



# Open Source Program Offices

Dirk Riehle <sup>ID</sup>, Friedrich-Alexander-Universität Erlangen-Nürnberg

*An open source program office is a company organizational unit tasked with managing the use of, contribution to, and leadership of open source projects from the company's perspective. by way of policies, guidance, and education.*

company's business model is used and that vulnerabilities are avoided or at least managed. This form of engagement is therefore mostly about license compliance and supply chain security. Tools and tactics are focused inward. They include, for example, education, software composition analysis, and component managers.

**A**n open source program office (OSPO) often starts out as a single person who is tasked with reigning in the use of open source software in projects and products. Once the scope of the challenge becomes clear, the OSPO is set up as a new organizational unit. An OSPO is typically a central function, often located in the office of the chief technology officer.

The overall mandate of an OSPO can be laid out along the three dimensions of a basic maturity model of engaging with open source. This model consists of three stages:

1. *Using open source software in projects and products:* The main challenge of using open source software is to ensure that only software that fits the

2. *Contributing code to existing open source projects:* The main challenge of contributing to open source projects is to avoid the outflow of intellectual property that is competitively differentiating and therefore should be kept closed. This includes avoiding any contribution that signals important information to markets and the competition about the company's product strategy. This form of engagement is mostly about managing the dependencies on open source software. Tools and tactics include education of the company's open source contributors and outreach to open source communities.
3. *Creating and leading open source projects:* The main challenge is to identify the business opportunities and justify the costs that result from creating and leading open source projects. This includes prioritizing and aligning open source leadership with the strategic goals of the company. This form of engagement



**FROM THE EDITOR**

Welcome back! This month’s “Open Source” column continues discussing the fundamentals of practical open source use in organizations. To this end, we discuss open source program offices, the common name for the organizational unit tasked with “taking care of open source” and “ensuring that nothing goes wrong.” Of course, this is just the starting point for an organization’s open source engagement, as we’ll also see. Enjoy this new career perspective and keep on hacking! –Dirk Riehle

is mostly about industry collaboration for market positioning and managing revenue streams. It requires understanding of how industry dynamics play out, including but not limited to which features, components, or layers of the stack are ready for commoditization.

A company’s open source strategy consists of strategies for these three forms of engagement. The open source strategy, initially designed or at least facilitated by the company’s OSPO, is part of and has to align with the company’s overall business strategy.

For each form of engagement, the OSPO may have to set up and operate tools and metrics, perform internal and external marketing, train its employees, ensure compliance, manage risks, respond to crises, etc. All of these tasks come with often nontrivial workflows.

The scope of an OSPO’s mandate widens with the growing maturity of the open source understanding and engagement of the organization: from initially just using open source software, through contributing to open source projects, all the way to creating and leading open source projects. All of these activities need competent guidance, policies, and tooling.

**USING OPEN SOURCE SOFTWARE**

The first stage of engaging with open source is typically to use the software.

As explained in a previous column<sup>1</sup> in-stance, there are two main categories of users: end-users and distributors.

- › End-users worry about using quality software that, for example, will be maintained for a sufficiently long time and is not riddled with vulnerabilities.
- › Distributors, in addition to worrying about software quality, also worry about fulfilling the open source license obligations of the code they are incorporating into their products.

An OSPO supports the employees of the company in making the decision of which open source software to use and which not to use. Support can range from advising what to use to outright forbidding the use of a particular open source software.

The associated workflows can get laborious and grow in volume quickly. Efficiency is key. To this end, most OSPOs, based on the dominant business models of the company, will provide initial guidance to developers searching

for open source software to use. Table 1 shows how open source software can be prequalified as allowed, must-ask, or verboten. Software that falls into the “allowed” category is likely to be allowed for use in projects and products, software that falls under “must ask” will likely trigger stringent review and resolution, and software that falls under “verboten” is unlikely to be allowed at all (Table 1).

Employees who would like to use open source software that passes this filter will then have to feed it into an approval process operated by an open source review board (on behalf of the OSPO). The review board typically consists of experts from the program office and other parts of the company.

In a comprehensive approval process, the review board will perform a thorough analysis of the open source software to understand what’s in the package. The two main aspects of interest are code quality and software licenses.

- › To review software quality, the review board or selected members will run metrics tools on the code and the community to get an indicator of their quality and longevity. Example indicators are the number of outstanding bugs, mean time to bug fix, and diversity and size of the project community.
- › To review the software licenses, the review board will run software composition analysis tools to identify the licenses in the code

**TABLE 1.** Simple prequalification matrix for selecting open source components.

	Allowed	Must ask	Verboten
Software	SQLite	glibc	CoreNLP
By license	MIT	LGPL-2.0	AGPL-3.0 or later
By origin	github.com/google	github.com/random	Stackoverflow

(which may be different from the licenses declared by the developers). Often, developers name one primary license, while actual code has been composed from code of many different licenses over time.

The review board makes a recommendation or casts a decision based on these findings and the use of the open source software is approved or rejected. The review of licenses of incoming open source code is also often called the *inbound licensing process*.

### The outbound licensing process reviews whether a particular open source engagement hurts or strengthens the competitive position of the company.

For actual use of the open source software in projects and products, the software, together with the results of the review process, should be put into an open source software management system for retrieval when building software inside the company. A company should never pull open source software straight off the web into their projects and products: It should only use reviewed, approved, and managed versions of the software from a trusted system. Otherwise, it significantly increases the risk of falling prey to software supply chain attacks.

### CONTRIBUTING TO OPEN SOURCE PROJECTS

The second stage of engaging with open source is typically to contribute to an open source project. Most people and companies start contributing by filing bug reports for components they use in the projects and products. Both people and companies may or may not move on to contributing source code. There are many other ways of contributing to open source projects.

Corporate reasons for contributing to open source projects range from the tactical to the strategic.

Tactical reasons are mostly about code contributions that the company does not consider competitively differentiating. The three common types of contributions are 1) bug fixes, 2) refactorings, and 3) new functionality. Nondifferentiating code is better maintained by the open source project than the company. This makes it easier for the company to catch up with new versions of the open source software because the company doesn't have to repeatedly merge their closed

modifications into the open source software as it updates to a new version.

Strategic reasons are mostly about ensuring technical compatibility with and managing the dependencies of the company's projects and products on the open source software. Sometimes, an open source project does not have enough interest in a particular feature or technology to maintain it. If the company's projects or products rely on it, it must step up and start supporting and maintaining it or see the open source software go stale for it. Also, without a voice in the project, the open source project might take a left turn where the company wants it to go straight. To manage its technical dependencies and to ensure that nothing goes wrong, a company must pay in by actively contributing. This way, the company maintains its interests.

An OSPO, like in the case of incoming open source software, typically also operates a review and approval process for proprietary software to be contributed to open source projects. This is also called the *outbound licensing process*.

A company should only ever contribute code to open source software

projects that are competitively nondifferentiating for it. As part of an overall strategy, it may decide to let go of closed code that was once considered differentiating, but under the new strategy isn't any longer.

The outbound licensing process reviews whether a particular open source engagement hurts or strengthens the competitive position of the company.

- › The obvious reason to not contribute a software feature to an open source project is that customers are paying for it.
- › Another reason may be that the open source license requires a free patent grant that the company is not willing to provide.
- › Sometimes information that a company is depending on a particular open source software, as laid open by a contribution, is hurting its competitive position.

It is a common beginners' mistake to assume that the open source project exists to serve the company. It doesn't. The project exists for its own purpose. A company can't demand a bug fix, for example, even if the bug causes major pain. Similarly, a company can't provide a bug fix to a project and expect the project to maintain it. The bug fix might just linger and never get picked up if nobody is interested in it.

There is no guarantee that an open source project will behave the way a company wants it to. The company can ask nicely, but there is no guarantee that it will receive a response.

Consequently, a company needs to build both capability and credibility with an open source software and its project community. Then, not only will the company be able to provide good contributions, it may also be listened to. For open source projects, on which a company's projects and products critically depend, building such capabilities and credibility is a must.

## LEADING OPEN SOURCE PROJECTS

The third and most advanced stage of engaging with open source is to create and lead open source projects. This often correlates with a taking on a larger role in the open source ecosystem, most notably by joining and actively participating, sometimes leading, open source foundations.

There are many reasons for why companies create and lead open source projects. Three important reasons for doing so are:

1. lowering the cost of development of nondifferentiating software components by sharing the costs of development with other interested parties
2. establishing de facto standards through widely used open source software that works well with a company's projects and products, saving more costs
3. tapping into broad-scale innovation by the open source community in such a way that it benefits the company's complementary products.

Small hobby projects on GitHub or GitLab are created as quickly as they are abandoned. Creating successful, long-term viable open source projects that fulfill the creator's needs is a significant long-term investment and needs to be thought through from the beginning.

An OSPO collaborates with the main lines of business to identify their strategic needs for new open source projects and then helps them realize these projects.

A particularly important case of creating and leading new open source projects is the open sourcing of existing internal (closed) software. The OSPO works with the line of business to determine:

- › a proper home (on a company managed site or at an open source foundation)
- › the extent of open sourcing (what and what not to open source)

- › the extent of intellectual property made available (trademarks, patents)
- › a timeline including staffing, launch, marketing, etc.

Open source foundations are non-profit organizations with the purpose of hosting and furthering open source software. Such foundations are created to establish a fair and equal playing field for all parties interested in a particular open source software. A well-run open source foundation ensures that the investment of the involved parties into some open source software is safe.

Creating successful, long-term viable open source projects that fulfill the creator's needs is a significant long-term investment and needs to be thought through from the beginning.

Open source foundations are therefore the natural place for companies to go to and create new open source projects. Some of the large open source foundations have effectively become the host of whole platforms or layers of the technology stack that operate modern software systems. For example, the Apache Software Foundation is host to most of the open source data processing components, and the Cloud Native Computing Foundation is host to most of the managed cloud services components.

Open source foundations are natural partners to corporate OSPOs. The OSPO often provides nontechnical guidance and staffing, establishes and supports the integration of line-of-business representatives into the open source foundations, and coordinates the interaction across the ecosystem, for example, between the components of an open source platform at an open source foundation.

## GOOD GOVERNANCE CERTIFICATION

As discussed, open source governance at its core consists of governing

- › how and which open source software to use
- › how and when to contribute to open source projects
- › how and why to create and lead open source projects.

The OpenChain project, hosted by the Linux Foundation, is an attempt by industry to specify good open source governance of companies to make the flow of open source software along

the software supply chain as smooth as possible.

To this end, the OpenChain project is defining a standard for good governance. Like any specification, it does not provide best practices, but rather focuses on requirements like "define open source use cases" or "have an open source approval process."

At the time of writing, the OpenChain 2.1 specification of 2020 was the most recent standard. Version 2.1 covers:

- › The OSPO. The specification covers requirements for
  - having a defined OSPO mandate
  - having posts and roles with defined responsibilities
  - having specific posts like the legal counsel or public contact
  - managing the evolution of this structure
  - having a defined and secured budget for operating the OSPO.

- › *Using open source software in products.* The specification covers requirements for having an open source usage policy and processes for ensuring license compliance.

The usage policy requirements cover:

- having a policy,
- creating awareness for the policy
- assessing a company's competence with it.

The license compliance requirements cover:

- defining use cases
- having a standardized license interpretation
- managing the open source components in your products
- tracking the corresponding license compliance artifacts
- responding to third-party inquiries
- remediation of compliance issues.

- › *Contributing to open source projects.* The specification only states that you should have a contribution policy.

Nothing is said about creating or leading open source projects.

The OpenChain specification is a work in progress and will likely keep evolving and extending its scope. However, this does not diminish its significance. Already today, certification agencies have set up OpenChain compliance marks and are offering certification with (their interpretation of) the OpenChain specification.

At the time of writing, no company was requiring that its suppliers provide

such a certification mark, but it may only be a matter of time until companies will be required by their customers to demonstrate proper open source governance, most likely by featuring an OpenChain compliance mark.

## THE OSPO LIFE CYCLE

OSPOs have a life cycle.

Most companies start out with tasking one employee, part time, "to take care of open source." This person will typically try to help product and project teams get license compliance right. As a side job, this person can't achieve much and is likely to get quickly overwhelmed by the number of requests as word gets out about their responsibility.

Next, companies create an OSPO. The initial mandate is usually to create an open source policy for the company and to ensure that nothing goes wrong with its intellectual property. This leads to a focus on open source governance and license compliance. In addition, firm-internal marketing leads to more and structured awareness of open source within the organization, ideally combined with training for personnel.


Then, as the OSPO grows, not only does it have to deal with an increasing volume of requests to approve and manage components for use in products and projects, but it also expands its scope. It helps teams review code to be contributed to open source projects. It may even decide to take a small or large leadership role by initiating and leading open source projects. This includes active engagement in organizations like open source foundations.

If the OSPO does its job right, it will create and transfer significant skills to project and product teams. The teams learn how to deal with open source: to

use it properly, to know how and when to contribute, and to know how and why to get active in open source foundations and lead open source projects. Over time, these skills become an entrenched capability of every project and product organization.

Consequently, after a phase of growth, OSPOs are likely to shrink as they transfer some of their strategic and tactical responsibilities to the project and product organizations directly affected by open source exposure and engagement. As a central function, the OSPO will retreat to supporting revenue-generating organizational units.

**A**t the time of writing, most companies have no OSPO. Those who do, are still in the early growth stages. However, we could also already observe how some OSPOs have shrunk in recent days.

A stable long-term state for OSPOs is likely to focus on coordination and support rather than leadership. Coordination includes maintaining foundation membership and ensuring that the different stakeholders within a company remain informed and coordinated. Support includes the central provision of tooling and training. 

## REFERENCE

1. D. Riehle, "Free- and open source software," *Computer*, vol. 57, no. 8, pp. 114-118, Aug. 2024, doi: [10.1109/MC.2024.3407268](https://doi.org/10.1109/MC.2024.3407268).

**DIRK RIEHLE** is a professor of open source software at Friedrich-Alexander-Universität Erlangen-Nürnberg, 91058 Erlangen, Germany. Contact him at [dirk@riehle.org](mailto:dirk@riehle.org).