



» The Linux Foundation

Establishing an Open Source Software Strategy: Key Considerations and Tactical Recommendations

.....
By Ibrahim Haddad (Ph.D.) and Brian Warner, The Linux Foundation

A Publication By The Linux Foundation
<http://www.linuxfoundation.org>

Introduction

Since the beginning of the software industry, nearly every software company has followed the same business model: source code was developed by their own employees or licensed from a third party, intellectual property was closely held, and software was delivered in binary format to clients. However, the availability of enterprise grade open source software is changing the way organizations develop and deliver products. With open access to source code and transparent development communities, software providers can reduce development costs while remaining active participants in the development process. In addition, end users of the software can also be active in the development process by contributing directly to upstream projects, rather than passive recipients of what the software vendor delivers to them. This development model enables organizations to think differently about how they procure, implement, test, deploy, and maintain software. In this article, we explore three basic strategies for organizations that utilize and integrate open source software in their products: consumer, contributor and leader.

Core Strategies: Consumers, Contributors, and Leaders

There are three core strategies to consider when deciding on your open source strategy:

- **The open source consumer** is the most common starting point, where an organization uses an open source component in place of undifferentiated, privately maintained code. In many cases, they also may be ad hoc contributors at small scale, contributing smaller bug fixes or features enhancements for the open source software they are integrating into their product.
- **The open source contributor** builds on the consumer model, where an organization adopts a policy of selective engagement with strategic open source projects. By contributing improvements, they are improving their ability to deliver a quality product while reducing development costs.
- **The open source project leader** builds on the contributor model, where a company earns a leadership role within a project. This scenario is possible when the company has become well established as a strong contributor within the project community, and should be considered when there is a strategic reason to assume a leadership role within a project.

It is important to note that no single strategy applies to all companies, and that a corporate open source strategy should be customized to your organization's particular needs, resources and wanted position. In most cases, an optimal open source strategy will be a blend of the consuming, contributing and leading, depending on the open source software at hand and its strategic importance to your product.

The Open Source Consumer

The decision to use open source software is often driven by a need for new features, expanded functionality, and to reduce internal development cost by leveraging the collective engineering effort and innovation of the open source community. It has been demonstrated repeatedly that using open source software increases an organization's bandwidth to differentiate, while reducing overall time and cost to deliver commercial products.

Furthermore, the tangible long-term business benefits can significantly outweigh the incremental short-term cost and efforts of managing a mixed closed and open source development portfolio.

When establishing a software strategy that encourages the use of open source software in commercial products, there are a number of actions that can be taken to ensure successful adoption of open source software:

- Communicate the strategy for the use of open source software.
- Educate your staff on open source compliance and license obligations, and the open source development model.
- Establish explicit criteria for determining which open source software is a candidate for inclusion into your products. Examples include availability of new features, the maturity of the project's source code, the size and composition of the project's development community, and other factors that measure the state of the code and the people who maintain it.
- Establish an open source compliance program to ensure you have the processes in place to meet license obligations of the open source software you are using in your products.
- Encourage your developers to identify and adopt open source development tools that can enable better internal collaboration, increased and transparent team communication, and faster development cycles.
- Encourage your staff to subscribe to open source mailing lists, subscribe to open source magazines, follow blogs, and participate in discuss forums.
- Encourage and fund staff to attend open source conferences for learning and networking opportunities, as much work and decision-making happens face-to-face.
- Join open source industry bodies and organizations such as the The Linux Foundation for opportunities to share development and legal best practices with other leaders in the industry.
- Hire developers from the open source community.
- Host local open source user groups and encourage your staff to get involved in local open source activities.
- Invite community members to present to your development team on topics related to the project.

The Open Source Contributor

Contributing to strategic open source projects can help organizations gain additional value, as code contributions can help shape future features in the project that better meet a company's needs and requirements. It can also provide substantial cost offsets, particularly when a large amount of re-engineering is required to integrate new versions of the component into the product. Contributing code back to the open source project, also called upstreaming, can reduce the amount of reintegration required when a new version is released, because the earlier changes are already incorporated into the mainline (or upstream) version of the code. Organizations that have a strategic reason for becoming involved as a contributor to one or more open source projects can begin by taking specific steps to build trust, increase their contributions and maximize value:

- Educate your team on community development best practices.
- Actively participate and drive technical discussions on the mailing list, IRC, discussion forums.
- Follow the open source community's established working methods and processes.

- File bug reports and contribute fixes to existing bugs.
- Contribute code to improve or extend functionality.
- Contribute code to implement new features.
- Contribute to documentation efforts.
- Contribute to testing and integration efforts (write test code, create test cases).
- Listen to feedback on your contributions, and act on it.
- Establish trust with the project maintainer and other project participants via your contributions and active participation.

The Open Source Leader

Leadership roles in open source communities are earned by establishing trust with the project members, and by maintaining a high level of continuous contribution to the project. Below are some of the tactical steps that can help steer you towards a leadership role within a specific open source project:

- Participate actively and openly within all aspects of the project, including planning, development, testing, and release management, thereby demonstrating your capacity to act as a good steward of the project.
- Achieve a higher level of participation and contributions:
 - » Engage with the various project participants.
 - » Contribute to solving bugs, adding new features, extend functionality in existing open source projects using the best practices outlined above.
 - » Demonstrate good faith by contributing (when relevant) proprietary source code from internal development to open source projects, under an appropriate open source license that makes it usable and useful to the community.
- Publicly acknowledge that the company has achieved tangible benefits by working with open source communities for critical software product development.
- Empower employees to seek maintainer status within the project.
- Sponsor events, provide financial support for project infrastructure, and consider hiring recognized open source developers from within the project.
- Increase participation in relevant open source organizations and foundations.
- Lead architectural and requirement gathering initiatives within the various communities and consortia to achieve commercial objectives.
- Establish an open source architect role to pro-actively guide the use of and contributions to open source software.

Conclusion

Open source methods are shifting software development practices into a new paradigm, moving from code developed behind closed doors and binary-only distribution to software assets that can be shared, modified and redistributed openly. Organizations that embrace the open source model are increasing their chances of retaining competitive advantage, increasing their agility, and diversifying development risks and costs.

This article described three core open source strategies: consumer, contributor and leader, and discussed best practices and tactical activities that can help organizations execute on these strategies.

If your company does not have an open source strategy, this article is a call to action. The Linux Foundation has a number of resources available that can assist with this process. Visit <http://www.linuxfoundation.org> to learn more.

Linux Foundation Resources

Linux Training

The Linux Foundation offers two training courses to enable organizations effectively work with open source developers:

- **LF 205: How to Participate in the Linux Community:** Working with the kernel development community is not particularly hard, but it does require an understanding of how that community works. This course is intended to bring attendees up to speed quickly on how kernel development is done and how to be a part of the process with a minimum of pain and frustration.
- **LF 271: Practical Guide to Open Source Development:** This course prepares organizations to maximize their effectiveness and shorten the time to value when participating in open source development projects. This course builds upon years of best practices and extensive experience in commercial participation in open source projects to help organizations approach the open development model in a structured and methodical manner, maximizing the likelihood of success. The course provides extensive examples from the Linux kernel community, and includes specific best practices for working with upstream.

Linux Foundation Labs

If you have a collaborative software project you need hosted at a neutral party, the Linux Foundation may be able to help. The Linux Foundation assists companies and communities by hosting collaborative software projects.

The Linux Foundation provides three main services to Lab projects:

- The technical, operational and legal infrastructure so that project leaders can focus on technological innovation.
- Guidance and consulting on open source best practices gleaned from the two decades of experience of Linux and the ability to collaborate and network with the large and growing Linux Foundation community.
- By providing these services to companies and developers, the Linux Foundation provides a much needed framework for advancing and accelerating technology that allows project hosts to focus on innovation.

There are two main criteria that must be met in order for the Linux Foundation to host a lab project:

- Use of open source governance best practices including license and contribution agreement choices in keeping with the ideals of Linux
- Project must either use Linux or have the potential to enhance the Linux ecosystem

If you have a project that may fit this criteria, please contact us:

<http://www.linuxfoundation.org/labs>.

Open Compliance Program

The Linux Foundation's [Open Compliance Program](#) was established to boost adoption of Linux and other open source by making license compliance ever-easier to achieve, to increase awareness and understanding of open source compliance responsibilities, and to make available free resources that can help companies establish their compliance programs. The program offers comprehensive training, compliance educational materials (white papers, compliance blog, webinars), compliance tools, an online compliance community (FOSSBazaar), a best practices checklist, a rapid alert directory of company compliance officers, and SPDX™, a standard to help companies uniformly tag and report software used in their products.

Linux Events

The Linux Foundation produces a number of [technical events around the world](#) that provide a venue to bring together developers to solve problems in a real-time environment.

Publications

The Linux Foundation produces a wide range of publications that are available for free download. These publications are divided into three categories: Open Source Compliance, Workgroups (such as Tizen, OpenMAMA, LSB, SPDX, FOSSology, etc.) and Community. The Linux Foundation publications are available from <http://www.linuxfoundation.org/publications>.

About the Authors

Dr. Ibrahim Haddad is the Director of Technology and Alliances at The Linux Foundation and Contributing Editor for the Linux Journal.

Brian Warner is Operations Manager at The Linux Foundation.

The Linux Foundation promotes, protects, and advances Linux by providing unified resources and services needed for open source to successfully compete with closed platforms.

To learn more about The Linux Foundation, or any of our initiatives, please visit us at <http://www.linuxfoundation.org/>.

