EXHIBIT 24

GitHub HQ (SF) | June 19, 2017

Sustain

A ONE DAY CONVERSATION

for Open Source Software sustainers

THE REPORT

Sustain

The Report

On June 19th, 2017, one hundred people gathered in San Francisco to create a cultural shift in how we think about the sustainability of open source software.

This document contains a synthesis of the sessions that were held, the conversations that were shared, the resources that were compiled and the working groups that were formed that you can join today. It also presents key recommendations and challenges for actors present in the development of open source software, including:

- **maintainers** who manage contributions and take responsibility for the quality and governance of their projects,
- **contributors** who contribute their time and experience to a project and
- **consumers** or **users** who utilize a project to achieve their own goals.

To these we would add a fourth actor: the ***sustainer***—the individual or organization who is concerned with the fragile state and future of highly-used and impactful open source projects.

Contents

- Context
- Key recommendations
  - Create sustainable communities
  - Free the maintainer
  - Optimise project governance for contribution and retention
  - Raise the value of non-code contributions
  - Commit to the commit
  - Use money as an incentive for open source
  - Recognise, value and invest in open source software
  - Lower barriers for open source projects to manage finances
  - Prepare maintainers for a sustainable future
- Session overviews and themes
  - What do we mean by sustainability?
  - The importance of good governance
  - Building diverse communities
  - Organisational structures
  - Managing financial contributions
  - Creating sustainable incomes
  - Supported Attendees
  - Conclusions and next steps
- Resources
  - Working groups
  - Foundations offering organisational support
  - Acknowledgments

Context

Most software today relies on a foundation of freely available and permissively licensed code—a foundation that is beginning to show signs of stress. The communities of people who maintain the open source projects underpinning today's technology are beginning to feel the weight of increased demand by consumers.

For the past five years, Free and Open Source Software (FOSS) has seen a near-exponential increase in projects exploring technical and commercial niches. This growth can be attributed to the success of the distributed revision control management software Git and GitHub, a commercial platform for hosting and working on Git-based software projects. The financial and cognitive cost of starting a project built on FOSS is now so low that we have seen the advent of seed-, micro- and accelerator-based investment models only possible due to the considerable value provided by Open Source Software (OSS).

The proliferation of projects solving increasingly niche problems has led to less potential reputational return for individuals—a key factor in attracting contributors to an open source project. We also see evidence of a shift from the egalitarian relationship between users and contributors once enjoyed to one in which too few contributors are overwhelmed by demands from developers more akin to a commercial consumer than a peer.

Key pieces of OSS, of which many applications depend on, are often supported by small groups of individual contributors with no financial support or contractual obligation to do so. This has created a landscape in which the goodwill of a few can no longer sustain the increasing demands the ecosystem places on them.

Our collective challenge is to support those working on essential digital infrastructure as a public good. Commercial organizations are beginning to see the value in giving back to the community, but no individual company or organization is incentivized to address the public good problem alone. In order to support our digital infrastructure, we must find ways to work together.

Nadia Eghbal's *Roads and Bridges: The Unseen Labor Behind Our Digital Infrastructure*, offers a concerned view of the open source ecosystem. Commissioned by the Ford Foundation, the study, published in July 2016, provides a comprehensive overview of how FOSS is developed, maintained and utilized. However, it stops short of providing a roadmap for solving the cultural, financial and institutional issues discussed, which, one year later, is becoming the focus of the conversation among open source project maintainers.

This conversation was and continues to be the focus of Sustain.

Key recommendations

Based on the contributions of the participants, we make the following key recommendations.

These recommendations draw on the generosity of all attendees and do not reflect the views of any particular participant, participating organization, host or sponsor. They offer practical steps that stakeholders engaged at various points in the production, consumption, governance and assessment of OSS can take to address immediate challenges and opportunities.

The overarching theme that emerged from all the conversations at Sustain is that **when we talk about sustainability, we are talking both and equally about the sustainability of resources and the sustainability of its people**.

Create sustainable communities

The community of contributors and maintainers is central to the ability of an open source project to serve its users. For an open source project to become a healthy environment that doesn't lead to maintainer burnout or become a burden for a very small group of people, it must put the needs of these communities first.

Free the maintainer

Many users of OSS rarely consider the needs of the community behind a project. This is an understandable and somewhat forgivable offence given that the norms established around software development and licensing have put the needs of users first. We should work to establish a new social norm that frees maintainers from the enormous weight of the responsibility that they feel to sustain a project so many users rely on. Developing and releasing software in the world does not bind them to it forever. Maintainers should have the freedom to choose how they will engage with contributors, whom they chose to engage with and, most importantly, when they will engage with a project themselves.

Optimize project governance for contribution and retention

Reciprocally, maintainers should acknowledge that they often design practices and processes that put themselves at the center of a project. Project maintainers often react to the influx of activity on a successful project by aligning practices to suit their needs. As a consequence, they raise the barrier to entry for new or novice contributors in order to reduce the workload of existing maintainers. While this makes sense from the perspective of optimizing for less maintainer workload, maintainers must recognize that centralization of responsibility results in more pressure on fewer people, leading to fatigue and eventual burnout. Instead we need to create and promote best practices projects that support new contributors and create a path for them to take a greater part in the decision making and direction of the project over time.

Raise the value of non-code contributions

The production of OSS deals almost exclusively with writing code. Consequently, the engineer is often the sole focus of a project and code contributions are valued more than other types of contributions (documentation, community guidance, etc.). Engineers therefore become the decision makers. They build the structures, processes and norms around the production of code which can neglect the needs of other types of contributors. Recognizing and elevating the contribution of all participants in a project will legitimize their concerns, raise their profile, leverage the same motivations that lead to developers to contribute and create stronger software that is the product of a more diverse range of perspectives and skills.

Commit to the commit

Open source contributions are often made on the basis of immediate and individual needs. Little thought is given to the long term maintenance these contributions require. Contributors should recognize this and accept a degree of responsibility for the maintenance of their contributions as long as they are in use. In exchange, maintainers should provide a clear way for contributors to accept this responsibility and engage with the community to do so.

Use money as an incentive for open source

There is a polarizing reaction to money within an open source project, specifically with regards to paying for a contributor's time. This perception is damaging to projects that lack some of the other traits that motivate contributors to engage with a project, including many mature, well-used, 'infrastructure-type' projects. Removing the cultural aversion to money in open source can enable code contributors to keep building software while incentivizing others to take on other equally important but less implicitly rewarding tasks like resolving issues and bug triaging. By sharing guidance on how to manage money as an incentive, we can provide a stable foundation of support while enabling contributors and maintainers to continue to build value within projects.

Recognize, value and invest in OSS

The support networks that currently exist for open source are severely limited because companies do not contribute financially to projects. Many companies fail to recognize the value that building upon open source software adds to their bottom line. And those who do acknowledge the value open source brings to their products find it very difficult to account for that value and are unable to justify any meaningful investment in it. Projects work around company policies to create ways in which support can be provided. Largely these come from marketing activities: event or site sponsorships that may distract from the goals of a project itself, or hiring engineers to work on those projects, which often brings conflicts when prioritizing milestones. The open source ecosystem will not be sustained for the future unless projects,

supporting organizations and companies work together to create a sustainable, influence-free source of finance to maintain their shared digital infrastructure.

Lower barriers for open source projects to manage finances

Open source communities are organic, disperse and loosely associated groups of contributors and maintainers that do not align with any territorial, legal or financial boundary. This means that even when projects have the financial support and governance processes necessary to pay for contributor's time, they must still navigate barriers around corporate structure, banking and employment law. We need to create institutions and organizations that will abstract and insulate projects from these burdens, allowing projects to accumulate and distribute funds as they see fit.

Prepare maintainers for a sustainable future

We must prepare maintainers today for the future that we wish to see tomorrow. They will need to model incomes, costs and make appropriate budgetary decisions. They will need to negotiate agreements on behalf of their community. They may even need to defend their projects legally. We must prepare maintainers for the business of open source.

Session overviews and themes

What do we mean by sustainability?

Sustain began by asking itself the simple, context-setting question: What do we mean by sustainability?

While several themes emerged during the sessions, two distinct problems always surfaced when talking about sustainability: the sustainability of resources and the sustainability of people. Yet both problems are resolved with one key property: resilience. The resilience to withstand changes in a project's resources, community and environment. There was also a consensus that strategies that embrace diversity — of contributors, maintainers and resources (technical and financial) — lead to a more sustainable project. Many of the sessions and themes centered around creating diversity within these areas.

The importance of good governance

A common theme throughout Sustain was the need for good governance within a project. Good governance can be the solution to some of the community-based sustainability issues that were highlighted including contributor retention and contributor/maintainer burnout.

Good governance in an open source project should be open and transparent, scale alongside a project's user base and allow contributors to take a more active role in the project as they gain experience. It should also represent the interests of the *potential* contributor, not just those who are currently engaged in a project.

Governance models should serve three distinct groups within a project. A broad contributor base, a group of maintainers with the right to accept, merge and distribute new versions of a project and, finally, a technical or leadership committee that defines the strategic direction of a project and breaks ties in decision making.

Governance should follow Elinor Ostrom's institutional design, defining a set of rules on how contributors will work together, who establishes these rules and how the rules are set. In addition, it should be clear how contributors can become maintainers and how maintainers can become members of the technical or leadership committee for a project.

Building diverse communities

There was a general acceptance that there's currently an unhealthy centralization around code and the role of the engineer in the production of open source.

An open source project needs a rich and varied set of skills that are very unlikely to exist within any one individual. The designer, the user researcher, the community manager, the accountant. It is surprising that we have reached this point despite the relentless focus on human-centered development in software more generally.

We need to bring a more diverse set of skills into open source by supporting projects with inclusive governance and tools that make it easy for people with skills outside coding to contribute. Then we need to amplify the input of these contributors in order to legitimize their views and empower them to become a more active participant in OSS. To sustain this, we need to equip developer-maintainers with leadership qualities that will enable them to distribute their responsibility and grow a supportive network around themselves and their projects.

Organizational structures

When thinking about organizational structures, we are thinking about how a community engages with legislative and financial restrictions in order to enable it to function—not about how it organizes to push a project forward. This usually requires some kind of legal entity to represent the interests of the project as a whole.

Projects typically need to think about creating a legal entity whenever they need to handle money and protect individual maintainers. Typically, this happens when they need to accept financial contributions, apply for a grant, charge for a product/service or pay for the services of others.

Using a legal entity in an open source project has many advantages. It can be used to house the assets of a project, protect members of the community from legal risks and accept and process payments for goods and services.

The type of entity that works for an individual project depends on the scale of the project and its community, how it creates financial sustainability, how it distributes funding and how it describes its mission. Some legal entities—public benefit, trade associations and charities—can operate tax-exempt, but there are tighter restrictions on accounting, auditing and governance practices that must be adhered to.

A number of foundations and institutions offer the legal and tax status to projects that fall under their mission. They support projects as part of an onboarding process to establish good governance, assist in financial and legal matters and to provide protection for project assets. More advanced legal structures can be used to ensure that projects are insulated from the influence of supporters—whether they are direct supporters offering grants or donations, or indirect supporters who pay for a product or service. The canonic example here is the Mozilla Foundation that owns the Mozilla Corporation, a subsidiary that operates a number of for-profit services, donating profits to the foundation.

Managing financial contributions

While there are many commercial providers offering ‘free for open source’ tools (often as a grassroots marketing initiative) there are comparatively few organisations offering direct financial support to open source projects. There are several challenges here that need to be addressed.

First, it is difficult for open source projects to handle money. Overheads are introduced and need to be managed. A legal entity may be needed (see Organizational structures), a bank account may have to be registered and taxes have to be accounted for. Most open source maintainers do not have the capacity to handle these additional pressures.

Second, it is difficult for projects to invoice a company for something as immaterial as ‘free software’. Often companies and maintainers must find workarounds to bring financial support to a project in exchange for a marketing opportunity on a website or at an event. This severely limits the ability for the organization to provide a meaningful level of support as the marketing team does not see the same level of benefit as the engineering team and can distract the project from achieving its goals.

Finally, it is difficult for project maintainers to decide on how and who to disperse money to. While it is easy to pay for technical resources or even the time of professionals who are not considered part of the production (legal representation, for instance), it is difficult for maintainers to decide *who* should be paid for their time and *what* they should pay. And yet it is already

evident that paid and volunteer contributors will happily work alongside one another when one is paid to do so by their company (typically a consumer of the project).

Creating sustainable incomes

We have established that accepting and managing financial contributions is a challenge. For projects that do, the further challenge of ensuring those contributions continue to flow in applies more pressure on already fragile maintainers. It is therefore essential that we support maintainers by charting a course through the murky depths of financial sustainability.

The key takeaway from our discussions was that while options available to maintainers are varied, each comes with a set of tradeoffs, incentives and influences that must be mitigated and navigated carefully to ensure the project is able to operate independently and in the community's best interests. One of the best ways of doing this is to ensure that a number of approaches are combined so no one supporter is able to exert pressure on the project over others. Of course, good governance can also mitigate some of these factors, particularly where a project roadmap has pre-approved many of the larger pieces of work that will be tackled.

Nadia Eghbal's work on financial sustainability was a key reference point for the discussion surrounding sustainable incomes and the incentives around them. Among them:

Donations
Being the purest form of financial support, the donation comes with the fewest risks and incentives. The challenges around managing donations concern the lack of individually material donations and the need to move to recurring subscription-style donations, both of which can be costly in marketing and PR. Through the lens of influence and incentives, most crowdfunding methods result in a donation. Open Collective, Gratipay, Patreon and Salt provide a means to accept donations to projects. A number of foundations can extend their charitable status to projects, freeing them from tax liabilities, some of whom are listed in the Resources section.

Subsidies
Subsidised income is the most common form of support for an open source project. Within this category we include paid part- or full-time employment by an employer and individual contributors working as a consultant or contractor to a company. This work may or may not involve work directly on the open source project that is being supported. The key factor in subsidized income is that the work that is free of incentives and influence is subsidised from that which is not. This speaks directly to the difficulty of insulating a project from the influence of contributors working in the interests of their employers. While it is possible to mitigate against the influence of employers through good governance and a project roadmap—it is rare for a project to act entirely in its own interests.

Grants and 'projects'

Conversely, open source projects that do have the necessary governance and foresight regarding the future of their work can seek sources to directly fund that work (which we have for the moment called ‘projects’). Typically, grant-making institutions provide the finances necessary to achieve targets that are in line with the aims of the project and funding institutions. With this form of support comes the overhead of administration of a grant, including reporting and tracking success. There is also the significant overhead of maintaining a charitable entity or finding an appropriate organization to extend its support to a project.

‘Products’

Products are not directly related to the project itself. Merchandise, books and guides are included in this category but it could be argued that sponsorship opportunities and events be included here too. While it is possible to raise significant portions of a project’s income from these activities, they are time intensive yet rely upon the success and attention of the project itself. These two interests are obviously in conflict, but ‘product’-based income does have a considerable advantage: ‘products’ are incidental to, and therefore distanced, from the interests of the project itself. This can do much to remove the influence of any one purchaser.

Successful examples of ‘products’ supporting an open source project include PyCon, the annual conference that uses a significant portion of its income to support the Python Foundation and, with it, the Python language and package environment.

Licensing

Many open source projects are licensed under terms that do not permit any commercial use without the products upon which these projects are built being licensed similarly. While it may be against the philosophy of open source software, for many there does exist a number of options to license parts or all of a project separately for commercial users in exchange for financial support.

The ‘open core’ model in which additional ‘enterprise’ features are sold on top of an open source project can work well for projects with distinct features, usage or deployment environments. Conversely, an entire project could be provided restriction-free for a fee that allows maintainers and contributors to work without having to consider the implications of where to place a feature or how to evolve a project.

While there will be many who reject the notion of using licensing or re-licensing, it is important to consider the needs of the maintainers and the project over the advantage that commercial usage provides for a partner that would otherwise be unwilling to use or contribute to the project.

Services

Finally, some open source projects may be able to offer paid-for services directly. Typically this includes access to a shared hosted service or a single-customer provisioned service. As a result of this, services are an option reserved for quite complex projects with direct access to

customers rather than smaller component projects (think database service versus HTML templating engine).

Services can provide an influence- and incentive-free source of income for a project but they can also suffer to licensing models. Maintainers and contributors must make a decision whether a feature is part of a paid-for hosted configuration or a local instance.

Conclusions and next steps

The first Sustain event was a resounding success. Over 100 participants came together to learn, to share and to explore the key issues surrounding the sustainability of OSS. Together we framed a lot of the concerns and challenges in a way that participants can take to their friends, colleagues and employers using a consistent and informed vocabulary. We also saw a number of participants sharing their experiences, mapping the route to solving some of these problems and the current solutions that they have employed successfully to solve them. Many of the key recommendations stem from these experiences. Finally, we charted a course toward solving those problems that still puzzle us. Commitments were made where interests and incentives were aligned to continue working on these issues outside of day's events. The organizers will follow up with each of these groups, support them in whatever way we can and ensure that their work is represented alongside the resources that the organizers will curate and publish on the sustainoss.org website.

We are already working toward holding a second event in 2018, most likely on the east coast of the US and we look forward to welcoming participants back to share what they have learned and to continue pushing toward a more sustainable future for OSS.

To the participants and sponsors of Sustain and to every contributor to open source software, we thank you.

- The Sustainers

Resources

Working groups

At the closing of Sustain participants were asked to propose informal working groups to help work toward solving some of the problems identified using the day. Organisers are committed to supporting these groups, maintaining contact and supporting them however they can.

Resource to help raise/give away a playbook and pitch deck for projects to ask companies for resources

Goal: Lower Barriers for projects to get open source resourcing
Contact: David Ryan

Open source categorization

Goal: Create repository with README and invite everyone interested as collaborators
Contact: Gregor Martynus

Open source vitality dashboard

Contact: Richard Littauer

Meetup at All Things Open 2017

Goal: Catch up on progress
Contact: Brian Bassett

Establish guidelines to forming an accountability org for corporations funding oss without money and project management

Goal: Establish guidelines to forming an accountability org for corporations funding oss without money and project management
Contact: Blaine

Articulate a set of core values, best practices, and design principles for open source institution design

Goal: To promote literacy of the pattern language of sustainable software / community development.
Contact: Karen Sandler

Be a guest on legal podcast to chat about open source sustainability

Contact: Andrea Goulet

Open source metrics

Goal: Compile a spec for describing an open source project
Contact: Ben Nickolls

Draft method for identifying sustainability pathways

Contact: Christie Koehler

Run a open source mentorship user group

Contact: Rachel Norfolk

Open source cryptocurrency

Goal: Research and info gathering to run experiment
Contact: Xavier Damman

Zulip chat group for open source sustainers

Contact: Lou Huang

Organise next Sustain (most likely in NYC)

Contact: Pia Mancini

Foundations offering organizational support

The following foundations have programmes that offer projects their legal and tax status, provide supporting human and technical resources and hold and protect the assets of a project:

- Apache Foundation
- Eclipse Foundation
- Centre for Cultivation of Technology
- Open Source Initiative
- Open Source Geospatial Foundation
- Free Software Foundation
- Linux Foundation
- New Media Democracy Fund
- NumFocus

- Software Freedom Conservancy

Acknowledgments

Report team:

- Ben Nickolls (Author)
- Pia Mancini (Data wrangler, editor)
- Justin Dorfman (Data wrangler, editor)
- Robert Gibb (Editor)

Event organizers:

- Chad Whitacre
- Pia Mancini
- Justin Dorfman

Special thanks:

- Brandon Keepers at GitHub
- Nadia Eghbal at GitHub
- Allen “Gunner” Gunn at Aspiration

Event Sponsors:

- GitHub
- StackPath
- 8th Light
- Waffle.io
- JS Foundation
- Iron.io
- Core Infrastructure Initiative (Linux Foundation)
- Google
- Alfred P. Sloan Foundation
- Lawrence Livermore National Laboratory
- Aspiration
- Open Collective
- Sticker Mule
- Unixstickers
- Gratipay
- Changelog
- Maintainer Mountaineer