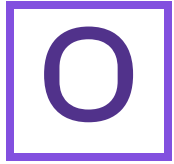


# An introduction to innersource



rganizations worldwide are incorporating open source methodologies into the way they build and ship their own software.

Adopting this modern approach to software development can be transformative—enabling collaboration and fostering the creation of high quality code. But understanding the principles, and potential challenges, is critical to success.

Many companies use the word “innersource” to describe how their engineering teams work together on code. Innersource is a development methodology where engineers build proprietary software using best practices from large-scale [open source](#) projects, like Kubernetes or Microsoft’s Visual Studio Code.

Large-scale open source projects require coordination and teamwork across thousands of contributors. The most successful ones are driven by a vision for their future, in addition to day-to-day user needs: speed, reliability, and functionality. The scale at which these operate can teach us a few lessons—and help your business build better software, faster using innersource.

In this article, we’ll cover the basics of how teams use innersource in their daily workflows and apply open source practices to work collaboratively and efficiently.

## What makes a project open source?

GitHub is home to the largest open source community in the world. What our community builds spans the interests and skillsets of millions of contributors. Together, they work on programming languages like Apple's Swift and frameworks like Facebook's React.js—and they don't just write code. Datasets, legal documentation, and more can also be open sourced.

Anyone can use open source software. They can also view, modify, and distribute a project for any purpose—as enforced by [open source licenses](#). Similarly, anyone can help build open source software. In general, projects are developed by a community of distributed contributors who share code, feedback, bug reports, and more.

Best practices for this kind of open collaboration are fundamental to innersource, and we'll spend the rest of this article going through a few key learnings, tools, and terms.

**To learn more about how people start and contribute to open source projects, check out [our guides](#).**

## Lessons learned from the open source community

If your team builds software, you're probably already building with or on someone else's open source project. The community is rich with lessons on how to collaborate across time zones, rally contributors, and manage large, complex projects. Here are a few to keep in mind as you set up an innersource practice:

### Open collaboration encourages more contributions

More contributions mean more opportunities. Open source projects can accept changes from anybody in the world. That's a lot of brainpower dedicated to advancing a project, meeting user needs, and finding and fixing bugs.

Similarly, innersource brings more ideas to the table. Teams can more easily innovate—and with more people inspecting code for errors and inconsistencies, they can build securer, more reliable software. Problems are found and fixed before the wrong person discovers them.

### **Developers don't always have to start from scratch**

Anyone can discover and reuse open source projects for nearly any reason. People can even use them to build other things or modify them to suit their specific needs.

Similarly, innersource code helps your team discover, customize, and reuse existing internal projects. They can also establish and build on a shared set of documented processes to optimize the way your company deploys and uses software. This can lead to lower cost, greater flexibility, and an end to vendor lock-in.

### **Transparent decision-making builds process, trust, and alignment**

Successful open source maintainers document their decisions by default. Because each conversation has its own URL and a history of comments for context, time zones matter less, and developers can work asynchronously without skipping a beat.

Opening up your projects brings a new level of transparency to your organization. Not only is the code visible but also the process and decision-making behind it. Well-documented conversations help developers on distributed teams get up to speed and start building. And with work happening in the open, collaboration can also include product managers, designers, security teams, and more.

## **Participation is critical**

The success of any open source project depends on participation. There are many built-in reasons to contribute—to improve skills, find a mentor, or build a reputation, for example—but project maintainers also need to create a community culture that welcomes and encourages participation.

Within an enterprise, individual developers can pursue their interests, share ideas on a level playing field, and more easily learn from their peers. However, innersource also requires a cultural shift. Your team's culture will need to encourage knowledge sharing and welcome contributions from across your organization.

## **Core development teams strengthen a project's process**

Open source projects may have thousands of contributors and community members, but a much smaller team is usually responsible for the project's overall direction. This speeds up decision-making and ensures that someone's always responsible.

For innersource projects, distributing control across a smaller group of participants frequently makes approvals and reviews more effective. Creating a small, cross-functional team of decision makers can also help teams stick to quality standards and gain executive support.

# An open source community behind your firewall

Innersource has been described in a variety of ways and developers in organizations that already engage with large open source communities may not use the term innersource at all. Instead, they think of it simply as the way they apply open source methodologies to software development behind their firewall.

Innersource does not imply reduced security or privacy. You don't necessarily have to share proprietary software publicly or invite any outside individuals to view source code or access innersource projects. Companies can rest assured that any nonpublic code will remain securely within their environment—and only developers with appropriate permissions will be able to contribute.



**What we're seeing now is the technology has caught up with all these ideas of innovation and collaboration, and that's really critical for us.**

Joan Watson, Research and Development IT, Emerging Tech and Business Engagement, DXC Technology

Adopting innersource practices is like starting an open source community within your organization. As with open source, transparent collaboration mobilizes a community's collective knowledge and skills to create better software. An innersource community, in contrast, contains the knowledge, skills, and abilities of people and tools within a single enterprise.

## Why do companies adopt it?

As businesses evolve and differentiate their products and services with software and data—or recognize software and data is their product or service—they quickly realize that traditional development methods and tooling don't quite work. The slow, systematic practice of gathering requirements, holding meetings, and developing in silos is not in step with the pace of technology today—or even the pace of customer demands. So how can businesses keep up?

Innersource helps teams build software faster and work better together—resulting in higher-quality development and better documentation. It also can help companies become more efficient by:

- Making it easy to find and reuse code on a broad scale, avoiding wasted resources and duplication
- Driving rapid development, regardless of company size
- Reducing silos and simplifying collaboration throughout the entire organization—inside and between teams and functions, as well as across teams and business lines
- Increasing clarity between engineers and management, as well as anyone else who's interested
- Creating a culture of openness, a precursor to open source participation
- Reinforcing the pride, growth, and job satisfaction felt by team members who help wherever there is a need

Leading companies like PayPal, Bloomberg, and Walmart use innersource to build software for their teams and their customers. It provides them with a unique competitive advantage and helps them stay relevant by adopting recognized best practices.



**We see innersource as a way to improve efficiency through code reuse. But even beyond that, it's an amazing conduit for learning and exchanging ideas and facilitating innovation within IBM.**

Jeff Jagoda, Senior Software Engineer, IBM

## **The anatomy of an innersource project**

The right mix of individuals, teams, and resources can ensure a project's success. Many open source projects follow a similar organizational structure—one you may want to consider when setting up cross-functional teams to manage your company's innersource projects. A typical open source project has the following types of people:

- **Maintainers:** Contributors who are responsible for driving the vision and managing the organizational aspects of the project. They are not necessarily the original owners or authors of the code.
- **Contributors:** Everyone who has contributed something back to the project.
- **Community members:** People who use the project. They might be active in conversations or express their opinion on the project's direction.

Bigger projects may also have subcommittees or working groups focused on different tasks, such as tooling, triage, and community moderation.

Innersource projects are likely to follow a similar structure. Many engineering organizations sort developers into teams like application engineering, platform engineering, and web development. Structuring organizations in this way can leave blind spots that exclude qualified people. Organizing a core decision-making group supported by teams across your organization can help rally the expertise necessary to solve a given problem faster.

Within an enterprise, “contributors” are developers across your company, while “maintainers” are a project’s leaders and key decision makers.

- **Maintainers:** Developers, product managers, and other key decision makers within your company who are responsible for driving a project’s vision and for managing day-to-day contributions.
- **Contributors:** Developers, data scientists, product managers, marketers, and other roles within your company that help drive software forward. Contributors are not necessarily part of the direct project team but help build software by contributing code, submitting bug fixes, and more.

## Tools of the trade

Here are a few tools that drive open source development on GitHub. They’ll also be key components of any innersource project.

- **Issues:** Issues are where developers bring up topics and start conversations. If someone finds a bug or has an idea for a new feature, an issue is a great place to start—and anyone with access to it can join in on the discussion. [Learn more about issues.](#)



- **Pull requests:** Pull requests are living conversations about changes that developers would like to make to a project. They're where people start working on solutions and review changes that are in progress. [Learn more about pull requests.](#)
- **Synchronous chat channels:** Sometimes teams need to make quick decisions. Synchronous chat channels like Slack are complementary to discussions and comments on GitHub and great for talking through problems in real time.

There are hundreds of tools available to use with GitHub that can help your team work better, from project management to [continuous integration and deployment](#) services. [See them all.](#)



**Innersource is not new to Bloomberg ... our competitive advantage is really in being able to innovate come up with new ideas and get them out to our specialized consumers on a regular basis at the speed that they require it to be competitive in the market.**

Panna Pavangadkar, Global Head of Engineering Developer Experience, Bloomberg

## Is it right for your organization?

Innersourcing is as much a cultural shift as it is a technological one—and it's important not to underestimate what a challenge this can pose to some organizations. Like their open source counterparts, innersource projects thrive in places where efforts naturally lean toward discoverability and reuse. It also helps to have small, cross-functional communities of the organization that share similar interests and expertise.

In general, there are a few factors to look for that can help you decide whether to start innersourcing your work—and kickstart the planning process:

- Your company has a vision for your projects that is both realistic and shared across teams. Projects should have clearly defined problems and opportunities to be addressed.
- Key participants (initiators, catalysts, evangelists) in projects you plan to innersource have experience working collaboratively.
- You have a plan to onboard and acclimate new “contributors” and other participants into your process.
- Your team has the tools and processes to communicate openly and build consistently.
- You’re able to start with an intra-organizational group of people with defined shared goals.

Innersource can shift how individuals see themselves and their responsibilities, making organizational structure an important consideration. An effective innersource process should be informal, mentored, self-selecting, and supportive of its participants.

To effectively adopt innersource practices, contributors need to be able to work easily across silos and other organizational divisions. The degree to which an organization supports knowledge-sharing initiatives can be a good indication of how well they’ll be able to adapt.

**For real-world examples, check out [Creating an innersource culture at Booz Allen Hamilton](#).**

These are just a few questions to help assess how ready your team is. If you're answering "yes" to many of these, you may be ready to start an innersource program at your company:

- Does our organization have an open and transparent culture?
- Do we develop software on a single, open platform?
- Are our engineering initiatives well resourced and supported by leadership teams?
- Do developers at our organization have enough autonomy to contribute to projects outside their immediate teams?
- Does our company participate in the open source community?
- Does our engineering team use continuous integration tools?
- Are there pre-existing, cross-functional communities that work across teams at our company?
- If yes, do these communities have built-in leadership?

Every company is different, and none of these should be seen as prerequisites for an organization to adopt innersource. In general, you should start innersource practices when you feel there's enough trust between teams across your organization to have others view, contribute to, and give feedback on their work. Many companies choose to begin with a pilot program and then decide what's best for them.



**Once you embrace it [innersource] and see new teams come on, you show examples of places where not only can people contribute, you unlock bottlenecks.**

Jeremy King, Executive Vice President & Chief Technology Officer for Global eCommerce, Walmart

## Ready to get started?

Companies often kick off their innersource programs by starting small. Pilot projects can help teams experiment with more open processes, democratize access to code, and document best practices before applying innersource more widely. Small successes can help show your internal community of developers how to make the most of their code and ship better software, faster.

The way software is built is fundamentally different than it was a decade ago. Innersource is one approach to modernizing your processes, speeding up development, overcoming organizational barriers, and improving the quality of your software. As the largest open source community in the world, GitHub is where open source best practices start. Together, we can change the way your team builds.

### TABLE OF CONTENTS

Lessons learned from the open source community

An open source community behind your firewall

Why do companies adopt it?

The anatomy of an innersource project

Tools of the trade

Is it right for your organization?

Ready to get started?

---

## FEATURED

### GitHub's DevOps Solution

Learn why 90% of the Fortune 100 use GitHub to build, scale, and deliver secure software.

Start your journey with GitHub