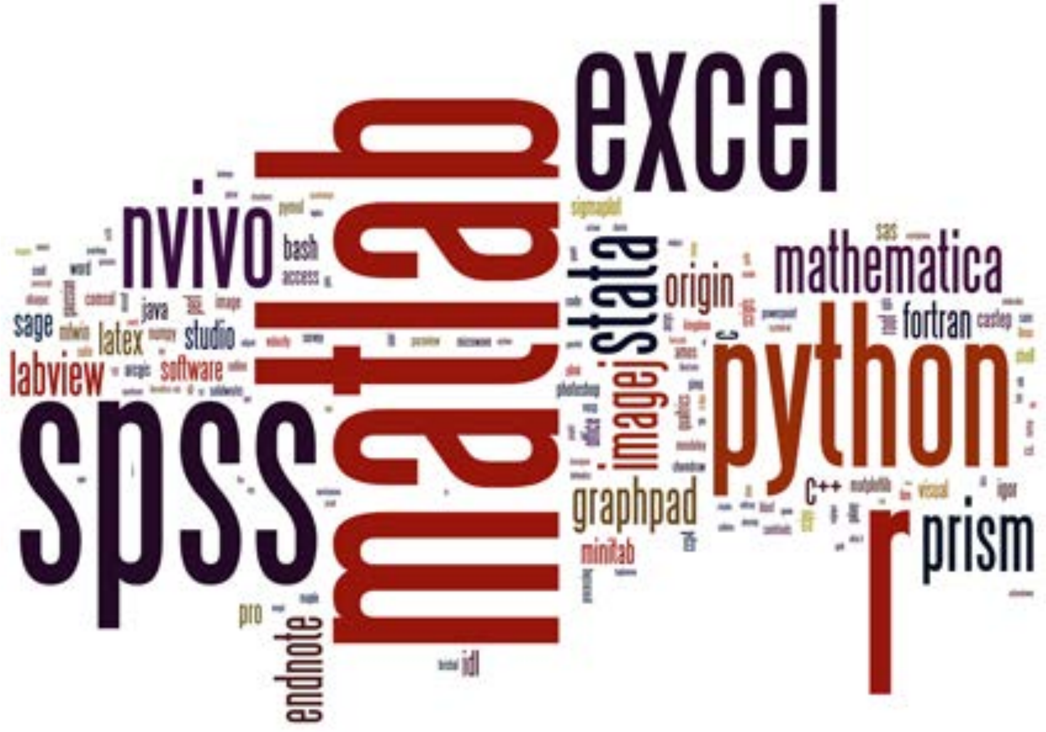
  - I use research software: 92% / 95% (UK/US)
  - My research would not be possible without software: 67% / 63%
  - My research would be possible but harder: 21% / 31%
  - I develop my own software: 56% / 28%

S. Hettrick; https://www.software.ac.uk/blog/2016-09-12-its-impossible-conduct-research-without-software-say-7-out-10-uk-researchers
S.J. Hettrick, et al.; 10.5281/zenodo.14809

U. Nangia and D. S. Katz; 10.6084/m9.figshare.5328442.v1

Why do we care about research software?

- 40 papers in Nature (Jan-Mar 2016)
  - 32 explicitly mentioned software
  - Average of 6.5 software tools/paper
  - Most of which were research software
- Top 100-cited papers:
  - 6 of top 13 are software papers
  - "… the vast majority describe experimental methods or software that have become essential in their fields."



Nangia and Katz; 10.1109/eScience.2017.78

"Top 100-cited papers of all time," Nature, 2014 10.1038/514550a

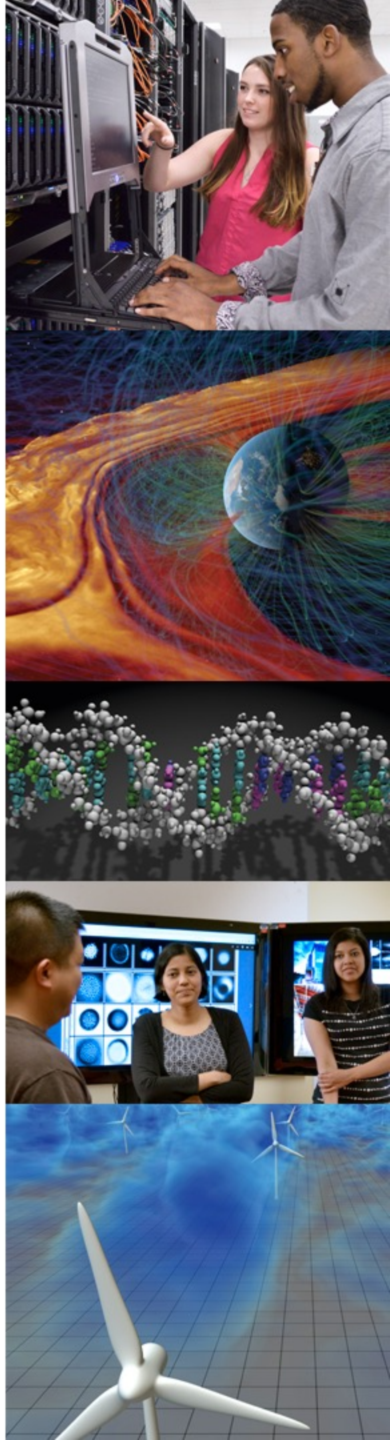
Research and research software vision

- All research software that can be is open Open Science
- All research software is high-quality and robust Software Engineering
- All research software is findable, accessible, and usable & used by others (for their own research) FAIR
    - And is cited when it is used
- All contributors to research software are recognized for their work Software Citation, JOSS
    - With good careers RSE +
- All research software is sustained as long as it is useful SSI, URSSI, AUSSI
- All research is reproducible Reproducibility

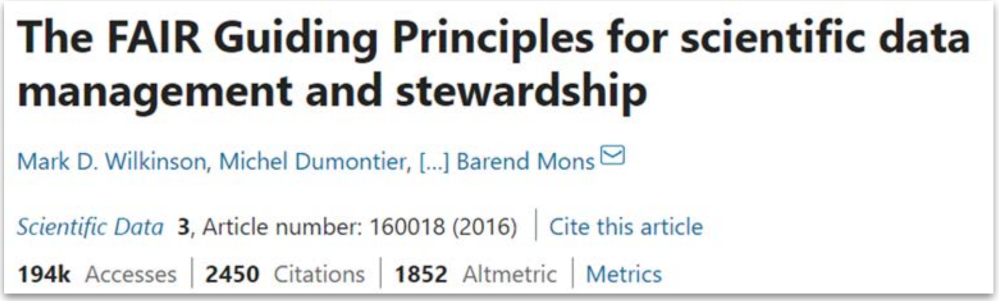
Note overlaps in terms of incentives and policies; all start with recognition of research software

FAIR for Research Software (FAIR4RS)

https://doi.org/10.5281/zenodo.5748175



The FAIR Principles



The FAIR Guiding Principles for scientific data management and stewardship

Mark D. Wilkinson, Michel Dumontier, [...] Barend Mons

Scientific Data **3**, Article number: 160018 (2016) | Cite this article

194k Accesses | **2450** Citations | **1852** Altmetric | Metrics

A set of principles, to ensure that data are shared in a way that enables and enhances reuse by humans and machines

Findable

F1. (Meta)data are assigned a globally unique and eternally persistent identifier.

F2. Data are described with rich metadata.

F3. (Meta)data are registered or indexed in a searchable resource.

F4. Metadata specify the data identifier.

Accessible

A1. (Meta)data are retrievable by their identifier using a standardized communications protocol.

A1.1. The protocol is open, free, and universally implementable.

A1.2. The protocol allows for an authentication and authorization procedure, where necessary.

A2. Metadata are accessible, even when the data are no longer available.

Interoperable

I1. (Meta)data use a formal, accessible, shared, and broadly applicable language for knowledge representation.

I2. (Meta)data use vocabularies that follow FAIR principles.

I3. (Meta)data include qualified references to other (meta)data.

Reusable

R1. (Meta)data have a plurality of accurate and relevant attributes.

R1.1. (Meta)data are released with a clear and accessible data usage license.

R1.2. (Meta)data are associated with their provenance.

R1.3. (Meta)data meet domain-relevant community standards.

https://doi.org/10.5281/zenodo.5748175

FAIR for non-data objects: some context

- FAIR Principles, at a high level, are intended to apply to all research objects; both those used in research and those that are research outputs
- Text in principles often includes "(Meta)data …"
  - Shorthand for "metadata and data …"
- Principles applied via dataset creators and repositories, collectively responsible for creating, annotating, indexing, preserving, sharing the datasets and their metadata
- What about non-data objects?
  - While they can often be stored as data, they are not just data
- While high level goals (F, A, I, R) are mostly the same, the details and how they are implemented depend on
  - How objects are created and used
  - How/where the objects are stored and shared
  - How/where metadata is stored and indexed
- Work needed to define, then implement, then adopt principles

https://doi.org/10.5281/zenodo.5748175

FAIR for non-data objects: support

- FAIR Principles, are intended to apply to all digital objects (Wilkinson et al. 2016)
- We focus on the adaptation and adoption of the FAIR principles to research software

***Recommendation n°5** :*

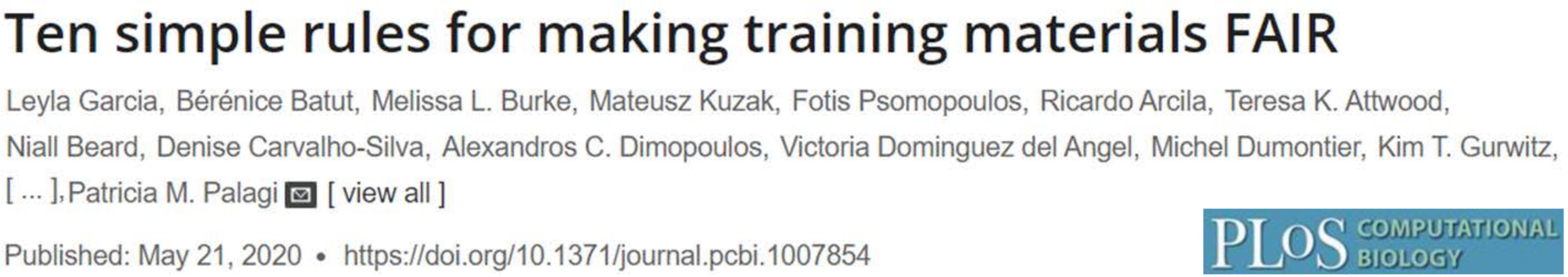
*Recognise that FAIR guidelines will require **translation for other digital objects** and support such efforts.*

2020: 'Six Recommendations for Implementation of FAIR Practice'

(FAIR Practice Task Force EOSC, 2020)

https://doi.org/10.5281/zenodo.5748175

FAIR for non-data objects: some efforts

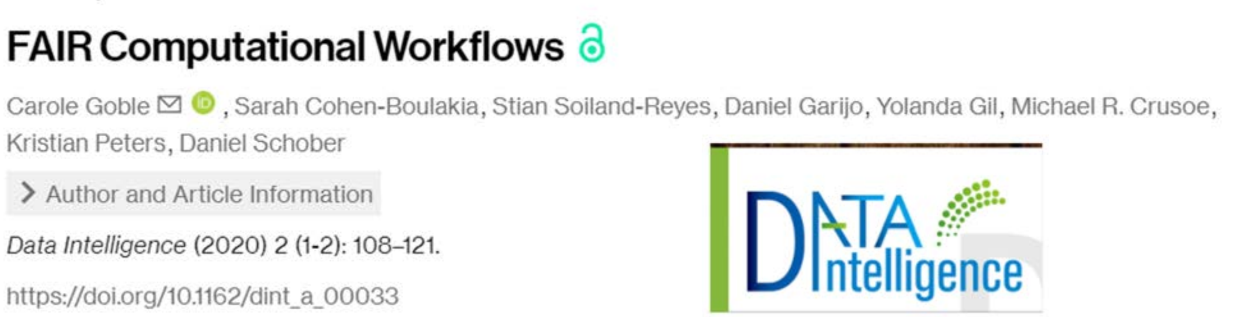


Ten simple rules for making training materials FAIR

Leyla Garcia, Bérénice Batut, Melissa L. Burke, Mateusz Kuzak, Fotis Psomopoulos, Ricardo Arcila, Teresa K. Attwood, Niall Beard, Denise Carvalho-Silva, Alexandros C. Dimopoulos, Victoria Dominguez del Angel, Michel Dumontier, Kim T. Gurwitz, [...], Patricia M. Palagi ✉ [view all]

Published: May 21, 2020 • https://doi.org/10.1371/journal.pcbi.1007854

January 01 2020



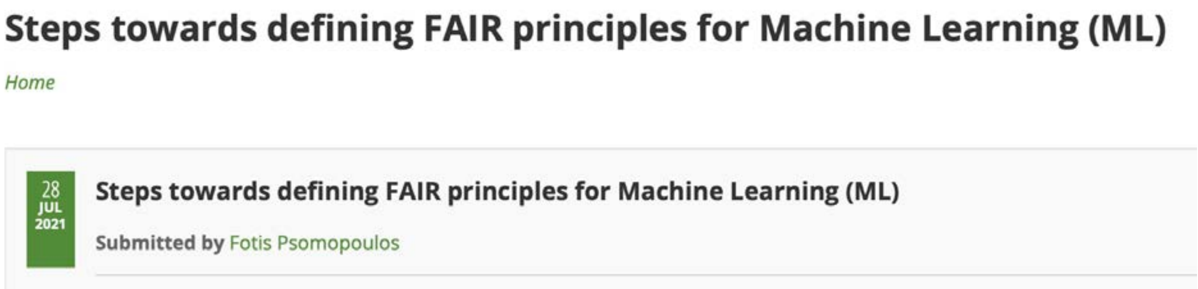
FAIR Computational Workflows

Carole Goble ✉, Sarah Cohen-Boulakia, Stian Soiland-Reyes, Daniel Garijo, Yolanda Gil, Michael R. Crusoe, Kristian Peters, Daniel Schober

> Author and Article Information

Data Intelligence (2020) 2 (1-2): 108–121.

https://doi.org/10.1162/dint_a_00033



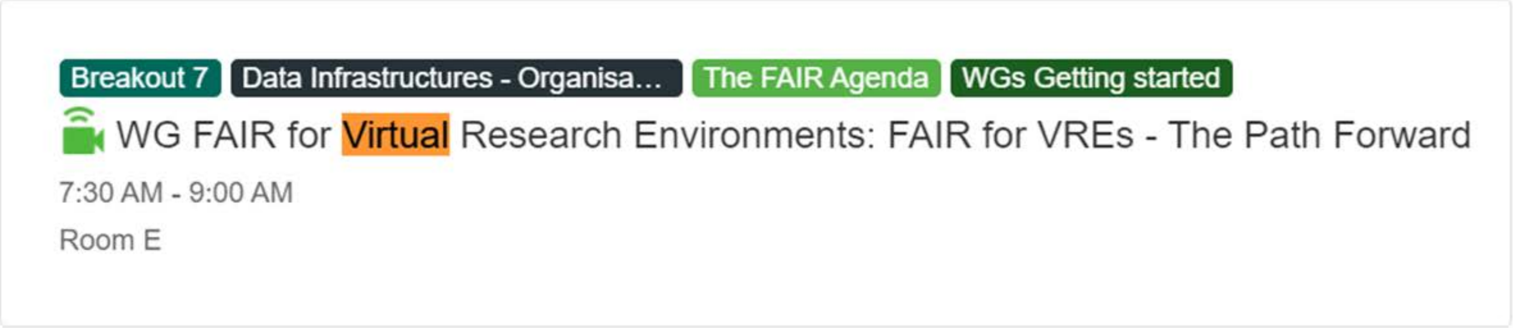
Steps towards defining FAIR principles for Machine Learning (ML)

Home

28 JUL 2021

Steps towards defining FAIR principles for Machine Learning (ML)

Submitted by Fotis Psomopoulos



Breakout 7 | Data Infrastructures - Organisa... | The FAIR Agenda | WGs Getting started

WG FAIR for Virtual Research Environments: FAIR for VREs - The Path Forward

7:30 AM - 9:00 AM

Room E

RESEARCH DATA ALLIANCE

ILLINOIS NCSA

https://doi.org/10.5281/zenodo.5748175

FAIR for Research Software (FAIR4RS)

- Working group defining FAIR principles for research software
  - Led by Michelle Barker, Neil Chue Hong, Leyla Garcia, Morane Gruenpeter, Jennifer Harrow, Daniel S. Katz, Carlos Martinez, Paula A. Martinez, Fotis Psomopoulos

FORCE11 The Future of Research Communications and e-Scholarship

RDA RESEARCH DATA ALLIANCE

<ReSA> Research Software Alliance

FAIR4RS initial subgroups

1. A fresh look at FAIR for Research Software
   - Examined the FAIR principles in the context of research software from scratch, not based on pre-existing work; published: Katz DS, Gruenpeter M, Honeyman T, et al. (2021). A Fresh Look at FAIR for Research Software. arXiv:2101.10883 [cs.SE], https://arxiv.org/abs/2101.10883
2. FAIR work in other contexts
   - Analyzed how FAIR principles are applied to research objects other than data/software – final report
3. Research software definition
   - Reviewing existing definitions and to specify the scope for the WG outputs – draft report
4. New research related to FAIR Software
   - Review recent research and studies around FAIR software
   - Via up-to-date identification of approaches that can help structure FAIR4RS work, in form of Zotero reading list and short report on important insights from review and survey – draft report

Recent working group status

- ~35 webinars and talks overall
- Jan – Feb 2021: Initial analysis of subgroup work led to a set of questions
- March 2021: Working group's input on these questions published
- April 2021: Group leads+ held writing sprint and assembled draft from subgroup products and initial community input
- 17 – 30 May 2021: Working group review of initial draft
- 11 June – 11 July 2021: Official community review (part of the RDA process) of second draft
  - ~280 people involved in process (subgroups, input to questions, review, …)
- Now: WG is drafting v1.0 FAIR4RS principles for preprint/journal
- Also now: starting new subgroups on adoption and future governance

https://doi.org/10.5281/zenodo.5748175

FAIR4RS principles

Findable: Software, and its associated metadata, is easy to find for both humans and machines.

F1. Software is assigned a globally unique and persistent identifier

- F1.1. Different components of the software are assigned distinct identifiers representing different levels of granularity
- F1.2. Different versions of the same software are assigned distinct identifiers

F2. Software is described with rich metadata

F3. Metadata clearly and explicitly include the identifier of the software they describe

F4. Metadata are FAIR and are searchable and indexable

Accessible: Software, and its metadata, is retrievable via standardized protocols.

A1. Software is retrievable by its identifier using a standardized communications protocol

- A1.1. The protocol is open, free, and universally implementable
- A1.2. The protocol allows for an authentication and authorization procedure, where necessary

A2. Metadata are accessible, even when the software is no longer available

Interoperable: Software interoperates with other software through exchanging data and/or metadata, and/or through interaction via application programming interfaces (APIs), described through standards.

I1. Software reads, writes and exchanges data in a way that meets domain-relevant community standards

I2. Software includes qualified references to other objects

Reusable: Software is both usable (it can be executed) and reusable (it can be understood, modified, built upon, or incorporated into other software).

R1. Software is described with a plurality of accurate and relevant attributes

- R1.1. Software is given a clear and accessible license
- R1.2. Software is associated with detailed provenance

R2. Software includes qualified references to other software

R3. Software meets domain-relevant community standards

FAIR4RS WG. (2021, June). FAIR Principles for Research Software

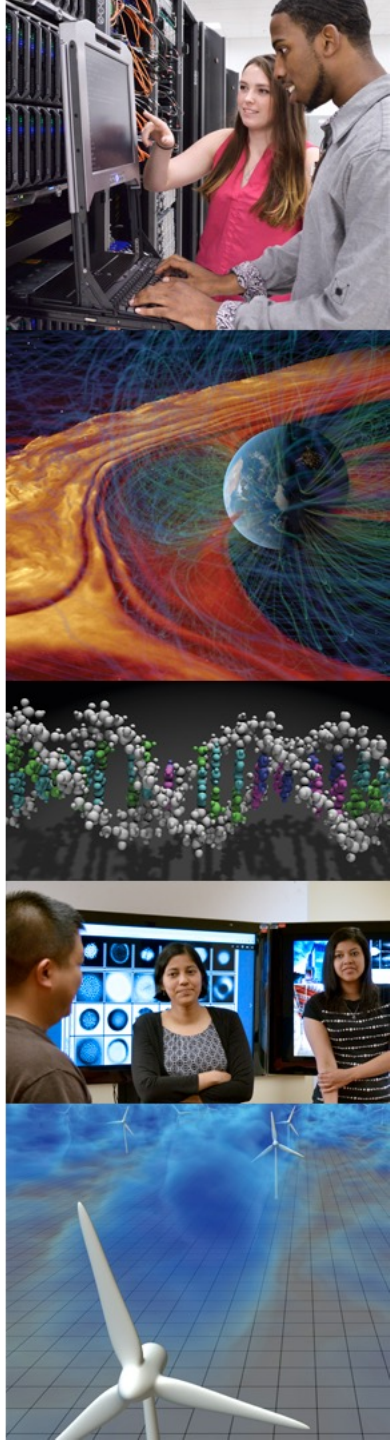
https://doi.org/10.5281/zenodo.5748175

Personal view of FAIR4RS status

- Original FAIR principles mixed metadata and data, e.g., “(Meta)data,” too strongly
  - Much of the metadata part translates directly to metadata about software
  - The data part doesn't
- F & A: basically not changed, but gaps appear
- I & R: multiple possible definitions that need to be resolved
- Lots of ecosystem gaps (open questions), particularly related to metadata, archiving, versions
  - Where is metadata stored? (in code repository for open source?, for closed source?, in archival repository?, in registry?)
  - Where is code archived? (GitHub/Gitlab are not archival, registries are not archival, repositories? Software Heritage?)
  - Different use cases need specific version, latest version, all versions

https://doi.org/10.5281/zenodo.5748175

Software Citation



https://doi.org/10.5281/zenodo.5748175

FORCE11 Software Citation Working Group (2015-16)

- Documented differences between software and data; defined software citation challenges
  - Katz DS, Niemeyer KE, et al. (2016) Software vs. data in the context of citation. PeerJ Preprints 4:e2630v1. DOI: 10.7287/peerj.preprints.2630v1
  - Niemeyer KE, Smith AM, Katz DS. (2016) The challenge and promise of software citation for credit, identification, discovery, and reuse. ACM Journal of Data and Information Quality, 7(4):16. DOI: 10.1145/2968452
- Created software citation principles
  - 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

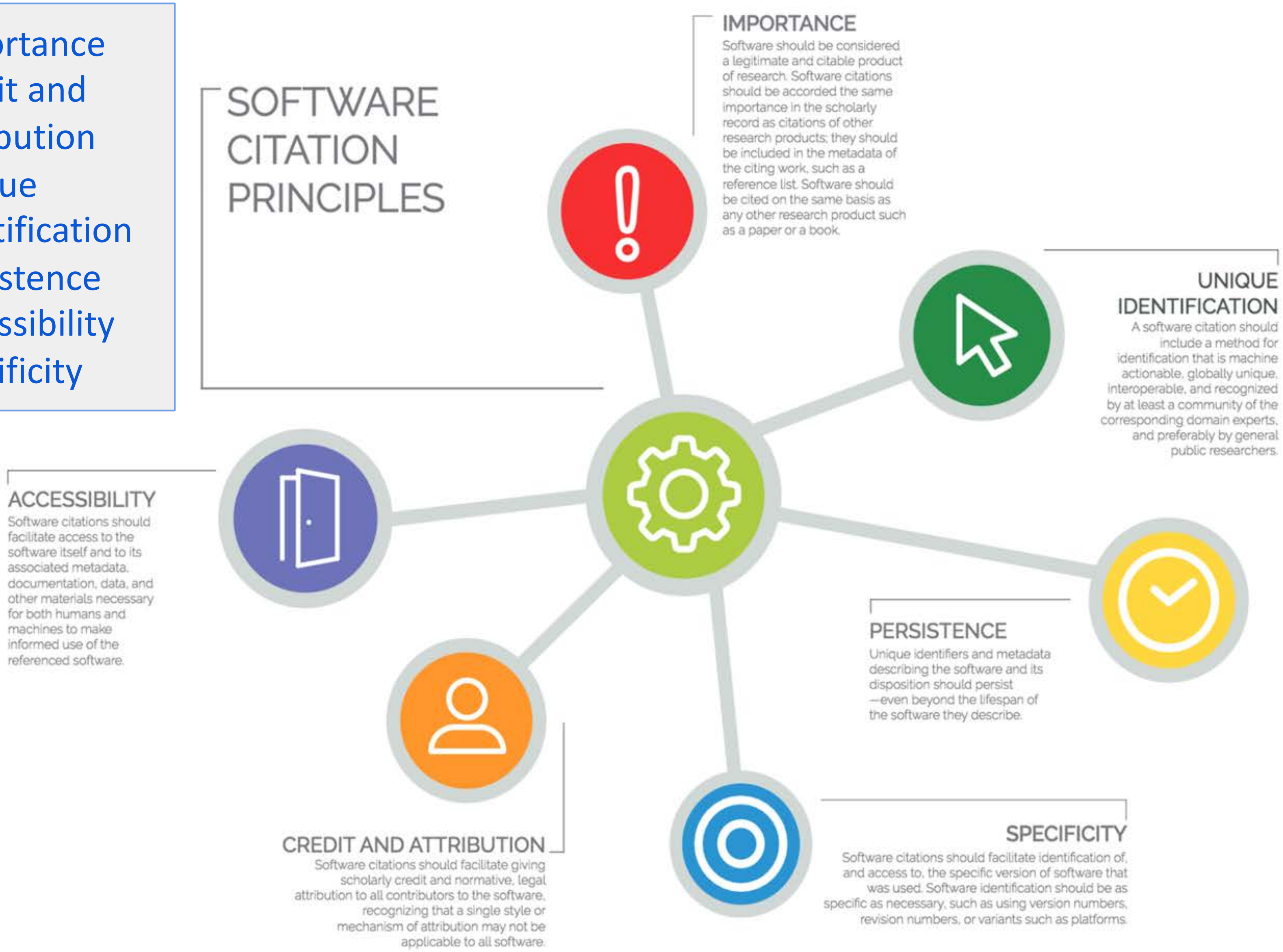
FORCE11
The Future of Research Communications and e-Scholarship

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

Co-Chairs: Arfon M. Smith, Daniel S. Katz, Kyle E. Niemeyer

Software is a critical part of modern research...

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



SOFTWARE CITATION PRINCIPLES

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; they should be included in the metadata of the citing work, such as a reference list. Software should be cited on the same basis as any other research product such as a paper or a book.

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.

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.

PERSISTENCE
Unique identifiers and metadata describing the software and its disposition should persist —even beyond the lifespan of the software they describe.

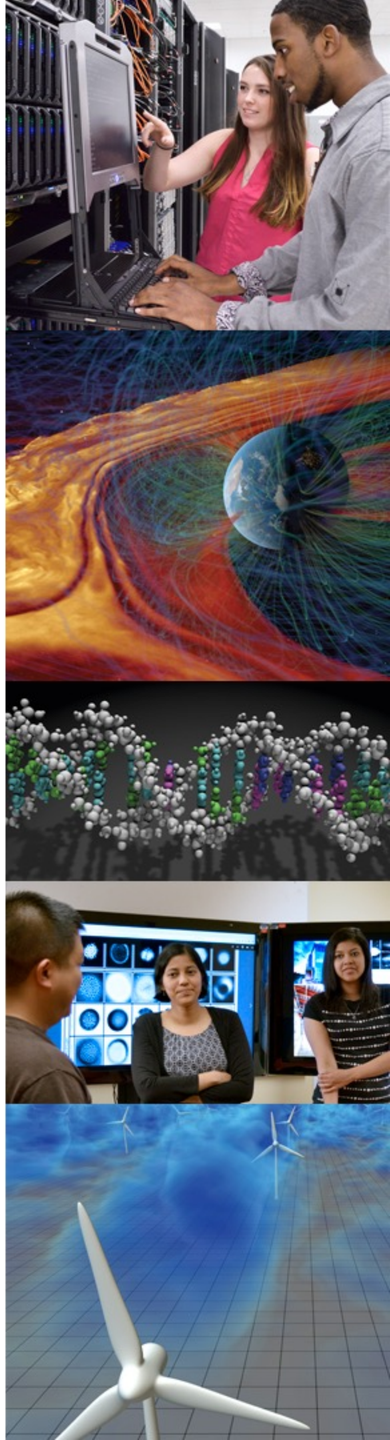
CREDIT AND ATTRIBUTION
Software citations should facilitate giving scholarly credit and normative, legal attribution to all contributors to the software, recognizing that a single style or mechanism of attribution may not be applicable to all software.

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.

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



FORCE11 Software Citation Implementation Working Group (2017-present)

- Initial goals:
  - Write out the “small amount” of detail needed to implement the principles
  - Coordinate research & other work going on in many areas
  - Work with communities to actually implement the principles
    - Publishers, conferences, repositories, indexers, funders, etc.
  - Co-chairs: Martin Fenner, Neil Chue Hong, Daniel S. Katz
- Quickly realized “small amount” of detail wasn’t small, scattered progress wasn't sufficient, underlying challenges not being addressed
  - D. S. Katz, D. Bouquin, N. P. Chue Hong, J. Hausman, C. Jones, D. Chivvis, T. Clark, M. Crosas, S. Druskat, M. Fenner, T. Gillespie, A. Gonzalez-Beltran, M. Gruenpeter, T. Habermann, R. Haines, M. Harrison, E. Henneken, L. Hwang, M. B. Jones, A. A. Kelly, D. N. Kennedy, K. Leinweber, F. Rios, C. B. Robinson, I. Todorov, M. Wu, Q. Zhang, "Software Citation Implementation Challenges", arXiv 1905.08674 [cs.CY], 2019.

https://doi.org/10.5281/zenodo.5748175

Challenges

Technical

- Complexity of software types: open source, closed source; published, unpublished; versioned, unversioned; developed by citer, not developed by citer; services, containers, executables
- How to uniquely identify software of each type (ideally as uniformly as possible)
- How to define and store citation metadata for each type
- How to access metadata and convert it as needed
- How to count citations across versions

- Realization: metadata is fundamental

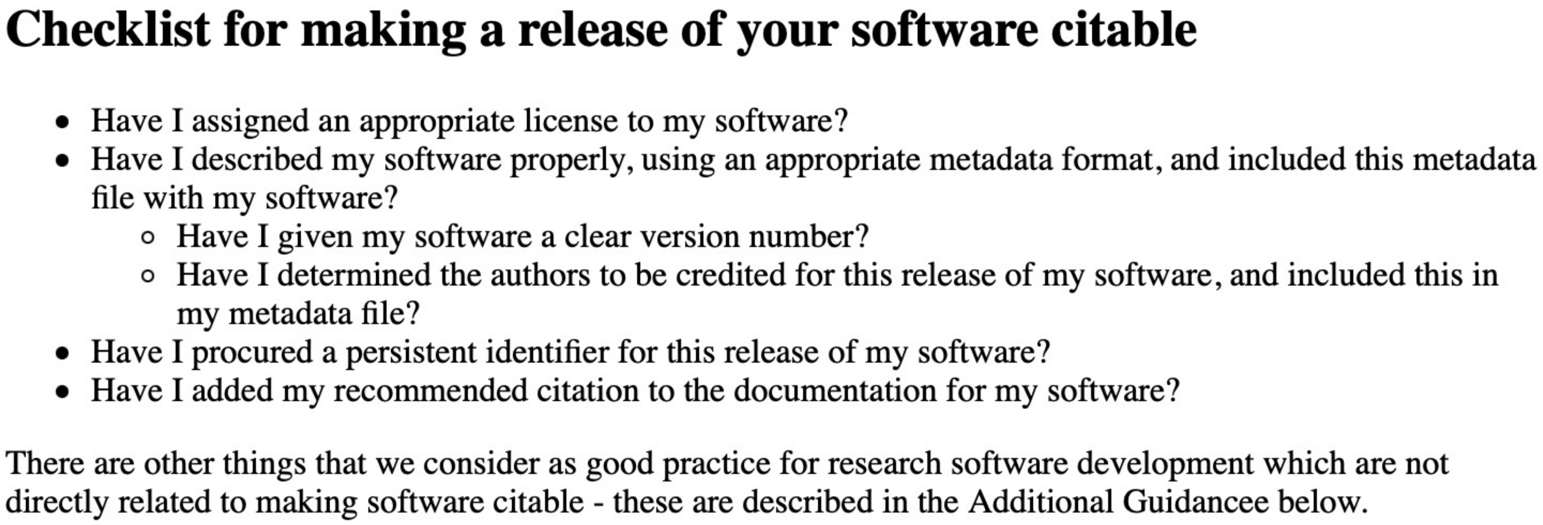
Social

- Need groups that work on implementation in context
  - Disciplinary communities
  - Publishers
  - Repositories
  - Indexers
  - Funders
  - Institutions
- Groups need to come together, run pilots to establish norms
- Example community effort
  - N. P. Chue Hong and D. S. Katz talked to the INCF/OCNS Software WG 22 Nov 2021
  - Will support the group taking this forward to the community

https://doi.org/10.5281/zenodo.5748175

Implementation task forces & outputs

- Guidance Task Force developed checklists
  - For paper authors who want to cite software
    - N. P. Chue Hong, et al., "Software Citation Checklist for Authors," Zenodo, 15-Oct-2019. 10.5281/zenodo.3479198
  - For software developers who want to make their software citable
    - N. P. Chue Hong, et al., "Software Citation Checklist for Developers," Zenodo, 15-Oct-2019. 10.5281/zenodo.3482768



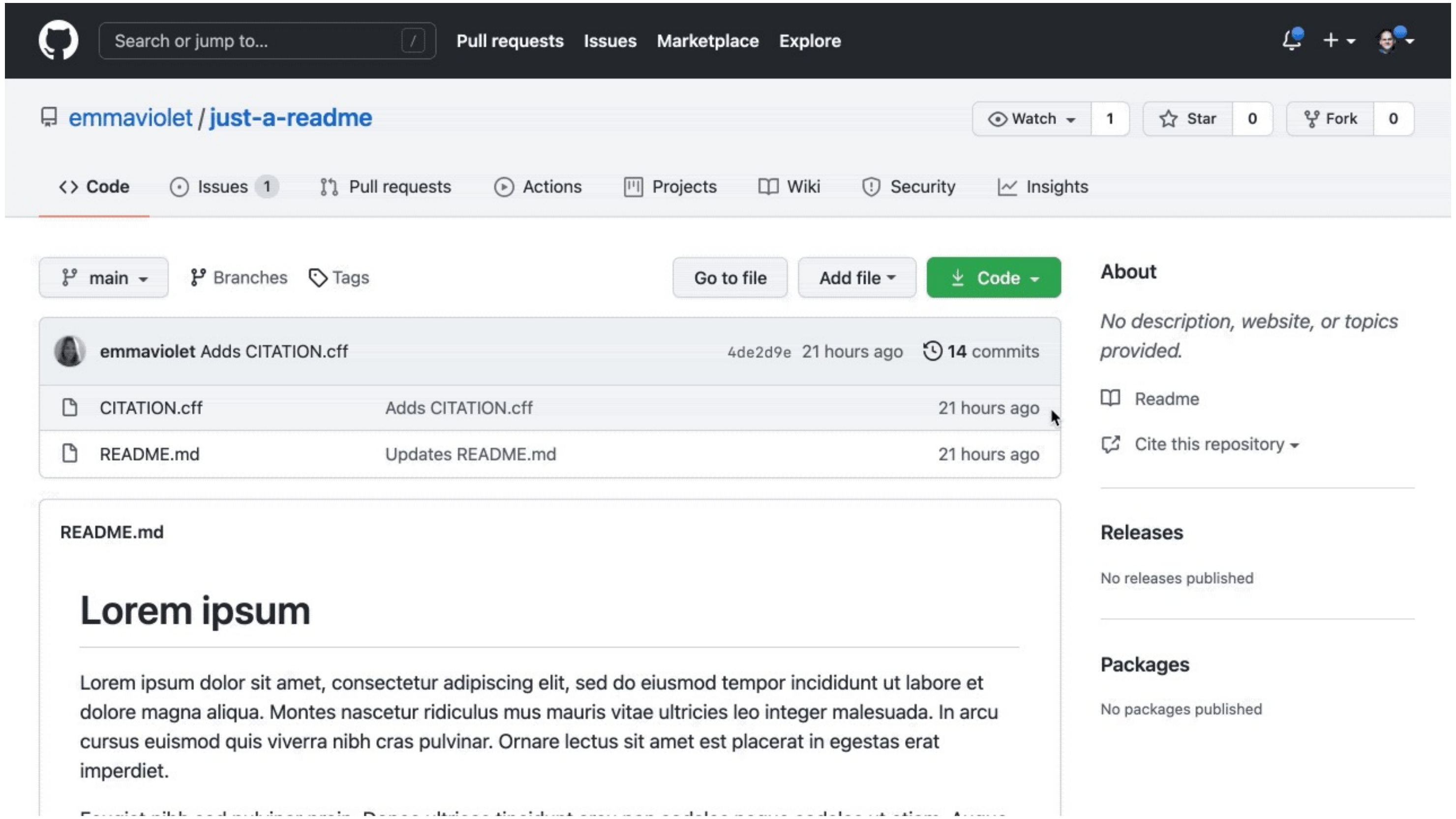
Checklist for making a release of your software citable

- Have I assigned an appropriate license to my software?
- Have I described my software properly, using an appropriate metadata format, and included this metadata file with my software?
  - Have I given my software a clear version number?
  - Have I determined the authors to be credited for this release of my software, and included this in my metadata file?
- Have I procured a persistent identifier for this release of my software?
- Have I added my recommended citation to the documentation for my software?

There are other things that we consider as good practice for research software development which are not directly related to making software citable - these are described in the Additional Guidancee below.

Citation files in repositories

- GitHub support for CITATION.cff files in repositories
  - also supported by Zotero browser plugin and Zenodo
- Note: implicitly supports versioning of citation files with software
  - Usable via Software Heritage



emmaviolet / just-a-readme

emmaviolet Adds CITATION.cff 4de2d9e 21 hours ago 14 commits

CITATION.cff	Adds CITATION.cff	21 hours ago
README.md	Updates README.md	21 hours ago

About

No description, website, or topics provided.

Readme

Cite this repository

Releases

No releases published

Packages

No packages published

README.md

Lorem ipsum

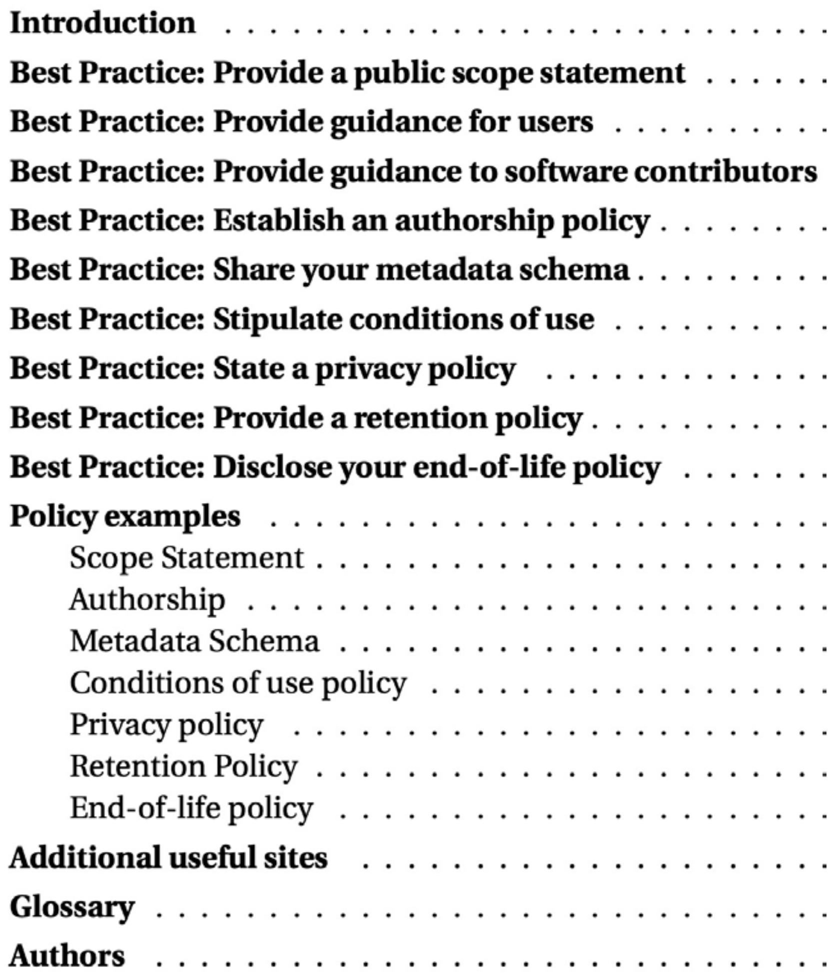
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Montes nascetur ridiculus mus mauris vitae ultricies leo integer malesuada. In arcu cursus euismod quis viverra nibh cras pulvinar. Ornare lectus sit amet est placerat in egestas erat imperdiet.

More on metadata

- CodeMeta project, 2015-2018
  - In parallel with Software Citation Principles & Implementation Working Groups
  - Some common membership
  - Aiming to understand metadata for software, not just for use in citation
  - Built a crosswalk of existing metadata standards for software
  - Then developed a CodeMeta standard to describe software based on these crosswalks
- CodeMeta Task Force (in Software Citation Implementation Working Group)
  - Updating the CodeMeta standard
  - Describing everything in CodeMeta using schema.org properties
  - Moving CodeMeta into a community group, with governance

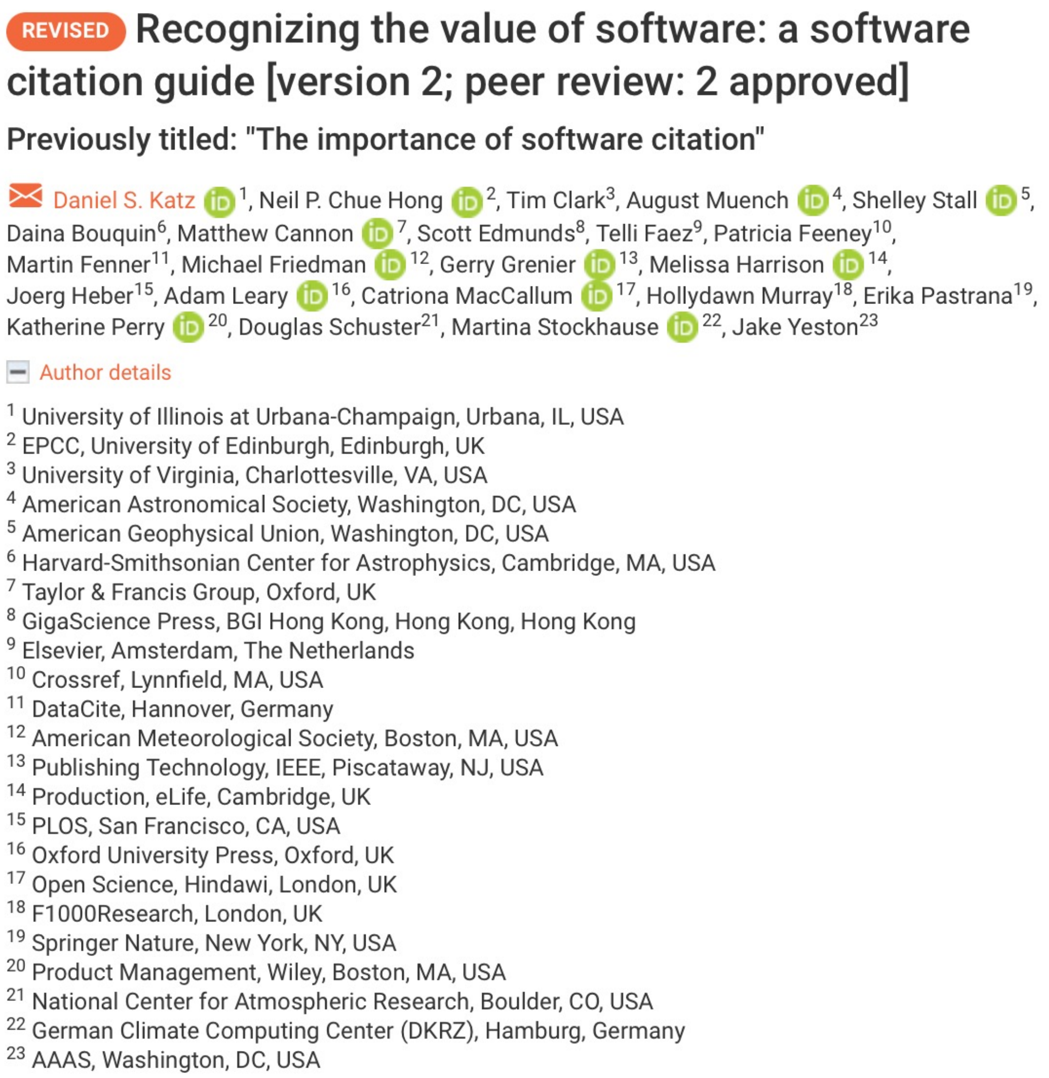
Registry and repository best practices

- Software Registries Task Force developed best practices
  - Task Force on Best Practices for Software Registries, "Nine Best Practices for Research Software Registries and Repositories: A Concise Guide," 2020. arXiv 2012.13117
- Community continuing in SciCodes: Consortium of scientific software registries and repositories
  - https://scicodes.net/



Journals task force

- Working with publishers to provide generic guidelines for journals and conferences to provide to authors
  - They then provide specific guidelines, with community-accepted language and examples
  - D. S. Katz, et al., “Recognizing the value of software: a software citation guide [version 2; peer review: 2 approved],” F1000Research 9:1257, 2021. 10.12688/f1000research.26932.2
  - Tracked by CHORUS in Software Citation Policy Index
- Also working on publication processing
  - How citation information moves from author provides to internal publisher/contractor systems and then to indices
  - S. Stall, et al., “Journal Production Guidance for Data and Software Citations”, in draft



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]

Author details

[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

Institutions

- Planning an Institutions task force
  - Institutions: places where people work
    - Universities, laboratories, industry, government, etc.
  - Want to affect policies and practices
    - How do they encourage software citation
    - How do they use software citation information in hiring & promotion
  - Collect and share examples
  - Help form communities

https://doi.org/10.5281/zenodo.5748175

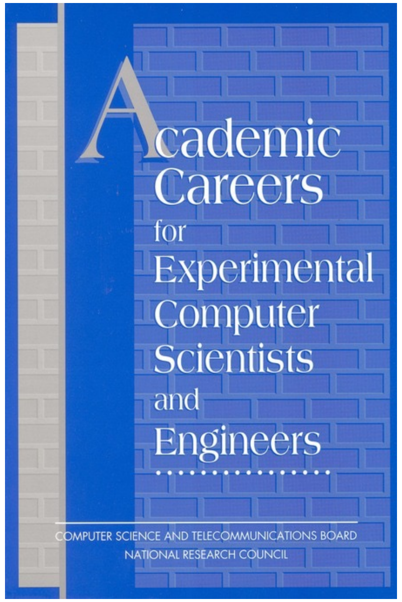
Promotion and evaluation

- Guidelines for promotion and evaluation important
  - Say what’s valued; shape activities people undertake
  - Promotion guidelines written by senior people, how can they be changed?
- We can influence these processes when we participate in these evaluations
- We can provide templates and guidelines for recognizing software contributions and encourage respected organizations to adopt them
  - Multiple groups working in this space

https://doi.org/10.5281/zenodo.5748175

Promotion and evaluation are not fixed

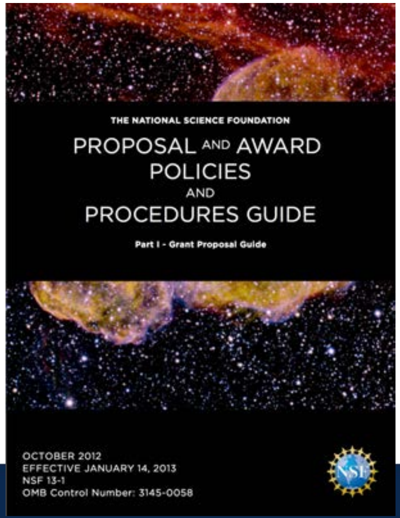
- National Academies (1994): “Academic Careers for Experimental Computer Scientists and Engineers”
  - Experimental artifacts are important in CS, should be part of evaluation
  - Intended to provide a reference point for change
  - Has been quoted in many tenure recommendation letters



Academic Careers for Experimental Computer Scientists and Engineers

COMPUTER SCIENCE AND TELECOMMUNICATIONS BOARD
NATIONAL RESEARCH COUNCIL

https://www.nap.edu/read/2236/



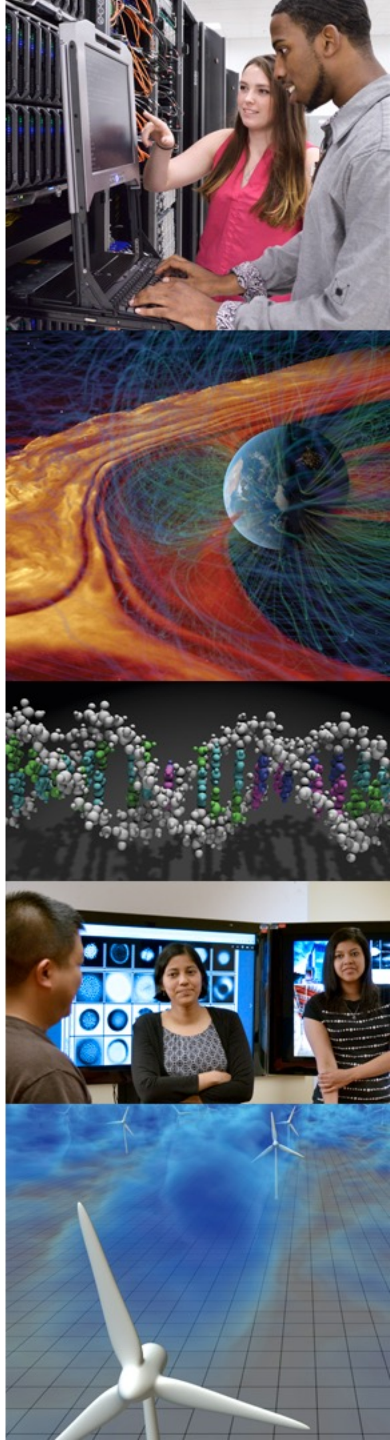
THE NATIONAL SCIENCE FOUNDATION
PROPOSAL AND AWARD POLICIES AND PROCEDURES GUIDE
Part I - Grant Proposal Guide

OCTOBER 2012
EFFECTIVE JANUARY 14, 2013
NSF 13-1
OMB Control Number: 3145-0058

- NSF 2013 biosketch change: products, not publications
  - Acknowledges software contributions as a primary research product
  - Intended to signal to universities that they should do the same

https://www.nsf.gov/pubs/policydocs/pappguide/nsf13001/gpg_index.jsp

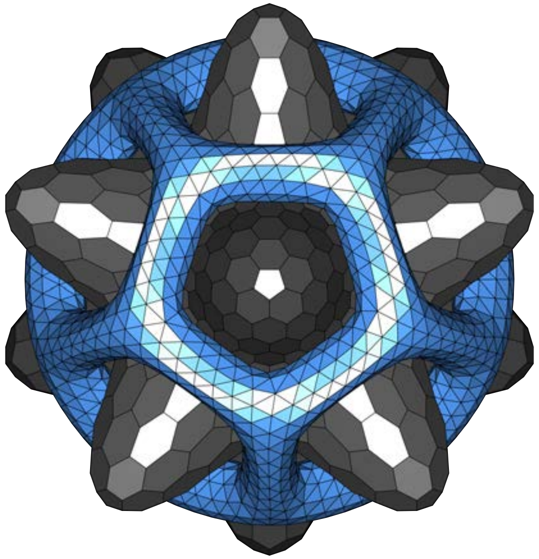
Journal of Open Source Software (JOSS)



https://doi.org/10.5281/zenodo.5748175

Journal of Open Source Software (JOSS)

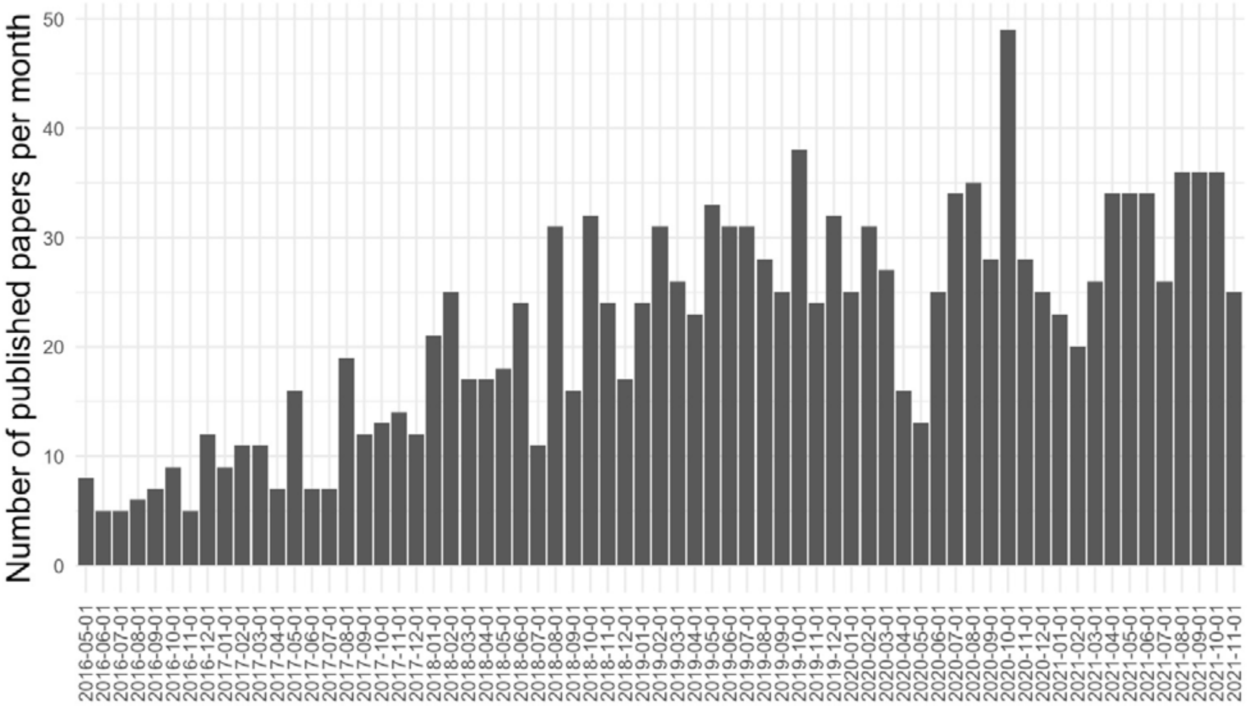
- A developer friendly journal for research software packages
  - “If you've already licensed your code and have good documentation then we expect that it should take less than an hour to prepare and submit your paper”
- Everything is open:
  - Submitted/published paper: https://joss.theoj.org
  - Code itself: where is up to the author(s)
  - Reviews & process: https://github.com/openjournals/joss-reviews
    - Adapted from rOpenSci
    - Expedited process for software already reviewed by rOpenSci & pyOpenSci
  - Code for the journal itself: https://github.com/openjournals/joss
    - Reused for Journal of Open Source Education (JOSE) and Proceedings of the JuliaCon Conferences



https://doi.org/10.5281/zenodo.5748175

Journal of Open Source Software (JOSS)

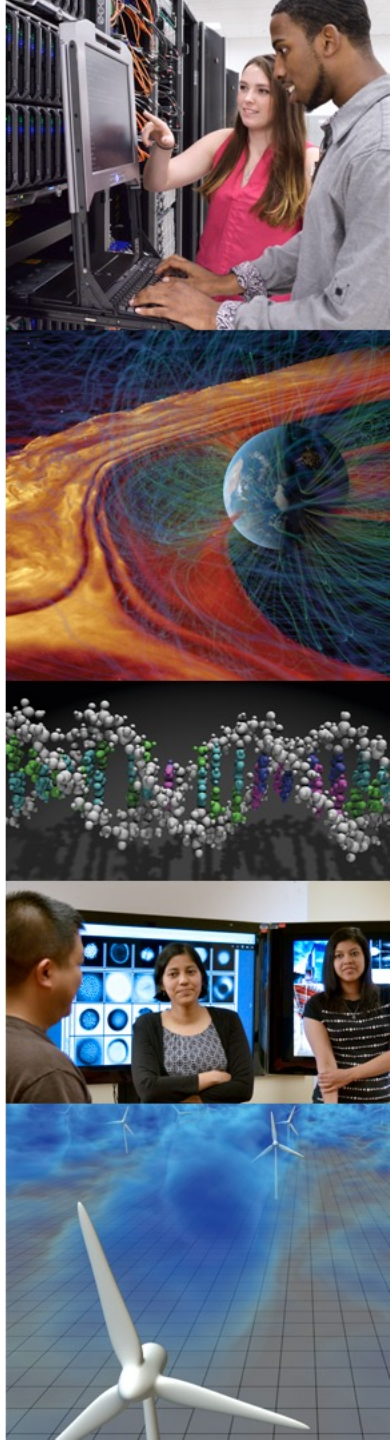
- JOSS papers archived, have DOIs, increasingly indexed
- First paper submitted 4 May 2016
- 31 May 2017: 111 accepted papers, 56 under review and pre-review
- 29 Nov 2021: 1465 accepted papers, 194 under review and pre-review
- Current publication rate: ~1 paper/day
- Editors:
  - 1 editor-in-chief and 11 editors at launch;
  - 1 EiC, 5 associate EiCs, 65 topic editors, 22 emeritus editors today



2021-11-29

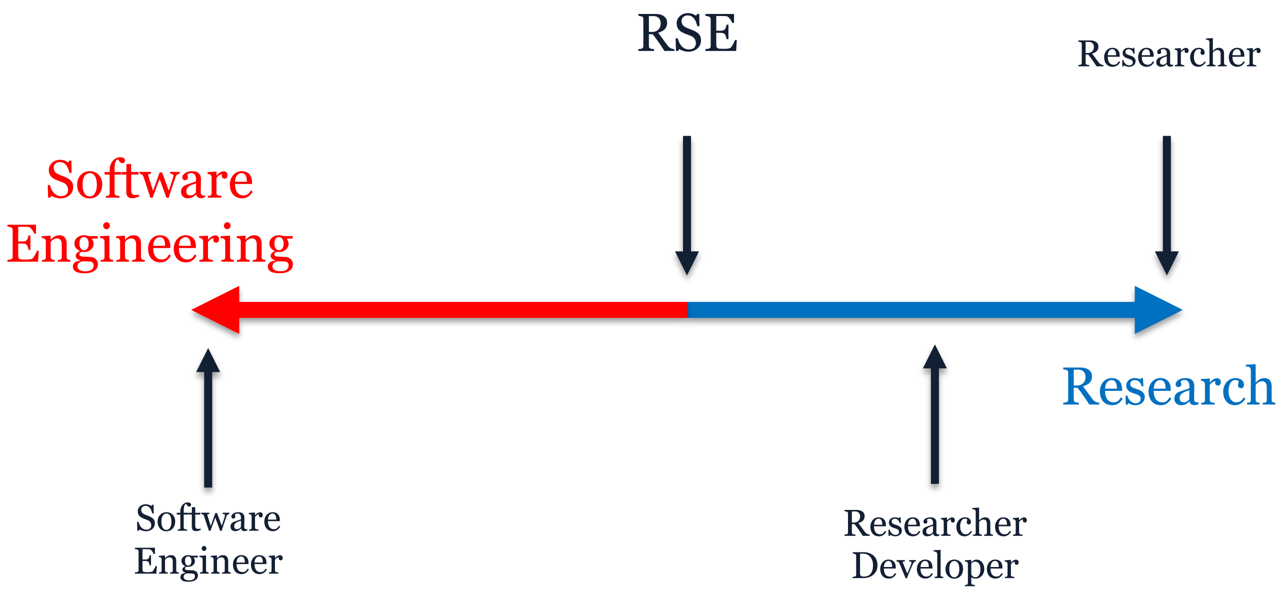
https://doi.org/10.5281/zenodo.5748175

Research Software Engineering (RSEng)



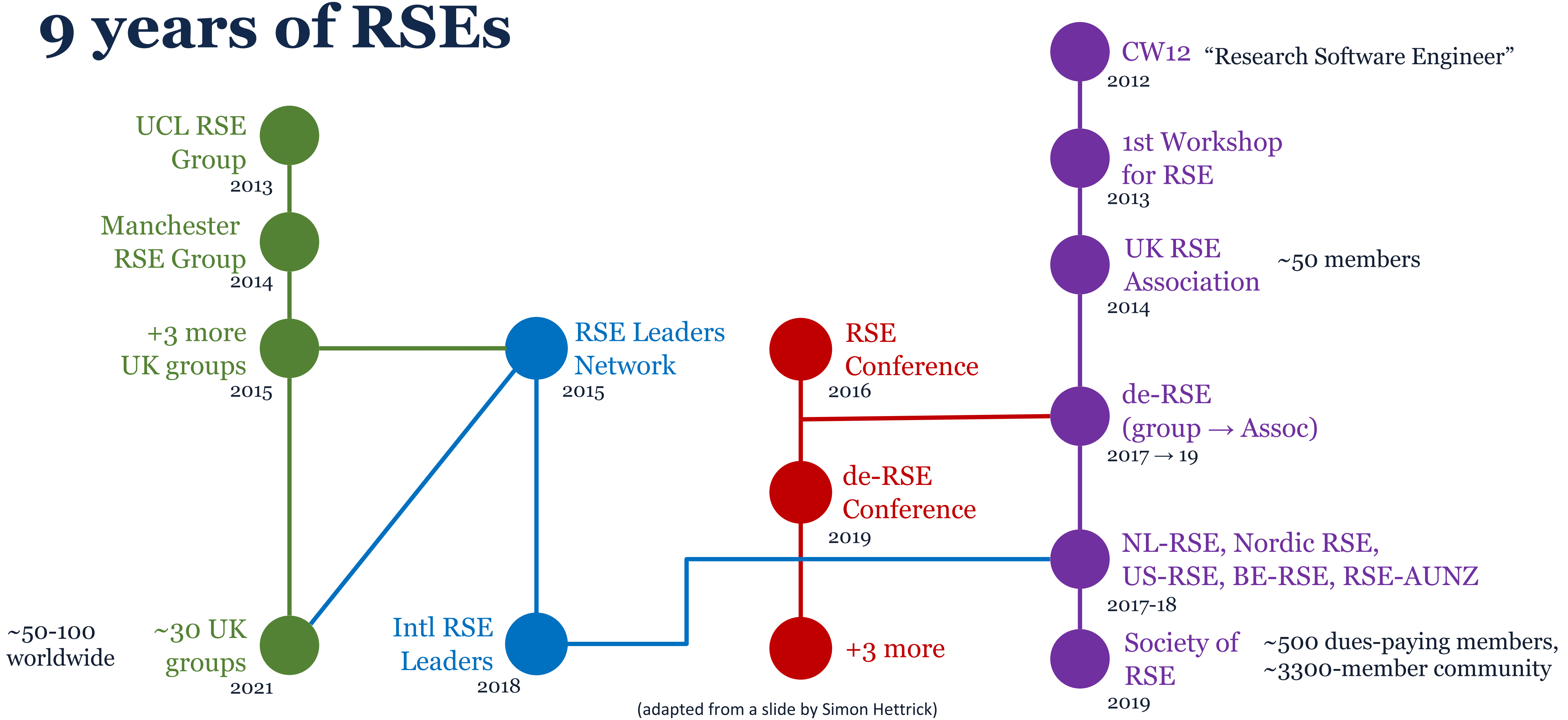
https://doi.org/10.5281/zenodo.5748175

RSEng



https://doi.org/10.5281/zenodo.5748175

9 years of RSEs



(adapted from a slide by Simon Hettrick)

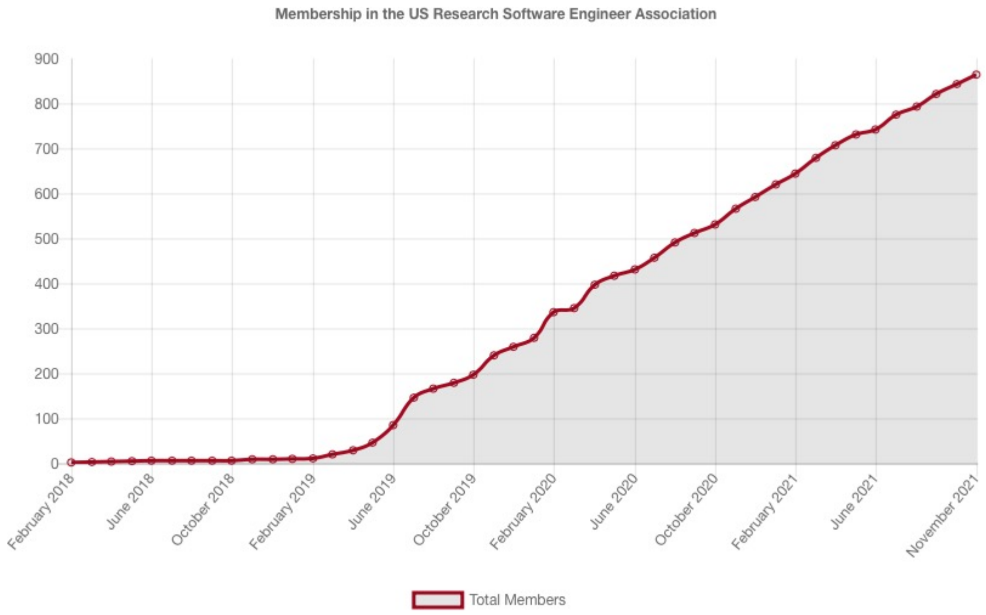
https://doi.org/10.5281/zenodo.5748175

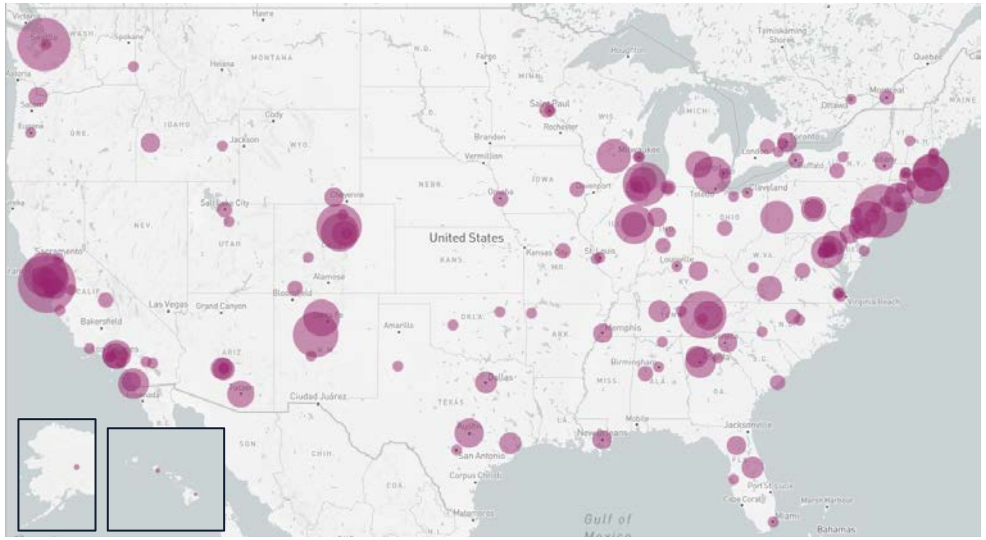
US-RSE Association

- A community-driven organization
- Members:
  - Writing and contributing research software at
    - Universities, laboratories, knowledge institutes, companies, & more
  - Interested in research software engineer careers
    - Students, researchers, software engineers
  - Identifying as RSE "allies"
    - Manage, sponsor, support

History

- Winter 2017-2018
  - US Survey of RSEs: ~175 responses, 12 responded with interest to build a national community
  - UK Sponsored *1st International RSE Leaders Meeting,* 5 US delegates
- 2018: Created Slack workspace, website, initial mission, steering committee
- 2019: Website redesign, logo, twitter, newsletters, code of conduct, events at PEARC19, SC19, Sloan Foundation funding for in-person workshop, RSE stories podcast
- 2020-21: More events, formal governance, elected steering committee, community calls, job board, non-profit financial sponsor, website & logo redesign





US-RSE Mission

- **Community**: Create a professional community to share knowledge, connections, and resources
- **Advocacy**: Promote RSEs impact on research, highlighting the critical and valuable role RSEs serve
- **Resources**: Provide access to information and material to support individuals and RSE groups
- **Diversity, Equity, and Inclusion**: Actively promote, encourage, and improve diversity within RSE community

- Activities and events that support the mission are underway
- https://us-rse.org/join

- Research Software Engineers International (http://researchsoftware.org) includes pointers to RSE associations in Belgium, Germany, Netherlands, Nordic contries, Australia and New Zealand, UK, US, with work underway to create associations in Asia, Colombia, Brazil, …

https://doi.org/10.5281/zenodo.5748175

Conclusions

- Overall, increasing recognition of software
  - And that it's not data, and that policies for data do not directly apply to software
- Moving forward in FAIR4RS, software citation, US-RSE
- Progress in any of these areas helps the others as well
- Lots of community building and implementation to do
- Remember: culture of science is us – we can change it
- What you can do
  - Work to make your own software FAIR – follow the principles
  - Make your software citable, and cite the software from others
  - When you review, insist that software is cited
  - Consider how your institution supports software development and maintenance
  - Point people who do this at the RSE movement
  - Work to make sure this is included in hiring and promotion
  - Overall, raise awareness of software as a key element of research

https://doi.org/10.5281/zenodo.5748175

Credits

- Thanks to Arfon Smith and Kyle Niemeyer for co-leadership in FORCE11 Software Citation WG
- And Neil Chue Hong & Martin Fenner for co-leadership in FORCE11 Software Citation Implementation WG
- And Michelle Barker, Neil Chue Hong, Leyla Garcia, Morane Gruenpeter, Jennifer Harrow, Carlos Martinez, Paula A. Martinez, Fotis Psomopoulos for co-leadership in FAIR4RS work
- And Jeff Carver, Ian Cosden, Julia Damerow, Charles Ferenbaugh, Sandra Gesing, Chris Hill, Christina Maimone, Lance Parsons, Jordan Perr-Sauer, for co-leadership in US-RSE
- And colleagues Gabrielle Allen, C. Titus Brown, Kyle Chard, Ian Foster, Melissa Haendel, Christie Koehler, Bill Miller, Rajiv Ramnath, Matt Turk
- And to the BSSw project (http://bssw.io) for a fellowship to pursue some parts of the citation work
- More of my thinking
  - Blog: http://danielskatzblog.wordpress.com
  - Tweets: @danielskatz

https://doi.org/10.5281/zenodo.5748175