

Why Open Source?

There are variety of good reasons to release something under an open source license, from "more perspectives make better software" to "establishing a standard." It is important to building a sustainable project to consider your reasons for publishing as open source and use these as guidance for decision making.

Engineering Economics

While open source work may have benevolent results, it is not an act of charity. Releasing work as open source and the corresponding contribution process eventually result in a higher return on the initial investment made versus the alternative closed source process. John Nash, a famous mathematician and subject of the Oscar winning movie "A Beautiful Mind", won the nobel prize in economics for his work on "cooperative games". He demonstrated that cooperating is not a zero sum game and that by working together all participants may yield higher returns than the investment they make. The best real world example of this may be open source software.

Stephen Walli (<http://linkremoved/>) in his blog post on open source motivations writes, "This wasn't contributing back out of altruism. It was engineering economics. It was the right thing to do, and contributed back to the hardening of the compiler suite we were using ourselves. It was what makes well run open source projects work."

One good example of this is Angular, a web application framework that is used extensively inside of Google. Angular saw rapid adoption by web developers who built extensions and tools which in turn increases the value inside Google as Google uses these extension and tools internally.

Sharing Your Work

PRX122

1:23:cv-00108

You may not have a grand strategy or any intention of shaping the industry with your project, you might just want to share your work to minimize the number of times the wheel needs to be reinvented. This makes it harder to justify investing company resources in growing a community or actively maintaining a project, but you can still "throw it over the wall" so that others are free to use it and adapt it.

Sharing when you can, even if you don't intend to steward the project, can generate good will for your company. However, it's vital that you set expectations so that you do not erode trust and disappoint people who open issues or submit pull requests. Make it clear in your README if you are actively maintaining the project or not.

Establishing or Supporting an Open Standard

Publishing your project under an open source license can encourage adoption of a standard. When a project becomes a standard, you benefit from a massive influx of outside contributors so the project, and the ecosystem around it, evolves more rapidly. This accelerates innovation across the industry and facilitates adoption of services and products you offer that are built on top of the project.

Take GNU and Linux for example: GNU and Linux drew inspiration from the Unix operating system and, being released under free and open source licenses, saw rapid adoption. Linux is now a de facto standard on servers, routers, and connected devices, and is increasingly common on consumer electronics. This has driven software vendors to support Linux and create tools that are familiar to Linux users, for example: support for Bash on Microsoft Windows.

You may also be able to provide leadership in an existing but fragmented field, like Protobuf or OpenThread, but it is not easy to achieve. The drawback in cases without broad adoption are more often than not greater fragmentation.

Disintermediation

Releasing open source software can serve to decouple things and create an opening to compete in a new field. Consider Kubernetes, software for automating management, deployment, operations and scaling of containerized applications.

Kubernetes is being used on Google Cloud, Microsoft Azure and Amazon Web Services. By providing a useful tool that happens to be an abstraction layer, Google made it easier for people to use multiple cloud service providers, or switch between them.

Drive Adoption with a Platform and Ecosystem

Releasing a project as open source allows others to adapt and build on top of your project. When people build on top of your project, they are invested in your success as well as their own. In cohesive communities there are frequently skilled participants who move across company boundaries with a primary affiliation to the project. This set of portable skill sets reinforces the ecosystem growth in addition to helping provide more diverse viewpoints to the project.

This is a particularly effective strategy when the core project has a clear and robust extensibility mechanism, as can be seen with WordPress. WordPress has a thriving ecosystem of contributors, designers, and consultants providing additional functionality through the plugin and theme APIs. The core WordPress product doesn't need to address every possible functional use case or design preference. Instead, the broader community has been able to extend the user base through both free and paid plugins and themes, which benefit all – Automattic (the company primarily behind WordPress), the WordPress project, the developer and designer ecosystem, and the end users.

Paradigm Shift

Technologies and architectures sometimes grow stagnant, and open source projects with fresh thinking can drive sea change. For instance, the release of MongoDB, Couchbase, and other "NoSQL" databases shook up the one size fits all approach taken with relational databases. We saw a sudden explosion of specialized databases and developers began investing time in finding the right tool for the job, choosing from relational databases, document stores, graph databases, etc.

As another example: in 2008, JavaScript was synonymous with slow, and websites that used too much of it were barely usable. When Google published Chromium, the project behind Google Chrome, as open source it also released the V8 JavaScript Engine which compiles JavaScript to machine code before executing it and employs an array of optimization techniques. This drove massive performance gains eventually seen in all major browsers, improving the user experience of existing websites, paving the way for the development of web applications, and enabling JavaScript to be used server-side with software like Node.js. Because V8 was released as open source, the entire ecosystem was able to move forward together, rather than just Chrome and its users.

Recruiting and Onboarding

Recruiting is rarely, if ever, a primary goal of releasing a project under an open source license. But it is almost always a useful side effect, especially when a company open sources things that are widely used internally.

Take Bazel, for example, which is the open source version of Google's internal build system, Blaze. By releasing it as open source it gets used by developers throughout the industry, rather than just those inside the company.

Once a tool is open source, it is possible to recruit from the pool of external contributors, having had the benefit of already seeing someone's work with technology. And onboarding, a long and expensive process, is that much easier when new hires are already familiar with some of the technology and the community building and supporting it.

Why not open source?

Companies are increasingly defaulting to using and publishing open source software as much as they can for many of the reasons discussed above. However, some things should not be made open source.

The number one reason not to open source a project is that it includes, or links to, your company's "secret sauce" or sensitive data because doing so can create security issues or erode your competitive advantage.

Except as otherwise noted, the content of this page is licensed under the [Creative Commons Attribution 4.0 License](https://creativecommons.org/licenses/by/4.0/) (<https://creativecommons.org/licenses/by/4.0/>), and code samples are licensed under the [Apache 2.0 License](https://www.apache.org/licenses/LICENSE-2.0) (<https://www.apache.org/licenses/LICENSE-2.0>). For details, see the [Google Developers Site Policies](https://developers.google.com/site-policies) (<https://developers.google.com/site-policies>). Java is a registered trademark of Oracle and/or its affiliates.

Last updated 2025-06-10 UTC.