

Open Source Governance Practices: A Path to Improving Sustainability

Dawn Foster , Independent Consultant

This article provides advice and resources for improving open source project sustainability through good governance practices.

Many maintainers and contributors are overworked and experiencing burnout due to the increasing expectations and pressures associated with their work in open source,¹ so sustaining open source projects over the long-term can be a challenge. There are many ways to think about sustainability, some of which were covered in a previous article, “From Data to Action: Building Healthy and Sustainable

when a project is in crisis, it will be more difficult to come to an agreement about governance.

Good governance isn't just tedious paperwork. Governance that clearly defines a project's decision-making processes and sets expectations can make it easier for people to successfully contribute. Much of the focus of open source project governance is on people: the roles they play, their responsibilities, how they make decisions, and what

Open Source Projects.”² Good governance practices^a are another way to increase sustainability and avoid threats that can disrupt open source projects.³ Being proactive about defining governance early before a crisis or another threat occurs can make an open source project more sustainable. The time to work on governance is as close to the start of a project as possible when things are going well. Waiting to adopt a governance model until there is a crisis within a community likely means that the project won't have processes for dealing with that crisis, and



FROM THE EDITOR

How do you avoid maintainer burnout in open source projects? It turns out to be a matter of good governance, as this column's author Dawn Foster explains. If you want to be successful as an open source project and still have a life, you need to be strategic and consistent about your project governance practices. Wise words, and practical at that. Be happy reading and healthy too. —Dirk Riehle

they should expect from each other as part of participating in an open source community. Having clear rules about how collaboration occurs, how decisions are made, and what types of contributions are in or out of scope helps community members make contributions that are likely to be accepted and embraced by a project. This facilitates the creation of an intentional culture and helps avoid wasting time from maintainers with contributions that aren't aligned with the project. It also helps create pathways to leadership where other people can better understand the processes for moving into leadership roles along with intentional processes for how the project promotes people into leadership.

GOVERNANCE MODELS

A good practice is to start with the simplest possible governance model and only move to something more elaborate when a project evolves to the point where additional complexity is needed. This is because overengineered governance processes create additional overhead and extra work that would take time away from other aspects of project development.

If an established project doesn't already have documented governance

and decision-making processes, a good place to start is by documenting how the project is already making decisions and reflecting on how the existing governance could be improved. It can help to reflect on a few questions when thinking about a project's existing processes:

- › Who currently makes decisions and approves contributions?
- › Who provides input?
- › How can new people be included in this process over time?
- › What works well already?
- › What processes could be improved?

Many open source projects start with some form of maintainer council governance model (for example, Jaeger^b), which is a simple form of governance. In this model, the existing maintainers make the decisions in the project, including decisions about promoting additional people into maintainer roles. As a project evolves, there are other, more complex forms of governance that might be used as a project grows to the point where it isn't practical for a single group of maintainers to make decisions across a large project. Many mature projects often evolve to the point where a steering committee that is elected by the contributor community is needed to make project-wide decisions that span across multiple groups (for example, Knative^c). Projects that have evolved to include a collection of quasi-autonomous subprojects might need a dual level structure where each subproject can have internal leadership along with leadership that helps oversee the collection of projects (for example, Operator

Framework Project^d). The Cloud Native Computing Foundation (CNCF) has developed a robust set of templates and instructions for each of these governance models^e that can be adapted for use by any open source project.

CONTRIBUTORS AND PATHWAYS TO LEADERSHIP

An important aspect of governance that can often be overlooked is the impact that good governance practices have on contributor sustainability.^f Many projects want to recruit more contributors, but projects may also underestimate the effort it can take for existing maintainers to review and approve contributions; thus, contributor growth should be paired with governance structures that make it easy to promote contributors into maintainer roles to handle this additional workload.⁴ Defining the roles and responsibilities for contributors, reviewers, and maintainers can help with recruiting new people into these roles.

It can help to think of this as a ladder where contributors climb up to become reviewers and those reviewers can become maintainers. A contributor ladder usually outlines the different contributor roles within the project, along with the responsibilities and privileges that come with them. Community members generally start at the first levels of the ladder and advance up it as their involvement in the project grows. This helps set expectations for the roles and encourages people to think about how they might take on increasing responsibility within the project. Existing maintainers can also actively encourage some of the best contributors to move up the ladder

^d<https://github.com/operator-framework/community/blob/master/GOVERNANCE.md>

^e<https://contribute.cncf.io/projects/best-practices/governance/templates/>

^f<https://chaos.community/practitioner-guide-contributor-sustainability/>

^b<https://github.com/jaegertracing/jaeger/blob/main/GOVERNANCE.md>

^c<https://github.com/knative/community/blob/main/STEERING-COMMITTEE.md>

into roles with increasing responsibility. As more people move into maintainer roles, it can help to reduce the workload for the existing maintainers. The CNCF has a contributor ladder template with instructions,^g and examples of implementation can be found in Kubernetes^h and Copacetic.ⁱ

Maintainers often experience an increasing demand to manage noncoding tasks as open source projects increase in usage and popularity, which puts additional pressure on maintainers who are already overworked.⁵ Maintainer roles can be created for tasks like documentation, community management, marketing, product management, and other important roles.^j Contributors with subject matter expertise can often do the activities associated with those roles more efficiently and with higher quality than a maintainer without that specific expertise.

CREATING INTENTIONAL CULTURE

A documented governance model along with pathways to leadership are key to creating an intentional culture for an open source project that encourages participation and contributions from others. As part of governance, the scope, vision, and values (sometimes called a charter^k) can help potential contributors understand what is/is not in scope so that contributions are more likely to be accepted to avoid wasting maintainer time on contributions that aren't aligned with the project. Decision-making processes are another aspect of intentional culture. Many projects operate primarily by consensus, but it can be helpful to also have a provision for calling for a vote when consensus cannot be reached or

in other specific situations. This all helps to avoid issues and misunderstandings by helping align the expectations of community members.

However, just documenting the governance process and setting expectations in writing isn't enough. Open source project leaders should also role model good behavior and help others understand what behavior is appropriate. Tolerating bad behavior unwittingly sets the expectation that this behavior is acceptable, which can drive new contributors away, so addressing concerns promptly before they can get out of control or become a crisis helps set an intentional culture and improve sustainability. A code of conduct is a good starting point for these conversations, but it's also important to think about how to approach code of conduct issues and how remediation and education can be a first step to help people meet the project's expectations, instead of taking an enforcement first approach.

Unfortunately, some projects eventually run into issues that can't be resolved via remediation or education, and in these cases, it may be necessary to remove someone from a project. Good governance documents should have a provision for involuntarily removing someone from a project, even if that person is in a top leadership position. Projects hope to never need to use the removal process, but a project will be glad to have this in place early if it's ever needed in the future.

PROJECT OWNERSHIP

Some people think of open source project ownership as a bit nebulous with projects owned by an undefined "community" of people; however, this is rarely the case from a legal or practical standpoint. While there are a few exceptions, open source projects usually have an individual, company, or foundation controlling the trademarks, project infrastructure, and other assets. This overall ownership structure often impacts how the project is governed on a day-to-day basis and how the project is perceived by others.

For projects owned by individuals, there are often concerns about succession planning and the impact on the project if something happened to that individual. This can result in concerns about the sustainability and risks associated with using projects owned by a single individual.

For company-owned open source projects, if the project is truly open, the governance processes should specify how people can move into leadership positions with the assumption that the projects will eventually have maintainers from outside of the leading company. However, some corporate open source projects do not allow people to move into leadership positions or make decisions from outside of the company. In this case, a company should be honest in their governance documentation about how others can or cannot participate, rather than making all the big decisions behind the closed doors of their company while claiming to allow anyone to participate.

Some open source projects are owned by foundations, which tend to be nonprofit organizations that are focused on providing the resources and organizational structures that projects need to get funding and legal support for their work. Neutral foundations provide a level playing field where contributors can contribute as equals regardless of whether they are contributing on behalf of a company or as an individual. This structure allows companies to collaborate together in a neutral environment where no single company is in control of the project.

In many cases, open source projects are donated to foundations by individuals or companies. The challenge with contributing open source projects to foundations is that those individuals and companies give some of their control to the foundation. Typically, the project's trademarks, repositories, websites, and other assets would be transferred to the foundation. Foundations often make sure that

^g<https://contribute.cncf.io/projects/best-practices/community/contributor-growth/incentivizing-contributors/>


^h<https://github.com/kubernetes/community/blob/master/community-membership.md>

ⁱhttps://github.com/project-copacetic/copacetic/blob/main/CONTRIBUTOR_LADDER.md

^j<https://github.com/goharbor/community/blob/main/MAINTAINERS.md>

^k<https://contribute.cncf.io/projects/best-practices/governance/charter>

the governance is set up in a way that makes it easy for anyone to participate and eventually move into leadership positions. While this does mean giving up some control over the project, putting a project under a foundation gives others more confidence that they can contribute as equals. Most foundations have so many projects that putting a new project into a foundation may not get as much attention as people expect, and the foundation's attention will be spread across all their projects. The biggest misconception about putting a project under a foundation is that the project will get an immediate boost with more contributors and more adopters, but this requires hard work from the people already participating in the project, so it's important to set expectations appropriately. It's also a permanent decision, and it's not likely that a project can be taken back after it has been put into a foundation, so it's worth taking extra time and doing some research to decide whether to continue to own the project or whether it would be better to put it into a foundation.

Open source project governance can have a powerful impact on sustainability and project success. Documented governance that includes pathways to leadership can improve the contributor experience while reducing the workload of existing maintainers. Using governance to create an intentional and welcoming culture with values that are upheld and role modeled by leadership encourages participation and builds community. Considering the ownership of a project can help projects evolve to better serve the needs of their users and contributors. Good governance practices can make the difference between project failure and success that is sustained over time. 

REFERENCES

1. C. Miller, D. G. Widder, C. Kästner, and B. Vasilescu, "Why do people give up flossing? A study of contributor disengagement in open source," in *Proc. 15th IFIP Int. Conf. Open Source Syst. (OSS)*, Montreal, QC, Canada, 2019, pp. 116–129, doi: [10.1007/978-3-030-20883-7_11](https://doi.org/10.1007/978-3-030-20883-7_11).
2. D. Foster, "From data to action: Building healthy and sustainable open source projects," in *Computer*, vol. 58, no. 6, pp. 74–78, Jun. 2025, doi: [10.1109/MC.2025.3530556](https://doi.org/10.1109/MC.2025.3530556).
3. V. Mindel, L. Mathiassen, and A. Rai, "The sustainability of polycentric information commons," *MIS Quart.*, vol. 42, no. 2, pp. 607–632, 2018, doi: [10.25300/misq/2018/14015](https://doi.org/10.25300/misq/2018/14015).
4. N. Eghbal, *Working in Public: The Making and Maintenance of Open Source Software*. San Francisco, CA, USA: Stripe Press, 2020.
5. J. Linåker, G. Link, and K. Lombard, "Sustaining maintenance labor for healthy open source software projects through human infrastructure: A maintainer perspective," in *Proc. 18th ACM/IEEE Int. Symp. Empirical Softw. Eng. Meas.*, 2024, pp. 37–48, doi: [10.1145/3674805.368666](https://doi.org/10.1145/3674805.368666).

DAWN FOSTER is an open source strategy consultant in London, U.K. Contact her at dawn@dawnfoster.com.



IEEE SECURITY & PRIVACY

IEEE *Security & Privacy* magazine features articles on practical application and research from leading experts in the field.

Discover Tools, Techniques, and the Latest Research
computer.org/security

IEEE COMPUTER SOCIETY

IEEE

Digital Object Identifier 10.1109/MC.2026.3690224