WHITEPAPER

How Legal, Engineering Teams Can Collaborate to Reduce Open Source Risk

See best practices to fine-tune the relationship between legal teams and software developers when it comes to the use of open source software.

FOSSA

Content

Introduction

The rise of open source software (OSS) has put a spotlight on the importance of building a productive, collaborative relationship between a company's legal department and its software development team. While open source offers a myriad of benefits, including being cost-effective and readily available, complexities related to OSS license compliance mean that the legal department must be keenly aware of how their organization is using OSS.

The sheer prevalence of OSS in modern applications makes license compliance an area that in-house counsel can't afford to ignore. Van Lindberg, an attorney at Taylor English Duma and an expert on open source issues, notes that "almost all code these days is overwhelmingly, primarily open source." Indeed, open source software is utilized by over 90% of organizations, according to Gartner. A separate survey, meanwhile, found that 72% of companies regularly use open source for internal or non-commercial purposes and 55% use it for commercial goods.

If attorneys are unaware of how software developers are using OSS at their company, they may be leaving the organization open to legal, reputational, and transaction-related risk. However, if the legal department imposes rules governing the use of open source that are viewed as onerous or arbitrary, software developers may not adhere to them, creating risk for the organization.

Although the relationship between in-house counsel and software developers may appear tricky, creating a collaborative, constructive relationship between them is entirely possible. By listening to developer feedback and establishing clear OSS guidance, the legal department can avoid being perceived as the dreaded "Department of No." Instead, it can build a highly effective partnership with the software development team and ensure that OSS compliance is built into all stages of software developers' work.

In this white paper, we'll explore how to persuade software developers to prioritize OSS compliance, what an effective workflow between developers and attorneys looks like, how to handle developer requests to contribute to OSS projects outside of the company, and how all of these strategies can help build general OSS compliance in your organization.

The Basics of Open Source

Open source software has source code that can be viewed by anyone, as well as modified for use in other software projects, as long as the user abides by specific terms (more on that below). It stands in contrast to proprietary or “closed source” software, whose owners may restrict access to the program’s source code and prohibit modifications to it.

Notably, the use of open source software is governed by licenses — meaning that while OSS may be available for free, there are certain requirements users have to comply with. License requirements generally kick in when the software is distributed.

Generally speaking, there are two major kinds of open source licenses: permissive and copyleft. Permissive licenses typically allow use of the code with minimal restrictions, such as requiring users to include the license text and the copyright notice with any redistribution of the code. In practice, a software developer is often able to grab code under a permissive license, change it to produce a new program, and then sell the program, while keeping its code to themselves.

Copyleft licenses, however, carry more conditions on use of the licensed code. They generally require that any derivative work be released under the same terms as the original. In other words, copyleft licenses require that anyone who releases a modified open source program must also release the source code for that program, which is likely not ideal for companies selling software products.

Over 100 licenses have been approved by the Open Source Initiative. The MIT license (permissive) and the GNU General Public License or “GPL” family (copyleft) are among the more popular.

“Some of them say, ‘If you use our code, you have to release your source code for free,’” Anthony Decicco, a principal at GTC Law Group who oversees the firm’s open source practice, says of copyleft OSS licenses. “So, you're a commercial software company, maybe even though your software might only be 20% ‘yours,' you don't want your source code out there for your competitors and your customers to access — you want them to buy that from you.”

"You can't use copyleft licensed components in a way that would trigger those obligations, otherwise, your business could be over."

— ANTHONY DECICCO, PRINCIPAL, GTC LAW GROUP

The alternative to source code disclosure might be unwinding and rebuilding your application without that specific open source component — which can be quite time-consuming and disruptive. That does beat the consequences of non-compliance, however, which can include potential litigation. Companies using OSS must therefore make a strong effort to maintain open source compliance, and software developers can be an important part of that process.

Initial Open Source Best Practices

Companies looking to become OSS compliant should consider a few initial steps. First, a company may want to perform a review of a “sample” codebase, according to Decicco and his colleague Brad Goldring of GTC Law Group. Auditing a codebase can reveal how much OSS an organization uses and which licenses are involved in that. An initial audit is also a chance for the company to determine its “risk tolerance” for various licenses and use cases, according to the attorneys.

Companies will also want to craft an OSS use policy, which provides guidance on open source licenses and how they can be used and outlines the procedure for having an OSS component reviewed and potentially approved. Keeping a policy short and easy to understand is a best practice so software developers aren’t tempted to set it aside.

> **“We often have a rule of thumb: If your policy is more than two pages — for the meat of it — you've done something wrong."**
>
> — ANTHONY DECICCO, PRINCIPAL, GTC LAW GROUP

Often, an OSS policy is developed by in-house counsel working with outside counsel who have expertise in open source software (although it’s also possible that in-house counsel with solid OSS experience could tackle this task). Certain OSS license compliance tools, such as FOSSA, also come with pre-built policy options, which can be useful for organizations that don't have the resources to create their own.

Importantly, it’s not just technology companies that can benefit from having an OSS policy. As Christopher Stevenson, an attorney at DLA Piper who advises companies on open source issues, notes, “pretty much every company out there is using some open source software.” Stevenson pointed out that he once put together a policy for a company that sells razors online, which turned out to be “a huge user of open source.”

“They have a robust website — it has all kinds of open source software in it — and they also develop a lot of software,” Stevenson says.

If a company already has a license compliance policy in place, it can be helpful to compare it against the results of the audit, to get a sense of whether the company’s practices are in alignment with its policy.

“We often find that the company is not in compliance with its own policies,” Goldring said, adding that sometimes this means it’s necessary to bring the code base into compliance, while other times, it means the policy should be updated (if, for example, it was too restrictive).

Finally, companies should also consider forming an OSS Committee, which will usually include individuals from the legal, business, security, and software development teams. Typically, the OSS Committee gives guidance when needed and reviews OSS use cases that can’t be auto-approved based on rules set by the committee, according to Decicco and Goldring. The OSS Committee can also be a key part of creating and maintaining the OSS use policy.

Typically, about 80% of the requested components are auto-approved (thanks to the use of software composition analysis tools and automatic approvals based on the OSS policy), while the remaining 20% will be reviewed by the committee, according to the GTC attorneys. Some typical situations in which an OSS Committee might review an open source component include:

- The possible use of a component licensed under an OSS license that hasn’t been assessed yet
- New usage scenarios involving previously approved components
- Using components from competitors
- Higher-risk use cases for valuable products.

“When a component comes in and we approve it from the legal point of view, it's not the end of the story."

— ANTHONY DECICCO, PRINCIPAL, GTC LAW GROUP

Decicco added: "The security people might nix it because it has known vulnerabilities or someone else in product development might say, ‘We don't want to use that component because our competitors are standardized on that.'"

Not all companies will opt to have an official OSS Committee and may instead simply have each relevant department (legal, security, etc.) review the OSS

question at hand. Additionally, Goldring notes that the legal department could serve as the “primary triage team” for OSS requests and flag riskier requests to other departments for additional review, as needed.

Not all companies will opt to have an official OSS Committee and may instead simply have each relevant department (legal, security, etc.) review the OSS question at hand. Additionally, Goldring notes that the legal department could serve as the “primary triage team” for OSS requests and flag riskier requests to other departments for additional review, as needed.

KEY TAKEWAYS

- Open source software has source code that can be viewed by anyone and modified for use in other software projects, as long as the user abides by its applicable licenses.
- Failure to comply with an open source license could result in a company needing to disclose its source code, rewrite its code, or relicense the component under a different license.
- Auditing a codebase can provide insight into how much OSS an organization uses and provide a chance for the company to assess its “risk tolerance” for certain licenses and use cases.
- Creating an OSS use policy is key, as it will provide guidance on open source licenses and the procedure for having an OSS component reviewed.
- Companies should consider forming an OSS Committee, which gives guidance when needed and reviews OSS use cases that can’t be auto-approved.

Getting Software Developer Buy-in

It’s also crucial to get buy-in from software developers on a company’s OSS policies and procedures, otherwise, there’s a risk that engineers will simply ignore them altogether.

“Sometimes, when we’re asked to evaluate existing policies, the first thing the engineers tell us is: ‘We deliberately don't follow our process because it's too cumbersome,’” Decicco notes.

Inclusion

To start, it can be helpful to include software developers in talks about crafting the company’s open source policy. Attorneys should be wary of issuing a procedure or policy like an “edict from a mountain top,” as Decicco put it, because that’s a recipe for failure.

“The more involved the developers can be in the development of those policies and understanding where they're coming from, the easier it is to avoid conflict down the road,” Goldring adds. “Because sometimes, that relationship gets a little contentious, where the developers really wanted to go down a certain path using certain components, but they maybe weren't aware of the risks associated with those components.”

Training

Training developers on these issues is also key, particularly since they might be coming from companies where there wasn’t as much legal oversight of their work. As Goldring notes, some developers may think that open source software is simply “free software from the internet” and they can do essentially whatever they want with it.

“When we do training sessions, it is amazing how many developers respond with, ‘Oh, I didn't know that the original developer could require me to do so many things,’” Goldring says, adding that after going through the training, you could see a “light turn on in their head” that this is something they need to pay attention to.

Additionally, companies may want to create open source training modules, likely with the help of outside counsel, so they’re able to easily onboard new engineers. It can also be helpful to have developers who are well-versed in open source essentially act as liaisons within the development team and field questions related to open source from their colleagues.

“The developers often appreciate hearing it from their own,” Decicco says, adding that, “they're more likely to comply, or think of it less like a roadblock.”

Communication

And of course, ensuring that your OSS policy is effectively communicated to all developers, new and old, is a crucial part of making sure that everyone is on the same page.

> **“The most important thing is to actually have a written policy that you have communicated so that people don't see this as an arbitrary hassle that they have to go through."**
>
> — CHRISTOPHER STEVENSON, ATTORNEY, DLA PIPER

KEY TAKEWAYS

- It’s crucial to get buy-in from software developers on a company’s OSS policies and procedures, otherwise, there’s a risk that they will ignore them.
- Attorneys should be wary of issuing a procedure or policy like an “edict from a mountain top,” and instead have the developers be involved in the creation of policies.
- Training developers on OSS issues is important and companies may want to create open source training modules, so they’re able to easily onboard new engineers.
- Communicating your OSS policy to all developers is a crucial part of making sure that everyone is on the same page.

Exploring The Workflow Between Legal and Developers

Once an OSS license compliance policy is crafted and the company's software developers understand the risks of non-compliance, the next step is to ensure a clear, collaborative workflow between developers and the legal department.

To begin, it's best to integrate the compliance process into the build phase (or, as early in the development process as possible) instead of waiting until the testing phase, according to Goldring. He noted that it's "incredibly valuable" to give feedback to developers as soon as possible in the development lifecycle.

"If a developer can receive license and security feedback — automated or otherwise — on the choices they are making for new components early in the development process, they can avoid having to remediate components that may not be acceptable for use, given license issues or vulnerabilities," Goldring said. "It is much easier for the developers to change course on a component choice before they have built around that component for the solution."

"It's important to always be seeking to reduce the barriers for your developers to gather feedback as they work through the component vetting process," Decicco said.

It's also crucial to try to automate things as much as possible, according to Lindberg, who cautions that if the legal department needs to be in the middle of every decision, "they are going to be overwhelmed and the developers are going to be frustrated."

"An ideal flow starts with automation tooling on the developer side, using built-in scans and services that help you understand what is going into your software at each moment," Lindberg says. "When an issue is noticed by the tool, it can be sent to someone who can put eyes on it and course-correct in real-time."

FOSSA is one such scanning tool, which can analyze the code your developers are writing and identify which components fall under various OSS licenses. It's also important to note that, In addition to building compliance into the development process as early as possible, compliance reviews should also happen in the testing phase to identify anything problematic that has been added to the code.

"Overall, begin the compliance review as early as possible to catch potential issues, and keep that review throughout to catch any incremental changes," Goldring says.

Companies and legal departments may also want to use a color-coding system, which outlines which open source licenses and use cases are automatically approved, need review, or are never allowed. Typically, this is laid out in the OSS policy itself, using a green category (automatically approved), a yellow category (needs review), and a red category (never approved).

“There are many permissive licenses,” says Ning Bao, a Senior Corporate Counsel at Juniper Networks specializing in open source issues, who spoke in his personal capacity. “They're much more popular and they account for the bulk of open source usage in most companies. Those are going to be automatically approved.”

In addition to using software tools to monitor codebases and flag potential license issues, software developers should expect to run into issues that they believe fall into the middle or “yellow” category by the legal department.

“Give a very clear chart to the engineers,” recommends Bao, so if they’re using code that falls in the middle ground, they “expect they need to talk to the lawyer.”

When an attorney is reviewing an OSS component, there are several key pieces of information they need from a developer, according to Decicco. Attorneys need to know what the OSS component is and does, its applicable license, how the developer plans to use it, and the organization’s ability to adhere to the conditions under which the attorney would approve the OSS component's use (such as, in some cases, ensuring it's never distributed or modified).

> **“It's a three-step loop for each OSS component: identify the component and the licensing and the use, analyze it, and then address any remediation or follow-up actions."**
>
> — ANTHONY DECICCO, PRINCIPAL, GTC LAW GROUP

Attorneys should also keep in mind that a quick turnaround time is important when reviewing developers’ open source questions, although the exact response time — whether a couple of days or a week — will likely vary from company to company.

“The faster you are, the more likely they will be to come to you with their questions,” Lindberg notes.

Depending on the company, the developer may also send their OSS component

questions to other internal teams, such as security. Overall, however, it should be the goal of the legal department to work constructively with developers, with Decicco noting that “we want to be the navigation system that helps developers around potential issues; we don't want to be the speed bump in the development process.”

“If you're in the legal department, you don't want to create a bunch of gates for your engineering teams,” he says. “You want to work with them to guide them through this. And we have found that a collaborative process produces the best results.”

The main point is to make sure you allow developers to receive guidance about their component selection process with “as little burden as reasonably possible,” according to Decicco. If you can accomplish that, your developers “will see the benefits outweigh the costs of complying with your organization’s review process,” he notes.

KEY TAKEWAYS

- Integrate compliance as early in the development process as possible, such as during the build phase.
- Try to make the legal team available on the communication platform used by the development team.
- Use automation tools to give you visibility into what is going into your software from moment to moment.
- Consider using a color-coding system that outlines which open source licenses and use cases are automatically approved, need review, or are never allowed.
- Quick turnaround time is important when reviewing developers’ open source questions.
- Try not to be a “speed bump” in the development process; rather, aim to work with developers collaboratively and help them navigate issues.

Navigating Potential Roadblocks

As software developers seek to build a product as quickly as possible, their need for speed may conflict with the law department's necessary legal precautions. As mentioned above, attorneys can address this by turning around OSS review requests as quickly as possible, using the tools that the engineers are already using, and generally doing as much as possible to fit into the developers' workflow process.

One possible point of contention that the legal department may need to navigate, however, is whether developers are allowed to contribute to OSS projects outside the company. On the one hand, attorneys may worry that some of the company's proprietary code could accidentally be released. On the other hand, allowing engineers to contribute to OSS projects can help with recruitment and build goodwill in the larger OSS community.

"People who are graduating now grew up with open source — they might participate as a student in certain open source projects," Decicco says. "So, when they get out in the workforce, you can't have a policy where they're not allowed to continue to do that."

If attorneys are concerned about engineers contributing to open source projects outside the company, they could also ask that the developers request permission to ensure there are no conflicts with the company's interests.

KEY TAKEWAYS

- Allowing engineers to contribute to OSS projects can help recruitment and create goodwill in the broader OSS community.
- Attorneys concerned about engineers contributing to outside open source projects can ask that the developers request permission beforehand to make sure there aren't conflicts with the company's interests.

Preparing for Transaction Due Diligence

Ahead of a major transaction, such as an IPO or a merger or acquisition, companies will want to examine their use of open source (or that of the company being acquired) as part of the due diligence process. Unfortunately, open source issues can complicate a deal, even resulting in buyers potentially walking away from a transaction.

Ultimately, having a system that ensures continuous compliance with open source licensing requirements will go a long way toward reducing risk. Ongoing OSS compliance can be achieved by using key strategies, such as automated tooling, strong collaboration between the legal and engineering teams, and smart policies.

“I think the way to think about any sort of M&A, IPO, etc. — any sort of due diligence event — is that with regard to open source, it is a lot easier to stay compliant than to identify whether you're compliant at a later date,” Lindberg says.

KEY TAKEWAYS

- It’s possible that open source issues could jeopardize or complicate a deal.
- Before a major transaction, companies should examine their use of open source or that of the company being acquired.
- Maintaining continuous OSS compliance is the best way to be prepared for a transaction.

Conclusion

Although it may seem difficult at the outset to establish a solid relationship between the legal department and software developers, it's entirely possible to create a productive workflow between the two departments. Clear communication is always key — from explaining the company's OSS policy to discussing requests to use specific open source licenses — and will lead to a collaborative relationship between the legal department and software developers.

This open channel between the two departments will also help ensure there's robust, continual OSS compliance at your company, allowing you to avoid future open source headaches down the road.

Appendix

https://www.gartner.com/doc/reprints?id=1-26OO2IJ6&ct=210630&st=sb

https://www.linuxfoundation.org/press-release/corporate-open-source-programs-are-on-the-rise-as-shared-software-development-becomes-mainstream-for-businesses/

https://fossa.com/blog/what-do-open-source-licenses-even-mean/

https://fossa.com/blog/all-about-copyleft-licenses/

https://fossa.com/blog/all-about-copyleft-licenses/

https://opensource.org/licenses/alphabetical

https://fossa.com/blog/the-huge-risk-that-most-ipos-miss/

https://www.technologyslegaledge.com/2015/08/doing-open-source-due-diligence/

https://www.lowenstein.com/media/5735/savareplussterba-don_t-open-yourself-to-problems-westlaw-mergers-acquisitions-4142020.pdf

About FOSSA

Up to 90% of any piece of software is from open source, creating countless dependencies and areas of risk to manage. FOSSA is the most reliable automated policy engine for security management, license compliance, and code quality across the open source stack. With FOSSA, engineering, security, and legal teams all get complete and continuous risk mitigation for the entire software supply chain, integrated into each of their existing workflows.

FOSSA enables organizations like Slack, Confluence, Uber, and Twitter to manage their open source at scale and drive continuous innovation. Learn more at https://fossa.com.