Challenges and Pathways to Sustainability in Scientific Software Ecosystems

James Howison
University of Texas at Austin
School of Information

i320D Open Source Software Development Class

This material is based upon work supported by the US National Science Foundation under Grant Nos. SMA- 1064209 (SciSIP), OCI-0943168 (VOSS) and ACI-1453548 (CAREER).


Sustainability: the condition that results when *the work needed to keep software scientifically useful is undertaken*
https://www.flickr.com/photos/kitch/5179039214
@jameshowison

Four Parts

1. Images of software in science
2. Why is sustainability difficult for software in science?
3. Routes to sustainability via peer production
4. Incentivizing sustainable ecosystems

An Image of scientific software work

How does a
a cubic km of ice
become a
scientific paper?

First find some ice

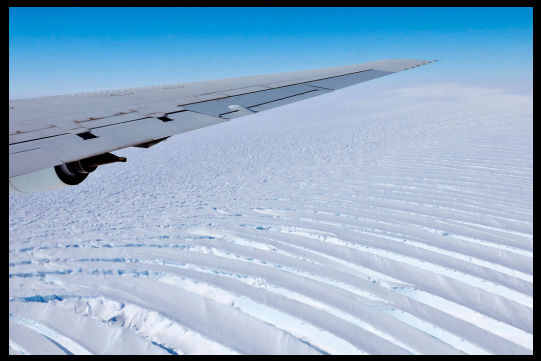


Image Credit: NASA

Build a big drill

Image Credit: IceCube

and some Digital Optical Modules

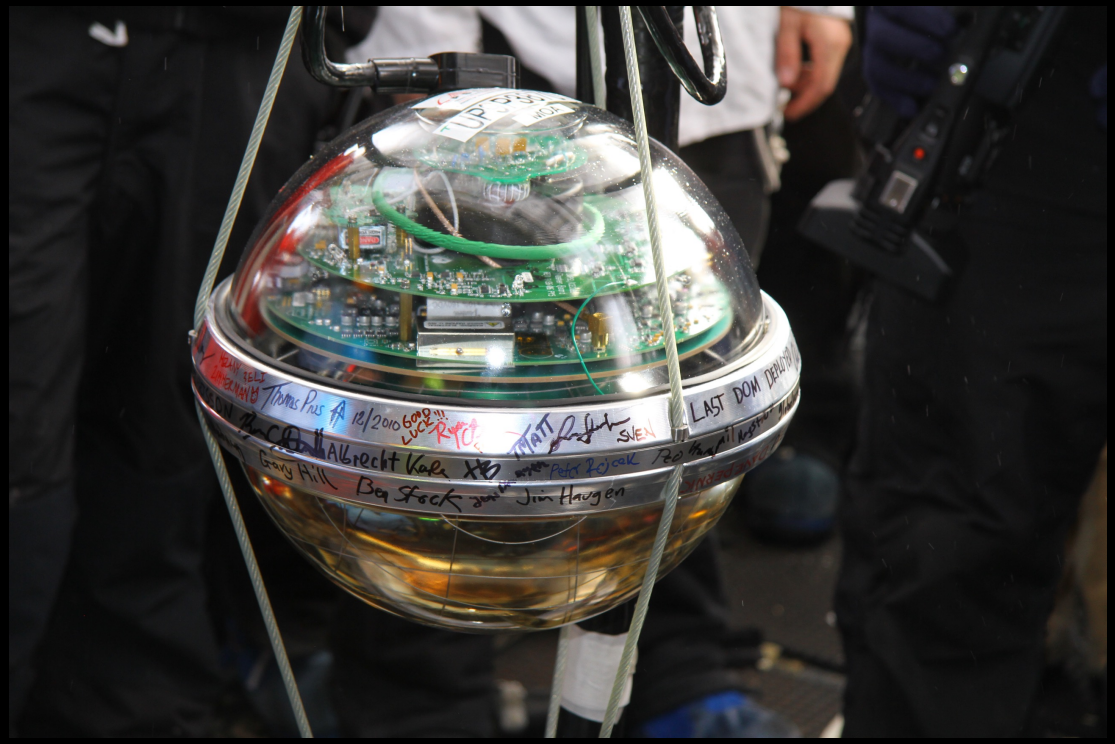


Image Credit: IceCube

Combine

Image Credit: IceCube

Collect and filter data

Image Credit: IceCube

Store and analyze it

Image Credit: http://www.flickr.com/photos/theplanetdotcom

Simulate light in ice

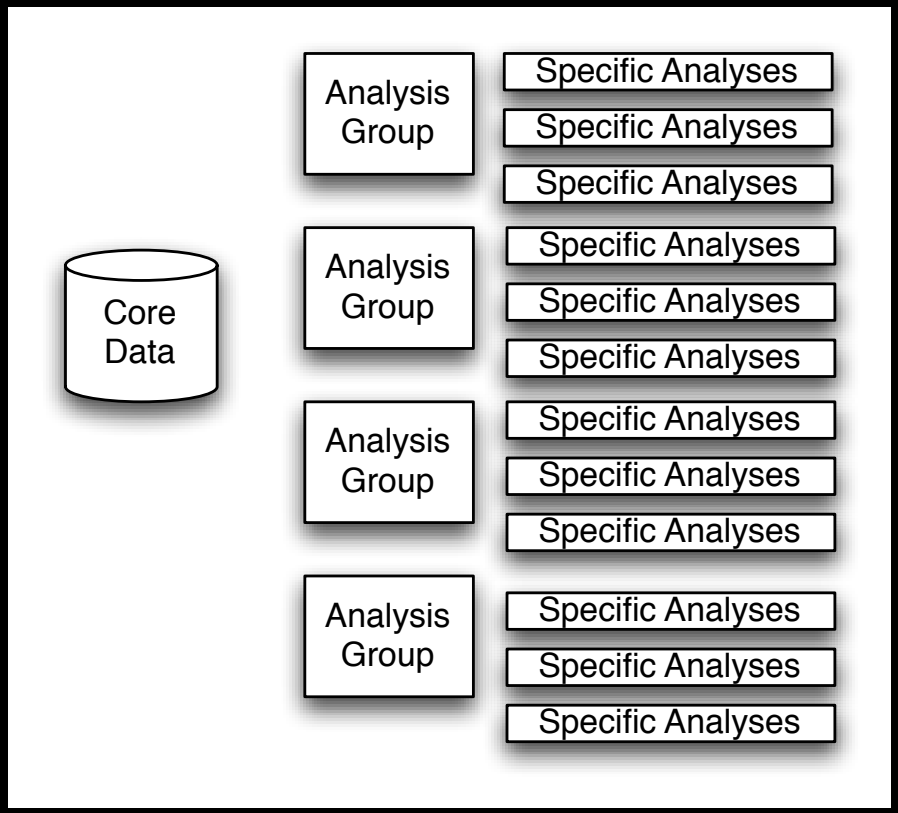
Photo credit: http://www.flickr.com/photos/rainman_yukky/

Simulate Atmosphere

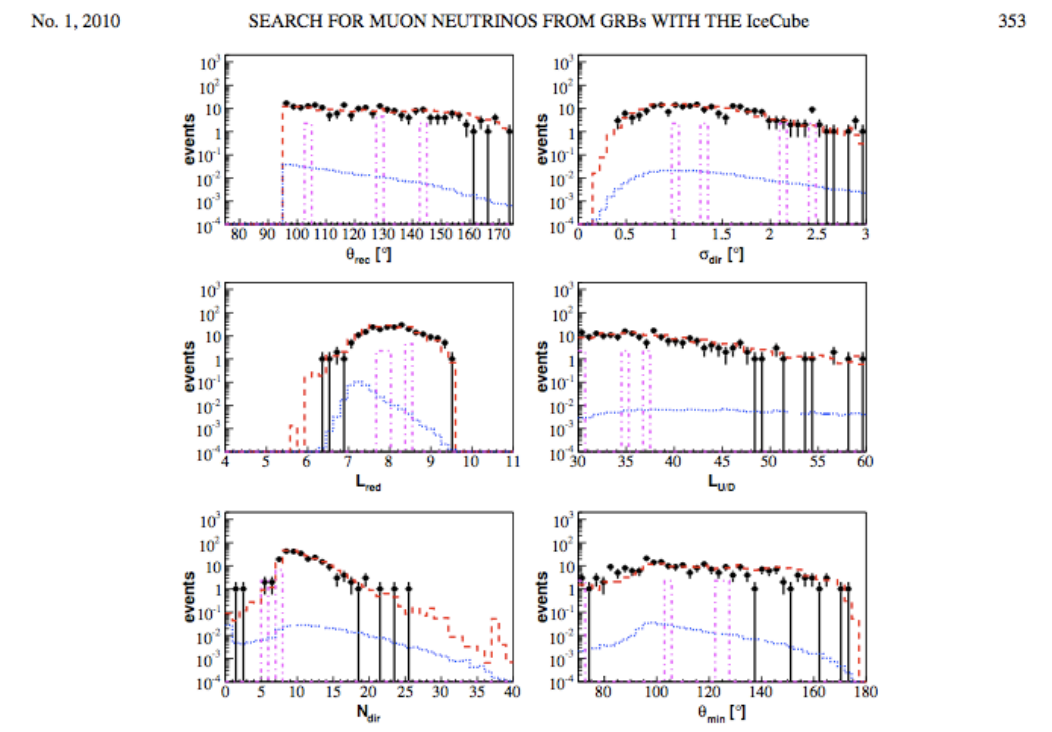
Image Credit: NASA

Model

Analyze


Core Data
Analysis Group
Analysis Group
Analysis Group
Analysis Group
Specific Analyses
Specific Analyses
Specific Analyses
Specific Analyses
Specific Analyses
Specific Analyses
Specific Analyses
Specific Analyses
Specific Analyses
Specific Analyses
Specific Analyses
Specific Analyses

Plots


No. 1, 2010
SEARCH FOR MUON NEUTRINOS FROM GRBs WITH THE IceCube
353
events
θ_{rec} [°]
σ_{dir} [°]
L_{red}
$L_{U/D}$
N_{dir}
θ_{min} [°]

Publish

The Astrophysical Journal, 710:346–359, 2010 February 10 doi:10.1088/0004-637X/710/1/346

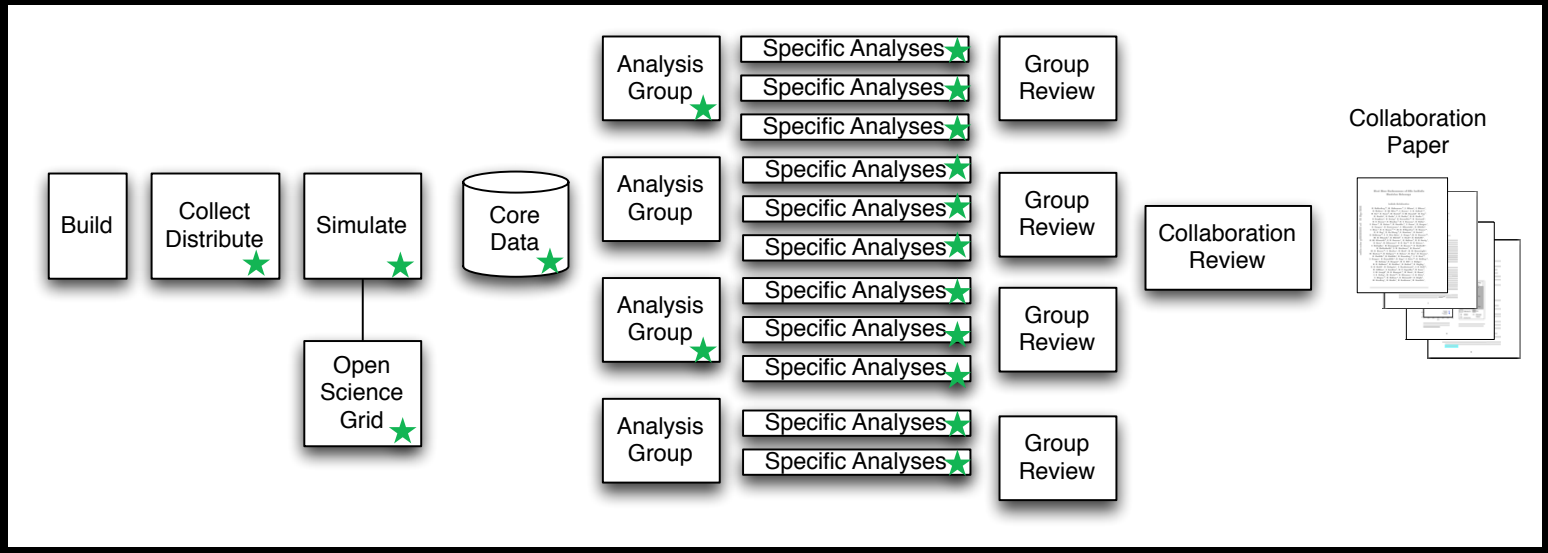
SEARCH FOR MUON NEUTRINOS FROM GAMMA-RAY BURSTS WITH THE IceCube NEUTRINO TELESCOPE

R. Abbasi[1], Y. Abdou[2], T. Abu-Zayyad[3], J. Adams[4], J. A. Aguilar[1], M. Ahlers[5], K. Andeen[1], J. Auffenberg[6], X. Bai[7], M. Baker[1], S. W. Barwick[8], R. Bay[9], J. L. Bazo Alba[10], K. Beattie[11], J. J. Beatty[12,13], S. Bechet[14], J. K. Becker[15], K.-H. Becker[6], M. L. Benabderrahmane[10], J. Berdermann[10], P. Berghaus[1], D. Berley[16], E. Bernardini[10], D. Bertrand[14], D. Z. Besson[17], M. Bissok[18], E. Blaufuss[16], D. J. Boersma[1], C. Bohm[19], J. Bolmont[10], O. Botner[20], L. Bradley[21], J. Braun[1], D. Breder[6], T. Castermans[22], D. Chirkin[1], B. Christy[16], J. Clem[7], S. Cohen[23], D. F. Cowen[21,24], M. V. D'Agostino[9], M. Danninger[19], C. T. Day[11], C. De Clercq[25], L. Demirörs[23], O. Depaepe[25], F. Descamps[2], P. Desiati[1], G. de Vries-Uiterweerd[2], T. DeYoung[21], J. C. Diaz-Velez[1], J. Dreyer[15], J. P. Dumm[1], M. R. Duvoort[20], W. R. Edwards[11], R. Ehrlich[16], J. Eisch[1], R. W. Ellsworth[16], O. Engdegård[26], S. Euler[18], P. A. Evenson[7], O. Fadiran[27], A. R. Fazely[28], T. Feusels[2], K. Filimonov[9], C. Finley[1], M. M. Foerster[21], B. D. Fox[21], A. Franckowiak[26], R. Franke[10], T. K. Gaisser[7], J. Gallagher[29], R. Ganugapati[1], L. Gerhardt[11,9], L. Gladstone[1], A. Goldschmidt[11], J. A. Goodman[16], R. Gozzini[17], D. Grant[21], T. Griesel[17], A. Groß[4,23], S. Grullon[1], R. M. Gunasingha[28], M. Gurtner[6], C. Ha[11], A. Hallgren[20], F. Halzen[1], K. Han[4], K. Hanson[1], Y. Hasegawa[33], J. Heise[30], K. Helbing[6], P. Herquet[22], S. Hickford[4], G. C. Hill[1], K. D. Hoffman[16], K. Hoshina[1], D. Hubert[25], W. Huelsnitz[16], J.-P. Hülß[18], P. O. Hulth[19], K. Hultqvist[19], S. Hussain[7], R. L. Imlay[28], M. Inaba[33], A. Ishihara[33], J. Jacobsen[1], G. S. Japaridze[27], H. Johansson[19], J. M. Joseph[11], K.-H. Kampert[6], A. Kappes[1,34], T. Karg[6], A. Karle[1], J. L. Kelley[1], P. Kenny[17], J. Kiryluk[11,9], F. Kislat[10], S. R. Klein[11,9], S. Knops[18], G. Kohnen[22], H. Kolanoski[23], L. Köpke[17], M. Kowalski[26], T. Kowarik[17], M. Krasberg[1], K. Kuehn[12], T. Kuwabara[7], M. Labare[14], S. Lafebre[21], K. Laihem[18], H. Landsman[1], R. Lauer[10], D. Lennarz[18], A. Lucke[26], J. Lundberg[20], J. Lünemann[17], J. Madsen[31], P. Majumdar[10], R. Maruyama[1], K. Mase[33], H. S. Matis[11], C. P. McParland[11], K. Meagher[16], M. Merck[1], P. Mészáros[24,21], E. Middell[10], N. Milke[15], H. Miyamoto[33], A. Mohr[26], T. Montaruli[1,35], R. Morse[1], S. M. Movit[34], R. Nahnhauer[10], J. W. Nam[8], P. Niessen[7], D. R. Nygren[11,10], S. Odrowski[30], A. Olivas[16], M. Olivo[20], M. Ono[33], S. Panknin[23], S. Patton[11], C. Pérez de los Heros[20], J. Petrovic[14], A. Piegsa[35], D. Pieloth[15], A. C. Pohl[20,36], R. Porrata[9], N. Potthoff[6], P. B. Price[9], M. Prikockis[21], G. T. Przybylski[11], K. Rawlins[30], P. Redl[16], E. Resconi[30], W. Rhode[15], M. Ribordy[25], A. Rizzo[25], J. P. Rodrigues[1], P. Roth[16], F. Rothmaier[17], C. Rott[12], C. Roucelle[30], D. Rutledge[21], D. Ryckbosch[2], H.-G. Sander[17], S. Sarkar[3], S. Schlenstedt[10], T. Schmidt[16], D. Schneider[1], A. Schukraft[18], O. Schulz[30], M. Schunck[18], D. Seckel[7], B. Semburg[6], S. H. Seo[19], Y. Sestayo[30], S. Seunarine[4], A. Silvestri[8], A. Slipak[21], G. M. Spiczak[31], C. Spiering[10], M. Stamatikos[12], T. Stanev[7], G. Stephens[21], T. Stezelberger[11], R. G. Stokstad[11], M. C. Stoufer[11], S. Stoyanov[7], E. A. Strahler[1], T. Straszheim[16], K.-H. Sulanke[10], G. W. Sullivan[16], Q. Swillens[14], I. Taboada[37], A. Tamburro[7], O. Tarasova[10], A. Tepe[6], S. Ter-Antonyan[28], C. Terranova[25], S. Tilav[7], P. A. Toale[21], J. Tooker[31], D. Tosi[10], D. Turčan[16], N. van Eijndhoven[26], J. Vandenbroucke[9], A. Van Overloop[2], B. Voigt[10], C. Walck[19], T. Waldenmaier[23], M. Walter[10], C. Wendt[1], S. Westerhoff[1], N. Whitehorn[1], C. H. Wiebusch[18], A. Wiedemann[15], G. Wikström[19], D. R. Williams[38], R. Wischnewski[10], H. Wissing[14,16], K. Woschnagg[9], X. W. Xu[28], G. Yodh[8], and S. Yoshida[33]

(IceCube Collaboration)

[1] Department of Physics, University of Wisconsin, Madison, WI 53706, USA
[2] Department of Subatomic and Radiation Physics, University of Gent, B-9000 Gent, Belgium
[3] Department of Physics, University of Wisconsin, River Falls, WI 54022, USA
[4] Department of Physics and Astronomy, University of Canterbury, Private Bag 4800, Christchurch, New Zealand
[5] Department of Physics, University of Oxford, 1 Keble Road, Oxford OX1 3NP, UK
[6] Department of Physics, University of Wuppertal, D-42119 Wuppertal, Germany
[7] Bartol Research Institute and Department of Physics and Astronomy, University of Delaware, Newark, DE 19716, USA
[8] Department of Physics and Astronomy, University of California, Irvine, CA 92697, USA
[9] Department of Physics, University of California, Berkeley, CA 94720, USA
[10] DESY, D-15735 Zeuthen, Germany
[11] Lawrence Berkeley National Laboratory, Berkeley, CA 94720, USA
[12] Department of Physics and Center for Cosmology and Astro-Particle Physics, Ohio State University, Columbus, OH 43210, USA
[13] Department of Astronomy, Ohio State University, Columbus, OH 43210, USA
[14] Université Libre de Bruxelles, Science Faculty CP230, B-1050 Brussels, Belgium
[15] Department of Physics, TU Dortmund University, D-44221 Dortmund, Germany
[16] Department of Physics, University of Maryland, College Park, MD 20742, USA
[17] Department of Physics and Astronomy, University of Kansas, Lawrence, KS 66045, USA
[18] III Physikalisches Institut, RWTH Aachen University, D-52056 Aachen, Germany
[19] Oskar Klein Centre and Department of Physics, Stockholm University, SE-10691 Stockholm, Sweden
[20] Department of Physics and Astronomy, Uppsala University, Box 516, S-75120 Uppsala, Sweden
[21] Department of Physics, Pennsylvania State University, University Park, PA 16802, USA
[22] University of Mons-Hainaut, 7000 Mons, Belgium
[23] Laboratory for High Energy Physics, École Polytechnique Fédérale, CH-1015 Lausanne, Switzerland
[24] Department of Astronomy and Astrophysics, Pennsylvania State University, University Park, PA 16802, USA
[25] Vrije Universiteit Brussel, Dienst ELEM, B-1050 Brussels, Belgium
[26] Department of Physics and Astronomy, Utrecht University/SRON, NL-3584 CC Utrecht, The Netherlands
[27] CTSPS, Clark-Atlanta University, Atlanta, GA 30314, USA

Software is everywhere



A point made better by many others, but particularly Edwards, P. N. (2010). *A vast machine: Computer models, climate data, and the politics of global warming*. MIT Press.

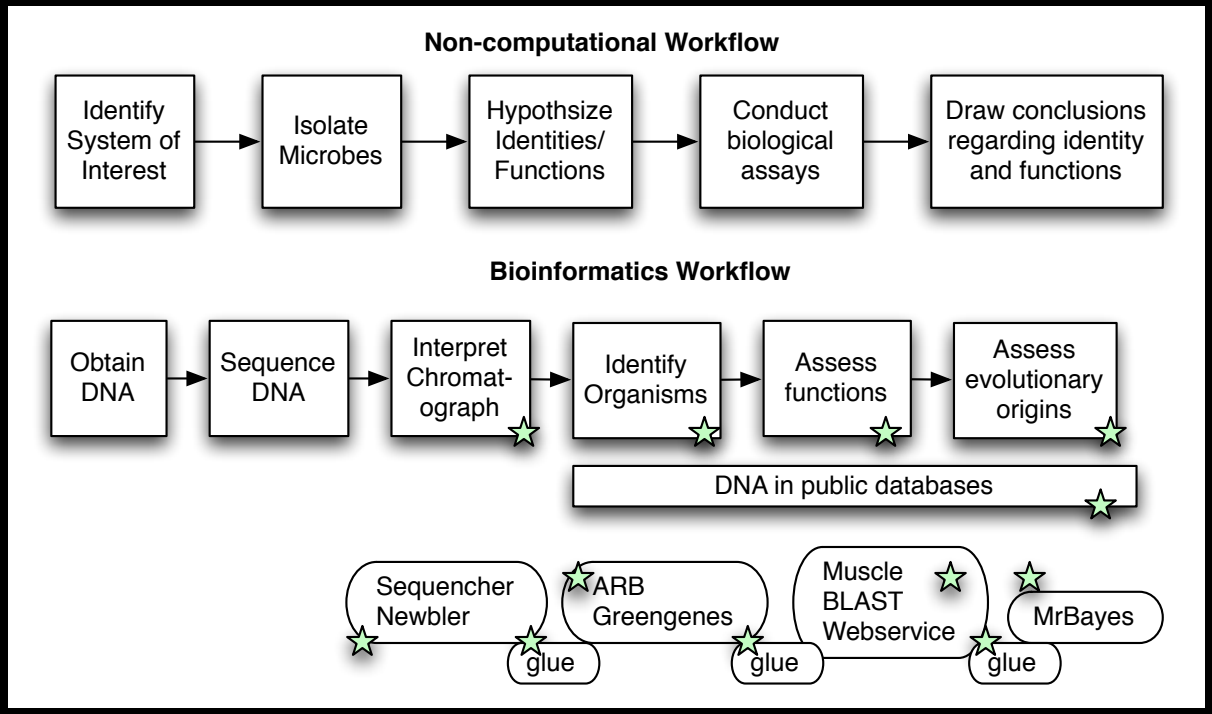
Three cases with focal papers

- STAR collaboration at Brookhaven national lab
- SBGrid (Structural Biology Grid software distribution)
- Bioinformatics study of leaf cutter ants

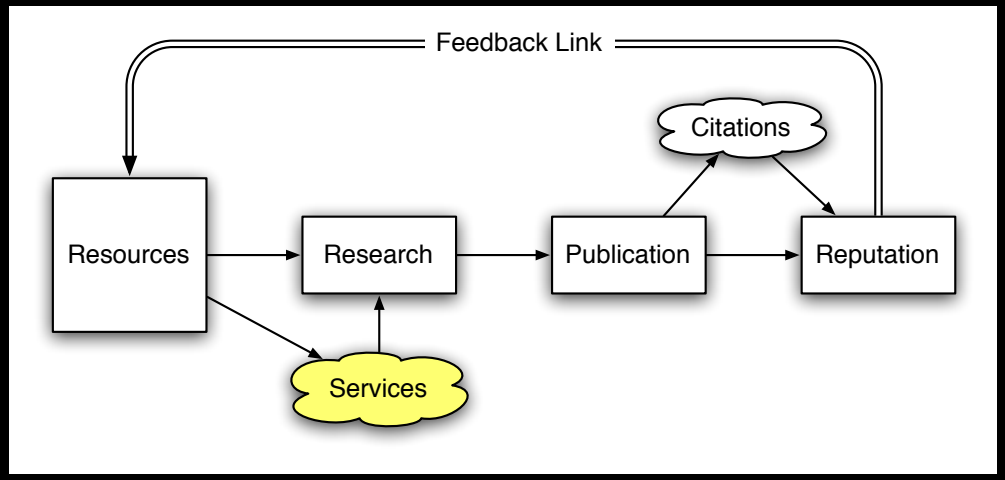
- We documented the workflows, identifying all the software involved, then interviewed the producers of that software, focusing on incentives for doing the software work.

Howison, J., & Herbsleb, J. D. (2011). Scientific software production: Incentives and collaboration. *Proceedings of the ACM Conference on Computer Supported Cooperative Work*, 513–522. https://doi.org/10.1145/1958824.1958904

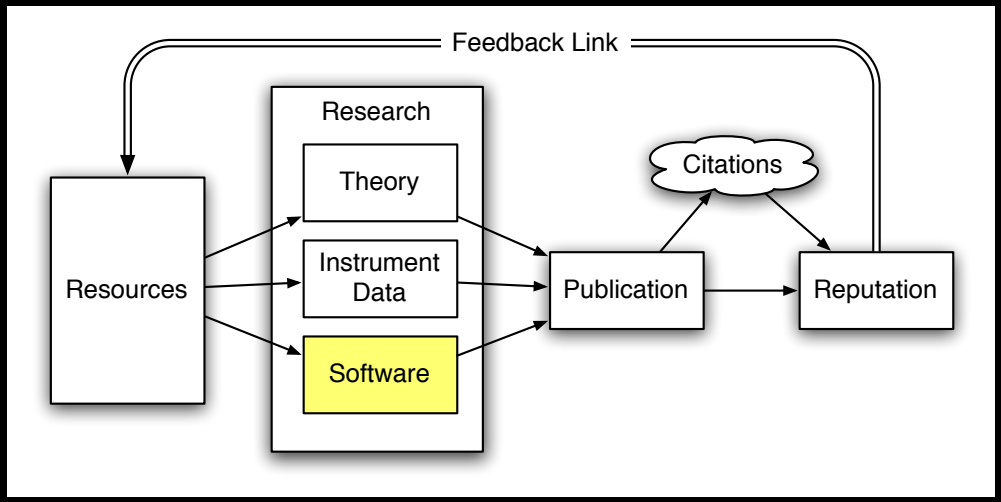
Example workflow with identified software components



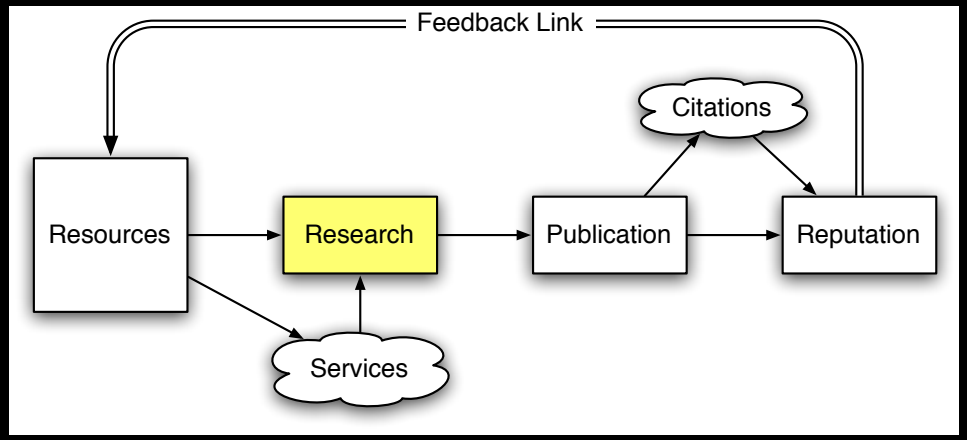
Software as support



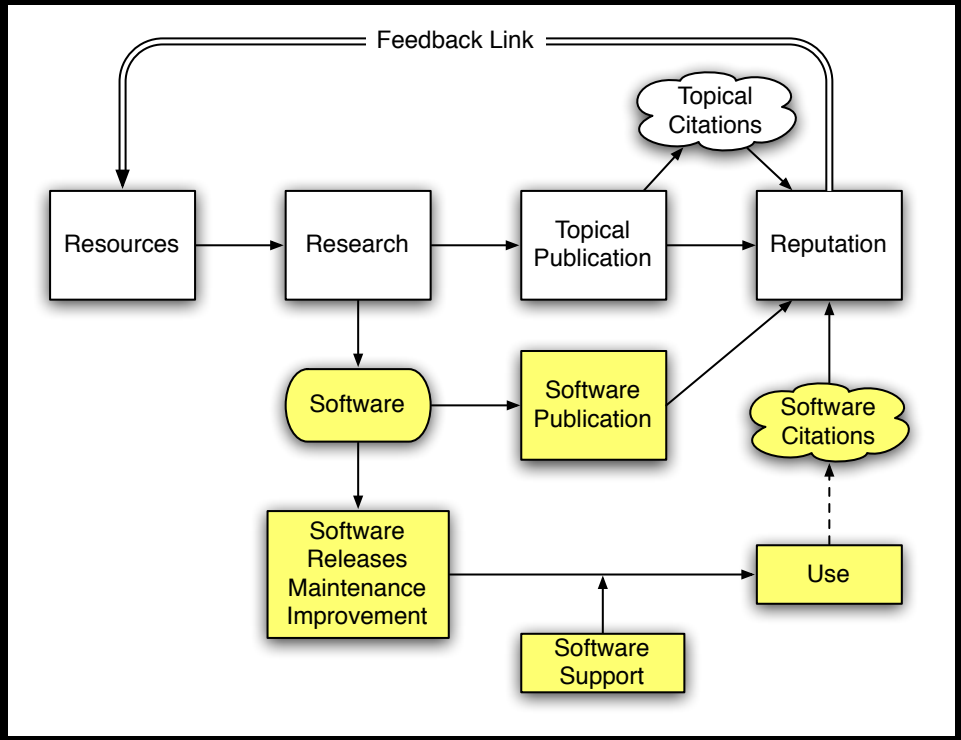
Collaboration service-work



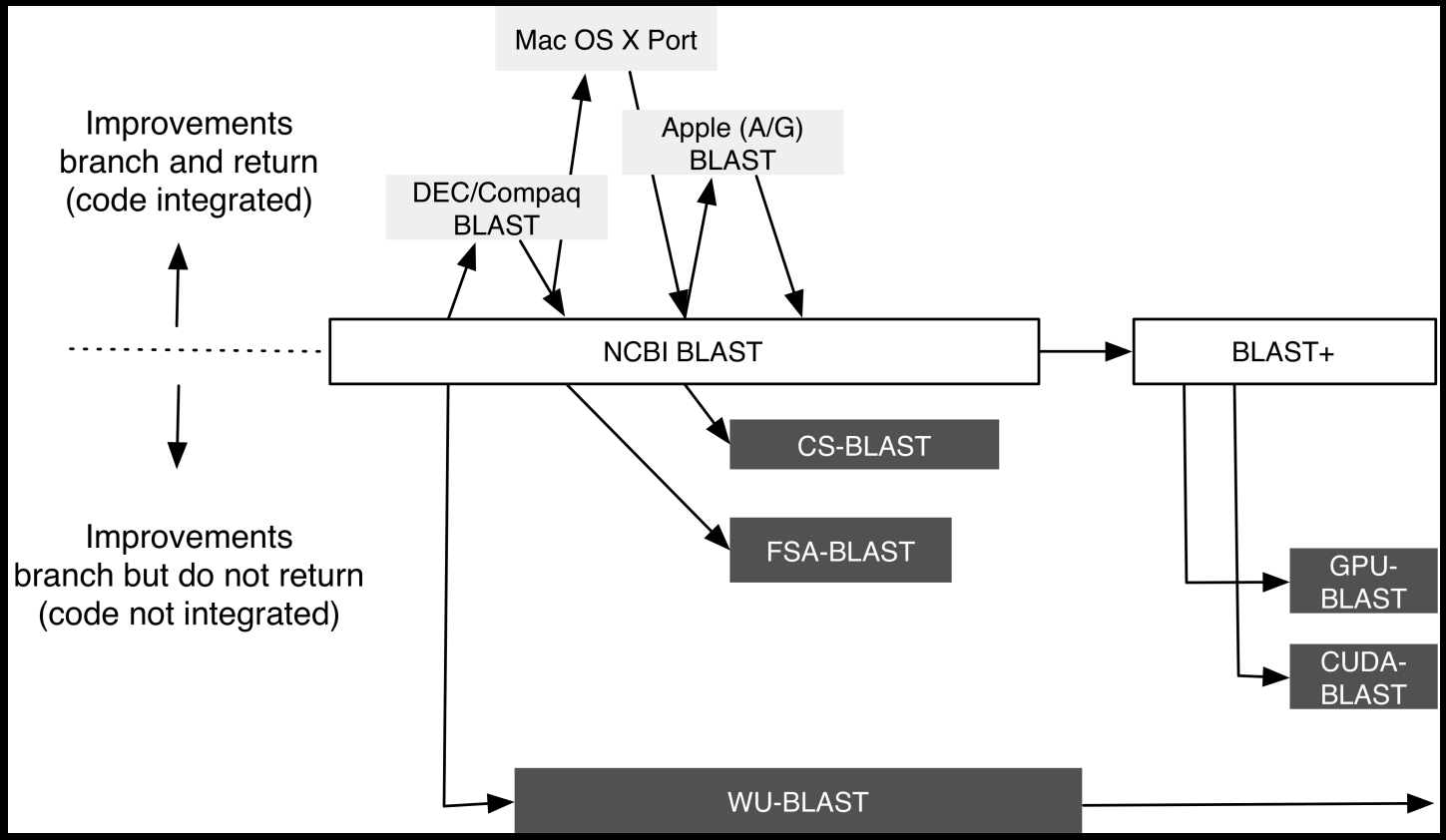
Academic credit: Incidental software



Academic credit:
Parallel software practice



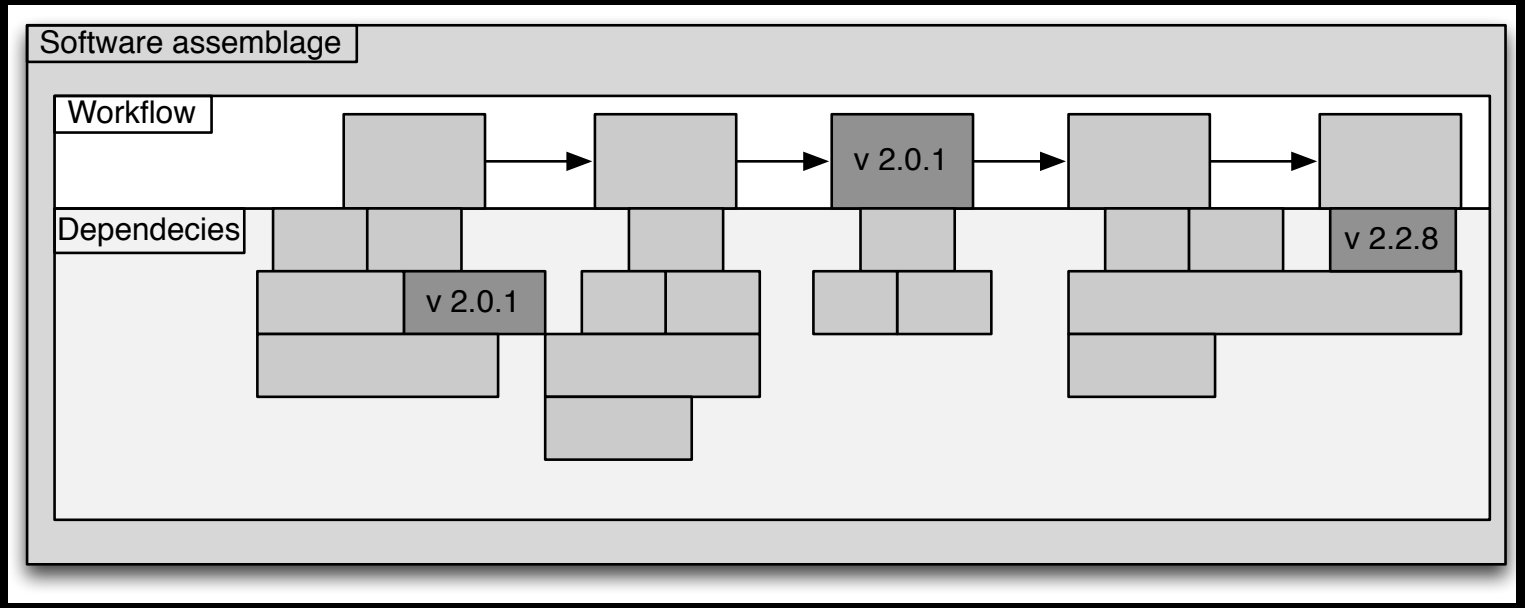
Academic reputation dampens integration



James Howison and Jim Herbsleb (2013) Sharing the spoils: incentives and integration in scientific software production. ACM CSCW

Part 2: Why is sustainability challenging?

How do scientists use software?



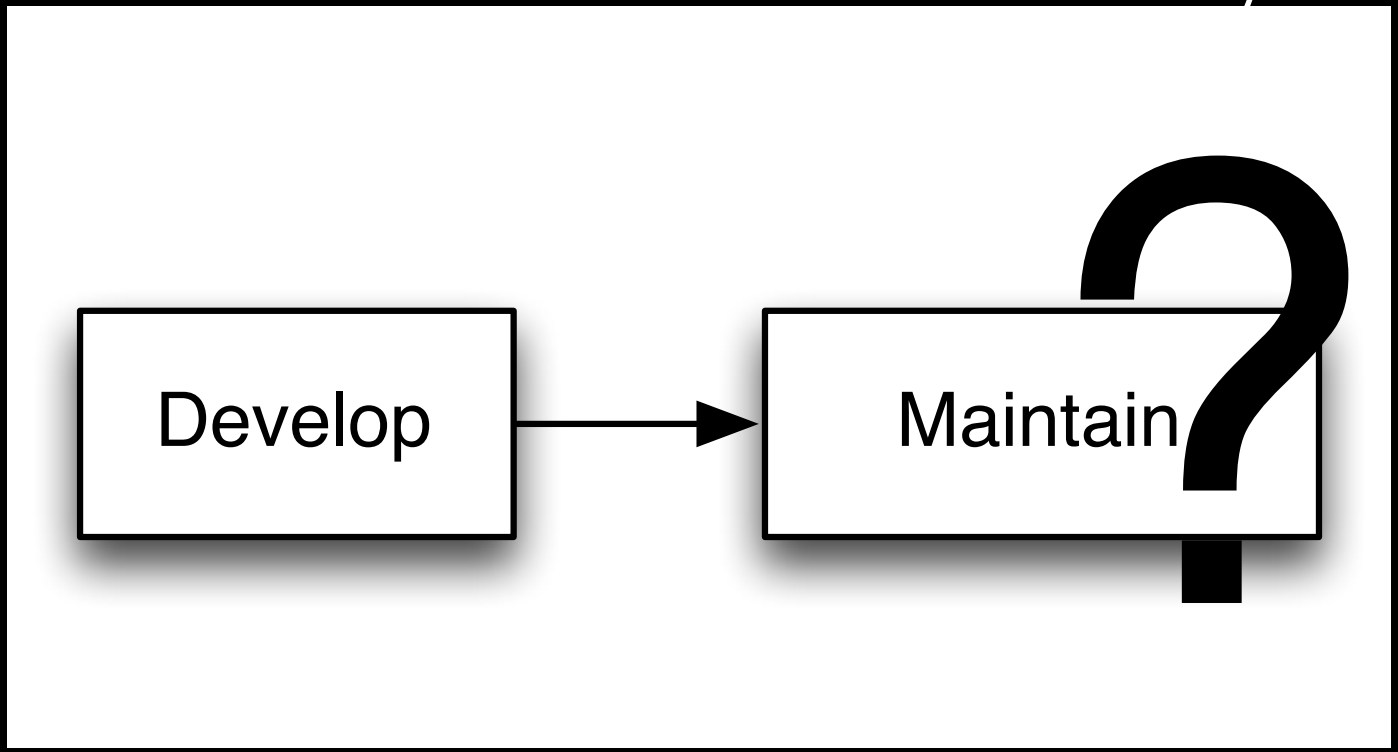
Edwards, Batcheller, Deelman, Bietz and Lee, Segal, De Roure and Gobels, Ribes and Finholt, Howison and Herbsleb

@jameshowison

Re-animating assemblages

- Scientists pull an assemblage together, “get the plots” and often then leave it, often for months or years.
  - The pace of work is generally much slower than in industry (where software is used day in and day out)
  - While infrastructure runs continually, end-users often encounter it only periodically
- When they return they return to *extend*; to use the software assemblage for new purposes, for new science, not simply to replicate/repeat.

What work is needed for sustainability?



What drives the need for work?

1. Difficulty of software production
2. Difficulty of software use
3. Changing scientific frontier
4. Changing technological capabilities (hardware and software)
5. Ecosystem complexity

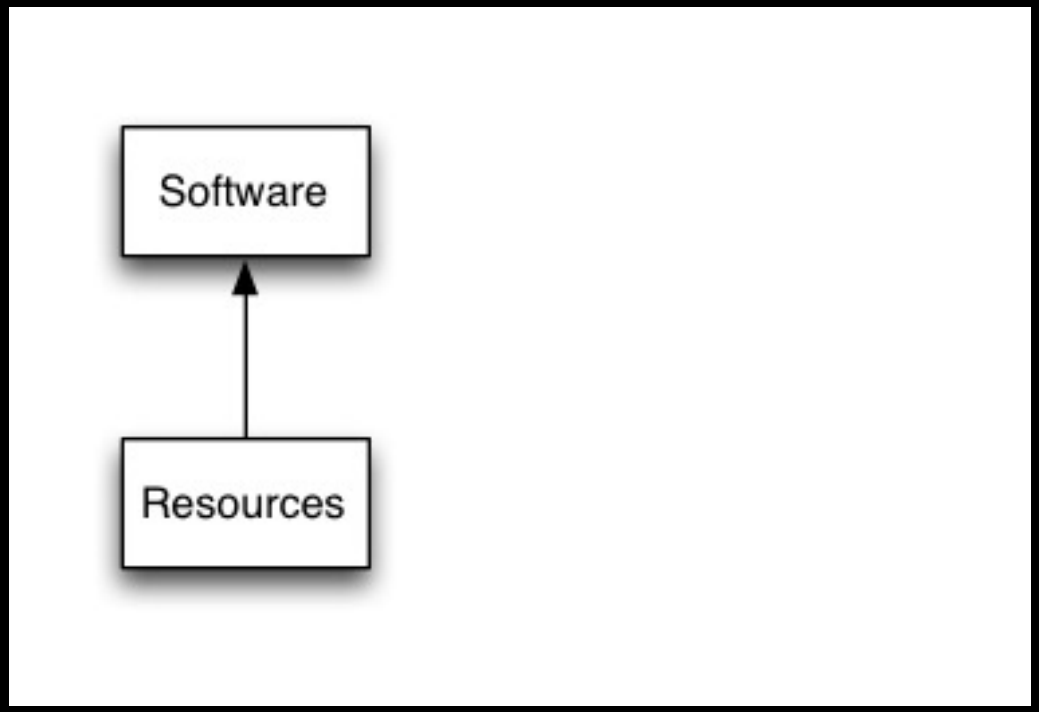
Strategies for reducing needed work

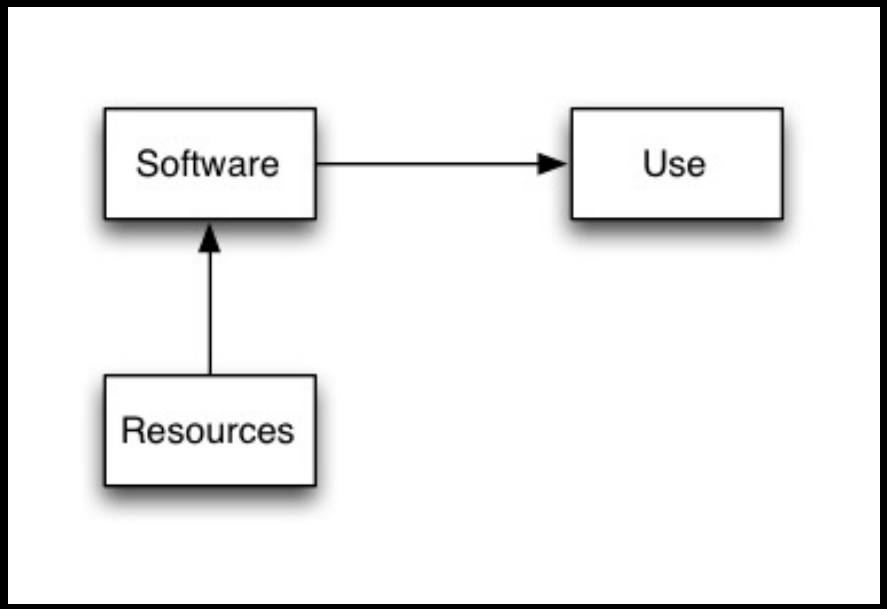
1. Suppress the drivers
2. Increase the efficiency of work (e.g., Becker et al)

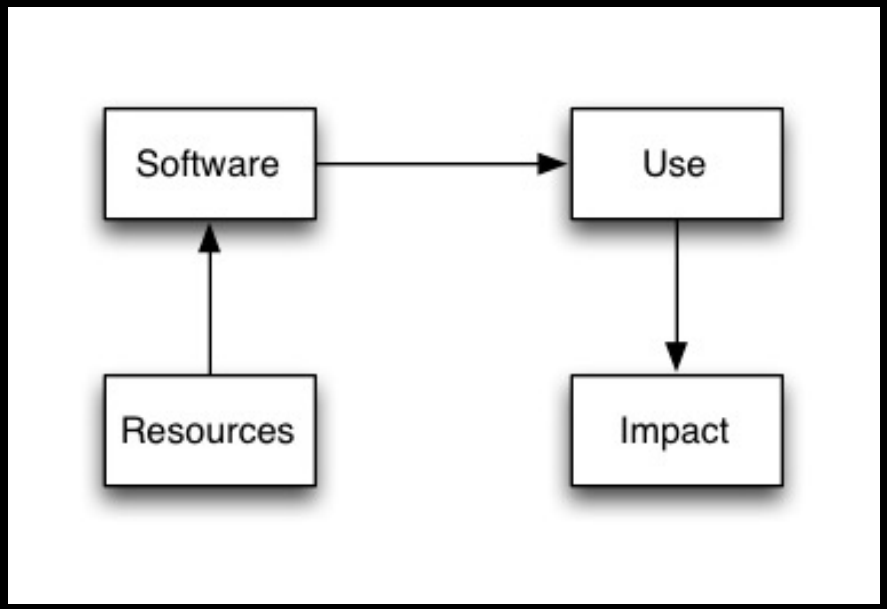
   … but, ultimately, work is *always* needed …

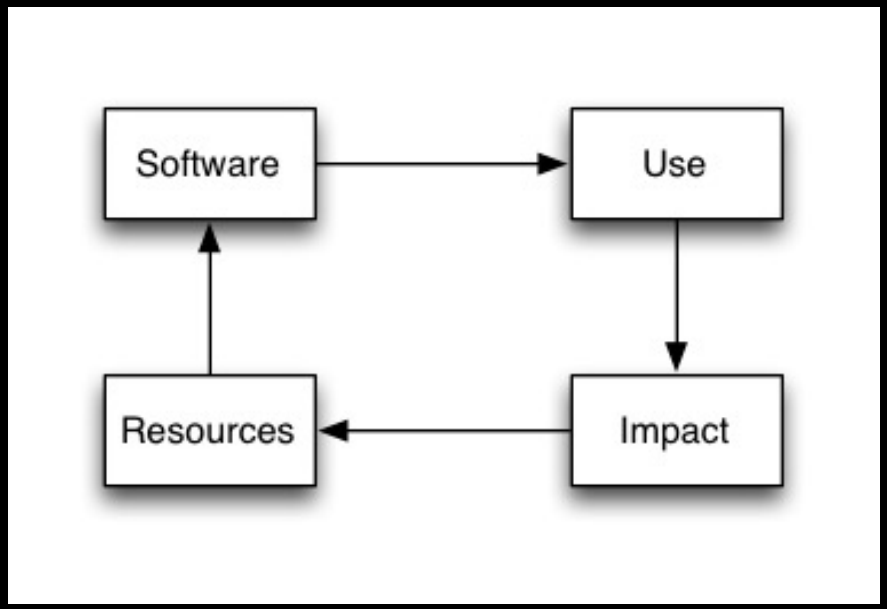
3. Attract resources willing and able to do the work

Resource attraction


Software
Resources


Software
Use
Resources


Software
Use
Resources
Impact

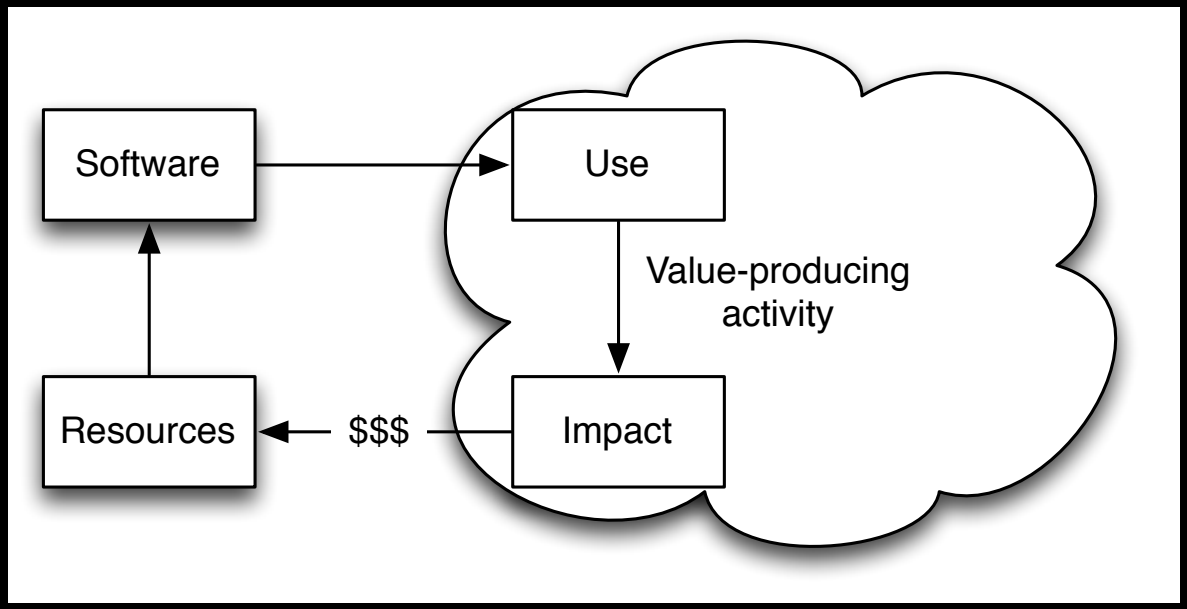

Software
Use
Resources
Impact

Modes of production

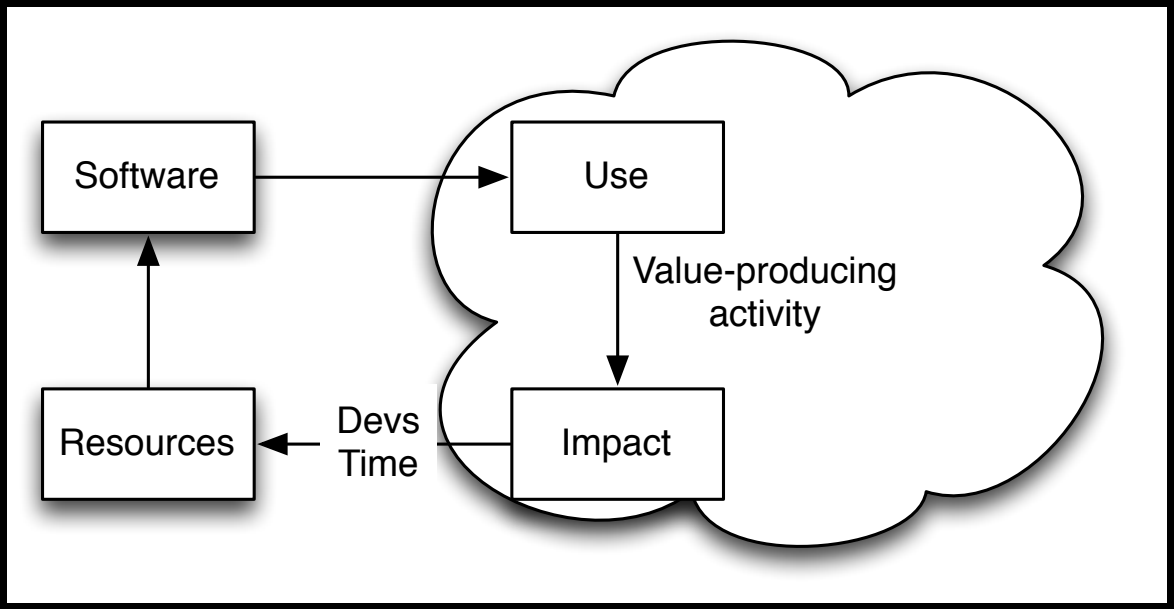
- How does a project attract resources?
  - Or, *how does use and impact turn into resources*?

- Three “ideal types”:
  1. A commercial project
  2. Open source peer production
  3. Scientific grant making

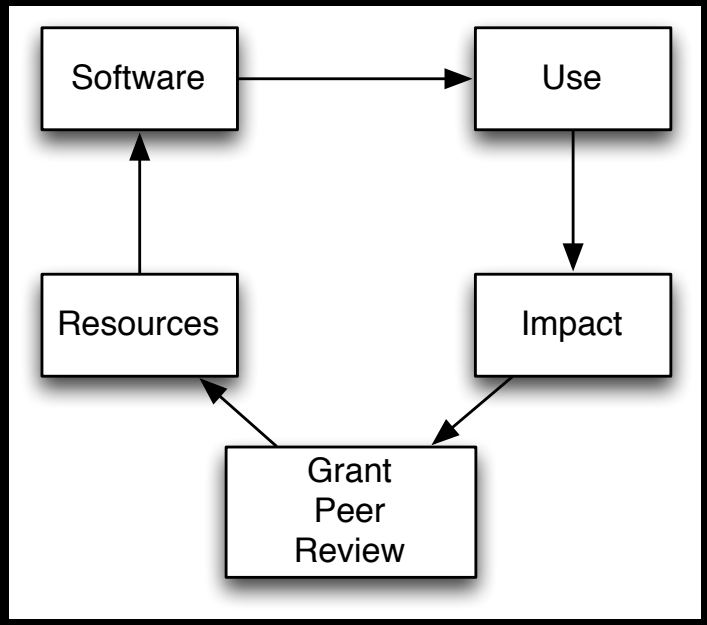
A commercial project



Open Source Peer Production



Grant-making



Ribes and Finholt 2007;
Howison and Herbsleb 2011

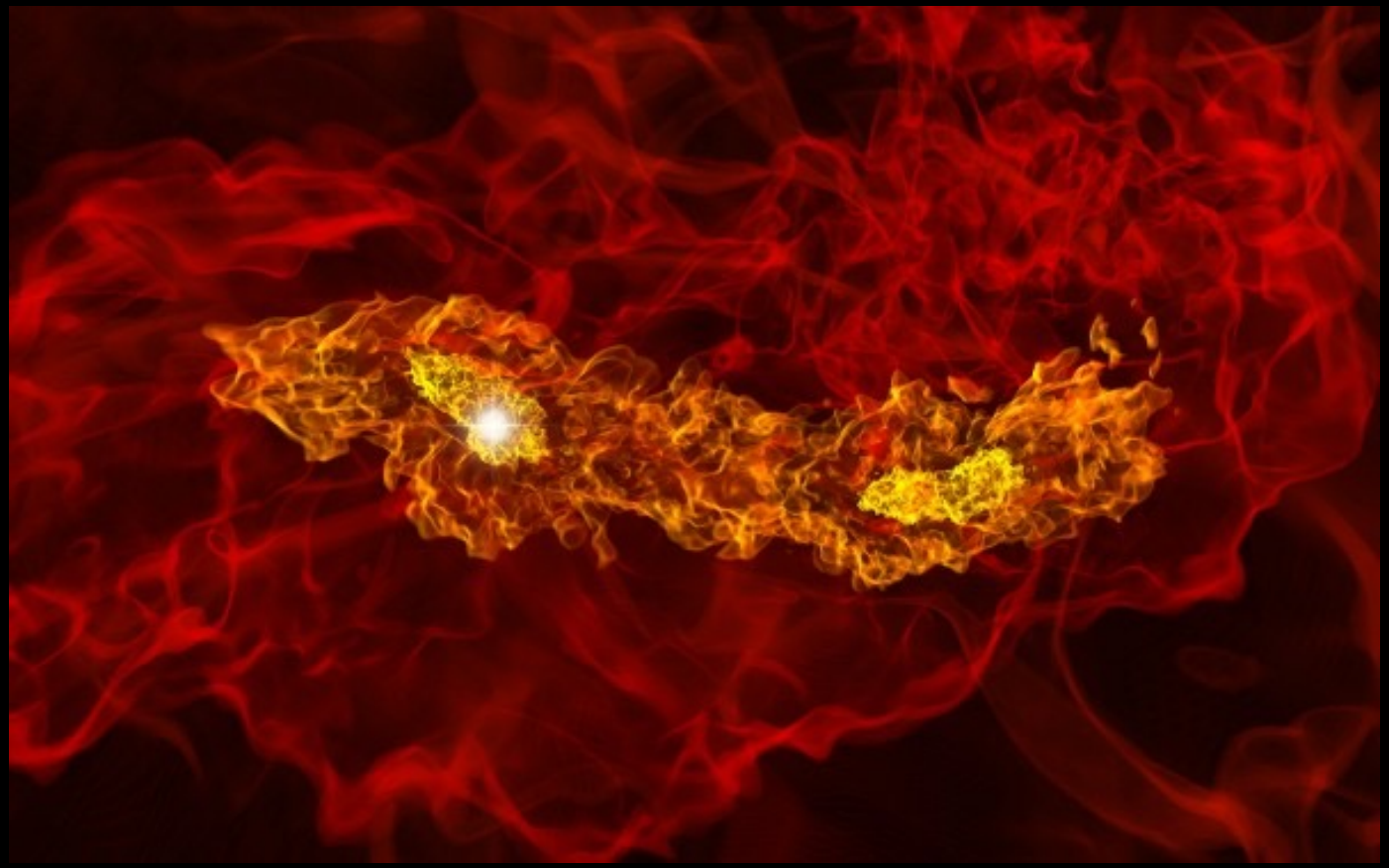
Summary: why so difficult?

- Indirectness of resource attraction
  - Reputation is a great motivator, but also problematic
- Incompatible incentives for needed work
  - Reputational rewards out of sync with needed work
  - Citations/literature impact rewards publication not maintenance
  - Grants create “service center” mentality
  - See also Nadia Eghbal “Roads and Bridges” for similar issues outside science
- Difficulty of work
  - Availability of skilled labor
- Science is unwilling to wait
- Time-scale of use
  - Longer cycles of use and re-use mean more change
- Ecosystem complexity
  - Will return to this later – different modes of production handle this differently.

Part 3:
Routes to Peer Production in Scientific Software

- Work with Eunyoung Moon, Hannah Cohoon, and Caifan Du
- Thanks especially to
  - Matt Turk (from the yt project) for countless discussions
  - Everyone that organized and participated in the WSSSPE workshops http://wssspe.researchcomputing.org.uk/

ENZO

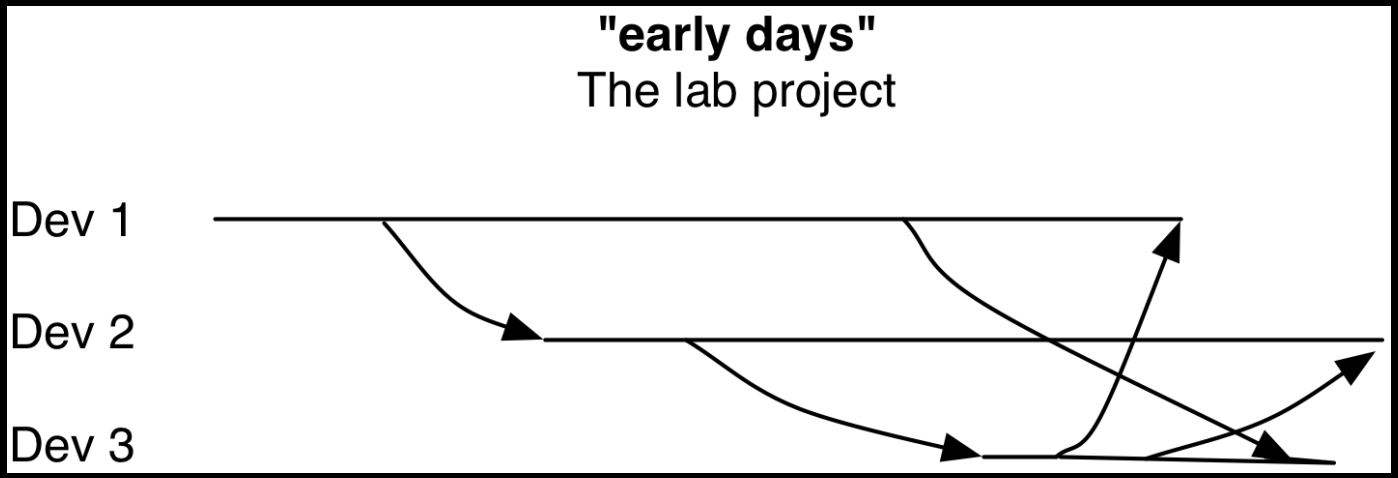


@jameshowison

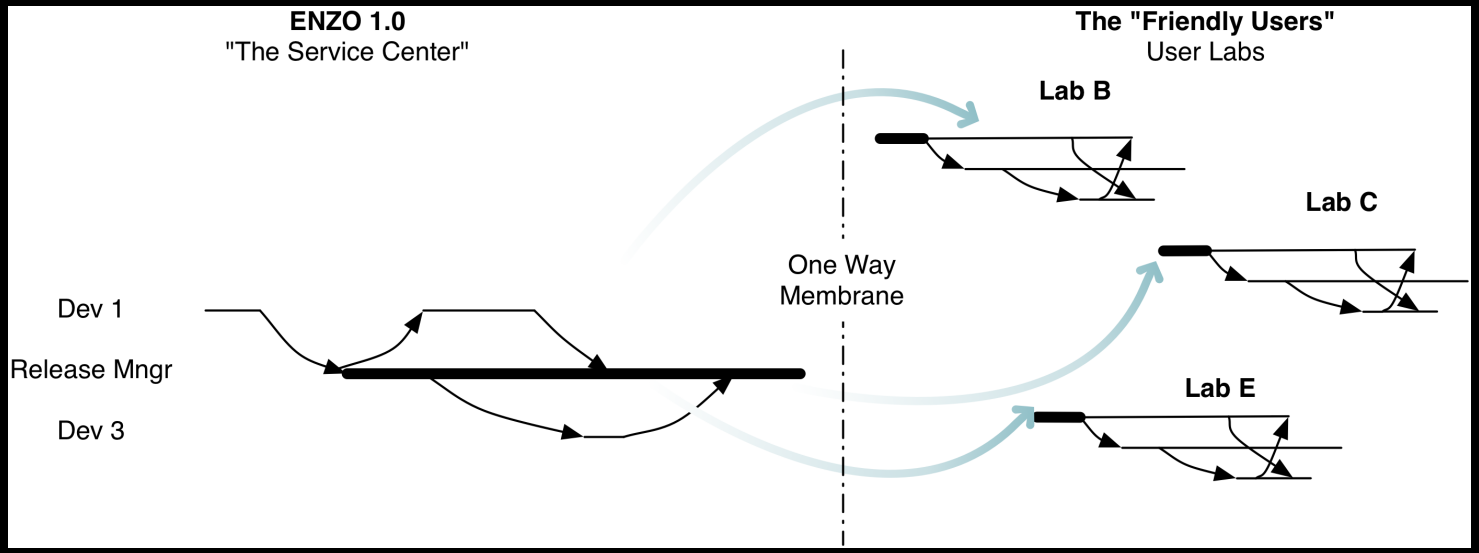
ENZO case study

Data:

- 5 interviews with participants
- Observation at related workshop
- Publications, websites, workshop websites, source code repositories
- Analysis:
  - Creation of timeline
  - Identification of episodes and 4 project phases (with their precipitating events)



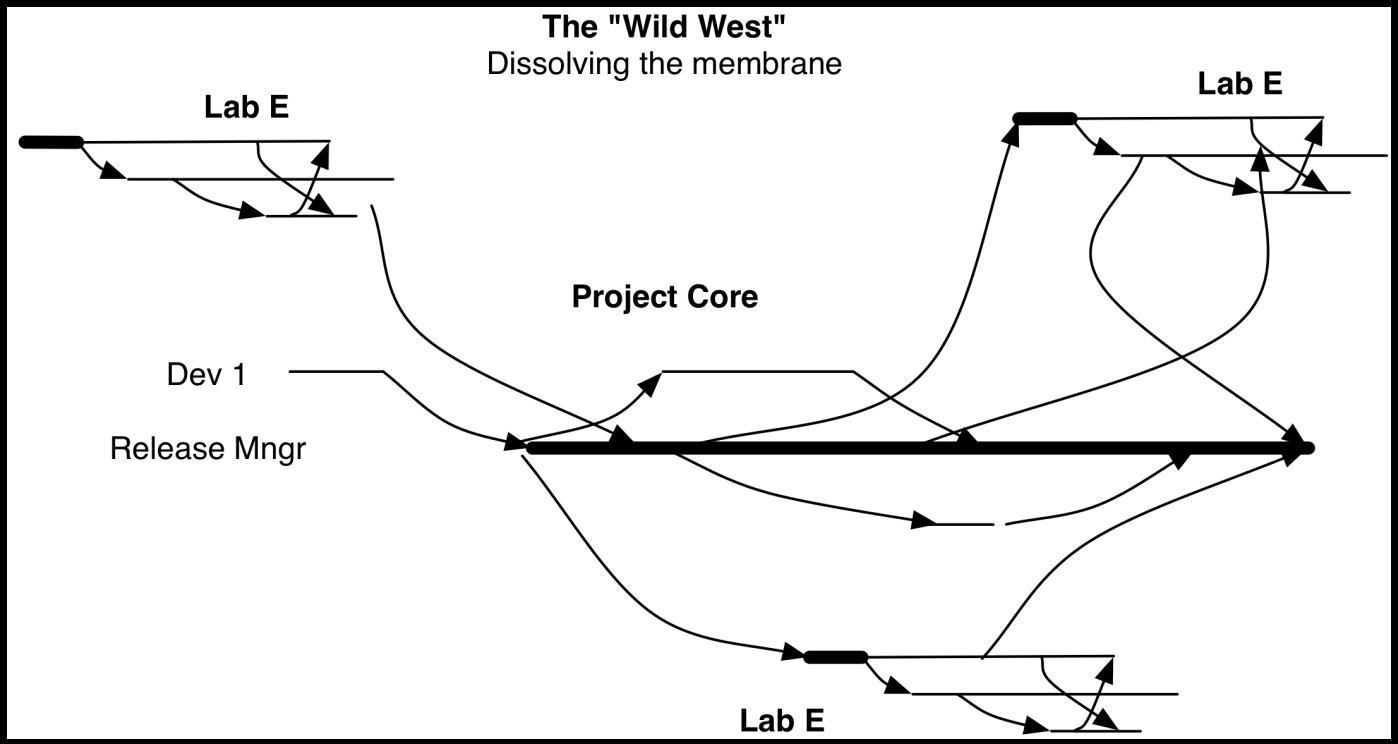
- No central base to which changes are coming and going
- Copy and pasting features across personal branches
- Single lab



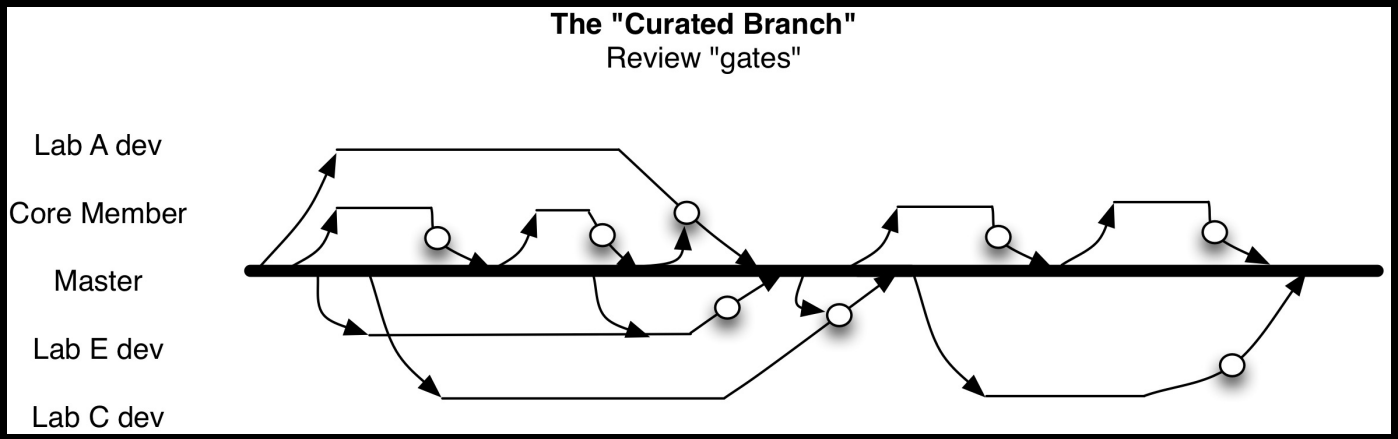
- ENZO lab reforms as “Service Center” (grant)
- Mainline branch internally, releases externally
- Little expectation of contributions coming back in
- “Friendly user” labs internally functioning like “early days”

The “Week of Code”

- Director of external lab (former post-doc) has new job at Stanford (with startup funds!)
- Learns of various versions through conversations at conferences and reviewing(!)
- Focus is on collaboration infrastructure, not governance.
- Begin with the code of those **not present**



- Central branch to which both core and outsiders contribute
- Development continues separately in external labs
- Called “Wild West” by participants, autonomy concerns.



- Introduction of “code revision” (pull requests)
- External lab members on similar footing to Core members
- Review helps members not “step on each other’s work”

Change

- What didn’t change:
  - Motivations (code is side-effect of scientific inquiry, papers first, code second), not focused on commercial value
- Challenges to change
  - Coordination after long periods working apart
  - Leadership’s feeling of responsibility and emotional connection, difficulty of passing on leadership (which eventually did happen).
  - Deep concerns about giving up autonomy (being “blocked” in one’s work)

What worked

- Collaboration technology **before** governance (perhaps contra "Collaboration Readiness" (Olson et al.) TORSC?).
  - A scaling of the set of features of collaboration technology over time, from version control to code review
- Social proof: visible action in public (Colfer and Baldwin "actionable transparency")
- Inspiration from open source
- Working alongside each other, rather than with each other. Superposition rather than Teamwork.

Part 3: Panel Study of SI2 funded software projects

- NSF funding through SI2 program contributed to over 350 grants
- Three step qualitative content analysis:
  1. Did the grant intend to create software
  2. What documents (URLs, Workshop reports, or Publications) are available?
  3. Read these, apply coding scheme
- Interviews to confirm and deepen (interviews ongoing)
- Today report on study of 92 grants that started by 2014, giving sufficient time for them to complete and observe transitions
  - Of these 92 grants, 84 were judged to intend to create software
- These 84 grants funded activity on 114 codebases, thus we present 114 projects observed multiple times over 5 years.
- Thanks to doctoral student Hannah Cohoon and Caifan Du

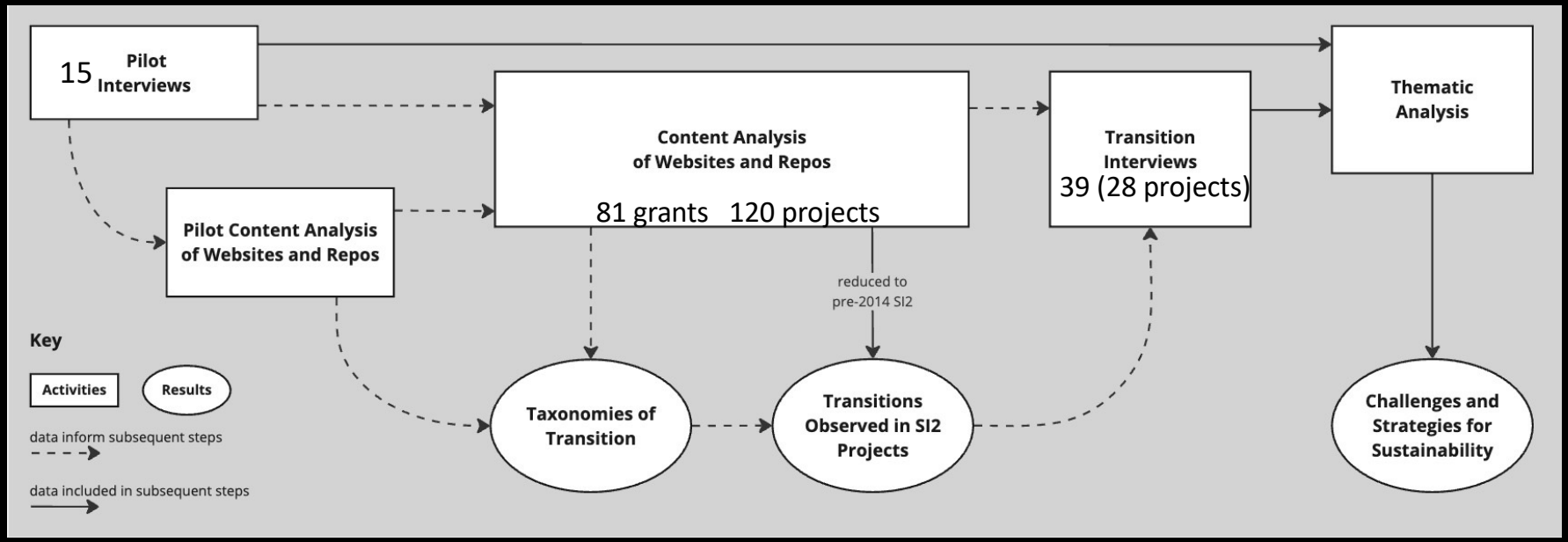
Research Questions

1. How were projects organized? How did that change?
2. What routes to peer production did our projects pursue?
3. What challenges exist, and how are they addressed?

See Cohoon, J., Du, C., & Howison, J. (2025). Tales of Transitions: Seeking Scientific Software Sustainability. *Proceedings of the ACM on Human-Computer Interaction*, *9*(1), 1–25. https://doi.org/10.1145/3701208

Method

Studied software projects funded by NSF through the SI2 program, over 7 years



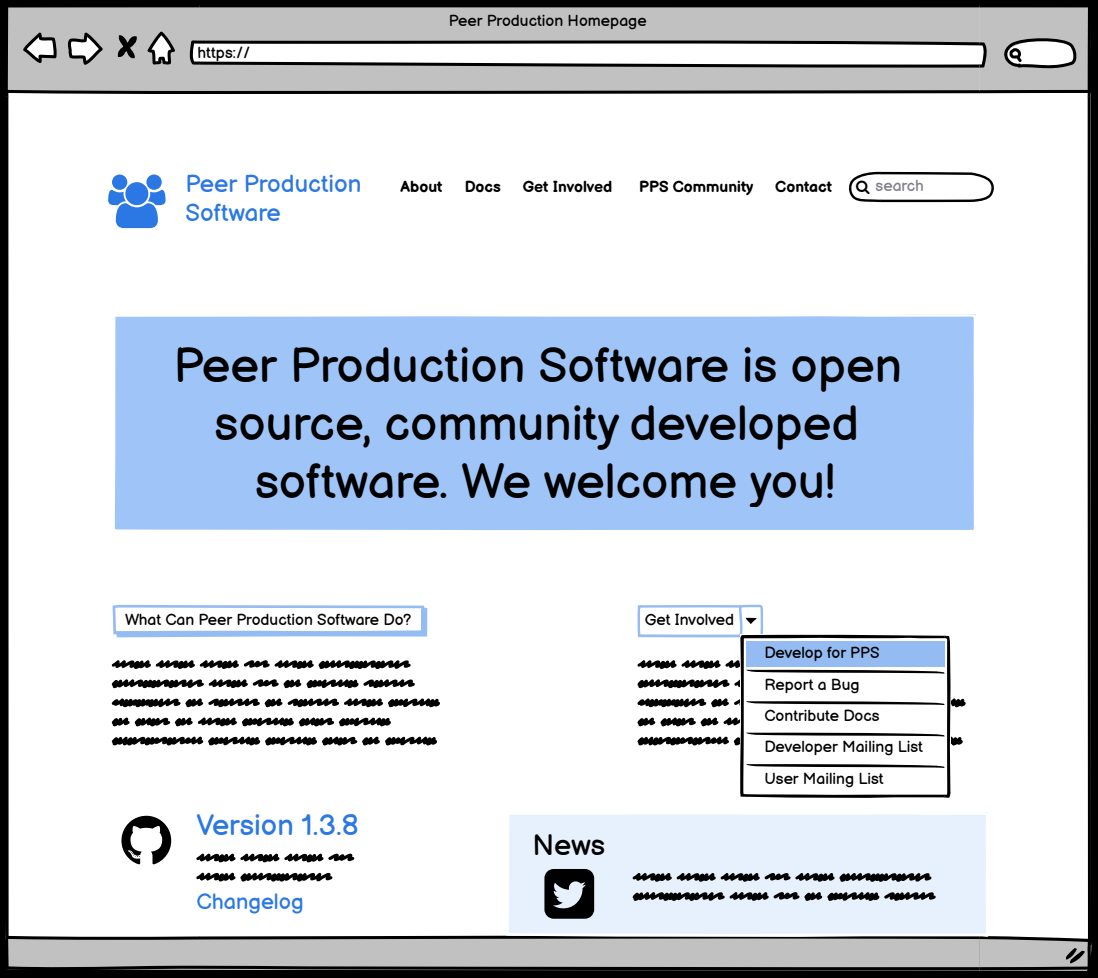
Results – How organized?

- We identified six organizational configurations
  - Peer production, Labs, Author Groups, Tool Groups, Businesses and Consortia

- We identified three kinds of organizational transitions between the
  - Reorganizations (same people, new form)
  - Migrations (different people, same form)
  - Hand-offs (different people, different form).

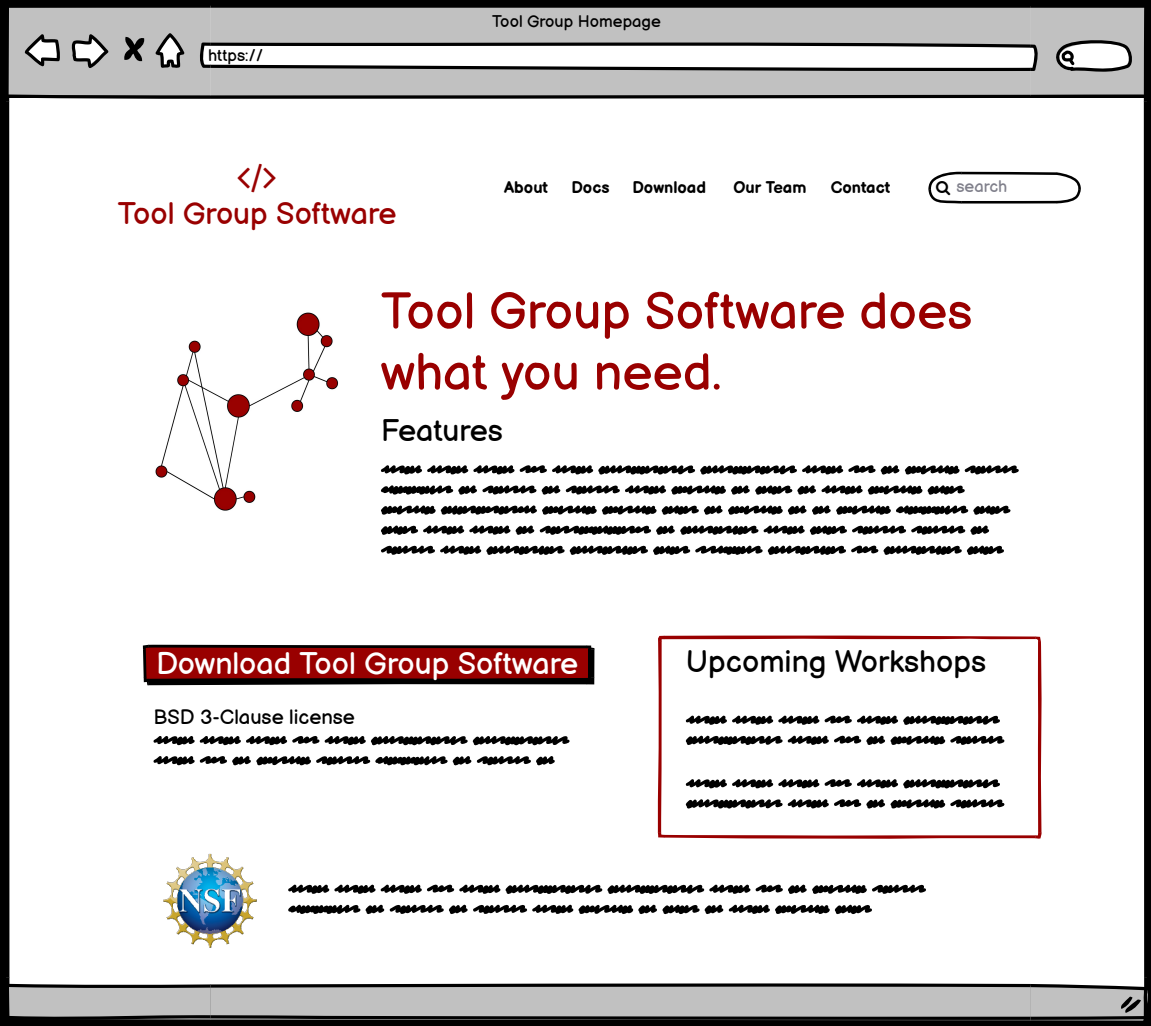
Peer Production

1. Group name
   - Identifies as community
2. Website
3. Singular interests
4. Outside contributions
5. Invitation to contribute
6. Very rarely Publications
   - Not featured if present
7. No Revenue



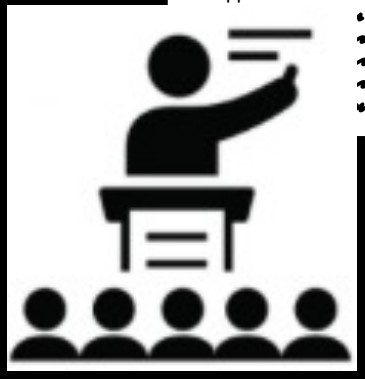
Tool Group

1. Group is Named
   - Rarely identifies as community
2. Website
3. Singular interests
4. Rarely Outside contributions
5. Rarely an Invitation to contribute
6. Sometimes Publications
   - Usually not featured
7. No Revenue



Author Group

1. Group is not named
   - Never identifies as community
2. Rarely a Website
   - Typically posters (SI2 meeting!)
3. Usually Singular interests
4. No outside contributions
5. No Invitation to contribute
6. Always Publications
7. No Revenue



University of Collegestan
University of Elsewhere

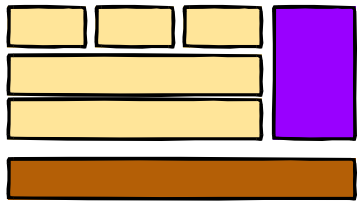
SI2-SSE: A Framework for Something Developed by an Author Group

Principal Investigator, Second Principal Investigator, Graduate Student, Graduate Student

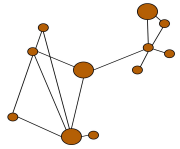
Problem

Approach

Architecture



Applications

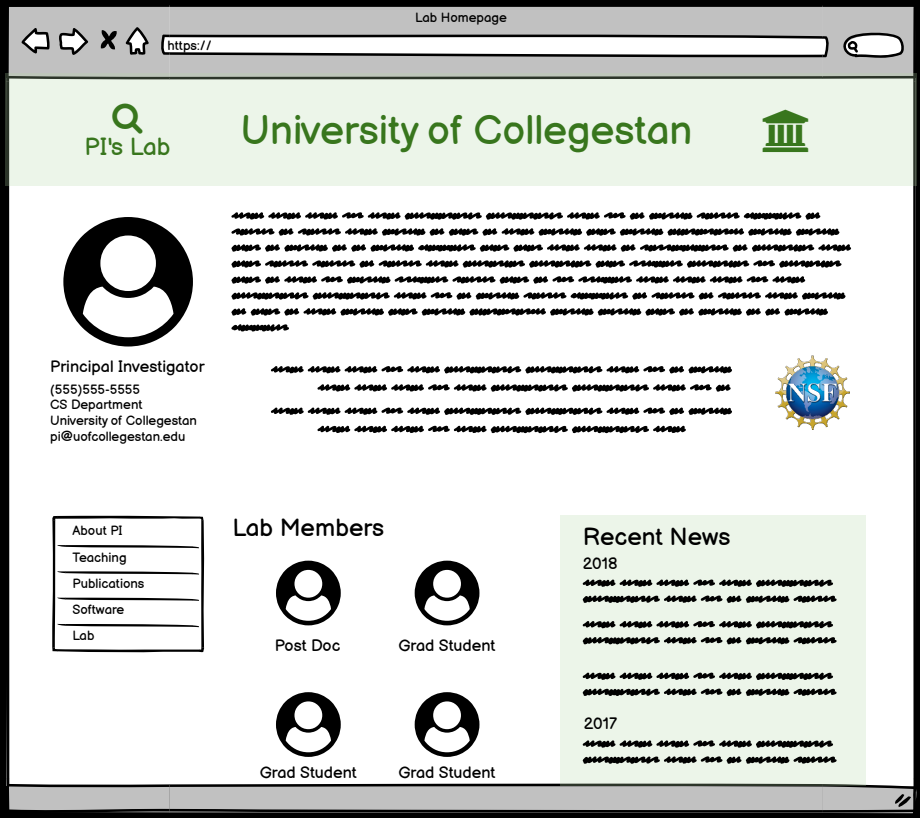


References

20XX NSF SI2 PI Meeting

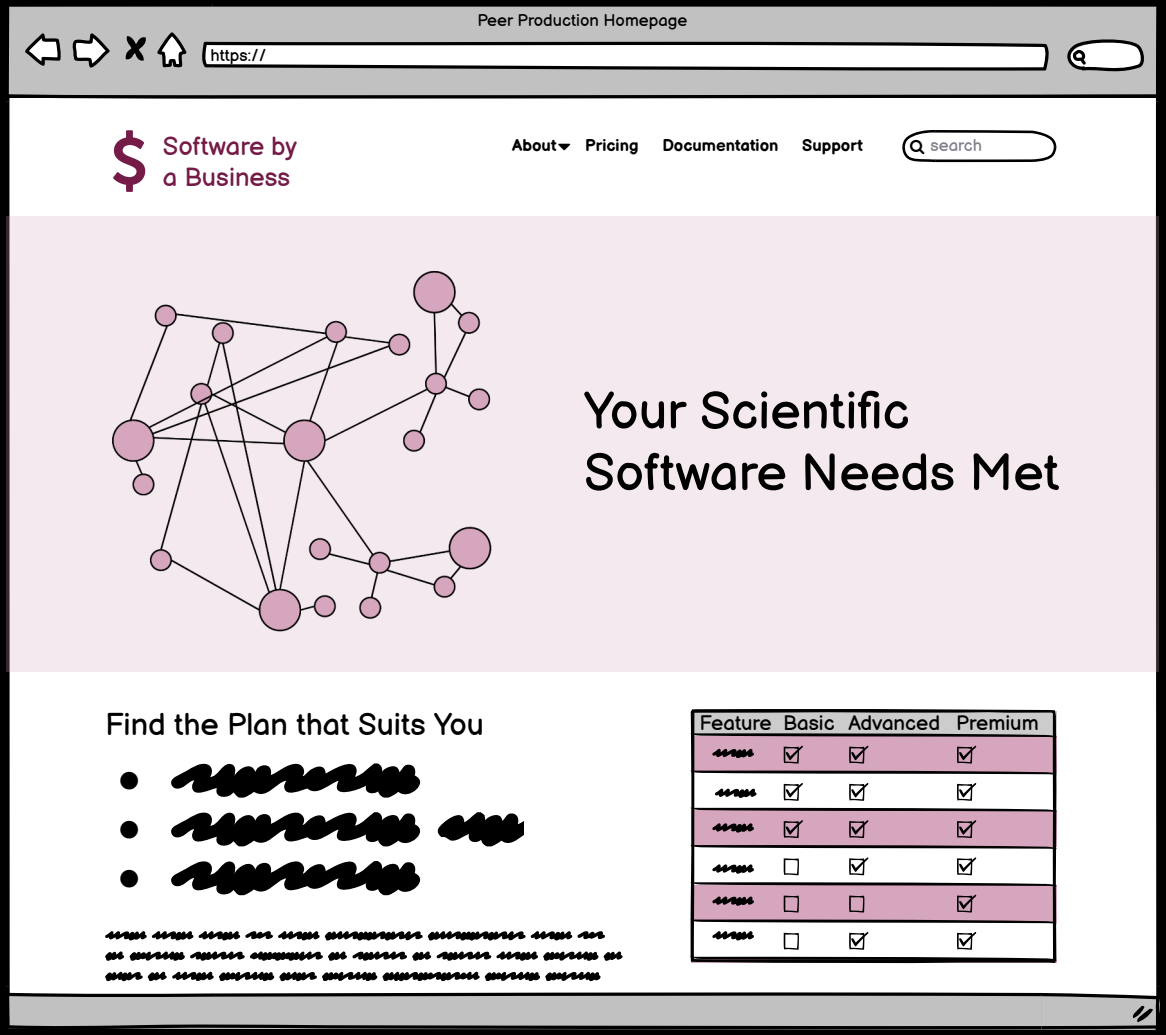
Lab

1. Group is named
   - Usually after the PI
2. Website
3. Diverse interests
   - Link to genres of outputs inc. software
4. Rarely Outside contributions
5. Rarely an Invitation to contribute
   - “Join my group as student or postdoc”
6. Always Publications
7. No Revenue



Business

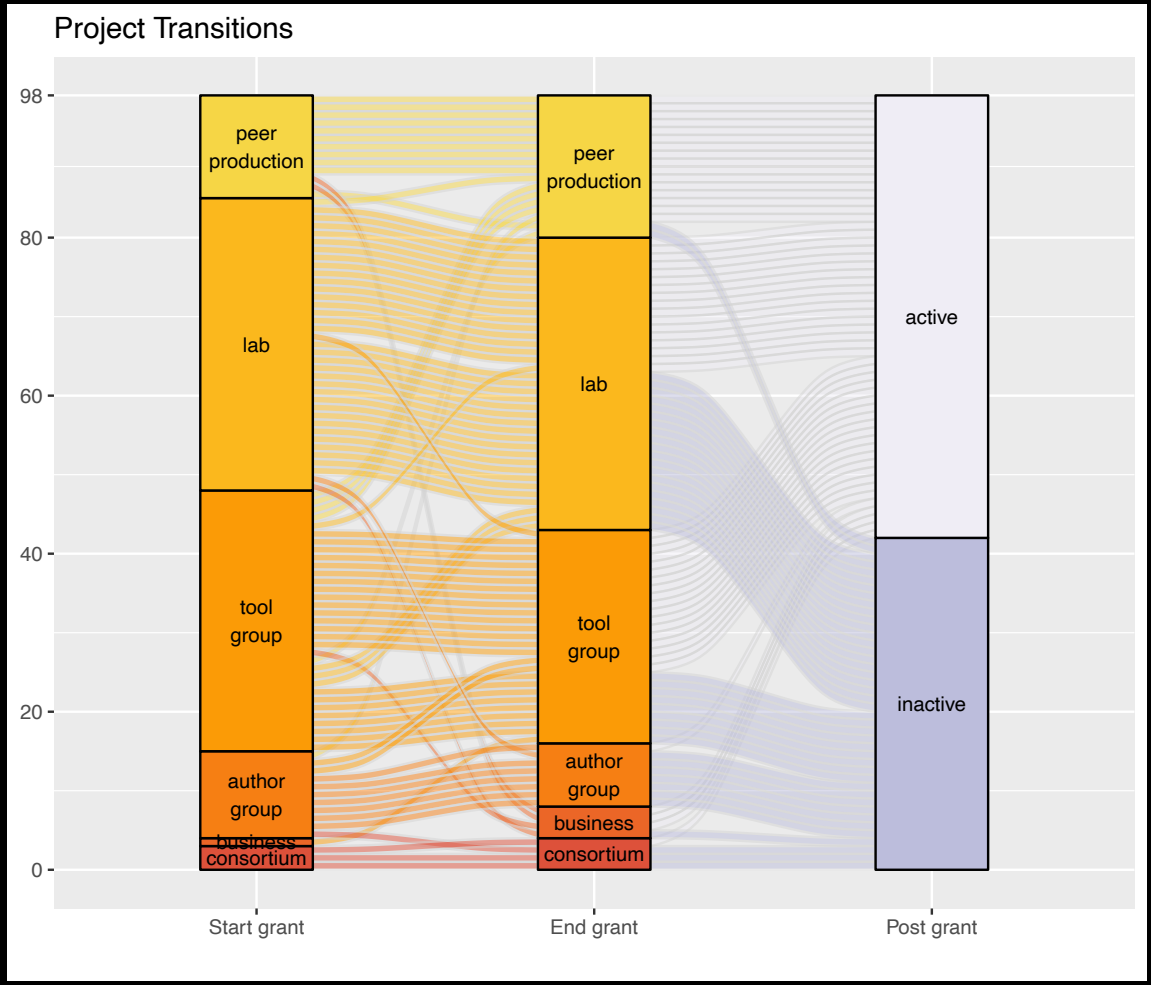
1. Group is named
2. Website
3. Singular interests
4. Never outside contributions
5. Never an Invitation to contribute
   - Sometimes a separate “community”
6. Rarely Publications
   - Not featured on website
7. Revenue (or at least “plans”)



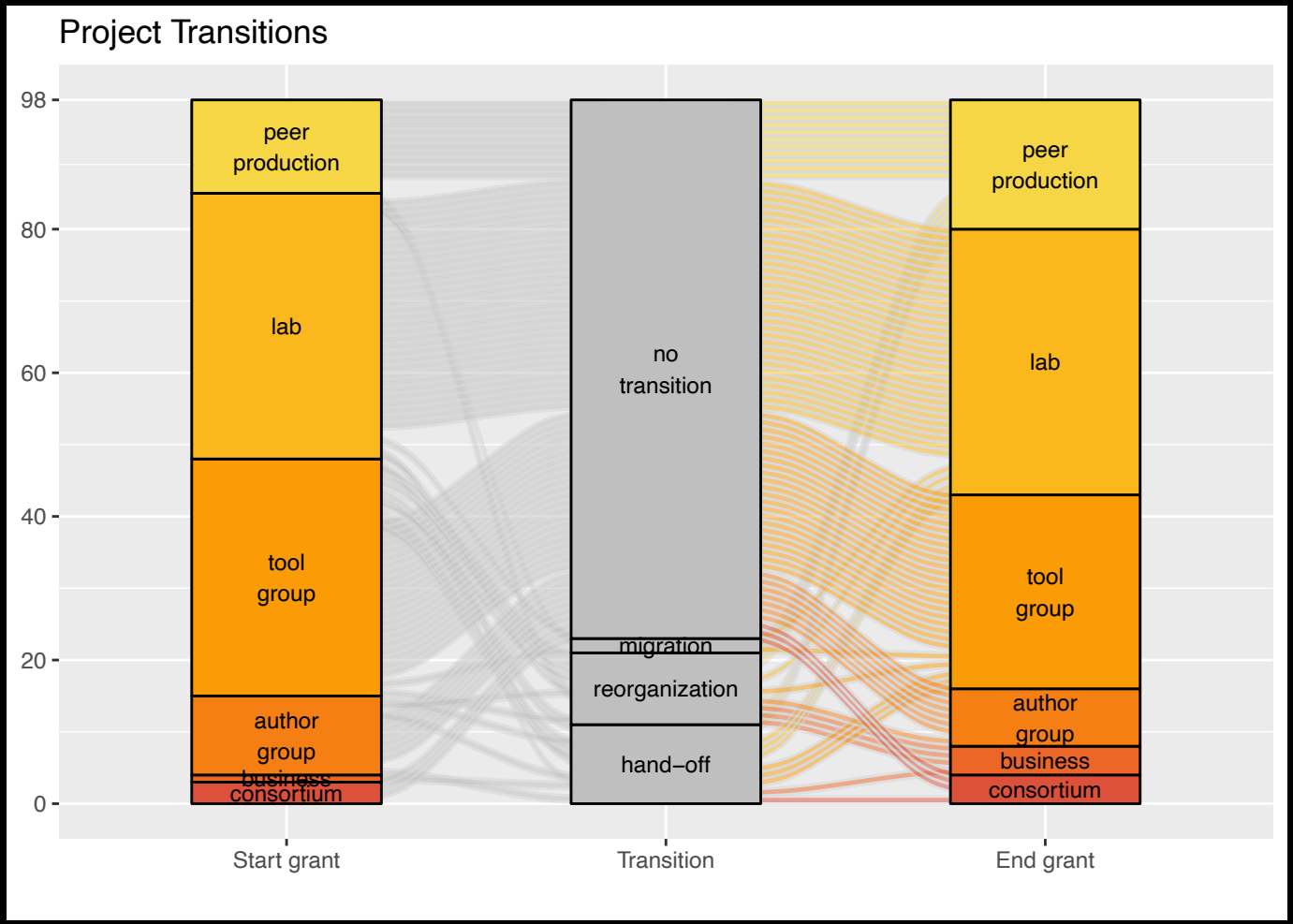
Result – How did organization change?

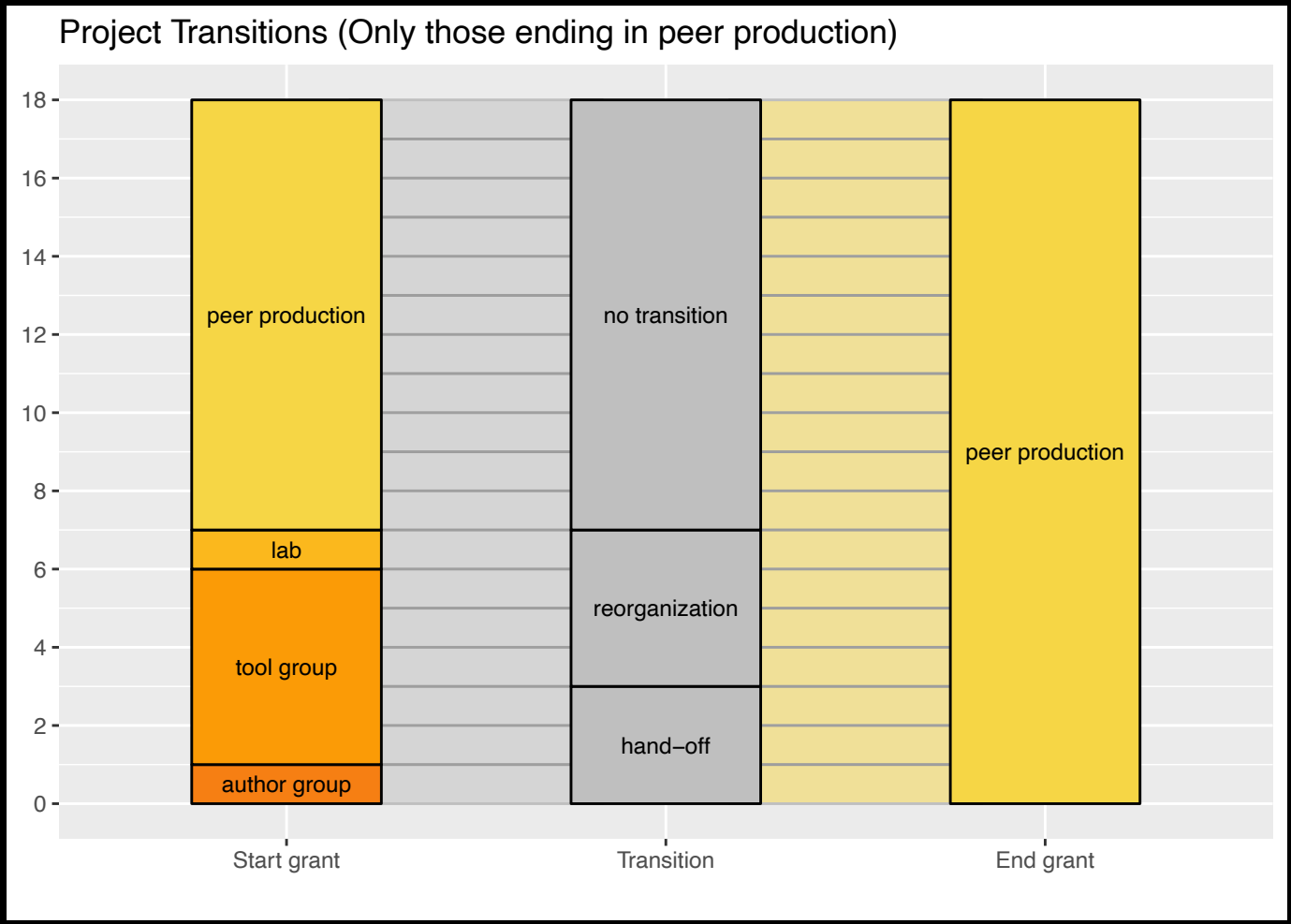
Here report a census of SI2 projects begun by 2014

98 projects funded by 73 grants



Figures produced using ggaluvial: Jason Cory Brunson. 2020. ggalluvial: Layered Grammar for Alluvial Plots. Journal of Open Source Software 5, 49 (2020) https://doi.org/10.21105/joss.02017


Project Transitions
98
80
60
40
20
0
peer production
lab
tool group
author group
business
consortium
no transition
migration
reorganization
hand–off
peer production
lab
tool group
author group
business
consortium
Start grant
Transition
End grant


Project Transitions (Only those ending in peer production)
18
16
14
12
10
8
6
4
2
0
peer production
lab
tool group
author group
no transition
reorganization
hand–off
peer production
Start grant
Transition
End grant

Results – Challenges thematic analysis

- Difficulty in upskilling contributors (long, long mentoring)
- Lack of deep well of possible contributors
  - Strong path dependence in software choice due to academic heritage
- Concern about extractive commits (Egbhal, 2016)

- Alternative route: the long center

Part 4: Ecosystem Complexity

https://www.flickr.com/photos/mrhayata/6933963596

@jameshowison

What work holds a software ecosystem together (if anything)?

- Sensing work
  - knowing how workflows/software “out there” are changing
- Adjustment
  - making appropriate changes to account for changing surroundings
- Synchronization
  - ensuring that changes in multiple components make sense together, avoiding cascades.

Improving Sensing

Fund, but more importantly develop reputational rewards to, **innovation in sensing the scientific software ecosystem:**

- Measure diversity of use contexts, understand how users recombine: Know what generates the work.
  - Go beyond single tool user-studies.
- Overcome concerns about visibility and scientific competition
  - Yes, privacy matters and research can be vulnerable, but users have responsibilities as well.
- Identify local ecosystems
- Increase visibility of software in publications
  - Make specific requests for citation, test their discoverability at http://citeas.org
  - Use machine learning to identify citations https://github.com/howisonlab/softcite-dataset

Improving Adjustment

- Accept that adjustment happens best at the edge.
  - But contributions require mentorship. Fund that mentorship.
- Incentivize projects to be open to gathering and rationalizing outside adjustments.
- Inculcate stewardship orientation within grant funded projects.
- Overcome the “service center” framing
  - Consider a funding program that only funds contributions to “other people’s” projects?

Improving Synchronization

- Projects must not just “be open” but contributing upstream and downstream.
- Fund software distribution work and innovation in distribution
  - Distributions can manage cascades of adjustment work
- Opportunities for research including simulating ecosystem impact of changes
  - If we knew how tools, data and questions were linked we could test possible changes.

Takeaways

- Recombination is a key affordance of software, but leads to complexity
- The work needed to maintain scientific usefulness is hard to incentivize
- Sustainability via transitions to peer production for ongoing sustainability are possible:
  - Funding existing peer production, especially projects competent in their ecosystems
  - Recognizing “the long center” and funding transition as a separate, mentored, project
- Ecosystem complexity makes all this much harder
- Two real options:
  - Suppress recombination (probably antithetical to science)
  - Gain visibility into recombination
- Grant-making seems weaker than markets or open source at supporting visibility (and thus incentivizing needed work)
- Agencies should incentivize and fund:
  - Existing peer production and transitions of “the long center” to peer production as distinct projects
  - Research infrastructure complexity and evolution; including “pushing upstream”
  - Workforce development in peer production competencies and ecosystem competencies

Papers at http://james.howison.name/publications.html#cyberinfrastructure @jameshowison

References

- Graphs produced with the tidyverse software running on Rstudio.
  - Wickham, H., Averick, M., Bryan, J., Chang, W., McGowan, L., François, R., … Yutani, H. (2019). Welcome to the Tidyverse. *Journal of Open Source Software*, *4*(43), 1686. http://doi.org/10.21105/joss.01686
  - Brunson. (2020). ggalluvial: Layered Grammar for Alluvial Plots. *Journal of Open Source Software*. http://citeas.org/cite/ggalluvial
- Evidence that scientific code has a distinct culture from other programming:
  - https://english.stackexchange.com/questions/20455/is-it-wrong-to-use-the-word-codes-in-a-programming-context

Batcheller, A. L. (2011). *Requirements Engineering in Building Climate Science Software.* (Ph.D. Dissertation). University of Michigan. Retrieved from http://deepblue.lib.umich.edu/handle/2027.42/86438

Bietz, M. J., Baumer, E. P., & Lee, C. P. (2010). Synergizing in Cyberinfrastructure Development. *Computer Supported Cooperative Work*, *19*(3-4), 245–281. http://doi.org/10.1007/s10606-010-9114-y

Borgman, C. L., Wallis, J. C., & Mayernik, M. S. (2012). Who's Got the Data? Interdependencies in Science and Technology Collaborations. *Computer Supported Cooperative Work (CSCW)*, *21*(6), 485–523. http://doi.org/10.1007/s10606-012-9169-z

Brown, D. A., Brady, P. R., Dietz, A., Cao, J., Johnson, B., & McNabb, J. (2007). A Case Study on the Use of Workflow Technologies for Scientific Analysis: Gravitational Wave Data Analysis. In I. J. Taylor, E. Deelman, D. B. Gannon, & M. Shields (Eds.), *Workflows for e-Science* (pp. 39–59). London: Springer. Retrieved from http://link.springer.com/chapter/10.1007/978-1-84628-757-2_4

Edwards, P. N. (2010). *A vast machine computer models, climate data, and the politics of global warming*. Cambridge, Mass.: MIT Press. Retrieved from http://site.ebrary.com/id/10424687

Edwards, P. N., Mayernik, M. S., Batcheller, A. L., Bowker, G. C., & Borgman, C. L. (2011). Science friction: Data, metadata, and collaboration. *Social Studies of Science*, *41*(5), 667–690. http://doi.org/10.1177/0306312711413314

Goble, C., De Roure, D., & Bechhofer, S. (2013). Accelerating Scientists' Knowledge Turns. In A. Fred, J. L. G. Dietz, K. Liu, & J. Filipe (Eds.), *Knowledge Discovery, Knowledge Engineering and Knowledge Management* (pp. 3–25). Springer Berlin Heidelberg.

Howison, J., Deelman, E., McLennan, M. J., Silva, R. F. da, & Herbsleb, J. D. (2015). Understanding the scientific software ecosystem and its impact: Current and future measures. *Research Evaluation*, *rvv014*(First published online: July 27, 2015), 17 Pages. http://doi.org/10.1093/reseval/rvv014

Howison, J., & Herbsleb, J. D. (2011). Scientific software production: incentives and collaboration. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work* (pp. 513–522). Hangzhou, China. http://doi.org/10.1145/1958824.1958904

Howison, J., & Herbsleb, J. D. (2013). Incentives and Integration in Scientific Software Production. In *Proceedings of the 2013 Conference on Computer Supported Cooperative Work* (pp. 459–470). New York, NY, USA: ACM. http://doi.org/10.1145/2441776.2441828

Howison, J., & Herbsleb, J. D. (2014). *The sustainability of scientific software production.* Working Paper, University of Texas at Austin.

- Lee, C. P., Bietz, M. J., Derthick, K., & Paine, D. (2012). A Sociotechnical Exploration of Infrastructural Middleware Development. Presented at the CSCW.
- Segal, J. (2009). Software Development Cultures and Cooperation Problems: A Field Study of the Early Stages of Development of Software for a Scientific Community. *Computer Supported Cooperative Work (CSCW)*, *18*(5), -606. http://doi.org/10.1007/s10606-009-9096-9
- Segal, J., & Morris, C. (2008). Developing Scientific Software. *IEEE Software*, *25*(4), 20.
- Baldwin, C. Y., & Clark, K. B. (2000). *Design Rules: The Power of Modularity*. Harvard Business School Press.
- Becker, C., Betz, S., Chitchyan, R., Duboc, L., Easterbrook, S. M., Penzenstadler, B., Seyff, N., & Venters, C. C. (2016). Requirements: The Key to Sustainability. *IEEE Software*, *33*(1), 56–65.
- Colfer, L., & Baldwin, C. Y. (2010). *The Mirroring Hypothesis: Theory, Evidence and Exceptions* (Working Paper No. 10–058). Harvard Business School Finance.
- Lakhani, K., & von Hippel, E. (2003). How open source software works: "free"' user-to-user assistance. *Research Policy*, *32*(6), 923–943.
- MacCormack, A., Rusnak, J., & Baldwin, C. Y. (2006). Exploring the Structure of Complex Software Designs: An Empirical Study of Open Source and Proprietary Code. *Management Science*, *52*(7), 1015–1030. https://doi.org/10.1287/mnsc.1060.0552
- Penzenstadler, B., Bauer, V., Calero, C., & Franch, X. (2012). Sustainability in software engineering: A systematic literature review. *16th International Conference on Evaluation Assessment in Software Engineering (EASE 2012)*, 32–41. https://doi.org/10.1049/ic.2012.0004
- Prlić, A., & Procter, J. B. (2012). Ten Simple Rules for the Open Development of Scientific Software. *PLOS Comput Biol*, *8*(12), e1002802. https://doi.org/10.1371/journal.pcbi.1002802
- Trainer, E. H., Chaihirunkarn, C., Kalyanasundaram, A., & Herbsleb, J. D. (2015). From Personal Tool to Community Resource: What's the Extra Work and Who Will Do It? *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing*, 417–430. https://doi.org/10.1145/2675133.2675172