Growing Open Source Projects with a Stable Foundation

Martin Michlmayr

April 2021

Preface

Starting an open source project is easy. Running a successful project, on the other hand, comes with a lot of work and responsibilities, especially if the project attracts a large user base.

While open source projects come in all shapes and forms, most projects encounter a similar set of growth issues throughout their life cycles. Because of this, various organizations have arisen to help projects handle these problems; these organizations are generally known as *FOSS foundations*.

This primer covers non-technical aspects that the majority of projects will have to consider at some point. It also explains how FOSS foundations can help projects grow and succeed.

This primer explains:

- What issues and areas to consider
- How other projects and foundations have approached these topics
- What FOSS foundations bring to the table
- How to choose a FOSS foundation

Contents

Background

1 Open source as digital infrastructure

Nadia Eghbal describes open source as digital infrastructure in her report Roads and Bridges:

> Much like roads or bridges, which anyone can walk or drive on, open source code can be used by anyone—from companies to individuals—to build software. This type of code makes up the digital infrastructure of our society today.

Open source has fundamentally changed the way software is written. Instead of starting a new software project entirely from scratch, programmers look for open source libraries that can be integrated to solve the problem at hand. Furthermore, the advantages of open collaboration through open source is increasingly being recognized by companies, and they team up with other companies in order to share development costs and accelerate the pace of innovation.

Open source hasn't just had a huge impact on the development community. Open source has become pervasive in our society, and we rely on it every day, even if we are not aware of it. The Linux kernel forms the core of all Android phones and Kindle e-book readers, numerous WiFi routers, and many other products we use on a daily basis. Other open source components are similarly widespread.

Modern life simply wouldn't be the same without thousands of open source libraries and programs written and maintained by a community of developers. Of course, this tremendous success of open source also raises important questions — since we rely on open source, how can we make its development more sustainable?

FOSS foundations, non-profit organizations that support open source projects, play an important role in contributing to the sustainability of open source.

2 Growth issues

It's easy to start an open source project. Popular platforms like GitHub and GitLab have reduced the barriers to entry even more.

The majority of projects stay limited in scope and reach. However, many projects attract the attention of other developers and users, and they quickly grow beyond the original scope envisioned by the project originator.

Such success can be rewarding on many levels: helping others solve important problems, receiving feedback and code from others, interacting with a wide range of people, and more.

But success and growth can be a two-edged sword as they are also associated with additional work, increased responsibility, and tasks that one may not be interested in.

As a project grows, the structures supporting the project must adapt. Decision-making is easy when you're the only one making decisions. The larger the project becomes, the more clearly defined the governance structures have to be. How can others contribute to the project? How are decisions made? These are some of the questions that need to be answered and codified in a governance structure.

Successful projects also need to deal with issues they never expected, such as non-technical decisions and administrative work. How can we accept donations? What can we do with the money? Do we need to register a trademark for our project name?

For the long-term health of the project, it's beneficial when the project's support structures don't rely on the project founder. An organization, such as a FOSS foundation, can provide an independent home for the project.

3 The role of foundations

FOSS foundations play an important role in supporting open source projects and contributing to their long-term sustainability. The people involved in FOSS foundations have considerable experience and often have encountered similar problems to those a project faces. They know from experience what works and what doesn't. Some organizations also provide specific mechanisms to help young projects, for example through mentorship or a formal incubation process for new projects.

A foundation can also serve as an independent entity that represents the consensus of a project. The process of creating a new organization, or selecting an existing organization to join, is a useful exercise because it forces a project to think about a number of issues, such as governance, asset management, culture, vision, and more.

There's a wide range of different FOSS foundations (as will be covered in more detail later). They support open source projects in a number of key areas, including governance, community, financial, and legal.

While there are a number of different types of organizations, the emphasis in this primer is on umbrella organizations. These are organizations that provide a nourishing environment and offer a set of services to multiple projects that operate under their umbrella.

Governance

4 Open collaboration

Open source is associated with open collaboration. Some projects rely on contributions from unpaid volunteers, other projects show a high level of corporate engagement, and many projects have a mix of volunteer and paid labor. Contributors work together in an open, collaborative manner to advance the project.

The Linux Foundation, which hosts many projects and organizations, describes five key principles of open collaboration:

- Organizational neutrality: assets of a project, such as domain names and trademarks, are held by a neutral organization rather than a single individual or company. While the organization owns the assets legally, the community decides how to use them through its governance model.
- Clear separation of funding and participation: providing financial support to the organization does not give companies or other donors the right to influence the technical direction of the project. The technical development is governed separately and all contributions are evaluated on their own merit.
- Open governance: projects operate in a transparent manner and have neutral and clearly defined governance models which set out how participation works and how decisions are made.
- Intellectual property clarity: the removal of uncertainty over licensing makes it easier to contribute and to use projects.
- Commercial support ecosystem: projects can benefit when a commercial ecosystem forms around them, which creates employment opportunities and provides funding to support the project, such as through sponsorship of a developer conference.

While the exact nature and operations of organizations and projects differ, these principles apply quite widely to the type of open collaboration that is supported by FOSS foundations.

FOSS foundations: protecting the definition of open collaboration

Mike Milinkovich, the Executive Director of the Eclipse Foundation, explains the role of FOSS foundations play in supporting open collaboration:

> We foster collaboration in an openly governed way that protects the assets that are created for all of the downstream consumers. We are institutionally mandated to resist those who try to redefine what openness means in our communities.
>
> I really believe that open source foundations are an absolutely integral part of protecting the definition of open collaboration, enabling open innovation, and making that sustainable for the long term.

5 Level playing field

Open source projects are increasingly being formed and led by companies. Some of these projects garner significant attention from other parties, such as individuals and other companies. In addition to using the software, those parties want to contribute and shape the direction of the project.

Unfortunately, some projects that are originated by companies do not offer a level playing field: contributions from the company are treated more favorably than those from ‘outsiders’. The company may also require a contributor license agreement that gives it broad rights to the contributions, further increasing its control over the project.

Good stewards recognize that you have to set a project free in order for it to flourish. FOSS foundations constitute a vendor-neutral venue where open collaboration among participants can take place on an equal footing. By moving a project to a neutral foundation, companies can signal that they are serious about creating a level playing field.

The project and foundation can put governance structures in place to ensure that no single corporation can exert undue influence over the technical direction of a project.

Kubernetes and the Cloud Native Computing Foundation

Kubernetes, the highly successful platform for managing containerized workloads and services, was started by Google. When version 1.0 was released, Google moved the project to the Cloud Native Computing Foundation (CNCF), a foundation operating within the Linux Foundation.

While Google is still a major contributor, it’s by no means the only driving force behind the project. According to developer data from CNCF, major contributors include Red Hat, VMware, Microsoft, IBM, Huawei, and others.

6 Technical governance

While FOSS foundations differ greatly in their operations and approach, the majority of organizations keep the technical governance separate from the governance of the organization.

Open source communities strive for technical excellence. FOSS foundations are there to support this effort, but the organization and the project are typically distinct entities, each focused on what they do best.

Technical governance at the Open Infrastructure Foundation

The Open Infrastructure Foundation supports the development and adoption of open infrastructure globally by hosting open source projects and communities of practice.

Thierry Carrez, VP of Engineering at the Open Infrastructure Foundation, explains:

> We do not require any organization to pay in order to successfully join our communities. Technical governance of our projects is disconnected from Foundation governance: one individual may be a leader in our projects without their employer being a Foundation member. This ensures the project is open to everyone, including individuals, maximizing participation and adoption.

7 Stability and institutional memory

Projects are in a constant flux. It's not just the code that changes, but also the collaborators. The project founder may leave after several years. The activity level of collaborators can go up and down depending on the circumstances in their lives (such as becoming parents).

Organizations can contribute to the longevity of a project by providing stability and institutional memory. While the people involved in an organization change over time, the organization itself can provide consistency through a stable approach supported by a common mission, guiding principles, and documented policies.

Best practices to support stability and institutional memory include:

- Clearly defined mission and vision
- Written policies
- Good record-keeping (such as keeping meeting minutes and archiving contracts)
- Onboarding and orientation information for new board members

Revision-control systems make it possible to understand years later why a certain code change was made. Mailing list discussions can similarly be used to track down the reasons why certain changes were made.

While the sensitive nature of some matters (such as legal issues) don't allow organizations to discuss and archive everything in public, similar principles can be applied to corporate records: maintain good internal documentation and records, and document why certain actions were taken.

The role of KDE e.V. in maintaining institutional memory for the KDE community

Adriaan de Groot, a member of the board of directors of KDE e.V., explains how the organization helps maintain an institutional memory:

> Very few of the original developers are still involved day-to-day with the KDE community. Having an institution that provides multi-generational stability is good. As old developers fade from view, their knowledge isn't typically entirely lost. We had a "lost knowledge from long ago" talk at Akademy 2020, the annual conference of the KDE community, specifically to carry forward some of the more obscure institutional bits.

Additionally, he emphasizes the role of guiding principles:

> We carry the KDE manifesto around; the manifesto is something the community agreed on, but it shapes the way KDE e.V. works by providing *moral* guidance. That means we have a 'constitution' that anchors us to a particular set of principles.
>
> It's important to have stability in the organization and innovation in the community.

8 Conflict resolution

Open collaboration is a wonderful way to develop software, but conflicts are bound to happen at some point in any project of significant size. There can be disagreements on technical decisions, the direction a project is heading, or other aspects of the project.

A solid governance structure helps to reduce and resolve conflicts by providing clear policies and decision-making processes. The culture of a project also has a big impact, especially if it encourages friendly collaboration.

Organizations can also run into conflicts, including conflicts of interest, especially in trade associations that have corporate members with different commercial interests.

Organizations employ a number of mechanisms to prevent conflicts in the first place. Florian Weikert, Dirk Riehle, and Ann Barcomb studied several FOSS foundations and identified five major areas of conflict prevention:

- Screening processes: new members and projects are only accepted when they pass a screening process. Such a process can identify common interests and the motivation of potential members. The technical, cultural, and strategic fit of projects can be assessed. Some organizations use an incubation process and new projects can only 'graduate' once they meet certain criteria.

- Governance structures and rules: foundations have formal governance structures and rules that are codified in their by-laws. They can include:
  - Transparent affiliations: contributors have to declare their corporate affiliations in public to highlight potential bias.
  - Decoupling money from control: the financial funding of the organization is separate from the technical decision-making.
  - Representation limits: some organizations limit the number of people from the same company who can serve on the board of directors.

  - Independent entities: key staff of the foundation, such as the executive director, may not be employees or consultants of a member company.

- Explicit strategies: organizations protect their culture and values through a number of explicit strategies, such as monitoring the behavior of participants.

- Common interests: even though member companies often compete in the market place, common interests allow them to collaborate. These common interests can be technical (e.g. a focus on technical merit over corporate agendas) or related to business (e.g. building a common platform against a dominant competitor).

- Culture and values: shared values prevent bad behavior and contribute to the success. Important values include openness, transparency, equality, and neutrality.

Culture and values: Python

The Python community expects its members to be "open, considerate, and respectful". Behaviors that reinforce these values include:

- Focusing on what is best for the community
- Acknowledging time and effort
- Being respectful of differing viewpoints and experiences
- Showing empathy towards other community members
- Gracefully accepting constructive criticism

Community growth

9 Incubation

Several FOSS organizations offer an incubation process, which is a formalized intake process for new projects (in particular for young projects). The aim of such an incubation process is to help projects attain more maturity in a number of ways.

Incubation can cover different aspects of a project, including the creation of well-defined governance structures with the help of experienced members from the organization. It can also involve mentorship and guidance on the diverse set of issues that a project may face.

Some projects can benefit from joining an umbrella organization: not only can they tap into considerable experience, but becoming part of a larger organization may also allow them to attract more attention quickly thanks to the brand of the organization.

Apache Incubator

The Apache Software Foundation (ASF) offers an incubation process for projects that wish to join the organization. It helps incoming projects adopt the Apache Way, Apache's style of governance and operation, and guides new projects to services the organization provides.

Apache provides an Incubator Cookbook that helps potential projects decide whether the ASF is a good fit for them and guides them through the steps required to join the incubation process.

CNCF Sandbox

The Cloud Native Computing Foundation (CNCF) groups projects into three stages according to their maturity: sandbox, incubating, and graduated projects.

The CNCF Sandbox has four goals:

- Encourage public visibility of experiments and other early work
- Facilitate alignment with existing projects
- Nurture projects
- Remove possible legal and governance obstacles to adoption and contribution

Projects can move to the next maturity level after demonstrating their sustainability according to clearly defined graduation criteria.

10 Mentorship

Several projects and organizations offer formalized mentorship programs to make it easier for those new to open source to get involved. Some organizations also pay stipends to mentees.

The Linux Foundation offer a mentorship program to help developers "to learn, experiment, and contribute effectively to open source communities". Projects that mentees can work on include the Linux kernel, Hyperledger, and GraphQL. The organization also provides tooling through which projects can run mentorship programs.

X.Org's Endless Vacation of Code (EVoC) is another program that pairs those interested in working on a project with a mentor.

Several organizations offer mentorship programs that are open to many open source projects, including Google Summer of Code and Outreachy. Being part of an organization may allow a project to participate in such outreach programs.

Outreachy: increasing diversity in open source

Outreachy intends to increase diversity in open source by offering remote internships to "anyone who faces under-representation, systemic bias, or discrimination in the technology industry of their country":

> Interns work with experienced mentors from open source communities. Outreachy internship projects may include programming, user experience, documentation, illustration, graphical design, data science, project marketing, user advocacy, or community event planning.

Mentorship and outreach programs are a good way for projects to attract new contributors and grow their community. Foundations can help by taking care of the legal paperwork and distributing funds.

11 Marketing and promotion

Many open source developers have a "if you build it, they will come" mentality, but active promotion can lead to more visibility. The more visible a project is, the more developers and users it can reach. Promotional activities include giving presentations at conferences, participating in podcasts, speaking to journalists, building an informative website, recording screencasts, and more. Such activities can target developers, users, or both.

In addition to promotion, a large part of marketing is understanding the target market better. Conferences are a good way for developers to interact with users to hear how they use the software and what problems they face. Meeting users and listening to interesting use cases can be a great source of motivation for developers.

Some projects have outreach programs in order to increase the size of the community, including cooperation with universities and mentorship programs.

FOSS foundations can assist with promotional activities in a number of ways. They can purchase merchandise for conferences, take care of the logistics of paid mentorship programs, negotiate contracts with graphical designers, and more. Many FOSS foundations have a good network of connections and can open new doors. Joining a prominent foundation can also give a project a boost in terms of visibility.

Finally, some FOSS foundations play an advocacy role, promoting ideas like software freedom and open standards.

Marketing efforts of LibreOffice

LibreOffice's marketing team leads a number of efforts, including:

- Communications, such as press releases
- Informing the community about what's going on inside the project
- Development of collateral
- Events management and participation

12 Culture

The culture of a project is an important factor that influences the health and growth of a community. Increasingly, projects put a great emphasis on creating a culture in which participants are welcome and can thrive. The culture of a project often determines whether contributors will stick around.

Many projects publish a formal code of conduct, so participants know what to expect. Such documents can emphasize the shared values as well as the expected norms of the community.

The Contributor Covenant is a code of conduct that has been adopted by many projects and communities, including Jekyll, Jenkins, and the Linux kernel.

Contributor Covenant

Some benefits of adopting a code of conduct like the Contributor Covenant:

> Adopting the Contributor Covenant helps makes your community's values explicit, and signals your commitment to creating a welcoming and safe environment for everyone.

A code of conduct is only effective when community leaders follow it themselves, providing a good example. There must also be an enforcement mechanism to make sure collaborators adhere to it. Problems with participants are often addressed through a formal process led by an organization within the project.

Django: enforcement of code of conduct

The Django project has an enforcement manual which clearly sets out how the project responds to an issue. Conduct violations are managed by a code of conduct committee, which is established by the Django Software Foundation.

> The committee will then review the incident and determine, to the best of their ability:
>
> - what happened
> - whether this event constitutes a code of conduct violation
> - who, if anyone, was the bad actor
> - whether this is an ongoing situation, and there is a threat to anyone's physical safety

13 Community health

Community health and growth can be measured, for example through metrics, and dashboards can be used to display and monitor the data.

Community Health Analytics Open Source Software (CHAOSS) is a Linux Foundation project focused on creating analytics and metrics to help define community health. It has working groups that focus on different areas, including risk, evolution, and diversity and inclusion.

CHAOSS: the importance of community health

CHAOSS gives several reasons for monitoring community health:

- Open source contributors want to know where they should place their efforts and know that they are making an impact.
- Open source communities want to attract new members, ensure consistent quality, and reward valuable members.
- Open source companies want to know which communities to engage with, communicate the impact the organization has on the community, and evaluate the work of their employees within open source.
- Open source foundations want to identify and respond to community needs, evaluate the impact of their work, and promote communities.

There are various dashboards available to show metrics of open source projects. One is Cauldron, which aggregates information from multiple collaboration platforms. Another is Bitergia Analytics, which is a project health dashboard that has been adopted by a number of foundations for the benefit of their member projects.

Financial

14 Fundraising

Significant projects typically need some funds to pay for expenses. Even if the developers are not paid by the project, there are other expenses, such as travel, hosting, and domain names.

Projects can have a number of sources of funding, including:

- Donations (from individuals and corporations)
- Merchandise (T-shirts and other items)
- Sponsorship (for conferences or other activities)
- Conference revenue (registration fees and sponsorship)
- Royalties (for books or trademark usage)
- Mentorship stipends (Google Summer of Code and Season of Docs)
- Grants (from non-profit organizations or corporations)

While it's easy to put a PayPal button on a web site, there is considerable paperwork and administrative work involved with the majority of these funding sources. Sponsorship and grants often involve legal agreements; many companies require a formal invoice and there are typically several steps associated with submitting invoices (registration as a vendor, creation of a purchase order, etc.); some donors wish to receive a written acknowledgment for their donation; and more.

Putting up a PayPal button also raises an important question: who does the money go to? If a project has a single developer, putting up a personal PayPal link might be acceptable. For larger projects, handling funds through a single developer is not an option. The practice is also problematic from a tax perspective as accepting donations may create tax liabilities for the individual holding the funds. This is where an independent organization can assist by keeping funds on behalf of the project and complying with all rules and regulations. Finally, some donors may prefer giving to an organization rather than to a specific developer.

Some umbrella organizations also have fundraising processes from which several projects benefit. The brand and reputation of an established organization often makes it easier to attract funding.

15 Spending

There are a number of ways money can be used to improve a project, including:

- Domain names
- Infrastructure
- Hardware for contributors
- Artwork
- Documentation
- Development
- Internships
- Training
- Travel stipends
- Conferences
- Hackathons and other development meetings
- Marketing materials
- Operations (including accounting and legal)

Some services, such as continuous integration (CI) infrastructure and pro-bono legal help, may be available to open source projects free of charge. In some cases, it might make sense to use project resources to pay for a service anyway (for example to get a faster response or more reliable service).

PSF Grants Program

The Python Software Foundation (PSF) operates a grants program which allows members of the community to apply for funding. The current focus is on virtual events, virtual sprints, and Python diversity and inclusivity efforts.

Events and projects that receive funding from the PSF are required to submit a report showing how grants were used.

16 Face-to-face meetings

There are many good reasons for projects to meet-to-face, such as hashing out technical solutions in an effective way, building social bonds, and increasing motivation for participants.

While it's fairly easy to meet at existing conferences, larger projects may wish to organize their own hackathons, developer meetings, or conferences. This comes with organizational issues that are often not fully appreciated. A venue needs to be selected and a contract signed, catering needs to be arranged, and an insurance policy should be taken out. The project may wish to reimburse attendees for their travel expenses, which typically involves payments to many different countries.

Hopefully the event is a success, but it's important to plan for all eventualities:

- What if the conference is canceled due to a pandemic? Can the venue be canceled for free or will the project incur a huge venue expense despite the cancellation?
- What if harassment happens at the event? Are policies in place for that?

Some organizations hold annual events and are well versed with all organizational aspects of event management. In fact, conferences are a substantial source of income for several organizations.

17 Development

Some projects pay directly for development work. This can greatly increase the amount of work that can be done in some cases, but it can also introduce challenges and conflicts.

One potential issue relates to motivation. Various studies have shown that a hobby that is turned into a job sometimes becomes a chore. Many open source contributors are driven by intrinsic motivation, such as a drive to learn and explore their potential. External factors, such as money, are not as motivating as those that come from within. Unfortunately, if external factors are introduced and later removed (e.g. funding stops), internal motivation may not fully come back, resulting in a loss of interest in contributing to the project.

There are also questions related to fairness. If there's not enough money to pay all developers, a decision needs to be made as to whose work will be funded. The project also needs to agree how much to pay, which can be difficult since salary expectations vary widely across the world.

Fortunately, many projects don't struggle with all of these problems. It's quite common that developers don't seek funding for their work, but are happy to see other people's work being supported financially. This can be because they already have a corporate sponsor (e.g. an employer who lets them work on the project) or because they prefer to stay in a volunteer capacity and work on the project as a hobby rather than a job. Some projects also successfully fund tasks that nobody else is willing to take on.

If a project decides to fund development, some paperwork is necessary, such as a contract setting out expectations and responsibilities for everyone involved. It's important to distinguish contracting work and employment, as the latter can lead to a lot of obligations on behalf of the project. Many foundations have experience with getting contractors on-board for projects and doing all of the necessary support work.

Django Fellowship Program

The Django Software Foundation runs the Django Fellowship program, which pays contractors to perform some of the administrative and community management tasks of the Django project:

> The Django Fellowship program has a major positive impact on how Django is developed and maintained. The Django Fellows triage 10-15 new tickets each week and review and merge around fifteen non-trivial patches a week from the community. Release blocking and severe bugs aren't postponed indefinitely. Major releases happen according to an 8 month schedule and bug fix releases occur monthly.

18 Asset stewardship

Successful communities usually obtain a number of assets to support their projects, such as:

- Money
- Servers and other technical equipment
- Intangible assets, such as trademarks and domain names

Over the years, there have been several reports about projects running into problems because key assets were held by individuals who no longer represented the project's best interests.

Neutral asset stewardship is one of the great benefits an organization can offer to a project. Instead of assets being owned by individuals involved in the project, all assets are owned by the organization. The project, through its governance process, decides what to do with the resources; this happens in cooperation with the organization, which ensures that all legal obligations are met.

Asset stewardship involves many responsibilities: trademarks and domain names must be renewed periodically; projects expect financial reports about project funds; and an investment policy may be needed if an organization holds substantial funds.

Projects with big balances

Some prominent open source projects receive substantial donations and sponsorship, but often have few opportunities to actually spend the funds to improve the project. Many collaborators are already paid by companies to work on the project, and only a small budget is needed to fund developer meetings and infrastructure. Since expenses are lower than donations, a significant fund balance is built up over the years.

Donations are made to support the project. Therefore, projects that accept donations should think about the best ways to use the funds to improve the project and community.

19 Transparency

Projects that accept donations have an obligation to be transparent about the ways the funds are being used.

Several projects and foundations publish annual reports that cover their activities for the year. Contractors funded to work on open source projects often publish monthly reports in which they document their accomplishments.

Many organizations also publish detailed financial reports about income and expenses. Charities that are based in the US have to publish an annual tax filing (form 990) if they receive a certain amount of donations that year. The tax filing contains information about the income and expenses, and it often gives a good overview of the activities of an organization.

Transparency also applies to other areas of an organization, such as the publication of by-laws and other governance documents.

Transparency of FOSS foundations

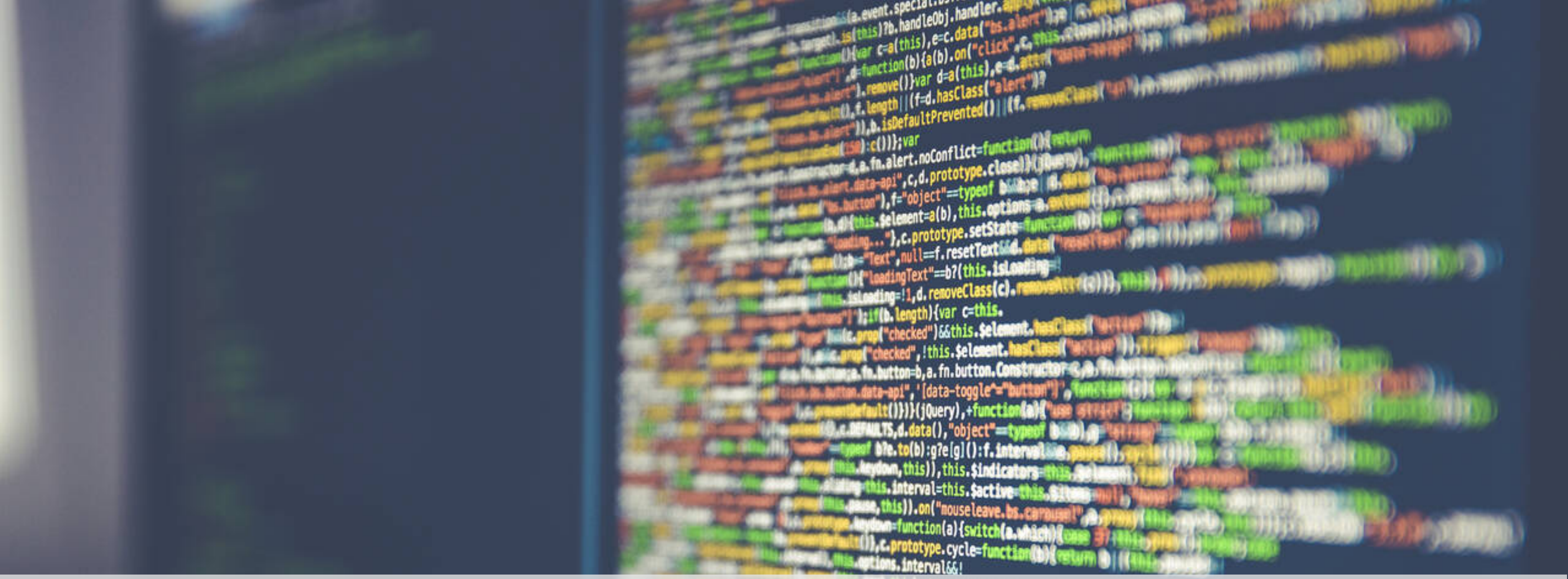
Many FOSS foundations publish annual reports, public filings, audited financial statements, and other materials.

Some examples include:

- Mozilla Foundation: annual reports, public filings, and audited financial statements
- NumFOCUS: annual reports and public filings
- Open Infrastructure Foundation: annual reports
- Software Freedom Conservancy: public filings and audited financial statements
- The Documentation Foundation: annual reports and accounting ledgers

There's also a repository where public filings from FOSS foundations are archived.

Legal



20 Licensing and copyright

The licensing of open source projects is a widely discussed topic. The choice of a license can greatly influence the impact and growth of a project. Many resources exist to help projects make the right decision, such as the "choose an open source license" site.

In addition to the license itself, there are other aspects related to licensing and copyright that projects have to consider, and where FOSS foundations can play an important role.

Some projects require contributors to sign a Contributor License Agreement (CLA), which gives certain rights to the project. If a CLA is required, a FOSS foundation would be a good home for the agreements. However, CLAs are often seen as harmful and the majority of projects rely on a system commonly referred to as inbound=outbound whereby contributions to the project are provided under the project's license without the need for another agreement. Some projects additionally use a simple self-attestation mechanism to confirm the origin of the contribution, such as the Developer Certificate of Origin (DCO) used by the Linux kernel and other projects.

Another area where FOSS foundations can help is with the enforcement of open source licenses. While the idea of open source is to share the code widely, there are certain conditions attached to the distribution of the software. Unfortunately, these conditions are sometimes not followed (many times inadvertently, although sometimes intentionally). FOSS foundations can work with those organizations to resolve these types of issues, or, if necessary, to initiate legal action.

Finally, while we don't like to think about death, it's important to plan ahead. A project may want to change its license or move to a new version of a license in the future, but this can only be done with the explicit permission of all copyright holders. A FOSS organization could act as a custodian and make decisions for deceased contributors who granted them such permission.

21 Trademarks

Trademarks play an important role in open source projects. Anthonia Ghalamkarizadeh, a lawyer, describes a trademark as an "identifier of origin", which gives brand protection in two directions:

- Protecting the reputation and values of the trademark owner
- Protecting users from confusion and fraud by conveying a clear message about a product's origins

Unfortunately, popular open source brands are sometimes used for nefarious purposes. For example, downloads of 'Firefox' were bundled with a costly subscription, which users did not expect. The Firefox trademark allows the project to remove illegitimate downloads that abuse its brand.

Trademarks are usually held by a FOSS foundation, which can take legal action if necessary. Trademarks can be registered, and the Madrid system allows registration in multiple jurisdictions. Registered trademarks need to be maintained and enforced. The GNOME project, for example, had to defend its trademark when Groupon wanted to use the name for an unrelated product.

It's important for projects to have trademark usage policies that clearly spell out acceptable and unacceptable uses of the mark. The Model Trademark Guidelines are a good starting point for this.

Trademark resources

FOSSmarks offers a practical guide to understanding trademarks in the context of open source projects, including basics on trademarks, choosing a name for a project, registering a name, and more.

The Apache Software Foundation maintains a trademark page, which lists not only its own trademark policy, but also contains a list of useful trademark information.

Finally, the Legal Issues Primer published by the Software Freedom Law Center contains a chapter on common trademark issues.

22 Patents

Patents are a way to register an invention and get exclusive rights to the invention for a period of time. They can be problematic for open source projects. Projects generally don't patent their inventions, but instead they implement their ideas in software that they share widely. However, projects can get attacked by companies that hold patents and allege infringement. A patent defense can be extremely expensive because it involves lawyers.

The Open Invention Network (OIN) plays an important role in protecting some open source projects by building up a pool of patents as a defensive strategy. Projects and FOSS foundations can join OIN as community members.

The Software Freedom Law Center has published a legal primer, which describes several defense mechanisms.

Patent case against GNOME

The GNOME Foundation successfully defended itself against a patent infringement claim. The outcome was remarkable: in addition to GNOME receiving a covenant not to be sued for any patent held by Rothschild Patent Imaging, they negotiated the same covenant for any software that is released under an open source license.

GNOME's defense was highly successful, but a positive outcome is not guaranteed. Even if the project is successful in its defense, it will lose a lot of time that could have been spent on other activities, such as improving the software.

23 Liability

Unfortunately, sometimes things don't go as planned. A conference may have to be canceled because of a pandemic, or a project may be involved in a lawsuit (for alleged trademark or patent infringement, or other reasons).

While such issues will always be painful and time-consuming to deal with, having an organization is beneficial in those cases. The organization will be better equipped to deal with these types of issues than either projects or individuals on their own. A legal entity may also offer some protection for individuals. For example, individual contributors wouldn't be liable for a venue contract signed by a FOSS foundation.

Projects and organizations can also plan ahead by putting strong governance in place. When drafting policies and decision-making documents, they should take into account scenarios where things go wrong.

A FOSS foundation should also secure directors and officers (D&O) liability insurance to safeguard directors who serve the organization. Other insurance policies, such as for conferences, are also a good idea.

24 Contracts

Some projects have to deal with a surprising amount of paperwork.

Contracts may need to be signed for a number of reasons, such as:

- Sponsorship agreements
- Grants
- Venue rental
- Employment contracts
- Intellectual property rights and provenance
- License enforcement agency

It usually doesn't make sense for individuals to sign contracts on behalf of a project. It's much better for this to be done through an organization that represents the project.

FOSS FOUNDATIONS

25 Classification

Javier Luis Cánovas Izquierdo and Jordi Cabot classify FOSS organizations according to three orthogonal dimensions:

- Geographical scope: local or global focus
- Coverage: focus on a specific project or domain, or a broader focus, possibly serving as an umbrella organization
- Transparency: the degree of transparency in governance (e.g. by-laws) and operations

In terms of geographical scope, there is a wide range of focus. The PostgreSQL community has several organizations, like PostgreSQL Europe and PostgreSQL.US, that serve a specific geography, typically to organize events in those regions. Organizations like FOSSASIA and the Free Software Foundation Europe have a fairly specific (albeit still fairly broad) geographical scope. Other organizations, like the Eclipse Foundation, serve a truly global community.

The coverage can also vary greatly. There are several organizations that provide a specific service or focus on a particular problem, such as the Software Freedom Law Center, which offers legal services to open source projects, and Linux Professional Institute (LPI), which promotes the use of open source by developing certification standards.

Several organizations focus on a specific project, such as the Krita Foundation and the Blender Foundation. Some organizations, like KDE e.V. and the Python Software Foundation, were originally created to serve a specific project; as those communities grew and expanded, so did the role of the organizations.

Finally, there are umbrella organizations that accept a wide range of projects. Some of these have a focus on a specific theme, such as scientific computing (NumFOCUS), biological research (Open Bioinformatics Foundation), and cloud computing (Cloud Native Computing Foundation). Other umbrella organizations serve projects from a wide range of backgrounds.

MARKET MENU
ASK ABOUT TODAY'S SPECIALS!
HUEVOS Local, Pastured Eggs from Heirloom Varieties
VERDE COCINA
Buenos Dias Breakfast $10
Two eggs, scrambled w/veggies over white bean garbanzo mash, topped with more veggies, ranchero salsa, and locally grown and smoked BACON. Tortillas on the side.
Chilaquiles (chee-la-kee-les) $8
Huevos Rancheros $8

26 Services

FOSS organizations aim to meet the needs of the projects they serve. Umbrella organizations typically offer a range of services, broadly reflecting the project needs described in this primer (such as mentorship, asset stewardship, and event management). Different projects use different services based on their needs. Typically, projects can choose *à la carte* from a menu of services. Umbrella organizations usually take a percentage of project donations in order to provide their services.

Of course, not every organization offers everything. For example, several organizations don't employ staff to write software. This can be for a number of different reasons, including:

- Philosophical reasons: an organization may wish to provide infrastructure for a project but stay out of development.
- Lack of demand: the ecosystem has enough companies that pay for development, so there's simply no need to employ developers.
- Organizational limitations: some organizations are not prepared for the paperwork involved with having employees.

Example services and support programs

The Linux Foundation offers a wide range of support programs. Their core support programs include project operations, tooling, training, certification, and event management. Additional support programs include legal and IP management, HR support, marketing operations, and community growth.

The Software Freedom Conservancy offers many member project services, including contract negotiation and execution, conference logistical support, community elections, basic legal advice and services, FOSS copyright license enforcement, and fundraising assistance, along with leadership mentoring, advice, and guidance.

27 Umbrella organizations

An umbrella organization is an association that coordinates activities and pools resources. In the context of open source, FOSS foundations that operate as umbrella organizations accept projects under their ‘umbrella‘.

Projects that join an umbrella organization benefit from the organization’s legal structure and can access services offered by the organization, such as operations (accounting, legal, etc.), conference management, mentorship, and more. Umbrella organizations typically take a percentage from incoming donations to cover their operational expenses.

The term *fiscal sponsorship* is often used in the context of umbrella organizations. This term is confusing because it does *not* mean that the organization provides financial sponsorship to a project (in fact, organizations take a percentage of project donations).

What fiscal sponsorship means is that the umbrella organization acts as the host of the project and performs administrative functions on behalf of the sponsored project. Crucially, the organization takes on the responsibility of receiving and administering donations and of complying with all legal obligations.

Fiscal sponsorship reduces the need for creating a new organization because a project can use the existing infrastructure of the umbrella organization. Essentially, projects with similar needs can join up via an umbrella organization instead of each creating their own entity.

There are two types of fiscal sponsors:

- Comprehensive fiscal sponsors: projects become fully integrated into the sponsor, similar to a merger and acquisition.
- Grantor/grantee fiscal sponsors: projects don’t join the fiscal sponsor but remain autonomous; fiscal sponsors accept funds on behalf of the project and provide these funds to the project in the form of grants.

Center for the Cultivation of Technology: a grantor/grantee fiscal sponsor

The Center for the Cultivation of Technology is an example of a grantor/grantee fiscal sponsor:

> The Center for the Cultivation of Technology is a "backend provider" for the Free Software community. Affiliated projects use the legal entity to collect donations and to manage their budgets and assets, while we take care of organizational matters such as accounting, reimbursements and financial reporting.
>
> The Center aims to become a lightweight scalable fiscal sponsorship "host", much like a "payment processor for unincorporated projects". But in most cases, we work closer with projects to identify funding opportunities, provide additional services (together with partner organizations), exchange knowledge about sustainable organizational development, and allow developers to focus on what they do best: write code.

Looking at FOSS foundations, there's a wide range of how close or loose the relationship between the project and the umbrella organization is. Furthermore, some organizations expect projects to operate a certain way while others are more hands-off. For example, the Apache Software Foundation (ASF) requires projects to use the Apache license and follow the Apache Way, its governance process and philosophy. Software in the Public Interest, Inc., on the other hand, puts few restrictions on how projects operate.

Each of these approaches have their unique pros and cons. Different projects have different needs and some FOSS foundations will fit their needs better than others.

Debian and its trusted organizations

Instead of operating a Debian Foundation, the Debian project works with several organizations (known as trusted organizations) that provide services and hold assets for the project. The project also has an agreement with Software Freedom Conservancy that lets contributors assign copyrights if they wish to do so.

While this may be an unusual example, it shows that a loose relationship with one (or more) organizations works for some projects; other projects may benefit from a closer, integrated relationship with one organization (possibly even their own organization).

28 Legal structure

The majority of FOSS foundations operate as non-profit organizations. While there are FOSS foundations in many countries, presently the majority are based in the US. The most common legal structures used by US-based FOSS foundations are:

- Charities: these organizations work toward a specific mission that has a benefit to the general public.
- Trade associations: these organizations allow companies to advance common business interests.

These two legal structures are often referred to as 501(c)(3) (for charities) and 501(c)(6) (for trade associations), as a reference to the section of the Internal Revenue Code where they are defined.

Stormy Peters highlights a number of differences between charities and trade associations. For example, donations to charities are tax deductible (for certain donors in the US). This is not the case for trade associations, although membership fees can be treated as business expenses. Also, while charities are only permitted very limited lobbying, trade associations can engage in political lobbying that is related to member interests.

There are various legal structures in other countries that have been used to set up FOSS foundations. The Document Foundation's use of a German *Stiftung* is notable. In part, it was used specifically to make certain elements of its rules unalienable, therefore reducing the risk that the Foundation's initial ideals would be changed over time.

FOSS foundations: what's in a name?

Organizations that support the activities of open source projects are often called *FOSS foundations* (as in this primer). However, the term *foundation* has a specific meaning in some jurisdictions and therefore some people prefer different terminology, like *association* or *non-profit organization*.

29 Geography

Open source projects typically operate globally, but the organizations that serve them have to establish a legal entity in some jurisdiction. One question is whether the country of that entity matters. While projects are used to operating via the Internet, setting up a legal entity often involves activities that have to be done in person (such as opening a bank account). Therefore, some FOSS foundations are established where they are simply because that's where a volunteer willing to do the work was located.

The location of the legal entity may matter in some circumstances. If the majority of donors are in a specific region, it makes sense to establish the legal entity there in order to make donations easier. For example, bank transfers are easy within the EU, but many US-based organizations don't have bank accounts there. The tax deductibility of donations across borders is also a big question. Similarly, if the majority of members in a trade association are from one region, it's probably best to form an entity there.

Finally, some grants may be restricted to organizations in certain regions. For example, the European Commission has a number of funding initiatives and some are limited to organizations based in the EU.

Apart from these factors, the jurisdiction doesn't seem to matter that much. The specific rules for non-profit organizations are different in each jurisdiction, but every place has regulations that the organization has to follow and comply with.

Eclipse Foundation's move to Europe

The Eclipse Foundation serves a global community. While the organization has an office in Canada and employees in Germany, it was incorporated as a trade association in the US state of Delaware.

In 2021, the Eclipse Foundation moved its corporate home to Europe by establishing an international non-profit association in Belgium:

> The Eclipse Foundation already has more than 170 members and over 900 committers based in Europe. The formalization of the Eclipse Foundation AISBL non-profit association will enable it to leverage its EU presence and recent international growth to foster global industry collaboration on open source projects in strategic technologies, such as the cloud, edge computing, artificial intelligence, connected vehicles, telecommunications, and the Internet of Things.

30 Governance

The majority of FOSS foundations have a board of directors that governs the organization. The role of the board is to define the vision of the organization and to provide oversight.

Dirk Riehle and Sebastian Berschneider describe three ways in which board membership is determined:

- Democratic: directors are elected by members of the organization
- Meritocratic: status is earned through continued project work
- Autocratic: the board selects new directors

Some organizations have members that can participate in elections and other decision-making processes.

Organizations with staff are usually led by an Executive Director (ED), a role similar to the CEO of a company. The Executive Director works closely with the board to define the vision, implements that vision, manages staff, and oversees the daily operations.

It's important to remember that the governance of the project and that of the organization are distinct, even though there can be a great overlap between the philosophy, approach, and the people who are involved.

Director responsibilities

Directors (members of the board of directors) have a fiduciary duty toward the organization. A fiduciary duty describes a relationship between two parties that requires one to act solely in the interest of the other. Directors have to act in the interest of the organization.

There are typically three legal duties:

- Duty of care: directors must show a certain level of competence and diligence, for example by attending meetings and staying up-to-date with the organization's finances and activities.
- Duty of loyalty: directors must act in the best interest of the organization.
- Duty of obedience: directors must ensure the organization follows and adheres to its mission, governance policies, and any laws or regulations.

The Charities Bureau of New York State published the document "Right from the Start" which outlines the responsibilities of directors.

Home

31 Life cycle

The needs of projects change throughout their life cycles. A young project typically has few needs (such as raising donations and holding a domain name). As projects grow, so do their needs: they may want to apply for trademarks, organize conferences and developer meetings, hire contractors, and more. Some projects that are part of an umbrella organization might also outgrow their host at some point and find enough reasons to justify setting up their own organizations.

Tony Sebro describes three different roles that umbrella organizations sometimes play:

- Playpen: incubation for young projects
- Apartment: a long-term living arrangement for growing projects
- Mausoleum: a place to put assets for inactive projects

It's important for projects to consider what needs they currently have and how those needs might change in the future.

jQuery and the jQuery Foundation

The jQuery project joined Software Freedom Conservancy, an umbrella organization, in 2009. In 2012, Conservancy and the jQuery board announced the creation of the jQuery Foundation. The jQuery Foundation was created as a trade association with the aim of supporting the development of jQuery, providing documentation and support, and fostering the jQuery community:

> "We are proud that the jQuery Board has built jQuery into a vibrant and successful Open Source community under Conservancy's mentorship," said Bradley M. Kuhn, Executive Director of the Software Freedom Conservancy. "Our mission includes helping member projects determine whether to form their own organization, and we're pleased jQuery is the first Conservancy project to take that step."

32 Costs and administrative burden

Developers greatly underestimate the administrative burden of running an organization. There have been several examples over the years of FOSS foundations running into various operational issues, such as problems due to unfiled paperwork.

The Linux Foundation observes that “overhead goes up exponentially“ and that “maintainers need to deal with issues they never anticipated”:

- Setting up a legal entity, which can take months of work and legal costs, then the ongoing maintenance of the entity and its associated state or federal filings
- Setting up a bank account, obtaining checks, handling accounting, and establishing payroll, as well as administering benefits and hiring staff
- Getting legal agreements signed by the sponsoring companies
- Filling out paperwork and forms to be set up as a supplier with a sponsoring company
- Dealing with invoices, purchase orders, and procurement departments
- Setting up a financial reporting process, so stakeholders have visibility into where the funds went

It’s important to take the costs and burden of running an organization seriously, to talk to lawyers and accountants to fully understand the responsibilities, and to think about the long-term implications and obligations. For the majority of projects, the costs of creating a new organization outweigh the benefits.

It’s therefore important to evaluate existing organizations first to see whether one of them would be a suitable home.

Foundation as a service

Umbrella organizations serve many individual projects, but what if several projects within a community want to join together as a foundation?

Several organizations follow a *foundation as a service* model whereby they provide operational support for other foundations.

The Linux Foundation is one example of such an organization. It hosts initiatives and organizations, such as Let's Encrypt, the Cloud Native Computing Foundation (CNCF), and the OpenJS Foundation. The Linux Foundation describes itself as "a federation of foundations" and it "provides a back-office infrastructure for projects and initiatives so they can focus on delivering value for their community".

Software in the Public Interest, Inc. (SPI) is another organization that hosts (virtual) foundations. The X.Org Foundation gave up its legal entity and became a virtual organization under SPI. It kept its own governance structure (including its board) intact while using SPI as the legal entity. The Open Bioinformatics Foundation also operates as a virtual organization with the support of SPI.

Finally, several FOSS foundations use the Open Collective platform for crowdfunding for their projects.

33 Build or join?

As part of deciding whether it's better to create a new organization or work with an existing one, the project needs to perform some self-reflection and think about diverse topics, including the project's vision, roadmap, culture, requirements, and more.

There are a number of specific topics that need consideration when evaluating whether to start a new organization versus joining an existing one:

- Needs: what needs does the project have now and how will those change in the future as the project moves through different phases? What services do different umbrella organizations offer?
- Culture: what are the values of the project? Are there existing organizations that have similar values where there is a cultural fit?
- Sustainability: does the project have the resources to run its own organization? What have other organizations done to ensure their sustainability?
- Level of integration: is there a preference to have a loose relationship with another organization, integrate closely with an existing organization, or maintain full control by operating a new organization?

These high-level questions may help guide the decision-making.

34 Finding a home

If a project decides to collaborate with an existing umbrella organization, there are a number of questions that should be asked to find a suitable organizational home for it:

- Joining
  - What obligations are there?
  - What fees are there?
  - What is the process to join?
  - Does joining offer alignment with other projects?
- Intellectual property
  - Does the organization require a specific FOSS license?
  - Does the organization impose a specific governance structure?
  - Does the organization require copyright assignment?
- Culture
  - Is there philosophical alignment in terms of software freedom and proprietary software?
  - What values does the organization promote?
  - Is there a cultural fit?
  - How loose or close is the relationship?
- Governance
  - Is the organization stable?
  - Does the organization have enough resources to support the project?
  - How transparent is the organization? Are governance structures documented?
- Services
  - Does the service menu match the needs of the projects?

  - Are contractors possible? Are employees possible? Are there limitations in terms of the location of staff?
  - Can projects participate in fundraising campaigns?
  - Does the organization offer marketing activities?

- Leaving
  - What is the process of leaving the organization?
  - Can assets (such as trademarks) be transferred out?

35 Resources

The FLOSS Foundations community is a useful resource to learn about FOSS foundations. It maintains a foundation directory, which is a great place to find organizations to examine. There's also a mailing list where people involved in FOSS foundations share their knowledge and experiences.

The Choose A Foundation website offers some guidance on choosing a foundation, but it currently only lists a small number of FOSS umbrella organizations.

The TODO Group offers a number of open source guides. Similarly, GitHub maintains some open source guides.

FOSS Governance Collection provides an indexed collection of governance documents from open source projects.

Appendix

A Acknowledgments

This primer was sponsored by Ford Foundation and Alfred P. Sloan Foundation. It's based on research on FOSS foundations that was conducted as part of their Critical Digital Infrastructure Research initiative, which investigates the role of open source in digital infrastructure.

The Center for the Cultivation of Technology provided operational support and insights throughout this research.

I'd like to thank the following people for valuable comments and insights: Moritz Bartl, VM (Vicky) Brasseur, Jake Edge, Gijs Hillenius, Carles Pina i Estany, Allison Randal, Karen M. Sandler, Daniel Vetter, and Stefano Zacchiroli.

I'd like to thank Jake Edge for copy-editing the primer.

I'd like to thank all FOSS foundations that participated in this research by sharing their valuable insights!

This primer was typeset with LaTeX using the kaobook class.

B Image credits

- Cover page: “Comouflage“, by Stefano Avolio, licensed under CC BY-SA 2.0, cropped from original
- Preface: “Flowers“, by Martin Höst, licensed under CC BY 2.0, cropped from original
- Contents page: “Artist book“, by Raphael Perez, licensed under CC BY 2.0, cropped from original
- Chapter 1: “Brooklyn, Manhattan and Williamsburg Bridges”, by John Cunniff, licensed under CC BY 2.0, cropped from original
- Chapter 2: “Matryoshka Madness”, by Michael Carian, licensed under CC BY-SA 2.0, cropped from original
- Chapter 3: “Ficus macrophylla (Moreton Bay fig), Botanic Park, Plane Tree Drive, Adelaide, South Australia”, by John Jennings, licensed under CC BY 2.0, cropped from original
- Chapter 4: “Carcassonne meeples, or followers”, by Júlio Reis, licensed under CC BY-SA 2.5, cropped from original
- Chapter 5: “Waterpass”, by Dmitry Makeev, licensed under CC BY-SA 4.0, cropped from original
- Chapter 6: “División, by candi..., licensed under CC BY 2.0, cropped from original
- Chapter 7: “Red elephant in Tsavo East National Park”, by Byrdyak, licensed under CC BY-SA 4.0, cropped from original
- Chapter 8: “Chestnut-tailed starling (Sturnia malabarica) on Indian coral tree (Erythrina variegata), Satchari National Park”, by Touhid biplob, licensed under CC BY-SA 4.0, cropped from original
- Chapter 9: “Wisdom is expecting”, by Kristina McOmber/Kupu Conservation Leadership Program & USFWS, public domain, cropped from original
- Chapter 10: “An older man at the 2019 Kumbh Mela”, by Balone21, licensed under CC BY-SA 4.0, cropped from original
- Chapter 11: “Spice Bazaar”, by Bit Boy, licensed under CC BY 2.0, cropped from original
- Chapter 12: “Kenya dancing”, by Kate Holt/Africa Practice, licensed under CC BY

2.0, cropped from original

- Chapter 13: “Doctors Chart – Hospital Chart”, by Weiss & Paarz, licensed under CC BY-SA 2.0, cropped from original
- Chapter 14: “Girls rolling coins for today’s tip share”, by Morgan, licensed under CC BY 2.0, cropped from original
- Chapter 15: “Money Laundering”, by Pixabay, licensed under CC0 1.0, cropped from original
- Chapter 16: “Group photo of the Wikimedia Summit 2019”, by Jason Krüger for Wikimedia Deutschland e.V., licensed under CC BY-SA 4.0, cropped from original
- Chapter 17: “Keyboard”, by Dan Foy, licensed under CC BY 2.0, cropped from original
- Chapter 18: “Ship’s wheel”, by liesvanrompaey, licensed under CC BY 2.0, cropped from original
- Chapter 19: “Mullioned window”, by David Brierley, licensed under CC BY 2.0, cropped from original
- Chapter 20: “Code on computer monitor”, by Markus Spiske, licensed under CC0 1.0, cropped from original
- Chapter 21: “Coke bottles”, by Toshiyuki IMAI, licensed under CC BY-SA 2.0, cropped from original
- Chapter 22: “Don’t stop”, by Martin Fisch, licensed under CC BY 2.0, cropped from original
- Chapter 23: “A car accident in Tokyo, Japan”, by Shuets Udono (derivative work: Torsodog), licensed under CC BY-SA 2.0, blurred plates (by Torsodog), cropped from original
- Chapter 24: “Writing in Russian cyrillic script with a fountain pen”, by Petar Milošević, licensed under CC BY-SA 4.0, cropped from original
- Chapter 25: “Illuminated wooden shelf with many glass jars containing cookies for sale”, by Basile Morin, licensed under CC BY-SA 4.0, cropped from original
- Chapter 26 “A menu for a restaurant”, by Jakub Kapusnak, licensed under CC0 1.0, cropped from original
- Chapter 27: “Souq Waqif, Doha, Qatar”, by Diego Delso, delso.photo, licensed under CC BY-SA 4.0, cropped from original
- Chapter 28: “Stonehenge”, by xavier33300, licensed under CC BY 2.0, cropped from original
- Chapter 29: “Table Bay, Cape Town”, by AerialcamSA, licensed under CC BY-SA 4.0, cropped from original
- Chapter 30: “Clet traffic signs”, by Project-128, licensed under CC BY 2.0, cropped from original
- Chapter 31: “Hmong Tribe”, by Rod Waddington, licensed under CC BY-SA 2.0, cropped from original
- Chapter 32: “Coolies pushing up a heavy load, Kolkata, India”, by Jorge Royan, licensed under CC BY-SA 3.0, cropped from original
- Chapter 33: “Modern Road construction”, by Tukheri Alfred, licensed under CC BY-SA 4.0, cropped from original
- Chapter 34: “Stretch”, by Julie, licensed under CC BY 2.0, cropped from original

- Chapter 35: “Globe”, by Rob Bertholf, licensed under CC BY 2.0, cropped from original
- Chapter A: “Thank You”, by Dennis Sylvester Hurd, licensed under CC0 1.0, cropped from original
- Chapter B: “Credit”, by GotCredit, licensed under CC BY 2.0, cropped from original

Copyright

License

Bugs

Please submit bugs and fixes to https://github.com/tbm/foss-foundations-primer