

<https://daffyduke.lautre.net/spip/?Understanding-open-source-governance-models>



Understanding open source governance models

- 6- Webographie -

Date de mise en ligne : dimanche 6 juillet 2025

Copyright © L'Imp'Rock Scénette (by @_daffyduke_) - Tous droits réservés

estimated reading time : 5 min

Open source projects usually operate according to rules, customs, and processes that determine which contributors have the authority to perform certain tasks. Understanding those rules can increase your chances of contributing successfully and positively to a project.

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 a project's governance model can help you make a successful, positive contribution to a project. To help you get started, then, we'll cover the following six open source governance models : "Do-ocracy" Founder-leader Self-appointing council or board Electoral Corporate-backed Foundation-backed

Note that many projects feature multiple governance models. For example, a project housed within a foundation may self-govern as a "do-ocracy" or have a board.

"Do-ocracy"

Open source projects adopting the "do-ocracy" governance model tend to ignore formal and elaborate governance conventions. Instead, they insist that those who do the work are those who make the decisions.

In other words : In a do-ocracy, participants who invest the most time, energy, and attention in specific aspects of the project have the most authority and influence over decisions in those areas of the project. Peer review is common under this model, but individual contributors tend to retain de facto decision-making power over project components they've worked on closely.

For this reason, some do-ocracies will claim to have no governance at all, but for most do-ocracies, the governance model is simply implicit in the everyday interactions of project members. As a result, joining them can be difficult and intimidating for newcomers who may not immediately know how to participate or seek approval for their contributions.

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, and controls permissions to merge code into it.

Some projects refer to their founder-leaders as "Benevolent Dictators for Life" or "BDFL" for short. In projects following the founder-leader model, lines of power and authority are typically clear ; they radiate from founder-leaders, who are the final decision-makers for project matters.

This model's limitations become apparent as a project grows to a certain size. Founder-leaders can become bottlenecks for project decision-making workâ€”and in extreme cases, the model can give rise to a kind of "caste" system in a project, as non-founders begin feeling like they're unable to affect changes that aren't in line with a

founder's vision.

Self-appointing council or board

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." These groups then maintain their own decision-making conventions and succession procedures. This model is useful in cases where a project does not have a sponsoring foundation and establishing electoral mechanisms is prohibitively difficult.

The model's drawbacks may become apparent in cases where member-selection processes spawn self-reinforcing leadership cultures. Occasionally, this model can hinder 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.

Electoral

Some open source projects choose to [conduct governance through elections](#) . They may hold formal elections, where people vote for candidates to fill various project roles, or conduct similar electoral processes to ratify or update project policies and procedures. Under this model, communities establish and document electoral procedures to which they 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. Electoral governance also tends to lead to precise documentation of elected project roles, procedures, and participation guidelines. Elections can have drawbacks when they become contentious, distracting, and time-consuming for project members. And unless a community has specifically codified term limits in its governance documentation, elections do not necessarily guarantee leadership turnover.

Corporate-backed

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 may not accept contributions from anyone outside it, or require a contributor agreement (CLA) to accept a contribution . For this reason, some commentators call this the "walled garden" governance model. Objections to this model arise if a project claims to support an open community but is in fact wholly controlled by a company or consortium. This can create mismatched expectations among adopters.

Foundation-backed

Some non-corporate-backed open source projects choose to be managed by a non-profit or trade association. Doing this ensures that a single project participant does not exert exclusive control of key project resources. 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, but other governance structures such as code approval or the management of the technical roadmap are handled by project leaders.

Funding and legal requirements normally limit this model to larger open source projects. However, some smaller projects choose to join larger "umbrella" foundations to reap some of the benefits of this governance model (in return for an administrative fee). This governance model can be 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.

Getting started

By analyzing an open source project's governance model, you demonstrate a commitment to acting as a conscientious contributor to that project. To learn more about open source project governance, [read our more extensive briefing document](#).