

The Open Source Program Office (OSPO)

Motivations, strategies and best practices for success



Table of Contents

Abstract & Introduction.....	3
OSS Overview	4
OSS as a Business Strategy	5
OSS as a Business Ecosystem	5
Understanding OSS Risks.....	5
Compliance/Legal.....	6
Financial/Business	6
Technological/Quality	6
Security	7
OSS Community Issues	7
How the OSPO Mitigates OSS Risk	8
What is an OSPO	9
Why OSPOs are Vital to the Open-Source Ecosystem	9
Responsibilities of an OSPO.....	9
Strategies and Best Practices for OSPO Success.....	10
OSPO Maturity Model/Stages of Adoption	10
Setting up Goals.....	12
Key Elements/Pillars of an OSPO.....	13
Key Contributors/Players.....	14
How to Start an OSPO	15
Conclusion	16
Further Reading and Learning	17



Abstract

As use of open-source software (OSS) continues to be pervasive and business critical in the corporate world, organizations face the challenge of managing its inherent risks. One approach that's gaining traction is the Open Source Program Office (OSPO) which helps organizations develop and manage their open-source strategy by establishing and enforcing policies related to OSS and third-party component usage, licensing compliance, legal issues, OSS community engagement, contributions to OSS projects, and more. This paper explores the makeup and potential of the OSPO, along with recommended practices for creating an effective OSPO—based in part on Revenera's experience in establishing and running its OSPO.

Introduction

The prevalence and importance of open-source software is not in doubt. BCG research reveals that 80% of IT departments plan to increase its use.¹ The challenge for most organizations is how to get the most technical and business value from OSS while avoiding its inherent risks. One answer, as demonstrated by technology industry leaders, is to establish an Open Source Program Office.

The OSPO helps organizations develop and manage their open-source strategy by establishing and enforcing policies related to OSS and third-party component usage, licensing compliance, legal issues, OSS community engagement, contributions to OSS projects and more. OSPOs exist on a spectrum of maturities. Some companies rely on spreadsheets. Others have sophisticated and well-resourced programs. Others still ignore the idea completely. This paper examines the OSPO and offers insights into why they are necessary for organizations that use OSS. It also explores some strategies and best practices for OSPO success, some of which are based on Revenera's own OSPO experience.

¹BCG, <https://www.bcg.com/publications/2021/open-source-software-strategy-benefits>

OSS Overview

It may seem unnecessary to define OSS, but it's a broad, hyped-up enough concept that it's easy to misunderstand several key aspects. This can be particularly relevant when it comes to how businesses use it. It's worth taking a brief moment to review the definition and its implications for a business before delving into the relevancy and elements of an effective OSPO.

In many cases, OSS is developed in a collaborative, public environment. Anyone can see the code and participate in its development. Potential users can download it for free. However, one common mistake is to equate OSS with "free software." OSS may not come with license fees, which are required for commercial, proprietary software, but there are costs associated with using OSS. And, most importantly, use of OSS involves a contractual relationship, because OSS is released under a license.

In most OSS licenses, the copyright holder of the source code grants others the right to use, study, change and distribute that code to anyone, for any purpose. The license typically creates certain obligations on the part of the licensee, such as providing attribution, or credit back to the author(s) or the duty to contribute any improvements in the source back to the OSS community that created it. And, the license restricts the licensee from taking certain actions, such as filing for a patent on the OSS source code.



WHAT IS AN OSPO?

OSPO is designed to be **the center of competency for an organization's open source operations and structure.**



BENEFITS OF OSS

57% Reducing overall costs

55% Improving development team efficiency

54% Shifting IT infrastructure or workloads to the cloud

Source: Forrester Opportunity Snapshot, April 2021

OSS as business strategy

The use of OSS is generally part of a broader business strategy. In some cases, this is an implicit rather than deliberate policy. However, much of the time, a software development team will make a conscious decision to use OSS for certain parts of its development process. Open-source is considered one of the most valuable tools for software development. It enables development teams to save time and money by using OSS components instead of buying or building software that has the same functionality.

By using OSS, a company can speed up its time to market. Instead of building software from scratch, developers can combine pre-existing, proven OSS components into an application. A further benefit is the ability to work closely with the open-source community. Developers can engage with peers, some of whom are regarded as the best at what they do. These relationships have the potential to drive better quality outcomes than are possible with commercially available software.

OSS as an ecosystem

The decision to use OSS means actively becoming a part of an OSS ecosystem. It's far more than just "using free software," as some might have it. OSS touches a collection of inter-dependent people and entities, all of which affect business outcomes. For one thing, OSS binds the company to the OSS community, which extends far beyond any one industry or region. The decision to license OSS can affect product quality, which affects the brand, which affects the ability to recruit and retain developers. The latter is a serious human capital challenge for most businesses, so the people aspect of the open-source ecosystem should not be overlooked.

Understanding OSS Risks

Open-source software comes with its share of business and technological risks. These include issues with license compliance and related legal difficulties. Poorly implemented OSS can lead to financial losses and other negative business outcomes. Software quality can suffer if OSS is not handled the right way. Security can be a major problem, especially if OSS is not subject to effective governance and security policies and practices. The open-source community itself also has the potential to be a source of risk. Unlike procurement of commercial software where you vet the vendor ahead of making the purchasing decision (top-down), OSS components are often sourced by individual developers (bottom-up), where often little regard is given to who the developers may be that wrote the code.

Compliance/legal

The use of OSS is almost always covered by a license agreement. Even unlicensed code, such what is often found on Github, may be subject to a default license in certain places, such as the European Union. OSS licenses fall into two general categories. A “permissive license,” such as the MIT or BSD licenses, typically let developers use code in proprietary software as long as they acknowledge the code’s original creator.

A “Copyleft License” like the Gnu’s Not Unix (GNU) General Public License (GPL) and Server Side Public License (SSPL) similarly permits developers to use and modify the code as much as they want. However, if a developer repurposes and distributes OSS, the developer must make the new version of the source code available to everyone. When that code is embedded in a proprietary application, this can be problematic.

Compliance with licenses terms is essential. However, an application might have code that’s governed by many different open-source licenses. It can quickly get complex and hard to manage. If a software maker does not comply with OSS license terms, even by accident, it can face litigation for copyright infringement from the licensor.

Financial/business

Implemented without regard to license terms, OSS can cause financial losses and other business difficulties. For example, if a software maker has to replace OSS code that has already been placed into an application, tested and released to the market, that can be an expensive, time-consuming remediation project. Litigation costs for misuse of license obligations could greatly add to that expense. The process would be a distraction for developers, slowing down other work. The matter could affect software vendors’ reputation, as well.

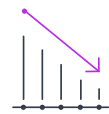
Technological/quality

In general, the open, collaborative process that creates OSS results in superior quality code. Indeed, OSS is known for attracting some of the best developers, who may be interested in working with other great coders on projects they consider interesting.

OSS RISK



If open-source licenses aren’t adhered to, there is potential compliance and legal risk



Financial losses as a result of litigation and loss of reputation



Possible quality issues in technology



Security breaches



Lack of participation in the OSS Community

However, that same open, community-based approach can lead to lapses in quality, especially if a small group of people is responsible for maintaining the code on a volunteer basis. For these reasons, use of OSS can result in software quality problems. Furthermore, unlike commercial software, open-source code has no guarantees of future development or timely responses to discovered issues. So it is imperative that component selection include assessing the future viability and activity of the community supporting it.

Security

Security is another double-edge situation for OSS. While on the one hand, the open nature of OSS lets everyone see the source code, and any security weaknesses it contains, the same openness creates risk exposure. The major security episodes unleashed by the Solar Winds supply chain attack and Log4j, to name two high profile examples, demonstrate just how serious OSS security needs to be. Security risks in open-source arise in two fundamental ways. One involves the insertion of malicious code into openly published libraries. A developer might download code components that contain malware and unknowingly embed them in a production application. This is known as a “supply chain” attack.

The other major security problem with OSS involves people simply not noticing that there’s an issue until it’s too late. The most egregious example of this was the Log4j crisis. Log4j is a Java-based software library used by millions of software programs around the world. It is the industry standard for logging functionality in Java applications. In December of 2021, it was revealed that Log4j contained a “Zero Day” vulnerability (CVE-2021-44228) that allowed malicious actors to execute remote code in many applications. Not only was the vulnerability extremely serious, the widespread use of Log4j, which occurred because it was free and an industry standard, compounded the crisis.

OSS community issues

Risks from the open-source community may not make a lot of sense to people outside the OSS world. It’s a real issue, however. Developers who use OSS are expected to participate in the OSS projects whose code they use. It’s a group effort and a reciprocal relationship. Developers who are perceived as taking more than they give are not treated with respect.

To understand why this is a serious problem, keep in mind that for many OSS stakeholders, the commitment to the community outpaces loyalty to a company or a proprietary product. And, in many cases, the OSS community is backed by major technology companies, so to be seen as shirking one’s duties can cause reputational problems on the levels of the individual, team and corporation.

How the OSPO Mitigates OSS Risk

The risks and complexities inherent in OSS make it unwise to adopt a laissez faire attitude toward the technology. Left alone, unsupervised use of OSS will almost certainly lead to disasters of one kind or another. To avoid this negative outcome, an organization that uses OSS should run some sort of organized open-source management program.

At a minimum, an open-source program embodies a set of policies that guide the use of OSS and compliance with OSS licenses. Open-source programs generally have three basic areas of responsibility: complying with licenses, contributing to open-source projects and giving back to the community. The latter involves activities like publishing open-source projects, sponsoring OSS developers or hosting OSS events.

A program can operate at different levels of formality, but the more organized and thorough it is, the better off all stakeholders will be. Nor is a program alone always enough to achieve success with OSS and mitigate the technology's various risks. The emerging best practice is to run an open-source program through an OSPO. According to Github, an OSPO "is designed to be the competency center for an organization's open-source operations and structure."²

Another recommended practice is to adopt Open Chain (ISO/IEC 5230), the industry standard for open-source license compliance. The OpenChain Project comprises open, international standards that enable OSS users to adopt the key requirements of a quality open source compliance program.



OPEN-SOURCE PROGRAMS BY THE NUMBERS

According to a Linux Foundation Survey of 1,141 business and IT stakeholders across multiple industries:

51% of respondents said an increase in funding for open-source initiative is very or somewhat likely this fiscal year

35% of OSPOs are located in software engineering and development departments (and another 18% are in office of CTO)

77% of respondents reported that their open-source program had a positive impact on their company's software practices

58% of OSPOs are formally structured

63% of those planning to create an OSPO are expected to initiate the process within a year

²GitHub, <https://github.com/todogroup/ospodefinition.org>

What is an OSPO?

An OSPO may not be an actual office. While it might be located in a physical space, it's more of an organizational construct. A person or team should be in charge of the program and have accountability for its operations and results. The leaders of the OSPO need to report into an executive sponsor of some kind, with “dotted line” relationships with key departments like legal, security, engineering, and product management. Or, the OSPO can interact with specialized teams staffed by members of those departments. The specifics will depend on the size of the organization and the level of its focus on OSS.

The OSPO's duties might include setting policies for code use, distribution, selection and auditing. Github's definition also added, “The OSPO is a designated function where the intake, compliance, contribution, and distribution of OSS is supported internally.” The OSPO might oversee the process of training developers, with the goal of ensuring legal compliance while also promoting and building community engagement that benefits the organization strategically.

Why are OSPOs vital to the open-source ecosystem?

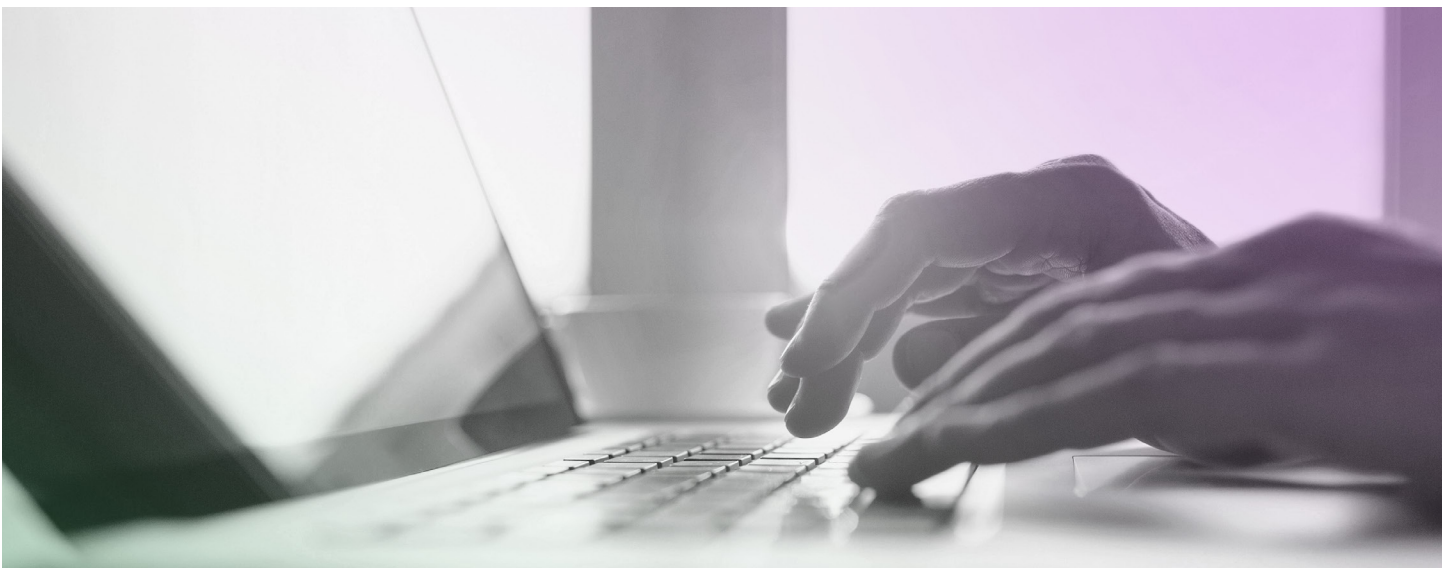
The open-source ecosystem, that interdependent group of people and entities that thrive on OSS, needs the direction and control provided by the OSPO. After all, managing an open-source program is about decreasing risk and enabling OSS to drive business outcomes. The OSPO facilitates the relevant processes. These include determining what OSS packages to use, making decisions about when developers should contribute to an OSS project, and so forth. The OSPO makes sure that decisions about OSS, along with OSS policies, align with business strategy—ensuring the success of the open-source ecosystem.

Responsibilities of an OSPO

OSPO responsibilities vary by organization. However, most have mandates to oversee the following areas of responsibility:

- Selecting which open-source packages to use. Decisions will be based on usage and licensing requirements, viewed in the context of business strategy and relevant policies.
- Enabling the use of OSS to accelerate software development to meet business goals.
- Managing OSS risk in accordance with the corporate risk profile, but also balancing risk management and OSS license compliance needs with release schedules.

- Serving as a steward of corporate open-source ethics and modeling “good citizenship” amongst the open-source community, including making recommendations about contributions to the community.
- Conducting training and education; getting all relevant people up to speed on OSS and the role of the OSPO in the organization.
- Establishing roles and responsibilities; enabling the long-term success of the program by establishing clear lines of responsibility and accountability.



Strategies and Best Practices for OSPO Success

An effective OSPO comes into being through a combination of strategy and best practices. While there is no “one size fits all” template, factors that affect the OSPO’s success include budget, executive level support and corporate culture. An organization’s open-source maturity is part of the equation, too, as are risk tolerance and overall business objectives.

OSPO Maturity Model/Stages of adoption

The Linux Foundation has developed a maturity model for OSPOs. It starts with “Stage 0,” which is equivalent to not having any OSPO or open-source program. It’s an ‘ad-hoc’ approach.

This is common, but not a great idea. From there the maturity model progresses:

- **Stage 1: Legal Education**—where the OSPO provides OSS compliance policies, delivers developer training and maintains an OSS inventory.
- **Stage 2: Community Education**—evangelizing OSS use and participation in the OSS ecosystem throughout the organization.
- **Stage 3: Engagement**—hosting OSS projects and focusing on engagement with the OSS community
- **Stage 4: Leadership**—becoming a strategic partner for the business, involved in making decisions affecting software development and technology



For a lot of organizations, even getting to stage 1 can be a challenge. Revenera, for example, is now between stages 1 and 2 in OSPO maturity, but our OSPO “origin story” is pretty typical. We went from loose coordination of OSS activities to pulling together some coherent policies and processes. However, we relied on email lists and unstructured processes to manage open-source use and compliance. We have since added more structure and organized process, culminating in the launch of a dedicated SharePoint site on the company’s intranet that holds all relevant documents, contacts and more.

Where a company is on the maturity scale, the idea should be to move up. It is important not to neglect community engagement, for instance. Leadership is a good goal, partly because resource allocation to the OSPO over time will depend on how the business management perceives the value of the OSPO in business terms.

Setting up goals

An OSPO is like any other serious business project. The process of creating one starts with establishing clear program objectives. Organizational culture will affect the way this process works and what the deliverables will be. In general, OSPO objectives should be clearly defined and documented. That way, others within the organization will understand their roles and why they are being asked to shoulder responsibilities in support of OSPO policies.

There may be something of an iterative loop at this stage, with goals, people and budget going through cycles until the scale and scope of the OSPO becomes firm. Some goals will have priority. Others will inevitably be pushed off. It's important to figure out where the OSPO is going to start. For example, the initial goal might be tackling open-source security. For other entities, the starting point could be an open-source project.

Success criteria go hand in hand with goals. For instance, the people running the OSPO may determine that they will measure their success by how much they decrease compliance risk for internal projects. Or, they might set a success metric that tracks the ratio of internal to external contributors an OSS project, which would demonstrate market adoption.

In Revenera's case, the goal was to establish consistent processes and policies across Revenera for the following purposes:

- Continuously assessing and remediating legal and security compliance issues as part of our development process
- Producing compliance artifacts (e.g., SBOMs, notices, etc.) for each Revenera product release
- Obtaining OpenChain ISO/IEC 5230 conformance
- Improving our ability to respond promptly to newly reposted security vulnerabilities
- Increasing our level of trustworthiness for customers and supply chain partners
- Improving software composition analysis (SCA) products and services

Key elements/pillars of an OSPO

An OSPO should establish a few key elements of operation. For maturity stages 1 and 2, these include:

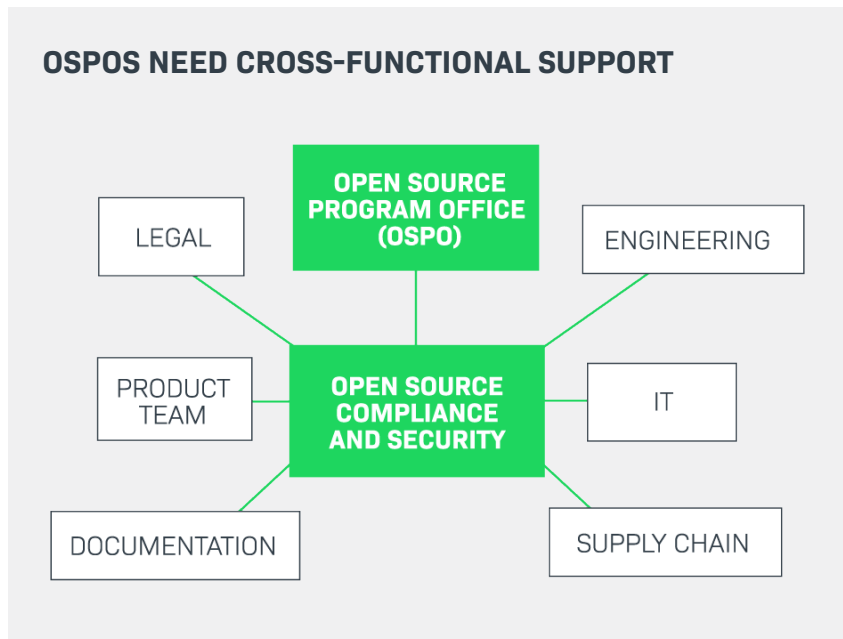
- **Managing legal risks and licenses**—the OSPO needs to track which component (with version) and governing licenses are in use for all OSS used in the organization. Its policies must then enable stakeholders to know if the development organization is adhering to license terms.
- **Educating developers**—developers require application security training as part of a shift-left initiative to minimize or prevent security issues to begin, as opposed to only remediating them once they've been discovered. Developers also need to learn about different open-source licenses, as well as the formal approval processes that take place when introducing new OSS products. Developers should also be made aware of risks of non-compliant OSS consumption.
- **Taking software inventory**—the organization that uses OSS needs to know what's in its code base. Creating an accurate software inventory usually involves building a Software Bill of Materials (SBOM). Specialized tools make this relatively easy to do. It's a continuous process, however, as code changes over time and the inventory needs to keep up with new releases.
- **Security vulnerability assessment**—continuous monitoring of vulnerabilities, coupled with threat intelligence for OSS code, must inform the use of OSS in the organization. This process should highlight the most relevant vulnerabilities so as to enable the effective management of OSS risk.

As the OSPO goes up the maturity scale, the pillars of the open-source program should embrace processes that foster community engagement. This has to be an active, deliberate project. The OSPO will ideally highlight opportunities for engagement and encourage specific people to take the initiative to get involved. The OSPO might also arrange for travel expenses and time off to attend open-source events. A similar effort should be made to make the organization into a “good citizen” of the relevant open-source communities.

The OSPO Alliance can be a resource in this context. They offer a platform for sharing materials on good governance for OSPOs. These include training materials and OSS governance frameworks.

Key contributors/players


People make an OSPO happen. And, some of the key contributors to the OSPO have to be specifically and ideally exclusively tasked with its operation and success. The OSPO will not work out well if it is a fraction of someone’s otherwise highly demanding job. In reality, this was how Revenera’s OSPO started. It’s not uncommon. As with some organizations, Revenera’s initial resource allocation for OSPO management personnel was limited.



When possible, however, there should be a dedicated Program Manager for the OSPO. At least one person from the legal department needs to be assigned, perhaps on a part-time basis, to the OSPO. This person must have, or develop, expertise in open-source licensing and compliance. Representatives from product management and engineering also need to have a presence in the OSPO as they will have to make room for additional backlog items coming from the governance program. In a

larger organization, they might be with the OSPO full time. IT staffers, security people and developers will need to be part of it, as well.

An executive sponsor is essential. This might be the VP of Engineering, the Chief Technology Officer (CTO), Chief Information Officer (CIO), or Chief Information Security Officer (CISO). The Chief Compliance Officer (CCO) is another possible choice.



The Revenera OSPO has a Program Organizer, in addition to representatives from the following departments:

- Executive
- Engineering
- Legal
- Product Management
- Security
- Audit services

How to start an OSPO

This paper presents a number of “best case” approaches to starting an OSPO. In all likelihood, most organizations will probably see their early efforts at managing OSS evolve into an OSPO. Revenera offers a good example of how this can work. Revenera’s first attempts at managing OSS use and compliance involved notifying groups of developers and other stakeholders about the use of a given OSS component by email. If a component was judged non-usable, an email would go out to the list.

This was not optimal for anyone initially, but it was a start. From there, Revenera established an open-source review board (OSRB), which is a common early step toward the creation of a formal open-source program and OSPO. An OSRB looks at requests to use OSS components submitted by developers. The board can give a thumbs up or thumbs down to the request.

As part of the process of setting up an open-source program, Revenera had key stakeholders in the process become certified on OpenChain. This was a positive, constructive step for the company, but it quickly became clear that more was needed.









Revenera made a conscious decision to launch an OSPO. While this may sound obvious, it’s a key step in the process. One might be surprised at how many companies drift along without a complete or coherent open-source program. There has to be a distinct move to get started.

One common challenge that Revenera faced was the assessment of a large amount of OSS code that had already been deployed. It’s one thing to review upcoming requests for OSS. It’s another to look backward and try to figure out just where all the OSS code has gone before. For Revenera, this was a matter of “drinking our own champagne,” so to speak. The Revenera SCA solution includes a SaaS solution for managing SBOMs, as well as an on-premise scanning solution for constructing and refreshing SBOMs as part of the development process.

Working with the Revenera SCA solution, the budding OSPO group at Revenera scanned 600 applications and began to build a substantial, detailed inventory of OSS at the company. The inventory, along with associated licensing information, is published on the Revenera OSPO SharePoint site. This site also contains documentation of the company’s open-source policies and compliance artifacts. With this information now on hand, and the scanning process ongoing, the OSPO is open for business.



CHECKLIST

-  **People**—Who will be on point to set up the OSPO, or evolve existing program elements into an OSPO.
-  **Money**—Is there budget for the OSPO? Or, at a minimum, is there management consensus that a portion of a full-time employee’s job will be committed to the OSPO?
-  **Plan**—Is there a working plan in existence for the OSPO?
-  **Organizational engagement and buy-in**—Are stakeholders from Legal, Engineering, Product Management, Compliance, Audit and so forth aware of the OSPO? Have they had a chance to provide input and feedback on the OSPO plan?
-  **Executive sponsor**—Has a senior level person, such as the CIO, taken on the role of executive sponsor?
-  **Form factor**—What will the OSPO look like at launch? Will it be an internal website, a physical location, etc.?
-  **Communication**—Are there clear processes for communicating with the OSPO, including the designation of a person or people who will reply to messages, etc.?
-  **Accountability and success metrics**—Are there clearly defined metrics for success for the OSPO, along with clear lines of accountability for achieving them?

Conclusion

An organization that uses OSS to any significant degree is well advised to start an open-program, and then move directly to the establishment of an OSPO. There is too much risk to take any other approach to OSS. An OSPO directs which open-source packages are suitable to use. Done right, this process enables OSS to accelerate software development and achieve business objectives. At the same time, the OSPO’s governance of OSS use reduces compliance risk. The OSPO further functions as a steward of open-source ethics and models “good citizenship” in the open-source community—a factor that is more important than business stakeholders may realize.

Each company will have to approach the OSPO in its own way. Typically, early efforts are scattered, and it can take some time and focus to get an open-source program properly organized. Ideally, though, an organization will endeavor to rise up the OSPO maturity scale. This means growing out of basic legal education functions and moving toward community education and engagement, culminating with the OSPO becoming a source of strategic leadership for technology decisions. ■

Further Reading

TODO Group

<https://todogroup.org/guides/>

The Linux Foundation

<https://www.linuxfoundation.org/resources/opensource-guides/>

CNFC

<https://www.cncf.io/>

OSPO Alliance

<https://ospo.zone/>

Learn from the Enterprise

Google

<https://opensource.google.com/>

Microsoft (GitHub)

<https://opensource.guide/>

Facebook

<https://code.fb.com/category/open-source/>

NEXT STEPS

Gain transparency and actionable insights
into the complexity of your software.

[LEARN MORE >](#)

Revenera provides the enabling technology to take products to market fast, unlock the value of your IP and accelerate revenue growth—from the edge to the cloud. www.revenera.com