A guide to open source project governance models

When initially engaging with an open source project, you will need to determine which of that project's participants has the authority to help you make a contribution. In open source software projects, the rules and customs that define who gets to do what – and how they are supposed to do it – is called a project's governance model. Understanding those rules and customs increases your chances of contributing successfully and positively to a project.

Every open source project and community operates according to a governance model, though some projects make their models more explicit than others do. This document describes some of the most common open source project and community governance models and offers guidance on getting started in projects that have adopted each model.

Do-ocracy

Open source projects adopting the do-ocracy governance model tend to forgo formal and elaborate governance conventions and instead insist that decisions are made by those who do the work. In other words, members of a do-ocracy gain authority by making the most consistent contributions. Peer review remains common under this model, but individual contributors tend to retain de facto decision-making power over project components on which they have worked most closely.

For this reason, some do-ocracies will claim they have no governance at all, relying instead on individual stakeholders' authority to make decisions on matters where they have done the most work. But such claims about an absence of governance are misguided. Every open source project has a governance model. In the case of most do-ocracies, the governance model is merely implicit in the everyday interactions of project members. As a result, joining them can be difficult and intimidating for newcomers, as would-be contributors might not immediately know how to participate or seek approval for their contributions.

To get started in a project with this governance model: Find an aspect of the project you feel you can improve and simply begin working. Review the recorded history of changes to the project to identify the participants whose feedback will be integral to your successful contribution. As the project accepts more of your contributions, you will gradually accrue influence in the community. Do not expect to influence decisions in a do-ocracy until you are able to demonstrate a history of successful contribution.

Founder-leader

The founder-leader governance model is most common among new projects or those with a small number of contributors. In these projects, the individual or group who started the project also administers the project, establishes its vision, controls all permissions to merge code into it, and assumes the right to speak for it in public. Some projects refer to their founder-leaders as *Benevolent dictators for life*. In projects following the founder-leader model, lines of power and authority are typically quite clear. They radiate from founder-leaders, who are the final decision-makers for all project matters.

This model's limitations become apparent as a project grows to a certain size. Separating founder-leaders' personal preferences from project design decisions eventually becomes difficult, and founder-leaders can become bottlenecks for project decision-making work. In extreme cases, founder-leader models can create a kind of caste system in a project, as non-founders begin feeling like they are unable to affect changes that are not in line with a founder's vision. Disagreements can lead to project splits. Worse, a founder-leader's disappearance, whether due to burnout or planned retirement, can cause a project to disintegrate entirely.

To get started in a project with this governance model: Browse project mailing lists or discussion forums to identify the project's founder-leaders, then address questions about participation and contribution to those leaders through one of the community's public communication channels. Founder-leaders tend to have a comprehensive view of the project's needs and will direct you to areas of the project that will benefit most from your contribution. Be sure to understand founder-leaders' vision for the project, as most founder-leaders will veto proposed changes they feel conflict with that vision. When starting out, do not expect to propose changes that will not serve the founder-leaders' vision for the project.

Self-appointing council or board

Recognizing shortcomings of the founder-leader model, the self-appointing council or board model aims to facilitate successful community leadership turnover and smoother succession. Under this model, members of an open source project may appoint a number of leadership groups to govern various aspects of a project. Such groups may have names like steering committee, committer council, technical operating committee, architecture council, or board of directors. Typically, these groups construct their own decision-making conventions and succession procedures.

The self-appointing council or board governance model is useful in cases where a project does not have a sponsoring foundation and establishing electoral mechanisms is prohibitively difficult. But the model's drawbacks become apparent when self-appointing governing groups grow insular and unrepresentative of the entire project community – as member-selection processes tend to spawn self-reinforcing leadership cultures. Moreover, this model can stymie community participation in leadership activities, as community members often feel like they must wait to be chosen before they can take initiative on work that interests them.

To get started in a project with this governance model: Because this governance model is typical of more mature open source projects, communities adopting this model will often curate *getting started* documentation aimed at assisting potential contributors. Find this documentation and read it first. Then read the project's governance documentation to determine how its governing bodies are composed. In many cases, you can locate a council or board governing the aspect of the project to which you would like to make a contribution. That body will be able to oversee your contribution and answer questions you may have.

Electoral

Some open source projects choose to conduct governance through electoral processes. They may hold elections for various roles, or hold votes to ratify or update project policies and procedures, for example. Under the electoral model, communities establish and document electoral procedures to which they all agree, then enact those procedures as a regular matter of decision-making.

This model is more common in larger open source projects where multiple qualified and interested contributors offer to play the same role. Elections are also common for projects with a sponsor, such as a foundation, because an electoral process can make the allocation of sponsor resources more transparent. Electoral governance also tends to lead to precise documentation of project roles, procedures, and participation guidelines. When election documents make these matters explicit, they help new contributors maximize their involvement in a project.

But elections also have drawbacks. They can become contentious, distracting, and time-consuming for all project members – whether or not those members are running. Some communities promote elections as a solution to the indefinite tenure of well-known project members, but elections do not generally cause turnover unless a project has explicitly codified term limits.

To get started in a project with this governance model: Communities appointing leaders through elections typically feature election results and a leadership roster prominently on their project websites. Review those documents to determine a point of contact in the project. Well-governed open source communities will make clear, also often on their project websites, their processes for proposing and reviewing items on which the community will vote. As you establish a reputation for making useful contributions to the project, you may eventually decide to be a candidate for a project leadership position. Be sure to interact productively and collaborate effectively with other contributors as they may be voting you into a leadership position some day.

Single-vendor

Occasionally, individual companies or industry consortia may choose to distribute software under the terms of an open source license as a way of reaching potential developers and users – even if they do not accept project contributions from those audiences. They might do this to accelerate adoption of their work, spur development activity atop a software platform, support a plugin ecosystem, or avoid the overhead required for cultivating an external developer community.

Under this model, the governing organization usually does not accept contributions from anyone outside it. Instead, open and closed source innovation occurs at the edges of the project. For this reason, some commentators call this the *walled garden* governance model. Occasionally, projects following this model will adopt licenses with strong copyleft requirements, which they see as a deterrent to commercial competitors benefitting from their work on the project. (The goal is to force competitors and customers with production requirements to purchase a non-open source license for the software – what some call a *dual license* approach.) This model becomes problematic in cases where a project claims to have an open community but is in fact wholly owned by a company or consortium.

To get started in a project with this governance model: First, consider any existing relationship between your employer and the company originating the project, if applicable. Next, assess the project's licensing terms and review its change history and bug tracker to determine whether you are able to contribute to the aspect of the project that interests you – and in the way you would like. Given the project's particular licensing stipulations, you may find yourself working alongside or on top of a particular project rather than contributing to it directly.

Foundation-backed

To exert greater control over resources and project code, some open source projects choose to be managed by an incorporated non-government organization (NGO), such as a charitable nonprofit or trade association. This approach allows the project to take ownership of resources like servers, trademarks, patents, and insurance policies.

In some cases, foundation leadership and project leadership can form a single governance structure that manages all aspects of the open source project. In other cases, the foundation manages some matters, such as trademarks and events, and other governance structures in the project(s) control other matters, such as code approval.

Extensive funding and legal requirements normally limit this model to larger open source projects. However, many smaller projects choose to join larger umbrella foundations, like the Software Freedom Conservancy or the Linux® Foundation, to reap some of the benefits of this governance model. This governance model is advantageous for projects seeking to establish legal relationships with third parties, like conference venues, or projects seeking to ensure successful leadership transitions following departure of key individuals. It might also help prevent the commercialization of the project under a single vendor.

High overhead – not strictly financial, but particularly in terms of contributor time, which can be substantial – is a significant drawback of the foundation-backed governance model. Some foundations are incorporated as industry consortia, in which sponsoring companies govern the organization. Different consortia allow different degrees of participation from individual project contributors. Some are fairly open groups, while in others, only corporate managers have authority.

To get started in a project with this governance model: If a foundation does not govern day-to-day project contribution activity, then locate the project's getting started documentation and follow it – see *Self-appointing council or board section*. Otherwise, note that individual projects under a particular foundation's umbrella will have their own sets of leaders, though some common guidelines may standardize basic contribution processes across all projects a foundation governs. To identify a specific project's leaders, consider addressing a request to the foundation members' mailing list. You might also examine the project's change history to identify frequent contributors (see *Do-ocracy* section) and contact them. As many foundations feature a contribution-based voting system, familiarize yourself with steps required to become a full voting member of the foundation. If the foundation is a members-only industry consortium, determine whether your employer is already a member. If not, talk to your manager about the importance of the project to your work and ask whether your employer might consider joining. In either case, foundation projects may require signing contributor paperwork. Your legal department should assist with reviewing and signing this paperwork.

To learn more, visit red.ht/OSPO.

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 integrate new and existing IT applications, develop cloud-native applications, standardize on our industry-leading operating system, and automate, secure, and manage complex environments. Award-winning support, training, and consulting services make Red Hat a trusted adviser to the Fortune 500. As a strategic partner to cloud providers, system integrators, application vendors, customers, and open source communities, Red Hat can help organizations 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