On principles for open scholarly infrastructure for software

Scott Chamberlain

2015-11-11

Geoff Bilder, Jennifer Lin, and Cameron Neylon wrote a blog post in February 2015 titled *Principles for Open Scholarly Infrastructures*. In it, they explore principles for open scholarly infrastructure, describe why we need them, and talk a bit about implementation.

They don't explicitly disuss an organization that primarily makes software, but I thought exploring these principles with respect to software would be a useful exercise. I discuss this with respect to rOpenSci specifically, but try to make broader points about software where possible.

The principles

They discussed three principles, and within each of the three principles they outlined a set of guidelines to follow. What follows is a short synopsis of each guideline, and I address some of the points they brought up under each principle (those that apply more to software).

Governance

> If an infrastructure is successful and becomes critical to the community, we need to ensure it is not co-opted by particular interest groups.

- **Be cross-disciplinary**: rOpenSci started out with a focus on ecology and evolution. Since then we've recognized many pain points across various disciplines that could be greatly ameliorated by good software. We've been accumulating colleagues from a variety of disciplines.
- **Be transparent**: We have strived to be open about our governance, and will continue to do so. The Jupyter and Dat projects are other good examples of this.

Sustainability

> An organisation that is both well meaning and has the right expertise will still not be trusted if it does not have sustainable resources to execute its mission.

- **Mission consistent revenue**: We're lucky in that we've found funding from foundations that share our values. This is critical to our long-term success so that we can continue to fulfill our mission.
- **Revenue based on services, not data**: In software, this means make open source software, which we do - all open licensed, free to use from academic to non-profit to industry. Instead of value-added services, we offer assurance of quality and long-term maintenance, both hard to do in academic software given the short-term nature of grants.

Insurance

> Long term trust requires the community to believe it retains control.

- Liberally licensed open source software goes a long way towards making sure that the software is owned by the community (even if proprietary derivatives are made).
    - **Open source**: Everything we do is open source. This is surprisingly not ubiquitous in academic software, due to a combination of some not bothering to add a license (essentially meaning it can't be re-used anywhere), or using a very restrictive license, either out of not considering the consequences or on purpose for competitive or monetary gain.
    - **Open data**: We're generally not a data provider, but if/when we do, we will provide free to use data. As an example, we do maintain a web API for the Fishbase dataset at ropensci/fishbaseapi.

rOpenSci as infrastructure?

> What would an organisation actually look like if run on these principles?

They list ORCID and CERN as examples. They didn't mention software per se. Can an organization that makes software be considered infrastructure? And if not, how do these principles apply to software, if at all?

Can we consider rOpenSci as infrastructure? I think we can in a sense. We are building a long-lasting framework for building and maintaining software, supporting the people that make the software, and maintaining the underlying tools to make it all run smoothly.

We provide a number of services that places us more into the role of infrastructure:

- Code review (see our onboarding process)
- Community standards for quality software (see our development guidelines and policies)
- General purpose software that addresses a wide variety of general use cases
- Openly licensed software that can be used in all settings (almost all our software is MIT licensed)
- Low level clients for many data science tools

Bilder et al. state:

> We have not addressed the question of how the community can determine when a service has become important enough to be regarded as infrastructure …

We can not determine whether rOpenSci is *important enough* - perhaps that will become apparent when it happens.

In some ways, rOpenSci could be considered to not be infrastructure. For example, we are specific to a single programming language. However, this could change in time as our focus changes. In addition, R is used widely across disciplines, making our community of users and contributors quite diverse.

Centralization

> Many of the consequences of these principles are obvious. One which is less obvious is that the need for forkability implies centralization of control. … Centralization can be hugely advantageous though – a single point of failure can also mean there is a single point for repair.

This is a key point about the rOpenSci model. We do centralize software of a particular type that fits our mission. Because we centralize in terms of physical location in our GitHub account, we can:

- Jump in to fix a problem quickly in case one of our community maintainers is away or unable to respond in a timely manner. This is possible because we have admin rights on all repositories.
- Maintain a certain level of software quality. Although all our repositories may not be at the highest level of quality now, we are trying hard to get there. Our review system is helping with this.
- Enforce a community that is welcoming to all, that encourages less experienced contributors, and marginalized groups. This is also done in our events.

We try to balance nurturing novice or inexperienced developers with . . . .(NOT DONE WITH THE THOUGHT YET)

References

Bilder G, Lin J, Neylon C (2015) Principles for Open Scholarly Infrastructure-v1, retrieved 2015-10-15, http://dx.doi.org/10.6084/m9.figshare.1314859