People in CSE: Incentives and Insight

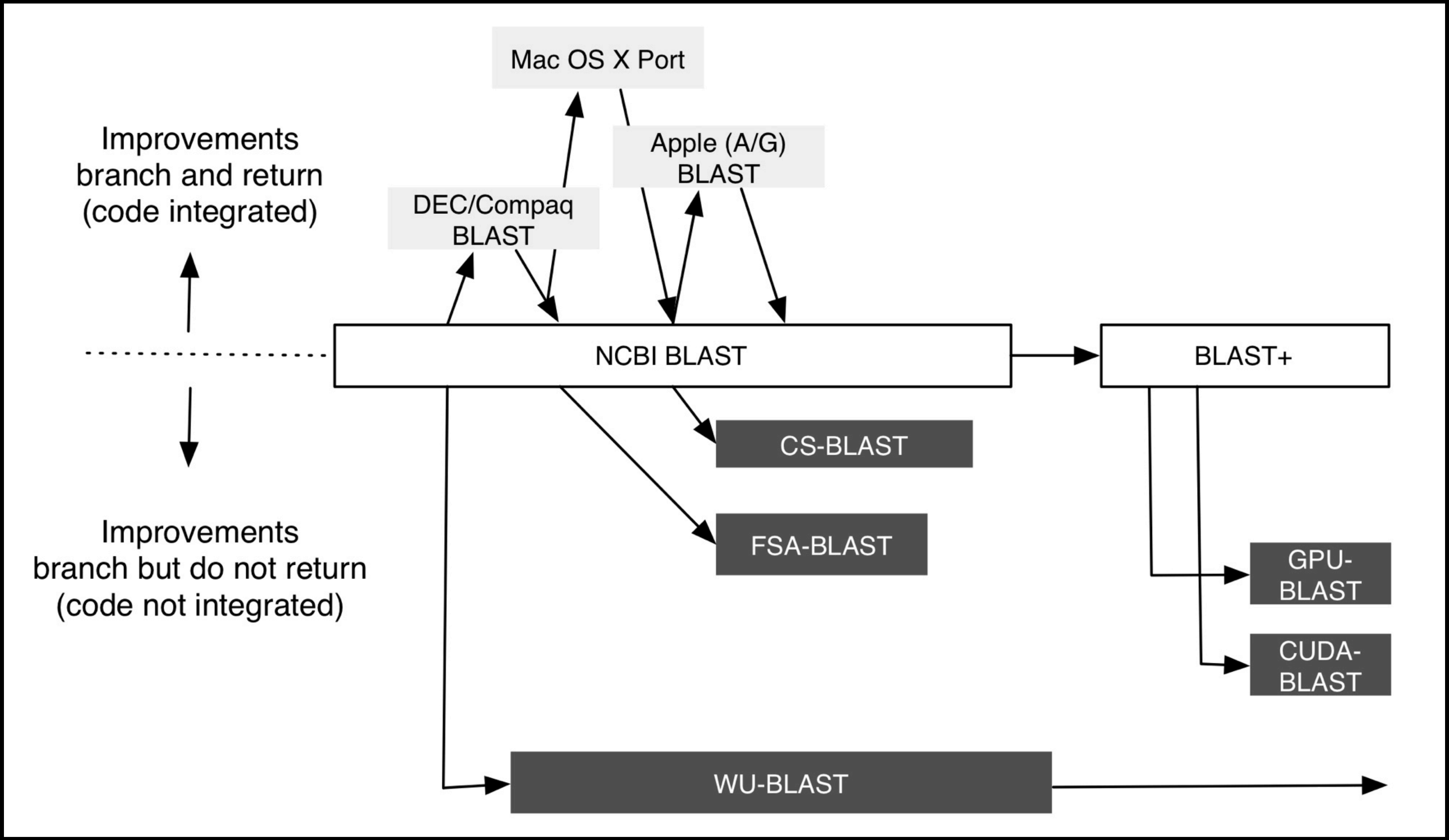
James Howison
School of Information
University of Texas at Austin

18 November 2015

Twitter: @jameshowison
(slides on slideshare, see twitter for link)

@jameshowison

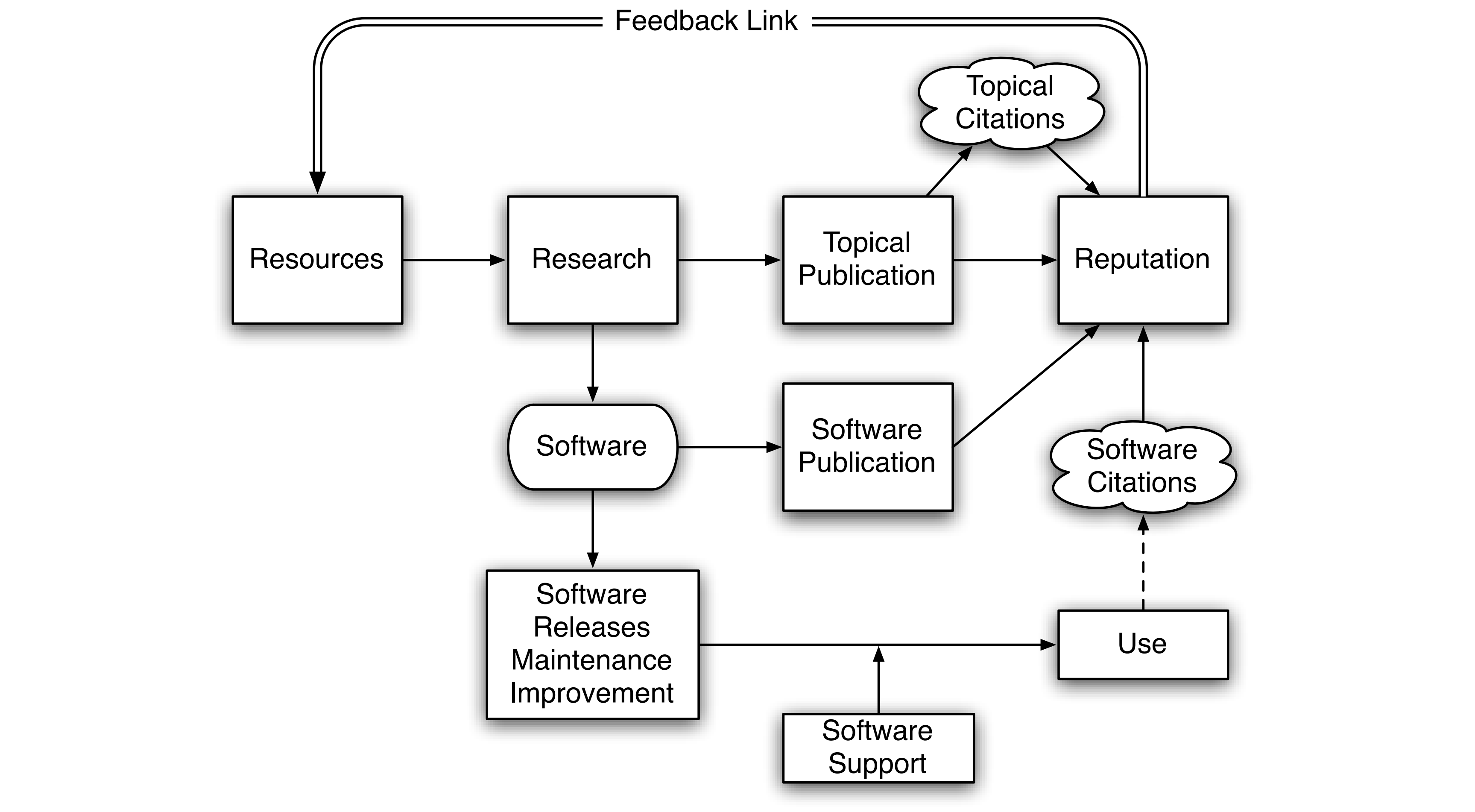
What's different about SE in CSE?


Mac OS X Port
Improvements branch and return (code integrated)
Apple (A/G) BLAST
DEC/Compaq BLAST
NCBI BLAST
BLAST+
CS-BLAST
FSA-BLAST
GPU-BLAST
CUDA-BLAST
Improvements branch but do not return (code not integrated)
WU-BLAST

Howison and Herbsleb (2013) CSCW, Howison (2014) SciSTS

@jameshowison

From Code to Impact

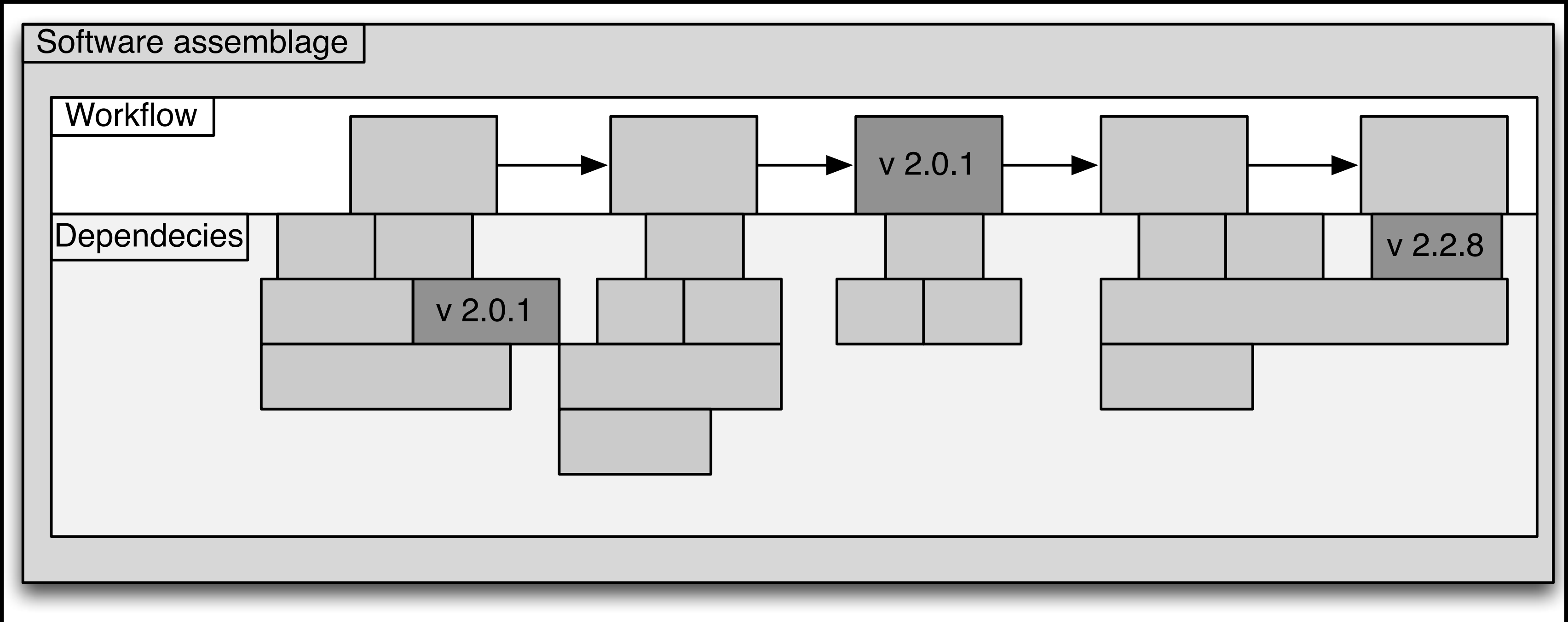


Incentives of your developers

Incentive	Challenge
Money	Keeping it coming; but how long can we do that?
Science	How can I keep my developers engaged with the science outcomes (e.g., end-user talks and conferences, code clearly visible in papers)
Reputation	Does the project founder or lead receive all the reputation? Can we share it around? Should we ask for citations to a paper or to a website and list contributors there?
Use value	How do we avoid after-paper codedumps and move towards small, regular coordinated contributions?

@jameshowison

How is CSE code used?



What insight do we have into how our users really use our code?

Howison and Herbsleb (2011)

@jameshowison

Insight into 2 kinds of dependencies

- Direct dependencies
  - Libraries you call on in your code
  - Who develops these? What motivates them? Have I met them? If not, why not?
  - Does my project push changes “upstream”?
- Complementary dependencies
  - Code our users use with ours in their workflows.
  - How do we know? (Use in CSE very hard to observe)
  - Can we incentivize our users to share this with us?

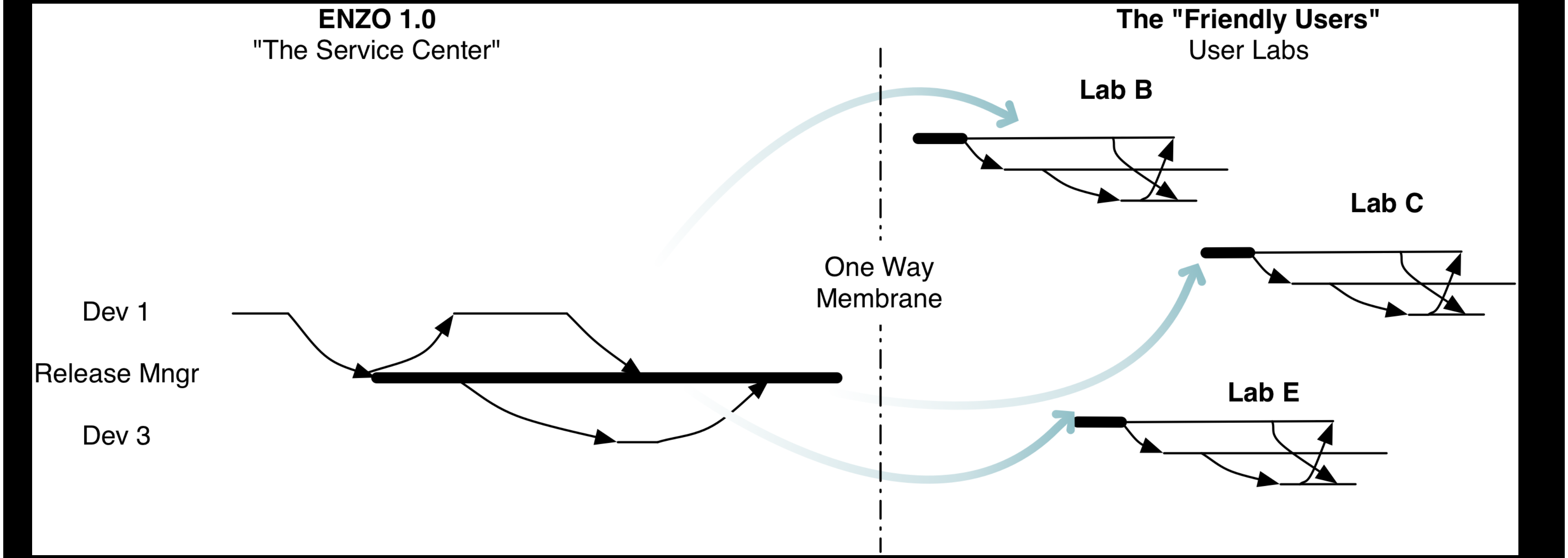
scisoft-net-map.isri.cmu.edu

Depsy.org

@jameshowison

How is CSE code developed?

Segal et al, 2005, 2007, 2009. Lee, Bietz et al, (2010, 2012)



- 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”

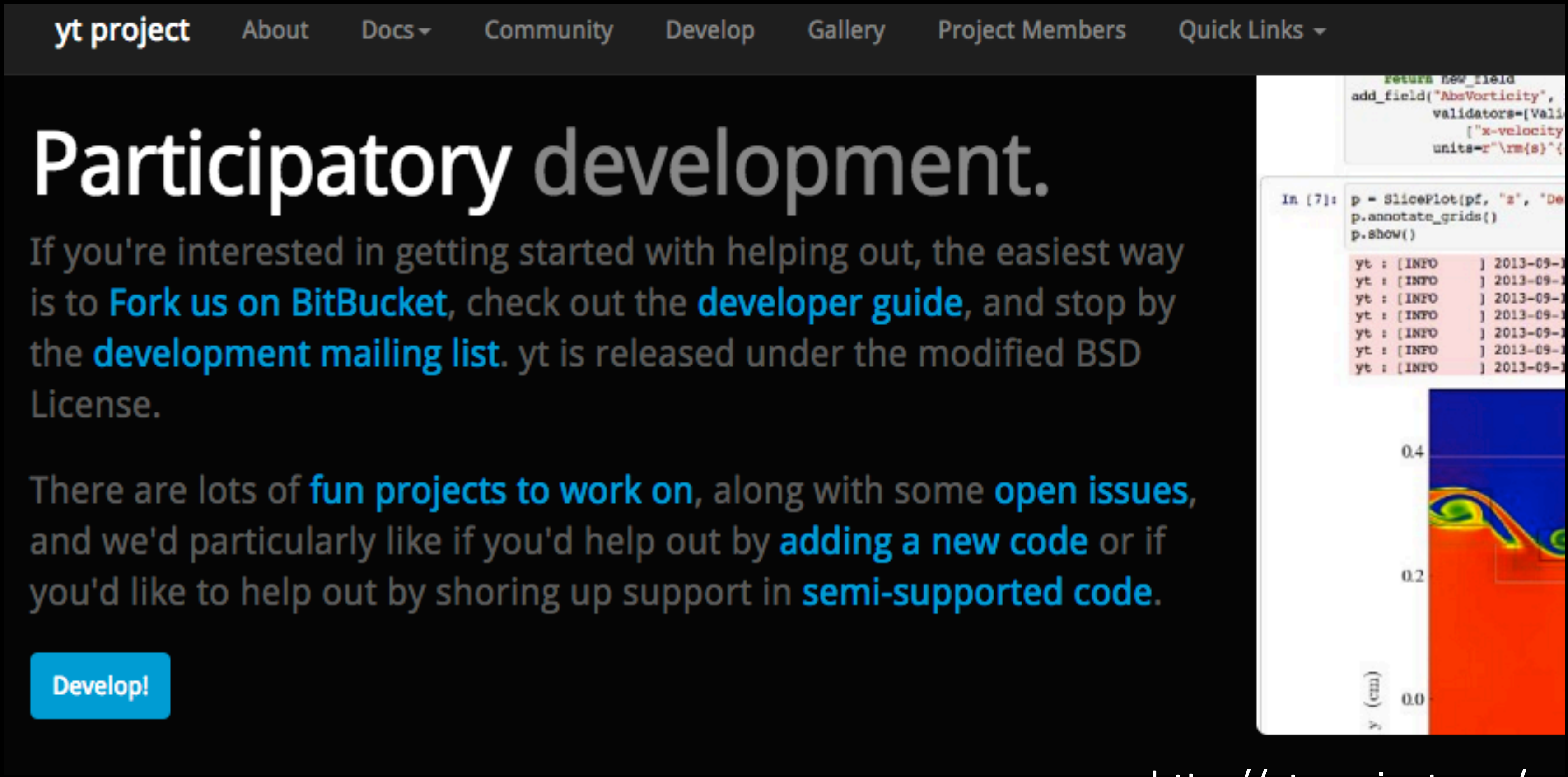
Howison (2015)

Converting users to co-developers

- Sustainability means figuring out:
  - What work will need to be done and who is motivated to do that work?
- Passive user → Active user
  - How can users learn while supporting each other? (Lakhani and von Hippel, 2003)
  - Set up and maintain a cohesive place (list, irc, stackexchange?)
- User → co-Developer
  - Are we inviting contribution and making it clear how to do it?
  - Can we integrate changes forked from the codebase long ago (long user development cycles)

@jameshowison

yt: A good template project



Participatory development.

If you're interested in getting started with helping out, the easiest way is to Fork us on BitBucket, check out the developer guide, and stop by the development mailing list. yt is released under the modified BSD License.

There are lots of fun projects to work on, along with some open issues, and we'd particularly like if you'd help out by adding a new code or if you'd like to help out by shoring up support in semi-supported code.

Develop!

```
        return new_field
add_field("AbsVorticity",
          validators=[Vali
              ["x-velocity
          units=r"\rm{s}^{
```

```
In [7]: p = SlicePlot(pf, "z", "De
        p.annotate_grids()
        p.show()
```

```
yt : [INFO      ] 2013-09-1
yt : [INFO      ] 2013-09-1
yt : [INFO      ] 2013-09-1
yt : [INFO      ] 2013-09-1
yt : [INFO      ] 2013-09-1
yt : [INFO      ] 2013-09-1
yt : [INFO      ] 2013-09-1
```

http://yt-project.org/

First step to community: ask for contributions (Kraut and Resnick, 2012)

@jameshowison

Takeaways

- Incentives:
  - What drives your developers and users?
  - What timescales does their work happen on?
  - How can I convert users to co-developers?
- Insight into users:
  - No sales data, poor tracking of downloads, inconsistent citation practices.
  - How do we see our users workflows/assemblages?
  - Can our code automate collecting use data?

Resources

- Great resources at UK Scientific Software Sustainability Institute
  - http://www.software.ac.uk/resources/get-speed
- Dan's blog:
  - https://danielskatzblog.wordpress.com/
- WSSSPE reports:
  - http://wssspe.researchcomputing.org.uk/
- Buzzing Communities (guide to establishing online communities):
  - http://feverbee.com
- “Team Geek” (O’Reilly book about teamwork in development

Papers

Bietz, M. J., Baumer, E. P. S., & Lee, C. P. (2010). Synergizing in Cyberinfrastructure Development (Vol. 19). Retrieved from http://link.springer.com/10.1007/s10606-010-9114-y

Howison, J. (2014). Sharing the spoils: the trouble with reputation as a motivation and reward for teamwork. Presented at the Science of Team Science, Austin, Texas, USA.

Howison, J. (2015). Sustaining scientific infrastructures: transitioning from grants to peer production (work-in-progress). Presented at the iConference, Irvine, CA. Retrieved from https://www.ideals.illinois.edu/handle/2142/73439

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 ACM Conference on Computer Supported Cooperative Work (pp. 459–470). San Antonio, TX. http://doi.org/10.1145/2441776.2441828

Kraut, R. E., & Resnick, P. (2012). Building Successful Online Communities: Evidence-Based Social Design. The MIT Press.

Lakhani, K., & von Hippel, E. (2003). How open source software works: "free"' user-to-user assistance. Research Policy, 32(6), 923–943.

Lee, C. P., Bietz, M. J., Derthick, K., & Paine, D. (2012). A Sociotechnical Exploration of Infrastructural Middleware Development. Presented at the CSCW.

Segal, J. (2005). When Software Engineers Met Research Scientists: A Case Study. Empirical Software Engineering, 10(4), -536.

Segal, J. (2007). Some Problems of Professional End User Developers. In Proc. IEEE Symp. Visual Languages and Human-Centric Computing (Vlhcc 07) (p. -118).

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