



Insights from planning a research software sustainability institute

Karthik Ram
 • @berkeley.edu

Karthik Ram

—————•—————@berkeley.edu



**Without software, modern
research would be
impossible**

Without research
software, modern research
would be impossible

Research software (as opposed to simply software) is software developed within academia and used for the purpose of research: **generate, process and analyze results.**

Research Software



Use

90%

95%

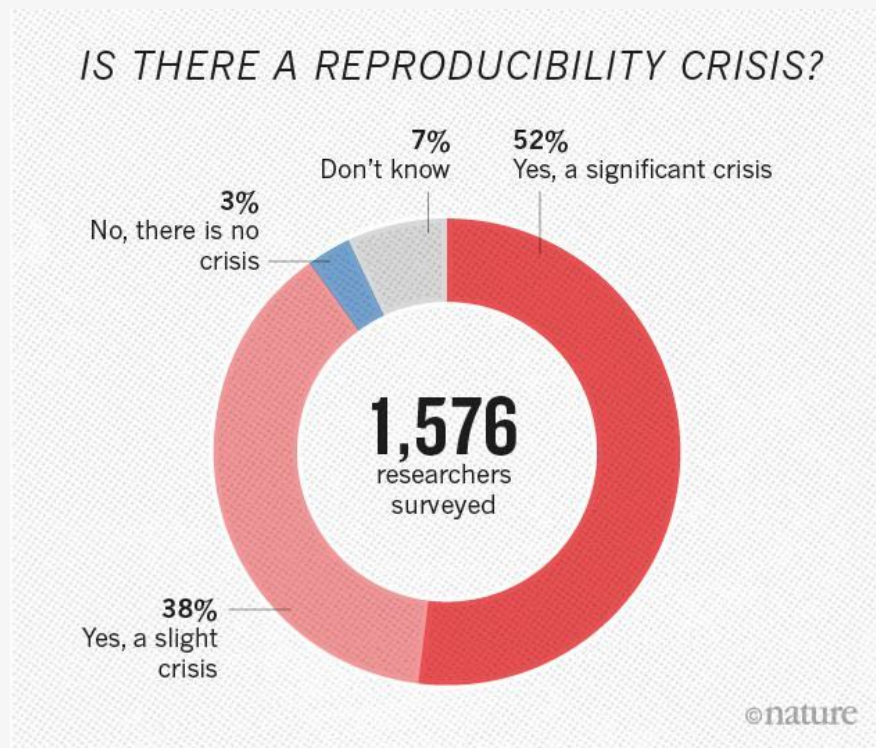
Can't
continue
without

70%

63%

Software is critical for
research but **we don't value
it as scholarship**

The reproducibility crisis is widespread



Baker, 2015. Baker & Dolgin 2017, Aschwanden, C. 2016, Casadevall & Fang 2010

**Lack of reproducibility is
quite widespread even in
applied computational
research**

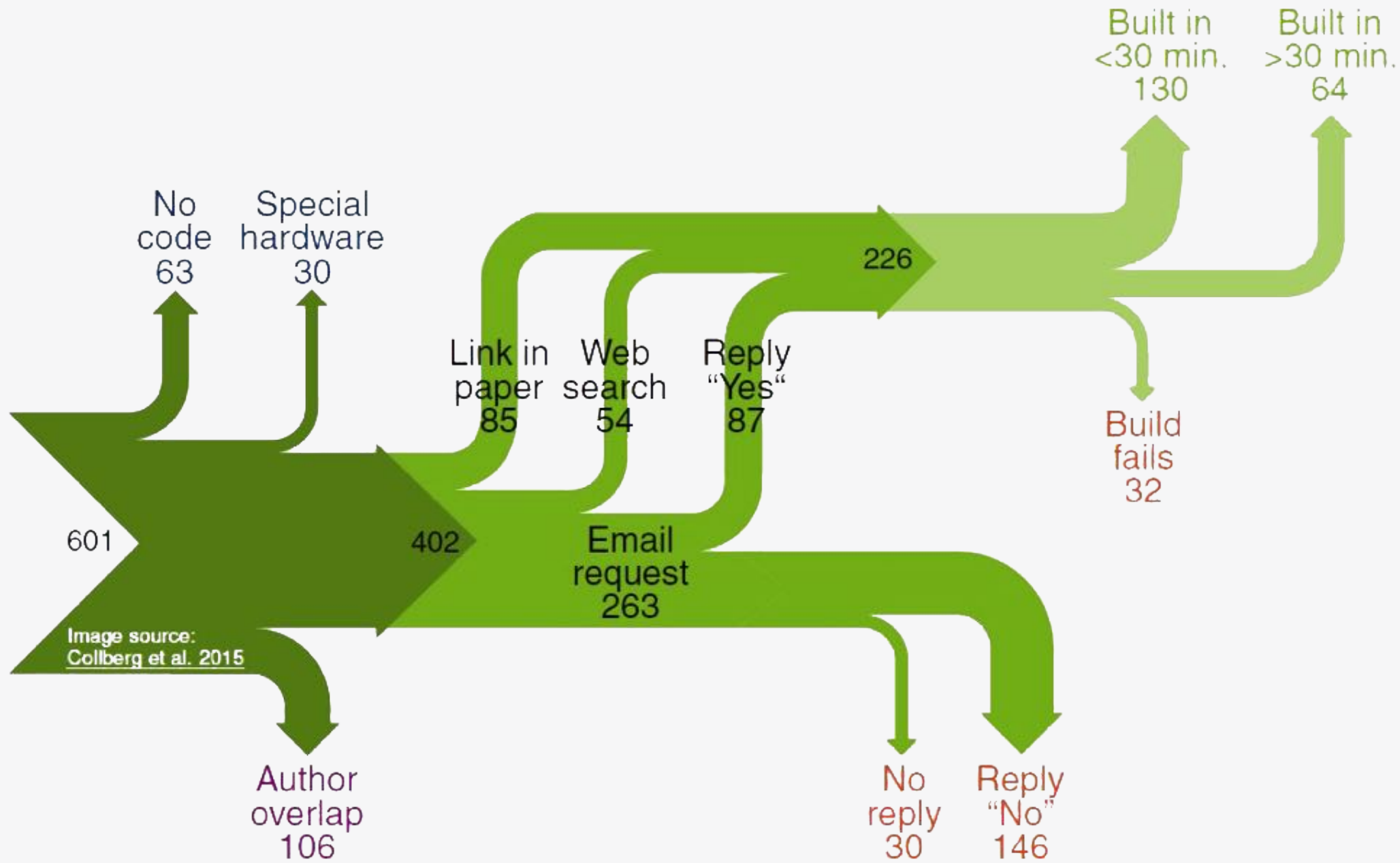


Image source:
Collberg et al. 2015

Prof. Daniel
Bolnick



“

*Recently, Dr. Tony Wilson from CUNY Brooklyn tried to recreate my analysis, so that he could figure out how it worked and apply it to his own data ... **he couldn't quite recreate some of my core results.***

“

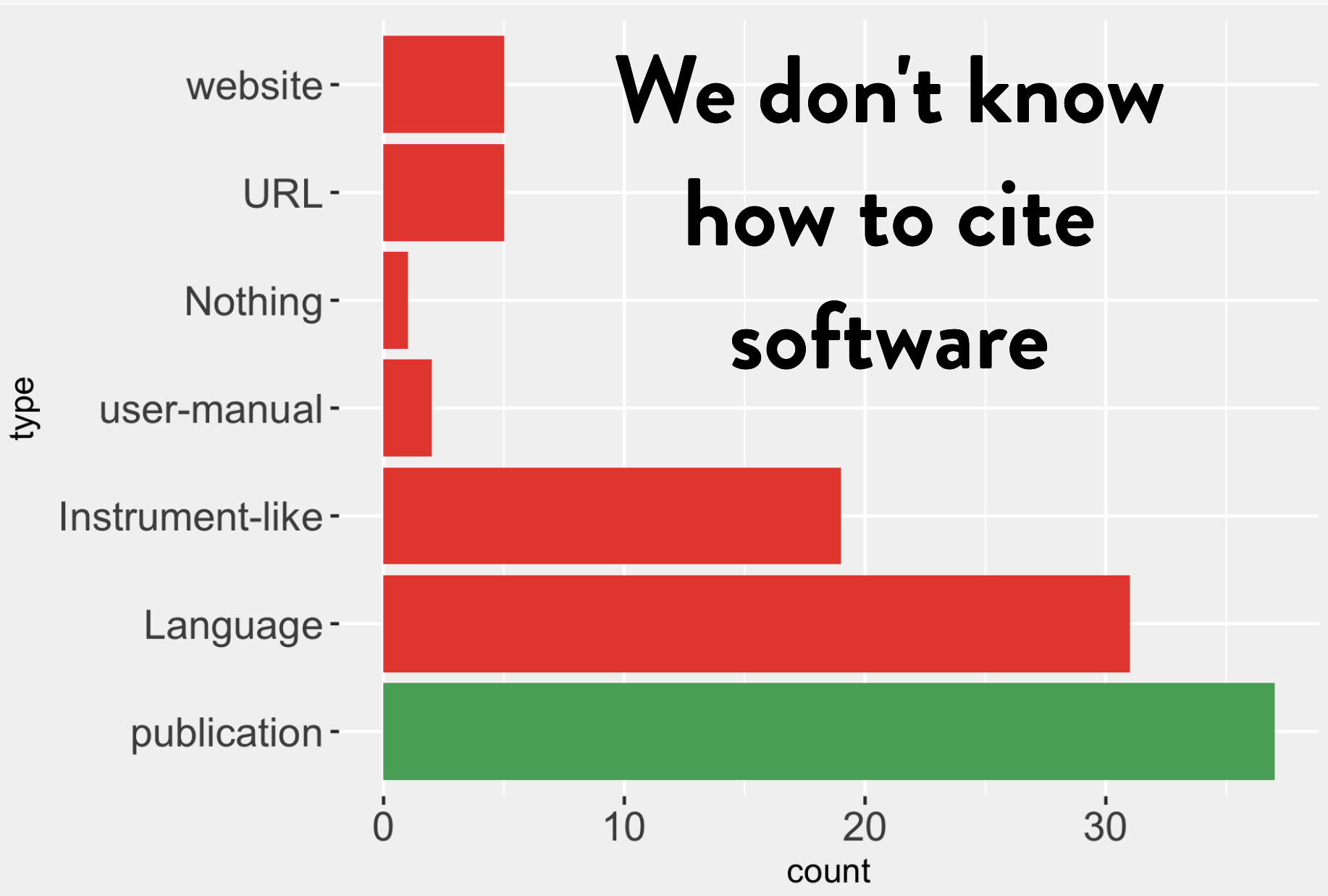
*So: how many results, negative or positive, that enter the published literature are tainted by a coding mistake as mine was. We just don't know. **Which raises an important question: why don't we review code (or other custom software) as part of the peer-review process?***

Training in
computational skills is
one of the **largest**
unmet needs

Barone et al, 2017

If we want the research
community to use software, we
need to **advocate for sustainable
software** and **provide access to
expertise**

We don't know how to cite software



Formal citations: 31% - 43%

**Informal mentions are the norm, even
in high impact journals**

Software is frequently inaccessible (15 - 29%)

Lack of visibility means that
incentives to produce high-quality,
widely shared, and collaboratively
developed software **are lacking**

Software sustainability describes the practices, both technical and non-technical that allow software to continue to operate as expected in the future

**Software sustainability is
strongly linked to
reproducibility and
transparency**



NSF funding 1996-2016

> **18k** awards totaling **\$9.6 billion**
related to research software.

Science & Eng →

URSSI

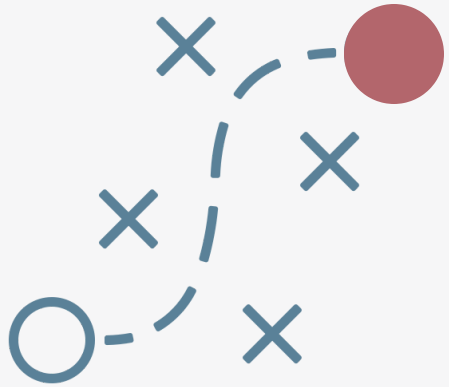
SGCI (Science Gateways)

MoISSI

HEP-S2I2

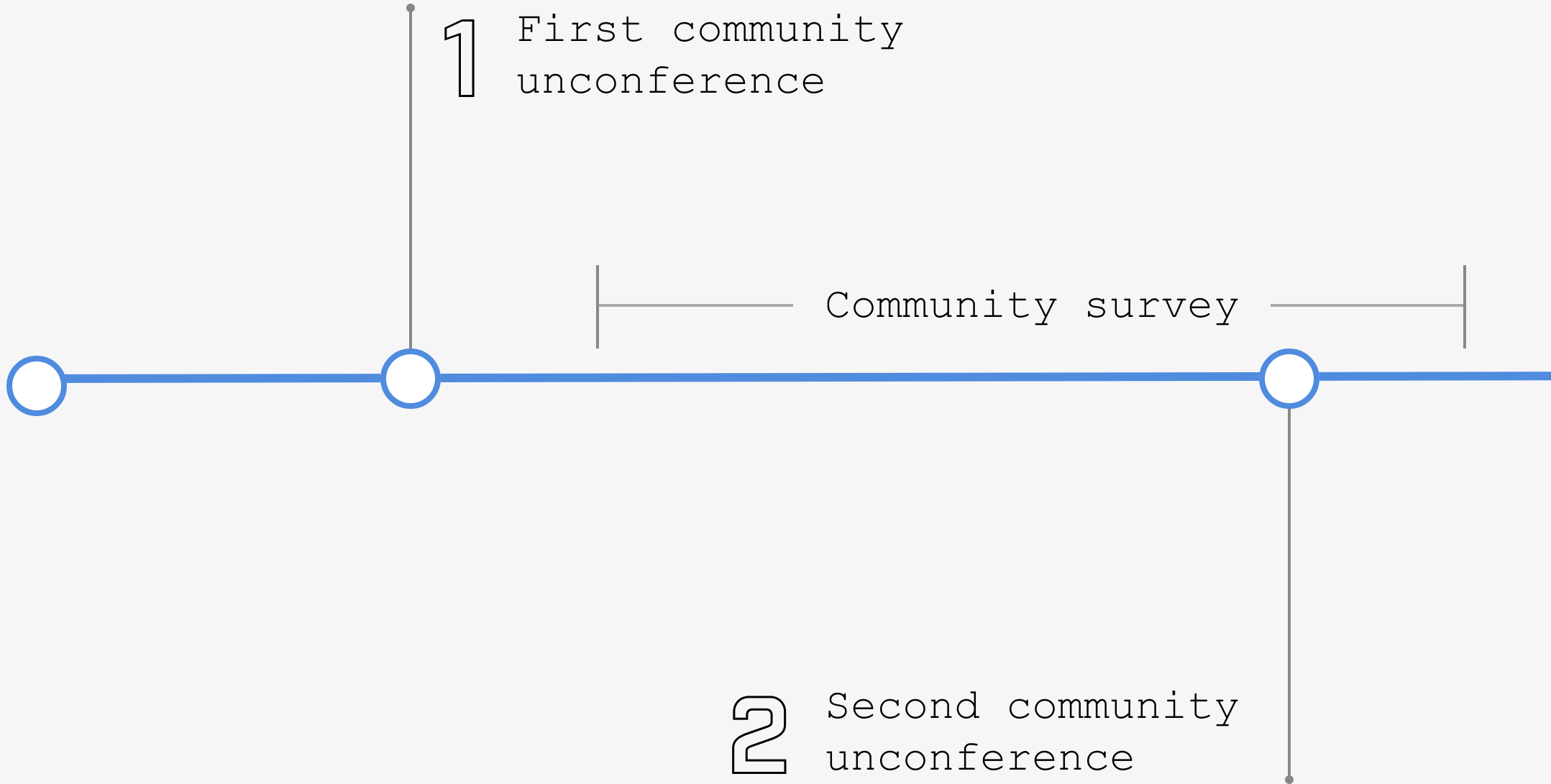
Tech & SW →

URSSI mission



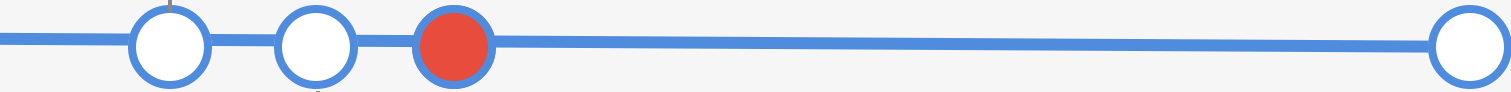
To improve the **quality**,
usefulness, and **sustainability** of
research software by **improving**
practices, and **increasing diversity**
of practitioners

**Funded
Jan 2018**



3

Software metrics,
citation



4

Research software
incubators

**Institute
plans by
December
2019**

Community unconferences

April, Oct 2018

Software

Critical, Discoverability,
Dev practices, Metrics,
Citation

People

Career paths, promotion,
tenure, diversity

Training

SWE carpentry, broad training
and mentoring initiatives,
URSSI Summer School

Org practices

Governance, collaboration



1

Training

Curriculum development,
partnering with existing initiatives,
assessment, outreach

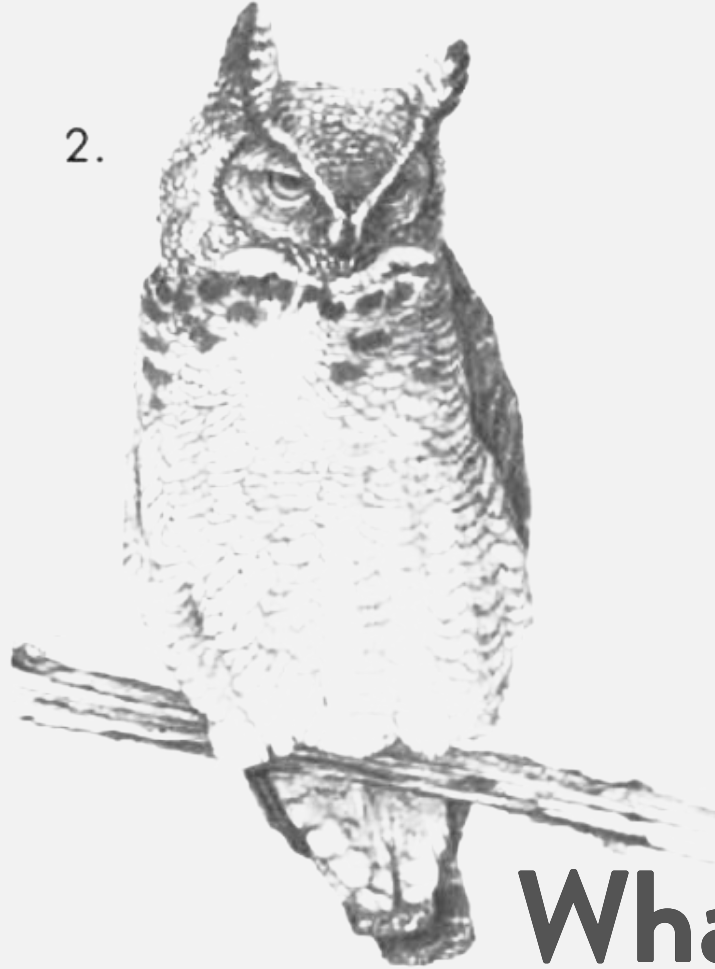
**What we train
people for**



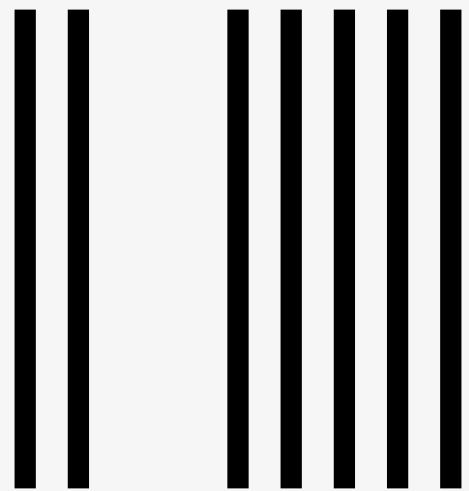
1.



2.



**What we expect
of them**



There are **large gaps in software engineering training** for researchers

Could there be a **RSE Carpentry?**

Training landscape



2

Supporting software

development

Incubate projects, provide
consulting support, offer startup
grants



A Ycombinator for science?

4 Software Incubators



Feb 2019





College Park, MD

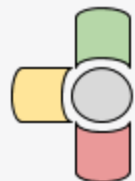
Schematic stages of open community for research software





Stage 0. Some code and a user of it. No sustained team.



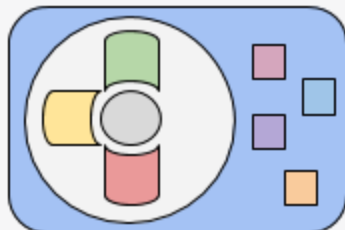
Stage 1. Software development team, internal use.  



Stage 2. Multiple software teams (different institutions) on same code (team is *community*), for internal use.  



Stage 3. Self-governing developer community deliberately supporting broad user community.



Stage 4. Self-sustaining organization dedicated to supporting user and dev community (e.g. through commercial support, events, software foundation, etc.).

3

Improving credit mechanisms

Credit mechanisms, Improving software citations, advocating for RSE roles





Credit when software is informally cited

Depsy text-mines papers to find fulltext *mentions* of software they use, revealing impacts invisible to citation indexes like Google Scholar.



Credit when software is reused

Citation is just part of the story—Depsy analyzes code from over half a million GitHub repositories to find how packages are reused by other software projects.



Credit for all software's authors

Depsy assigns fractional credit to contributors based on designated authorship, number of commits, and repo ownership—supporting a fairer, more software-native reward system.



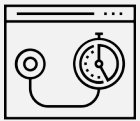
Writing software that is easier to sustain



Complete documentation



OSI compatible license



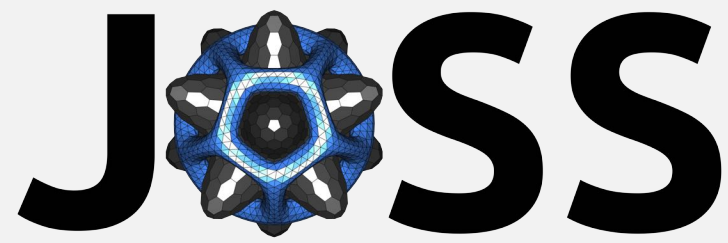
High test coverage



Readable code



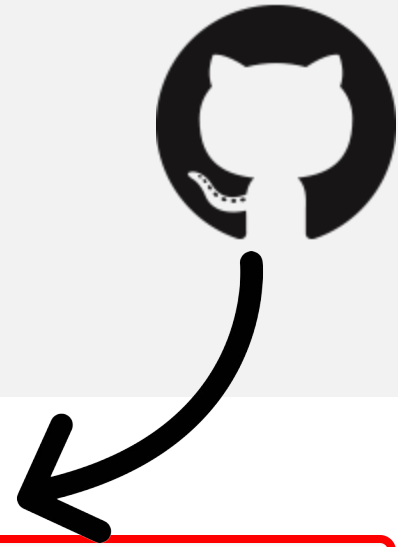
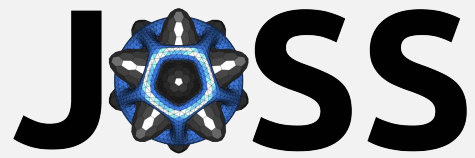
Usability



Journal of Open
Source Software

`joss.theoj.org`

JOSS is a **free, developer friendly,**
open access journal for research
software packages



Submit software for review

Title

Repository address

Software version

Suggested editor. [View editors here »](#)

Description

Please give short (1-2 line) description of your software.

I certify that I am submitting software for which I am a primary author

I confirm that I read and will adhere to the JOSS [code of conduct](#)

What does JOSS look for?



General Checks



Documentation



Functionality



**A really simple
manuscript**



labarba commented 14 days ago

Member



@whedon accept



whedon commented 14 days ago

Author

Member



```
Attempting dry run of processing paper acceptance...
```



whedon commented 14 days ago

Author

Member



Check final proof  [openjournals/joss-papers#489](#)

If the paper PDF and Crossref deposit XML look good in [openjournals/joss-papers#489](#), then you can now move forward with accepting the submission by compiling again with the flag `deposit=true` e.g.

```
@whedon accept deposit=true
```



whedon commented 14 days ago




Author

Member

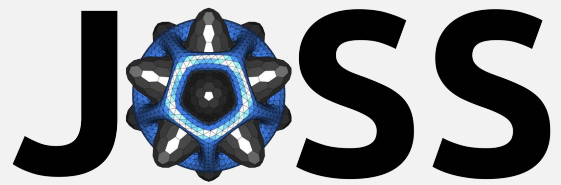


 **THIS IS NOT A DRILL, YOU HAVE JUST ACCEPTED A PAPER INTO JOSS!** 

Here's what you must now do:

0. Check final PDF and Crossref metadata that was deposited  openjournals/joss-papers#490
1. Wait a couple of minutes to verify that the paper DOI resolves <https://doi.org/10.21105/joss.01167>
2. If everything looks good, then close this review issue.
3. Party like you just published a paper!      

Any issues? notify your editorial technical team...



500 published papers

\$0 APC

\$3.50 to publish

SOFTWARE PAPER



testing, docs, containers,
contributor guidelines,
license, archive

4

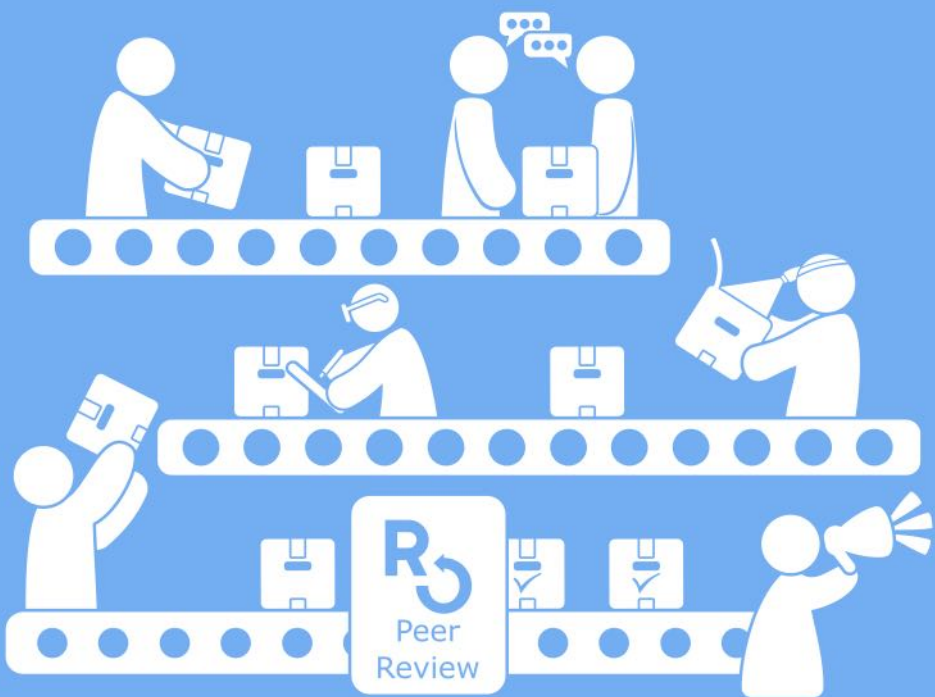
Building community



**Disseminate best practices,
governance etc.**

rOpenSci Packages

*Development, Maintenance,
and Peer Review*



ropensci.org/ software-review

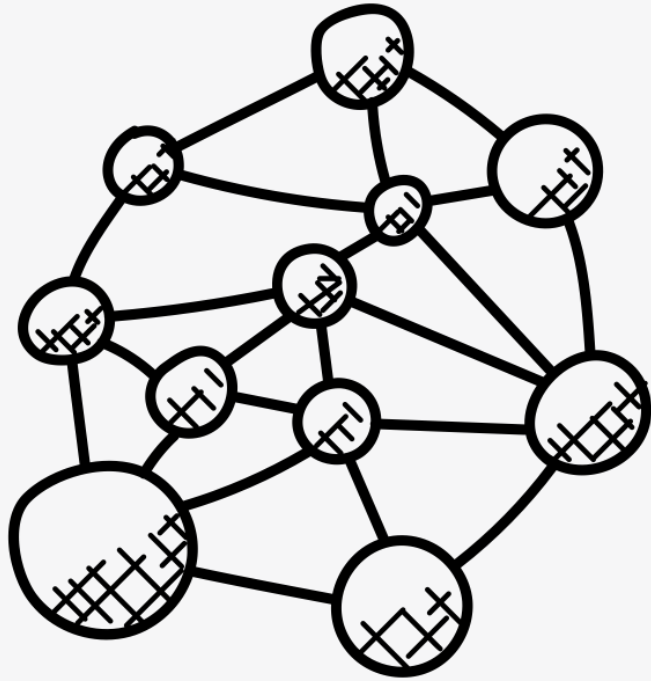


The UK has successfully
campaigned for **RSE**
roles

Could the rest of us do something
analogous?

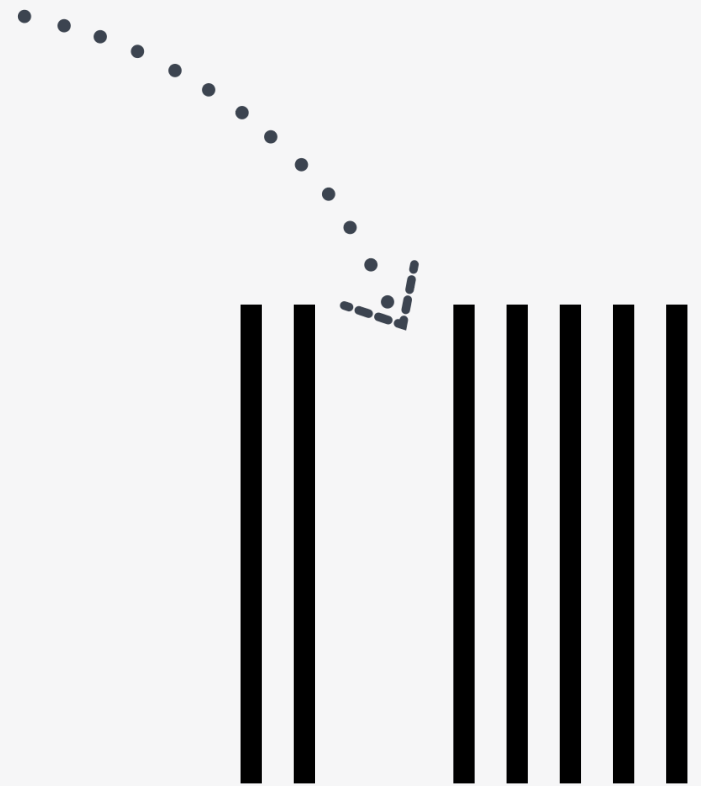
Developing a **community of research software engineers**, and the next generation of research software mentors.



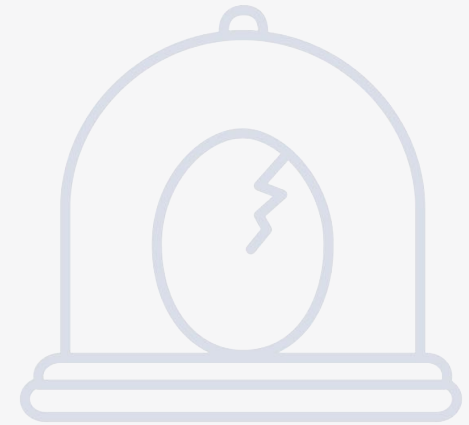


**A set of core
activities for
URSSI**

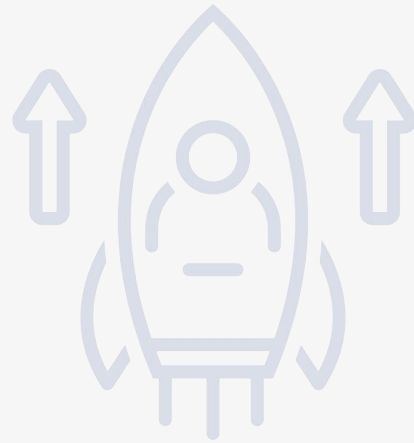
There is a considerable training gap to be filled. We plan to **train new developers** and **improve the diversity of the pipeline**



Incubate software projects and help them **grow**, become **sustainable**, develop a governance model.



Disseminate **expertise in software sustainability** and help researcher get credit for their software work



**Build a community around
research software and
make software a first class
research output**



Collaborate with URSSI

urssi.us

discuss.urssi.us

github.com/si2-urssi

bit.ly/c3dis19