GitHub

How to become an open source enterprise

Transform the way your team builds software with open source code, tools, and best practices.

Table of contents

Introduction

Open source is changing the way we build software. From hotel bookings to banking, almost every new application begins with code that anyone can inspect, modify, and enhance. According to the 2019 Open Source Security and Risk Analysis (OSSRA) report, nearly 100 percent of application code bases contain open source software (OSS), comprising 60 percent of the code analyzed.

While enterprises widely use open source in their applications, few have tapped into the full advantages that the OSS community has to offer. Businesses are looking to OSS and its community best practices for new ways to differentiate themselves from competitors, yet they often lack experience in working with external communities or breaking down internal silos. They try to hire more developer resources or find the best software tools—but still can't keep pace with the millions of developers in the OSS community.

Developers are at the core of innovation, making your organization's approach to in-house software development more important than ever. The new software leaders aren't the ones who simply choose the best tools or have the most developers. They're the ones on a journey to change their development culture; they're learning how to encourage collaboration and other OSS best practices found in some of the world's biggest systems—including the Linux operating system, Facebook, and Google. These leaders have transformed from companies who use some open source code into a new type of organization: **open source enterprises**.

What is an open source enterprise?

Open source enterprises are organizations that recognize OSS as essential, not optional, for modern software development success. They transform their businesses by making open source best practices fundamental to the way they build software, and outpace competitors with the help of the world's most innovative communities. Their relationship with OSS is balanced, and most importantly, secure. As a result, open source enterprises experience their own rapid innovation, faster speed to market, and reduced costs—while minimizing risk.

To become open source enterprises, organizations must:

- Utilize OSS, collaborate with, and contribute to open source communities
- Adopt innersource by bringing open source culture, tools, and practices into the workplace
- Leverage OSS and its communities safely to create software that is secure and compliant

All three ingredients help companies thrive digitally and accelerate innovation through open source. In this guide, we'll share how you can make the shift to an open source enterprise—and how GitHub can help.

1. Engaging in open source

An open source enterprise's success starts with the way they participate in open source communities. Instead of reinventing the wheel, open source allows you to use code that's already been made—freeing your developers to focus on the proprietary software that actually differentiates your products. But the true value comes in both consuming and contributing to open source. By engaging in open source communities, teams are able to bring new ideas back into your organization and have a stake in the future of the OSS projects your business depends on.

Connect with the community

Hundreds of organizations use popular open source packages like React and TensorFlow to power their applications. While you may never meet the maintainers and contributors behind the open source code you use, their work becomes a crucial part of your software development process—just like any other component in your supply chain.

In the same way you know and vet the components you bring into your supply chain, you'll want to better understand the communities responsible for the OSS you depend on. For your most critical dependencies, it's important to think of the relationship even more strategically. What are the open source communities' goals and future direction? How do they prioritize work? What's the long-term viability of their development model and where are they falling short?

The decisions open source maintainers make can have outsized implications for your business that go far beyond a few lines of code. Their community is now your community, and you should take an active role in shaping what happens next (and maybe even suggest some new features of your own).

Collaboration in action

Each month, about 39 percent of GitHub Enterprise Cloud users engage with OSS projects and 44 percent of Enterprise Cloud organizations actively contribute to OSS.

Help your team level up their skills

When you use OSS, everyone benefits—but how much is up to you. Harvard researchers found companies that contribute to open source get up to two times more productivity out of open source compared to companies that only consume open source. Actively contributing to open source communities like React Native and Ruby helps your team better understand how to apply that same code to your projects. It gives them firsthand experience with the risks involved in using OSS and how to manage them, as well as ideas for new enhancements that can help both the community and your team.

Open source best practices for enterprises

- Read the project's README and CONTRIBUTING files for guidance on how to participate effectively.
- Subscribe to notifications, read and comment in issues, and suggest solutions to problems with pull requests.
- Explore open source projects in your preferred programming language. Look for projects that both interest you and align with your organization's strategy.
- Standardize your open source library usage as much as possible for easier maintenance and reduced exposure.
- Contribute your local improvements upstream to open source projects—decreasing your maintainer burden and providing valuable feedback to the community.
- Review your business's open source investments and pick the most critical items for strategic focus. This may include increased time, effort, or even financial assistance.
- Use the `-on-behalf-of` commit tag when contributing back to open source projects to denote your organization's support.

Become an open source leader

Participating in open source doesn't end with contributing to existing projects. Large companies like SAP, Capital One, and Netflix have moved from open source consumers to leaders in OSS by creating and sharing their own open source projects on GitHub. Now software teams of every size can save time and resources by taking advantage of this reusable code—a move that's elevated their reputations in the open source community and with others across industries.

Sharing non-proprietary, non-business-critical software as open source creates better OSS for us all, but also establishes your role as a leader in the open source community. Your open source contributions are a key way to build ecosystems, promote the technologies and standards you depend on or support, and strengthen the connection between your team and the open source community.

From the GitHub experts

With GitHub features like contributor and repository graphs, you can see who's contributing to your preferred open source projects and get top-level summaries of recent repository activity. And since GitHub was built with collaboration and communities in mind, it's easy to interact with the open source communities you care about. Using pull requests and issues, you can collaborate with other contributors, suggest code changes, and help your community stay healthy.

Leverage the dependency insights graphs to understand where your organization is consuming open source, as well as what vulnerabilities or licenses they contain. If you discover a vulnerability, follow the project's security policy in their SECURITY file to let them know about the bug.

2. Building an innersource culture

Collaboration, transparency, and code reuse aren't unique to open source communities. Successful open source enterprises also adopt these same best practices to build internal software, an approach commonly referred to as "innersource."

Innersourcing helps organizations become more efficient by standardizing their code and development process on consistent platforms, reducing duplicate effort, and breaking down barriers so teams can share skills across the entire organization. It also creates a collaborative company culture where big ideas thrive—and get shipped.

Improve efficiency and collaboration

Just like in OSS, more contributors mean more opportunity for innovation. Innersource encourages teams to share their expertise enterprise-wide and build better software, together. Using the same tools and systems GitHub has built for the open source community, team members can easily discover and reuse internal code, or get help from your own subject matter experts.

Encourage transparency

Making projects available for everyone in your organization encourages a culture of openness and helps ideas spread. If your teams aren't comfortable sharing their own code internally, contributing to OSS communities will quickly become an uphill challenge.

Full transparency is a baseline for any open source enterprise. And not just for your code—teams should be able to access documentation, track discussions, and understand your decision-making process. Promote "innersource visibility" and grant all employees (and none of your outside collaborators) permission to view the same repositories across your entire organization.

Three tips for successful innersourcing

1. Make sure projects have clearly defined problems and opportunities to be addressed, which are shared across teams.
2. Give your teams the tools and processes to communicate openly and build consistently.
3. Put together a plan to onboard and acclimate new contributors and other participants into your process.

Break down silos

OSS is quickly created and consumed. Over one billion code contributions take place on GitHub each year and more than 2.1 million organizations use GitHub to collaborate. As we've seen at GitHub, the typical siloed development process slows down the pace of innovation, requiring more time to get applications into customers' hands. Instead, an innersource culture gives your developers more opportunities to work with other teams and across business lines.

Change is hard, and your organization's current incentives may not reward group wins as much as individual effort. The move to a more open environment may require a larger shift across the entire enterprise that you shouldn't expect to happen overnight. But nurturing your own internal communities is just as important to your organization's success as the open source communities you rely on. As you continue your journey to becoming an open source enterprise, encourage your teams to share knowledge, ask questions, and find the people who inspire them, regardless of the department they officially work in.

From the GitHub experts

Sharing your code in repositories on your private GitHub Enterprise instance allows teams to use collaborative tools like forks, branches, and pull requests to find what works for them. Forks and branches help developers freely experiment with changes without affecting original projects, so they can use internal solutions as a starting point instead of building brand new code each time.

GitHub issues also offer a place for everyone to join in on discussions about your code—and encourage transparent conversations. Mention a teammate to instantly add them to an ongoing discussion, link to other open issues, and coordinate your work in one place. And if your team needs more collaboration tools, head over to GitHub Marketplace to find hundreds of free and paid applications that can improve your workflows.

3. Staying secure and compliant

When it comes to enterprises, secure use of open source is especially crucial. Adding an OSS library to your project means adding thousands of developers to your team that aren't on your payroll—and haven't completed a company background check.

Successful open source enterprises innovate at scale with OSS communities while still maintaining strict security and compliance standards. That doesn't mean ignoring or downgrading potential risk. Instead, being an informed OSS consumer means staying up to date about the open source code you use, and using best-of-breed tools and security best practices to keep your team safe along the way.

Manage dependencies and vulnerabilities

When you build on open source components, you inherit the risk that comes with them. Even OSS that's been reviewed by multiple contributors can still have security issues or dependency vulnerabilities. For example, the Heartbleed bug was introduced into the OpenSSL cryptography library in 2012 and not publicly disclosed until 2014. Managing this risk requires visibility into what and where OSS is used in your projects—which can be hard to do alone.

Sit down with your team and create a plan for identifying and updating vulnerable components. Many organizations rely on tools to help track OSS dependencies, making it easy to see which projects depend on your code and vice versa. Other applications flag vulnerabilities and prevent your team from adding them in the first place. But safely consuming open source also depends on the human factor: set policies ahead of time for how you'll decide an open source package is secure and who approves it's safe to be used in your organization.

Best practices for managing OSS licenses

- Create an approval process for when developers want to introduce new OSS into your organization.
- Put together a list of pre-approved OSS licenses so developers know which OSS is appropriate to consume.
- Continue working with your company's legal department to regularly evaluate OSS licensing.

Understand open source licensing risks

An open source license provides rules and guidelines for the ways you can view, use, modify, and distribute an OSS project. Even if a project is public, you can't legally use any part of that project in your code unless you have explicit permission to do so. Open source licenses grant this permission—with limits.

There are important legal distinctions between customizing OSS for internal use by your team and commercially redistributing software that contains OSS components. Depending on the license, proprietary software that uses OSS code may also have to be released as open source in order to comply with the original OSS project's license terms (like under GNU GPL licenses). That doesn't mean you can't use any OSS in proprietary code, just that you have to ensure the OSS you use has the right license for your application's goals.

Maintain compliance

Compliance is an important part of using open source securely. Ignoring internal policies can lead to ad hoc OSS usage, and vulnerability or licensing blind spots. Development teams should build process gates into their workflows to help them use (and stick to) repeatable, trackable processes. This also creates a digital paper trail for auditors. Using statuses and other checks, you can automate compliance workflows and easily review your audit log to find out who performed an action, what the action was, and when it was performed.

In order to keep up with the volume of OSS code, larger organizations often have open source program offices—including legal staff—to ensure they maintain company policies while participating in OSS. Along with managing the OSS licenses and code coming into your organization, open source program offices can govern the policies for contributing to OSS projects.

Looking forward

Security is an ongoing effort for everyone who consumes and contributes to OSS. As more companies become open source enterprises, they'll also be faced with new security challenges. What happens if an OSS project you need is taken private, or even worse, is taken over by a malicious actor? How do you track the overall health and status of the OSS communities you rely on?

GitHub is committed to tackling these and other new open source security questions that come our way—and we're already working with top OSS maintainers and organizations to address them.

From the GitHub experts

Millions of open source projects live on GitHub.com. You can see all of the projects your organization's code connects to with dependency insights, as well as any detected vulnerabilities and licenses.

To get started, just turn on security alerts for vulnerable dependencies. When GitHub finds a vulnerability, we'll immediately send you an alert with suggested fixes from the GitHub community. Since launching our vulnerability alerts, we've sent alerts for more than 28 million vulnerabilities and seen over three million resolved.

Token scanning is enabled for public repositories to identify credentials that could lead to a security breach. GitHub searches for tokens for popular cloud services and alerts the vendor and repository owner if one is found in the public domain.

Our newest feature, GitHub Package Registry, also helps you discover and safely publish public and private packages in one place, next to your code. Your team can establish a private registry of approved software packages to safely consume within your organization.

Customer story

SAP

SAP is one of the largest providers of business software in the world. You may not find their products for sale at your local store, but 77 percent of all transaction revenue touches one of SAP's systems.

As clients expect more from software, SAP solutions need to rise to the challenge, expanding uptime and features without compromising speed or security. To keep up with competition, stay flexible as they scale, and support their 35,000 developers, SAP turned to GitHub Enterprise.

With GitHub Enterprise, SAP can build right alongside the open source community and securely integrate leading open source technologies into their own, saving time building proprietary solutions. And they can start contributing code to the projects they use most both internally and externally.

By working on one platform, developers' best ideas, teachable moments, and the conversations behind them all make it to GitHub. To take advantage of the growing software community within SAP, developers are also encouraged to innersource work, opening up projects to feedback and ideas from the rest of the organization.

Investing in open source, encouraging an innersource culture, and using a secure platform have all redefined the way SAP builds software. SAP has created a flexible infrastructure that grows with their team, incorporates open source code, and builds on best practices from the open source community—completing their transformation into an open source enterprise.

> *Now many of our projects start as open source code on GitHub.com...The benefits of engaging the community are clear: like fast fixes and new ideas in a collaborative environment.*

-Ingo Sauerzapf
Cloud Development Tools Manager, SAP

As an open source enterprise, GitHub helped SAP:

- Increase speed and innovation by leveraging open source code
- Build a reliable DevOps pipeline
- Attract and retain top developer talent
- Support their cloud transformation

Ready to explore how GitHub can help your organization become an open source enterprise?

Contact us!

sales@github.com
github.com/enterprise
+1 (877) 448-4820