

Setting up governance for an open source project

Executive summary

Governance is sometimes perceived as a large set of rules to apply to a project, but while governance for communities can be large and complex, often it helps to remember that governance is, simply put, “what do community members do when something must be decided?” Successful open source project communities will want to establish a governance early in the process. A few basic steps to take to structure a community’s governance—whether launching a new project or evolving one that is already active—can be found in this whitepaper.

Governance for Communities

Most governance models comprise two primary dimensions: roles and policies and procedures. The basic requirements here are actually quite spartan and can be evolved as the project grows. This document outlines the *minimum viable product* for project governance.

Defining Roles

As mentioned, a project will have a variety of real roles, but only a handful need to be defined to start out. Those basic roles are:

- ▶ Member
- ▶ Contributor
- ▶ Leader

Most existing projects already feature all these roles in some form. Each one of them should be recorded in a role document, either in a project’s documentation or in its main source code repository. This helps to create explicit guidelines, both setting expectations for and allowing more people to participate in the project. For each role, define who they are, how they qualify for that role, what they are expected to do, and what are their rights and privileges.

Members

This is possibly the least-documented role in open source projects, despite being the most pervasive. Members are the people or organizations who take part in the project and are recognized for it. Depending on how the project is run, these can be subscribers on a mailing list, sponsoring companies, known end-users, participants at an event, or members of a foundation. In some projects, member is synonymous with contributor, but this is not always the case. Most projects have a much larger group of people who are involved with the project, but are not actively contributing code or content to it.

Defining who members are requires deciding who the project is actually serving, which is always a critical discussion to have. Are customers of the main sponsoring company automatically project members? Can companies be members or only individuals? Are end-users members, or can only contributors be members? More than anything, defining members means defining whom the project leaders need to listen.

For almost all projects, specify what rules members are subject to (usually a code of conduct and not much else) and what they can expect from leaders and contributors. It is particularly helpful to explain how members should participate in the project, such as “Members file bugs against this repository, and use the ‘new bug’ template.” Most people, given explicit instructions, are happy to channel their participation in the routes shown to them.

Contributors

Far more projects have a written definition of contributors, but fewer than one might expect. It is often assumed, in the age of publicly hosted source code control, that anyone in the GitHub or GitLab statistics is counted as contributors. But defining “who is a contributor to this project” can be deceptively hard.

Is it anyone who posted on a mailing list, or does one need 100 merged pull requests to qualify? Is it just code contributors or contributors of any kind? What about folks who do events and advocacy? Are staff who work for a contributing company automatically considered contributors, or do they have to earn it individually? What about someone who contributed a lot of code three years ago but not since then? Who gets listed in project release credits, and how?

The contributor role is also one for which many more expectations will need to be set for what contributors receive for their work. This not only explains the intellectual property rules of the project (e.g., does the contributor still own their code), but also questions like how soon a contributor can expect their submissions to be reviewed and accepted or rejected. How the contributor will be credited for their participation should also be explained.

Leaders

As noted, every project has leadership, even when those leaders are not clearly identified. At a minimum, it will be necessary to identify who the leaders are so that decision-making processes can be clear. Many projects also explain the qualifications and procedure to become a leader, whether it is selection by a committee, election, or simply based on a specific job. If working on a more politically sophisticated project, then those should be written in a selection or election procedure document, but if it's simple, selection can just be part of the role document.

What fewer projects put into their leadership role documents are the other parts: the powers and limitations of the leaders, their duties, and how people leave the role (voluntary or otherwise). It is very important that everyone knows exactly how far a leader's authority extends, as well as what they are responsible for, or a project will end up with a lot of conflict between leaders and other project members. Having a set of written duties helps immensely when a leadership team has to decide to remove a project leader who has stopped participating but does not want to resign.

If a project is trying to recruit new and additional leaders, then it is also important to have a detailed set of qualifications a leader needs to meet. Contrary to some expectations, having detailed qualifications gives people who want to move up in the project a goal to aim for.

Setting policies and procedures

In addition to some basic role documentation, there is a certain amount of basic paperwork that each project should create for itself. These policy and procedure (P&P) documents are considered a kind of minimum for what a community needs in order to grow and mature a project. A project may, and eventually will, have other P&P docs as its contributor base expands, and the number of processes needed to write down will expand as well.

Some of these will be mostly technical (like a release process or a support policy), and we will not be exploring those here.

However, there are three minimum governance P&P documents that every project should have:

- ▶ Code of conduct
- ▶ Contribution process
- ▶ Communication information

Developing a code of conduct

Creating a code of conduct (CoC) for an open source community is one of the simplest and most powerful ways to influence the project's governance model. A CoC is a description of expectations for community members' behavior when they act within or on behalf of the project. It might outline the values a community agrees to uphold, articulate the behaviors community members expect one another to exhibit in the service of those values, and identify the consequences of violating the code. The most effective CoCs are those written through collaborative processes that involve participants across the community (not just project leadership). In this way, constructing a CoC can become a compelling community-building exercise.

Here are the core items that every CoC needs to have:

- ▶ A statement about what kind of behavior is encouraged
- ▶ A statement of what kinds of behavior are prohibited
- ▶ Contact information for reporting violations
- ▶ A description of the enforcement mechanism

When starting out, both the report recipients and the enforcers of the CoC are likely to be the project's founders. As a project grows, forming a specific CoC committee may have to happen, but not right away.

Contribution process

In order to recruit contributors, they will need to be told the basics of how to contribute to the project. For projects on GitHub or GitLab, this is placed in a document called `CONTRIBUTING.md`, but it can go anywhere as long as it is linked from the project's home page. If the contributor role is documented, just use that for the contribution docs. If not, here are a few things to be covered in a contribution document:

- ▶ Where to communicate with other contributors
- ▶ How to submit first code, documentation, or other contribution
- ▶ Any testing or formatting requirements

- ▶ What to expect from the review process
- ▶ When they qualify for membership and contributor status

Some projects have paperwork that needs to be submitted before any contributions can be accepted, such as a Developer Certificate of Origin or Contributor License Agreement, certificate of identity, or GNU Privacy Guard (GPG) keyring. Spell these out with step-by-step instructions in the contribution document.

Communication information

Most open source projects have multiple ways that project members talk to each other, including email, chat, issues, code reviews, video conferencing, and even in-person meetings. The channels a project uses and how to join them need to be spelled out. It is also important to keep this information up to date.

If available, it is useful to list both user and member forums and the channels used for contributors so that people know where to take their questions. Distinguish the media used for official project business as opposed to unofficial channels used for general discussion. It is extremely frustrating for contributors to be told about decisions made on the mailing list if they were unaware there was a mailing list. Any regular meetings should link to a calendar or at least information about the next meeting. And if the community has any important events, such as an annual conference, mention those as well.




Author biography

Brian Proffitt is a Senior Manager within Red Hat's Open Source Program Office, focusing on content generation, community metrics, and foundation relationships. Brian's experience with community management includes knowledge of community onboarding, community health, and business alignment.



About Red Hat

Red Hat is the world's leading provider of enterprise open source software solutions, using a community-powered approach to deliver reliable and high-performing Linux, hybrid cloud, container, and Kubernetes technologies. Red Hat helps customers develop cloud-native applications, integrate existing and new IT applications, and automate and manage complex environments. [A trusted adviser to the Fortune 500](#), Red Hat provides [award-winning](#) support, training, and consulting services that bring the benefits of open innovation to any industry. Red Hat is a connective hub in a global network of enterprises, partners, and communities, helping organizations grow, transform, and prepare for the digital future.

 facebook.com/redhatinc
 @RedHat
 linkedin.com/company/red-hat

North America
 1 888 REDHAT1
 www.redhat.com

**Europe, Middle East,
and Africa**
 00800 7334 2835
 europe@redhat.com

Asia Pacific
 +65 6490 4200
 apac@redhat.com

Latin America
 +54 11 4329 7300
 info-latam@redhat.com