

Conceptual Mismatches - Final Report

Mel Chua and Stephen Jacobs (RIT)

Executive Summary:

Our project investigated two questions:

1. What kinds of conceptions might be variously held by FOSS community members about well-maintained and sustainable digital FOSS infrastructure?
2. What tensions exist between the implicit/explicit contracts, expectations, and assumptions that may be behind these conceptions of maintenance and sustainability?

To investigate this, we looked at PyPI (Python Package Index) as a case study. PyPI's narrative is not uncommon in the FOSS world; it began as an individual's personal project and then inadvertently became development infrastructure - in this case, for millions of people. The project struggled with sustainability for years, but had recently run a successful effort to get itself into a far more maintainable state. We felt this sometimes-rocky journey towards sustainability would make PyPI an excellent case to study.

We did a series of three open, round-robin, qualitative interviews with three maintainers and three users of PyPI, the Python Package Index. In keeping with the FOSS ethos, we took a radically transparent research approach. Open-licensed interview protocols, transcripts, memos, and other artifacts (including this report) can be found in our [github repository](#).

We had originally set out to look at mismatched conceptualizations of sustainability and well-maintainedness between upstream maintainers and downstream users, and how those mismatches interacted to affect a digital FOSS infrastructure project's sustainability. As we progressed, we found that the mismatches were not between groups, but *within digital FOSS infrastructure culture itself*. These emergent insights shifted us to the two research questions listed above.

In terms of the first question on conceptions, we found that our PyPI maintainers and users all primarily conceptualized maintenance and sustainability in terms of technical capacity, or the ability to continue to grow and modify the code base. Other types of capacity (financial, legal, project management, etc.) were largely noticed when they blocked the scaling of technical capacity. However, redirecting resources toward building non-technical capacity was interpreted as *stalling* project progress, because it reduced the effort put into code development. When this happens, maintainers may attempt to steer resources back toward technical capacity, which continues the cycle.

In terms of the second question on tensions, we found that the mismatch wasn't that users thought one thing about "sustainability" and "maintenance," and maintainers thought another. Rather, all participants had similar expectations of how digital FOSS infrastructure should be run, but those shared implicit contracts conflicted in under examined ways. Additionally, since the community had prioritized technical capacity over other kinds (as previously discussed), they had limited resources for navigating these tensions and the frustrations they engendered.

These results suggest that different strategies are needed for addressing the non-technical capacities of digital FOSS infrastructure projects. While capacity for things such as documentation, outreach, project management, design, legal work, etc. are often acknowledged as needs within FOSS communities at-large, they are rarely addressed proactively. Rather, they are often addressed only when a project is in crisis. If additional project communities are investigated, further work should be done on successful non-technical capacity building efforts and how they might be replicated. In addition to expanding the pilot study to look at whether these results generalize across projects with different histories and resource situations, future work might include an investigation of what we call "infrastructure privilege," which is explained in the "directions for future research" section of this document.

Research questions

Original research question

How do mismatched conceptualizations of sustainability and well-maintainedness between upstream maintainers and downstream users of a FOSS digital infrastructure project interact to affect that project's community health and thus sustainability?

The research question evolves

As interviews progressed, we learned that (at least in the case of our pilot site, PyPI) users and maintainers largely had shared definitions and understandings of what a sustainable and well-maintained project would entail. Although some factors may not have been front-of-mind for participants, the stories of others fit into their existing schema once they read them. The mismatches were not about misunderstandings between the two groups; they were elsewhere.

Instead, we learned that there were a number of expectations and cultural tensions limiting how users and maintainers could work towards project sustainability together. This caused us to revise our research questions.

Revised research questions

1. What kinds of conceptions might be variously held by FOSS community members about well-maintained and sustainable digital FOSS infrastructure?
2. What tensions exist between the implicit/explicit contracts, expectations, and assumptions that may be behind these conceptions of maintenance and sustainability?

Method

Radically transparent research artifacts

In keeping with the FOSS ethos, we used a radically transparent research approach (developed by Chua) for applying FOSS philosophies, processes, and tools to qualitative research. As part of this, the research project design and interim artifacts, including interview protocol, consent forms, analytical memos etc. are available in our [github repository](#).

Pilot site - PyPI, a project from the Python community

We chose PyPI, the Python Package Index, as the pilot site for our first case study. Python is one of the most popular programming languages in the world (as of 2019), and PyPI is the package index that helps people find and install software for Python development. As with many FOSS projects, PyPI began as an individual's personal project and then inadvertently became development infrastructure - in this case, for millions of people. While not all digital FOSS infrastructure projects are this way, and this narrative is not unique to digital FOSS infrastructure projects, it is a common narrative among them.

As the initial PyPI project was not intended to support millions of programmers, neither the initial code nor the founding maintainer was prepared to scale. Although a few other maintainers joined the project, the increased demand caused the team to struggle with outages and updates until a recent rewrite specifically aimed at addressing the project's sustainability concerns. This rewrite, called Warehouse, was launched as production PyPI in 2018. A grant from the Mozilla Foundation funded contract work from a few Python community members that significantly contributed towards Warehouse's completion. Another significant factor was the lead maintainer finding full-time employment that allowed work time to be dedicated towards PyPI development.

Interview subjects selected

One of our team members (Chua) travelled to PyCon North America 2019 to discuss this project with Python community and recruit our study participants. The six narrators consented to being publicly identified as part of our radically transparent research approach: they are Nick Coghlan, Donald Stufft, and Ernest Durbin III (PyPI maintainers) and Naomi Ceder, Terri Oda, and Jackie Kazil (PyPI users). We are grateful for their generosity in sharing their time and stories.

Participants in the "user" group were purposefully selected for their extensive experience as developers and maintainers of other software projects, as well as their deep involvement in the Python community (serving on the board, running conferences, etc.). This was done so that any perspective differences regarding PyPI would be more likely to be the result of upstream/downstream roles with respect to *that specific project*, as opposed to differences in skill, experience level, etc. between users and maintainers.

Data collection process

We conducted narrative interviews with a publicly viewable dialogue structured over multiple rounds. The first interview round for each participant asked them to tell their version of the Warehouse rewrite story, explain their definitions of sustainability and well-maintainedness, and comment on their perceptions of PyPI's sustainability over time. Subsequent interview rounds involved participants seeing excerpts from each other's Warehouse stories and then commenting on them and/or re-telling theirs. Five participants completed the full course of three interviews; the sixth was only able to complete one interview before life intervened.

All interview transcripts were reviewed by participants and then open-licensed (CC-BY) and posted publicly with their consent. Stories were shared both within-group (maintainers-to-maintainers, users-to-users) and across groups (maintainers-to-users and vice versa) We also discussed our ongoing analysis and theorizing with participants, who contributed ideas of their own.

Analysis process

The two team leads met in Atlanta in September 2019 to review the preliminary narrative analysis and do an initial round of thematic coding. This occurred between the second and third interview rounds. In parallel with the third and final round of interviews, external consultants

with Python community expertise (Sumana Harihareswara and Shauna Gordon-McKeon) were brought to Rochester, NY in November 2019 for a weekend-long session to complete and triangulate thematic coding and begin axial coding. Axial coding continued remotely thereafter.

Writing process

The team leads attended the January 2020 writing retreat with some of the other Ford/Sloan FOSS Digital Infrastructure grantees, which helped us refine our ideas and connect them to other projects in the cohort. We then reconvened in Rochester in February to build an initial report draft. This final report was completed in March 2020 while COVID-19 forced us (and the rest of the world) to work remotely.

A note of thanks regarding accommodations

One of our team leads is Deaf. While both team leads are fluent in American Sign Language (ASL), our study participants and research cohort members are not. Consequently, we relied on ASL interpreting for larger research meetings and data collection events as well as real-time stenographic captioning (RTSC, also known as CART) for remote interviews.

Support for these accommodations was provided outside of the formal award and exempted from institutional overhead. This is a significant and necessary component of creating equitable opportunities for disabled researchers. Providing separate resources for access allows research funds to be entirely devoted towards research, which in turn avoids penalizing the research budgets of disabled scholars simply because they have access needs.

The research team is deeply thankful for this approach, which made a world of difference to our ability to conduct this case study. We are hopeful that it will become the rule, rather than the exception, for accessibility in institutions across the board.

Preliminary results

Question 1: What kinds of conceptions might be variously held by FOSS community members about well-maintained and sustainable digital FOSS infrastructure?

Digital FOSS infrastructure projects require multiple kinds of capacity for development and maintenance (technical, financial, legal, project management, etc). However, for PyPI, both user and maintainer conceptions of well-maintained and sustainable digital FOSS infrastructure may implicitly focus *only* on technical capacity (the ability to continue to grow and modify the code base) as indicators of maintenance and sustainability. Non-technical capacities were generally observed and/or discussed primarily in terms of their impact on technical capacity.

A few illustrative examples from our interviews are below (more examples available in this [memo on “aggregated definitions”](#)). Note that these examples describe both technical and

non-technical things, but are all *oriented towards* developing technical capacity.

1. A well-maintained project has people to make sure a project is keeping up with changes in the ecosystem (ex: maintaining compatibility when dependencies are updated, etc.)
2. A well-maintained project is following best practices regarding software development (tools, code style, tests, etc.)
3. A well-maintained project cares about backwards compatibility.
4. A sustainable project has new people onboarding [primarily for technical development].
5. A sustainable project has clear entry points for new contributors so they know how to get started [with technical development].

Our PyPI participants' focus on technical capacity stands in contrast to non-digital and/or non-FOSS infrastructure projects, which may be part of organizations that monitor and maintain a broader range of capacities. Since technical progress was the primary thing being monitored for PyPI development, it was also the main thing that was evaluated and resourced proactively. Other types of capacity (project management, documentation, design, etc.) lagged behind *until* they block technical development, at which point they were resourced reactively.

PyPI is a good example of some dynamics common to many digital FOSS infrastructure projects. For instance, internal and external messaging around progress may focus on technical development and technical development capacity, hiding gaps in other areas. Although building these other kinds of capacity also takes considerable skill and effort and also constitutes progress, a digital FOSS infrastructure project that primarily measures progress in terms of technical development may see a major shift toward building out these capacities as the *whole project* “stalling,” since *technical* progress is slowed or stopped. In response, maintainers may continue trying to steer their time and energy back toward technical aspects of development and maintenance, which continues the cycle.

Question 2: What tensions exist between the implicit/explicit contracts, expectations, and assumptions that may be behind these conceptions of maintenance and sustainability?

Different people bring different expectations to digital FOSS infrastructure. These expectations may come from various technical development cultures: FOSS projects, infrastructure projects (both digital and non-digital), corporate or government requirements, and so forth. When these expectations are assumed to be shared rather than explicitly discussed and agreed-upon, they can turn into implicit contracts. Each implicit contract is reasonable on its own, but can unknowingly come into tension with other expectations in play in digital FOSS infrastructure spaces, as they did in our PyPI case study.

A few illustrative examples of tensions from our PyPI interviews are below (more examples available in this [memo on “tensions”](#)):

Example of tension: attention management	
Statement A	Statement B
Users should be able to maintain an uninterrupted focus on their own work without being forced to read information from someone else's project. • Autonomy of contributors (FOSS culture) • Infrastructure should be a "black box" kept out of sight and out of mind (Infrastructure culture)	Users should have an ongoing awareness of what is happening "behind the scenes" with maintenance of the FOSS digital infrastructure they rely on. • Transparency (FOSS culture)

Example of tension: degree of cultural intervention	
Statement A	Statement B
Project culture and tools should emerge from the preferences and habits of existing maintainers. • Autonomy of contributors (FOSS culture) • Decisions about actions are made by those who carry out those actions (FOSS culture)	Leaders should intervene and set project guidelines that make it easier for new contributors to join. • Explicitness (FOSS culture) • Inclusion (FOSS culture)

Note that a single person might think statements A and B are both true, and not realize there is an underlying conflict between them. The mismatch here is not that users understand sustainability one way, and maintainers understand it in a different way. Rather, both groups are grappling with mismatches between different development cultures that have influenced the digital FOSS infrastructure space. These internal cultural mismatches can be compared to a child entering a bilingual space for the first time after growing up in a household that uses only one of their languages and attending a school that only uses the other. Multiple notions of "how things should work" suddenly collide and need to be made explicit.

When unspoken cultural expectations lead to the creation of implicit contracts, there are often no well-defined processes for what to do when these contracts are violated. Although users and maintainers are grappling with the same mismatches, they experience the consequences of these conflicting implicit contracts in different ways. Regardless of the specific consequences at play, one of the first manifestations of violated expectations is frustration. These tensions are not fundamentally bad, but when they are not acknowledged and navigated intentionally within the community, the resulting frustrations and communication breakdowns can lead to decreased

productivity (technical and otherwise) and even resignations from the project and a distancing from its community.

Conclusion

The findings for each of the two main research questions interact with each other. When implicit contracts collide, capacities in multiple domains (as well as time and energy) are needed to address those tensions in productive ways. However, if a project's focus has been on technical development capacity, the community may not have sufficiently developed the other capacities required to address them.

In the case of PyPI, the technical development needed to sustain the project could not move forward until other capacities were brought up to speed. In other words, the lack of non-technical capacities was a limiting factor that could not be overcome by simply adding more technical capacity. For instance, the Python Software Foundation (PSF's) capacity to financially and logistically manage grants had to develop sufficiently before they could receive a grant from the Mozilla Foundation. In turn, the grant allowed the PSF to hire project management and design capacity in the form of contract work.

The challenges and potential solutions here have to do with culture change and capacity-building, which are no easy task. These challenges are not unique to PyPI, or even to FOSS. The software industry has many well publicized failures of not supporting non-technical capacity development, including a paucity of diversity and a devaluing of non-technical work.

Some places are specifically developing training programs for non-technical FOSS contribution types where capacity is generally scarce. They include the Linux Foundation's class on FOSS Software Management, Brandeis University's certificate in Open Source Technology management, and the Linux Professional Institute's development of a certification in the Business of Open Source Software. These are only pieces of the puzzle; design, law, finance, translation, and other areas continue to be underserved.

We hope our research will spur communities to reconsider how they conceptualize and evaluate the many interacting components of infrastructure sustainability. We also hope that those who resource digital FOSS infrastructure maintenance (foundations, private industry, etc.) will consider the kinds of resources they provide, because additional technical capacity will sit unused until the other capacities are sufficient to match it. There are existing models for outside organizations to contribute technical capacity to a digital FOSS infrastructure project: for instance, donating computing resources, or allowing a maintainer to do FOSS work as part of their day job. There are fewer models for contributing non-technical resources without simply making a financial donation; these, too, can be developed. Each is one of many pieces of a puzzle.

Directions for future research

Extending the pilot study to additional project communities:

Thus far, these results are based on a small, pilot-sized study of one project (PyPI) within one community (Python). We chose PyPI as a case study of an individual developer's project that struggled with sustainably supporting unexpected large-scale adoption.

We'd like to follow the same methodology with two additional project communities so we can compare different "origin stories" and different arrangements for supporting various types of capacity development:

1. An individual's project that scaled more smoothly into a sustainable pathway for maintenance that balanced technical and non-technical capacity development. Btrfs (a filesystem project) is a potential contender here; it has been incorporated into the Linux kernel, and its lead developer and maintainer has been paid as a full-time employee by both Oracle and Facebook to devote their full time to the project.
2. A project that began with the intention to work at scale from the start, ideally by a team who foresaw the need for project skills and infrastructure beyond the technical aspects of development. We currently do not have a case in mind here, but would look for one that met these criteria.

Investigating the effects of "infrastructure privilege" on digital FOSS infrastructure projects

One concept that emerged during our work is something we are currently calling "infrastructure privilege" - the ability to take certain things for granted because one or more infrastructure systems meets one's needs without constant intervention.

The mismatches and tensions we found during this pilot study have complex cultural and historical roots. As is often the case with digital FOSS infrastructure projects, the original developer and the team that grew around him had deep technical backgrounds. They also often had multiple kinds of privilege (white, straight, male, non-disabled, cisgender, financially stable, formally educated, living in the global north, etc.). Privileges of this sort can manifest in terms of relationships to infrastructure: are basic food, water, shelter, healthcare, etc. needs covered by an infrastructure system that is taken for granted? Is access to technical infrastructure (power, connectivity, etc.) a given? If so, a person has infrastructure privilege.

Lack of infrastructure privilege can manifest in various ways. For instance, a contributor might live in a country where continuous electricity and internet are not available. They may have challenges in regularly accessing reliable power or computing devices, or be dependent on paratransit that unpredictably runs late. While infrastructure reliability issues can occur at some

point to any digital FOSS infrastructure contributor, it is assumed in general that these are rare, temporary, and potentially voluntary (i.e. the decision to raise a child, go back to school, etc.) and/or fixable if they are unwanted. Our group of study participants and researchers had a variety of relationships to infrastructure privilege, which enabled us to spot and articulate it to one another as an influencing factor in digital FOSS infrastructure sustainability.

The concept of infrastructure privilege relates to existing work on diversity and equity within FOSS communities, including work being done by the current research cohort. However, we wish to focus here on the impact that infrastructure privilege has had on how mainstream FOSS culture understands the concept of "infrastructure." Essentially, the notion of "infrastructure" within digital FOSS infrastructure projects has largely been defined by *people with high degrees of infrastructure privilege*. Everyone else is expected to hew to this notion of "what infrastructure is" regardless of their relationship to infrastructure privilege. In turn, this affects who is able to take up key leadership roles, as well as the range of allowable ways to inhabit those roles.

We brought these emergent thoughts to the January 2020 NYU writing retreat. The concept (phrased as "infrastructure as a privilege") seemed to resonate with other members of our research cohort and warrant further investigation. We would be interested in collaborating with other cohort members in investigating infrastructure privilege within digital FOSS infrastructure projects. Potential research questions include:

- What is the nature of the distribution of infrastructure privilege across various types and levels of contributor and leadership roles in digital FOSS infrastructure projects?
- How do various types of infrastructure privilege among maintainers correlate with sustainability-related outcomes for digital FOSS infrastructure projects?
- How does the concept of infrastructure privilege challenge the narrative of "meritocracy" in FOSS?
- ...and others.

Appendix: Dissemination of initial case study results

Moving forward, we expect to take on a variety of efforts to point the Critical Digital Infrastructure community to the case study and what has emerged from it.

Industry Conferences, Meetings and Periodicals:

- **Accepted:**
 - “Mismatches and Tensions: What PyPI Can Teach Us About Open Source Infrastructure Sustainability,” PYCON, April 15-23, 2020, Pittsburgh, Pennsylvania.
 - Physical conference canceled due to COVID-19. Conference will be offering previously accepted presentations for distribution via YouTube. Details pending of this writing but we intend to submit a presentation.
- **Rejected:**
 - “Sustainability Tensions in Open Source Digital Infrastructure Projects: Lessons From PyPI,” O'Reilly Open Source Software Conference, July 13-16, 2020 in Portland, OR.
- **Planning to Submit:**
 - **Conferences** ((all dependent on COVID-19)
 - ToDo Group Quarterly Meeting, 7/20
 - Sustain Open Source 2021
 - FOSDEM 2020
 - **Industry Publications** (timing dependent on Ford/Sloan press embargos and approvals)
 - OpenSource.Com

Academic Conferences and Journals such as:

- ACM Conference on Human Factors in Computing Systems 2021
- ACM Conference on Computer-Supported Cooperative Work & Social Computing 2021
- Journal of Systems and Software
- Journal of the Association for Information Systems
- Issues in Informing Science and Information Technology Public Culture