How to accelerate innovation with innersource
## Table of contents

**Intro** 3  
**Defining innersource** 4  
**Five ways innersource can accelerate innovation** 4  
1. Collaborate with context across teams and time zones 4  
2. Get new ideas to customers faster 6  
3. Discover and reuse code across your company 8  
4. Step up code quality and security 9  
5. Save time and encourage innovation with self-service 11
Introduction

Open source has changed the way we build software. From hotel bookings to banking, almost every new application is built on free, reusable code created by a worldwide developer community. According to Gartner, over 87 percent of the IT enterprises across the globe use open source for their mission-critical IT workloads, whether they're aware of it or not.

But process is just as important as output. As open source communities build and grow, they’ve found ways to collaborate quickly and effectively across regions, time zones, and languages. They’ve produced code built by distributed teams at scale, made distributed version control systems an industry standard, and even established the fundamentals behind modern DevOps. If you’re using Git, Python, or sharing and reusing code across your team, you’re participating in open source best practices.

As we’ve seen with DevOps, when enterprise development teams adopt these best-practices they can achieve the same velocity and innovation. By mirroring how open source communities work, enterprises can build software in the way their developers already know and love—creating an environment that’s familiar, free of unnecessary distractions, and focused on innovation through rapid incremental iteration.

What we’re describing is known as “innersource.” Like open source, even if you’ve never heard the term, you’ll probably still recognize some of the principles behind it. Innersource can look different from company to company, but always shares the same goals: adopting open source tools, practices, and culture to build proprietary software.

As home to the world’s open source community, GitHub has seen the impact of innersource firsthand. For example, in our 2021 State of the Octoverse report, we’ve seen that teams that follow innersource principles like code reuse can see productivity increase by a staggering 87%. In this ebook, we’ll explore how organizations like Ford and 3M use these and other innersource best-practices to improve velocity and accelerate innovation—and how GitHub can help.
Defining innersource

Innersource has been described in a variety of ways. Many companies use the word “innersource” to describe how their engineering teams work together on code. 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 at work.

Innersource is a development methodology where engineers build proprietary software using best practices from large-scale open source projects. 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.

Innersource brings best practices found in open source, and thus, an immersive DevOps culture. These best practices include collaborating with developers outside your core team, reducing duplicative work with code that’s shareable and reusable, encouraging transparent discussions and collaboration, and rapid iteration and feedback loops. All combined? Better customer experiences, delivered faster.

Five ways innersource can accelerate innovation

Collaborate with context across teams and time zones

The way people work has changed with globally distributed teams and remote work becoming the norm. Making the most of a worldwide workforce means that no matter where work happens, from HQ to the living room couch, it happens in a collaborative, transparent, and open
way. It also means that organizations can adopt more inclusive and
flexible hiring practices: hiring the right people for the work, regardless
of their location.

Zendesk’s global offices are a great example. The company values
transparency, especially on the engineering side, so that an engineer
in Copenhagen can understand and leverage work happening in San
Francisco. Jason Smale, Zendesk’s VP of Engineering, describes the
team’s internal setup as similar to one you’d find in an open source
project, with maintainers and contributors collaborating on the
company’s ecosystem.

Clear and documented ownership of projects allows each team to
manage the chaos of different workstreams without shutting them off
from great ideas. They leverage the company’s collective brain power
by allowing anyone on any team to contribute to, build on, or reuse
existing work. As Smale explains, “Our engineering culture is open and
centered around teams owning services and being responsible for
running them in production.”

Shared ownership is an important aspect of code quality and
collaboration at Spotify as well. “You need both openness and
ownership,” Product Manager Laurent Ploix said. Giving teams
ownership over particular projects increases their pride in their work
and simplifies decision making because one individual or group
has the final say in approving changes. “With strong ownership, we
avoid code piling up where no one knows whether it’s being used or
not,” Ploix said. But allowing and encouraging non-owners to submit
improvements increases the number of eyes looking for bugs and
opens those projects up to innovative new features. Or, as Ploix said:
“Someone out there might find a better solution than I can.”

Innersource doesn’t just improve cross-team functions, it also helps
teams work asynchronously across geographic divides and varying
schedules—an increasingly important ability for any organization.
“GitHub helps us collaborate with people who are remote,” explains
Erika Moreno Sierra, Senior Software Engineer at Deliveroo. “There are
a lot of discussions happening within pull requests.” Senior Software
Engineer Florian Thomas’ team is currently working with data scientists
who write machine learning models and can quickly link him to new
versions on GitHub. “We have the same access, so it’s easy for me to
follow.”
Even with teams working across locations and regions, GitHub and innersource still provide “a common language” for international organizations like Otto Group to share code and resources. With GitHub as their source of truth, Otto Group has advanced innersource programs within 18 of its subsidiary companies. “This development was driven by a group-wide transformation process that stands for a new era of collaboration,” says Dr. Hanna Huber, Otto Group VP of Technology Strategy and Governance.

And like the open source collaboration that innersource emulates, scalability is built-in from day one. According to Engie’s Tooling and CI/CD Leader Jean-Hervé Laveau, innersource initiatives encourage “sharing capacities and knowledge and sources to scale.” With employees spread across 70 countries, Engie’s developers can “share at scale, all around the world.”

GitHub Tip: To keep work moving across time zones, utilize GitHub as a collaboration platform to discuss ideas, changes, and perform reviews asynchronously with GitHub Issues and pull requests.

Learn more

Getting new ideas to customers faster

Opening up a project invites more ideas. It can also help teams step up their code quality, remove friction from their processes, and get work to production faster. Tom Erickson, Supervisor of Global Software Tools and Processes at Ford, sees the value in not only encouraging more ideas about code but also about process and work styles. “Developers can make suggestions and adopt a style of working that’s more open and fits their needs,” he says. “If you want to ship higher quality software faster, innersourcing just makes sense.”

Speed and innovation go hand-in-hand with innersource, turning the process of code-sharing on platforms like GitHub into a two-fold
benefit for developers and customers: For example, DXC can build applications faster—especially important in times of fast-changing market dynamics—and give developers a single index for all the code they create. For example, DXC technical lead Tom Halpin says that when a colleague had a question about the web testing framework Cypress he was able to simply search the company’s GitHub and find an in-house expert on the technology. “It helps us unlock the intellectual capacity of DXC,” he says.

Now, says DXC Fellow, VP and CTO of Modern Apps and Cloud Native Chris Swan, products they assumed would take six months ended up taking six weeks. “Customers were expecting to have to lay a lot of foundations for themselves before they could build their house,” Swan explains. “But we were able to come along and say ‘We’ve done all that already. Just put up the timber frame and sides.’” As a result, DXC’s customers can get to market faster and be more competitive.

For organizations like Otto Group and 3M, which have multiple divisions and subsidiaries, innersourcing allows developers, not just customers, to get faster access to internal solutions and technology. “You do it once, provide for many,” says Kevin Truckenmiller, Lead DevOps Engineer in Corporate Research Systems Lab (CRSL). Countless 3M divisions use Docker containers and develop in AWS. Since they all require access to AWS, that also means managing thousands of AWS credentials. Before GitHub, individual divisions would create their own credential tooling, resulting in multiple tools for the same purpose. Instead, CRSL created one federation tool to help teams access their AWS accounts using credentials from their corporate directory. Now, teams can easily open a pull request to download the tool’s code from a GitHub repository and put it to use. “Velocity is key. Instead of someone asking if we can build a feature for them, they can open a pull request and add it themselves,” says Truckenmiller. This self-service model enables people to unblock their own dependencies.

Still, successful innersourcing comes down to balance—moving at top speed while not skipping the crucial steps between. To avoid slowing projects down as teams grow, thoughtful processes and documentation take on an equally important role. For example, teams at Stripe frequently work on each other’s code, but no matter where it comes from, it always goes through the same review process. (And
with automation tools, like GitHub Advanced Security code scanning and linters, that review process moves quickly and efficiently.) Documentation on contributing to and reviewing code takes the burden of figuring out “what’s next” off individual contributors, making it easier to get involved on a project’s terms. Stripe Developer Advocate Michael Glukhovsky finds the ability to contribute without friction empowering: “If you see something that needs to get fixed, you can submit a pull request, and start a conversation.”

GitHub Tip: Make sure collaborators know vital information about the codebase and how to participate by adding README and CONTRIBUTING files to your repositories.

Learn more

Discovering and reusing code across teams

It’s an all-too-familiar moment: Your team plans, designs, and implements a new UI pattern for your product. It’s usable, effective, and iterated to perfection over a few months. But then you stumble upon a pull request with a similar feature, developed independently by another team. Not only could your code have spared its creators some work, but now there’s an inconsistency in your product’s user experience. Adapting and reusing field-tested features seems obvious. Making your code discoverable for other teams isn’t.

For ENGIE Digital’s Innersource Project Lead Florent Zara, “[The goal of innersource] is to break down silos and encourage people to work together.” He explains that innersource gets projects out in the open, making it easier to discover, reuse, and build on code across the organization. Charline Grenet, Head of Digital Communities and Communications, agrees. “Everything we do has to be industrialized to be reusable,” says Grenet. “We want to have projects that will be 100 percent innersourced.”
The practice holds up even in larger companies like Autodesk and Ford, where Ford Chief Engineer Florian Frischmuth feels that keeping code in one place, like on GitHub, makes it easier to discover. “Our environment allows developers to find solutions that have already been developed. They can collaborate on those, and then reuse them.”

Previously, Autodesk engineers would build a complex piece of software many times over “because the one that existed only did 90 percent of what we needed,” says Senior Director of the Build Platform, George Swan. They’d copy the branch, make it their own, and add the final 10 percent. Now, employees have access to nearly 100 percent of the Autodesk IP. Of 19,000 repositories, only about 10 are private. “We are open to everybody.”

Through code sharing and reuse, some teams start to resemble the open source communities where innersource and DevOps best practices first began, like at Spotify, where developers who own projects take on roles similar to some open source maintainers. They receive and triage new bugs, ideas, and code. They also might be responsible for deprecating or even archiving their projects. According to Ploix, “They become maintainers, for sure. And with strong ownership, we avoid code piling up where no one knows whether it’s being used or not.”

GitHub Tip: Enterprise accounts can set a default permission level for all GitHub repositories in their organization. Help developers easily find and share code by setting default repository permissions to open or “none.”

Learn more

Stepping up code quality and security

There’s a logical tension between “open” and “secure.” As people release software with a growing number of open source dependencies,
it’s more important than ever that software teams make security a priority from the start. In practice, enterprise teams have found that more collaborators create more opportunities to catch bugs and other issues. They also find that this approach invites security expertise into the process sooner.

At Nationwide, teams once kept projects close to the chest, but opening them up, at least internally, proved integral to higher quality code at scale. Cindy Payne, Nationwide’s Associate Vice President of IT Application Services explained that the projects teams were most hesitant to share often benefited most from outside perspectives. The practice also helped address security vulnerabilities, allowing the team to quickly assess impact, then triage the situation accordingly.

Innersource may happen behind the company firewall, but scalability is always at the forefront. “3M’s core values are not necessarily just company-centric,” says Truckenmiller. “We really do have the rest of the world in mind. We want to protect IP, but at the same time, we don’t want to solve the same problem five ways.” By shifting to a more open innersource model, a single solution can be shared amongst several researchers—with impacts that ripple all the way to consumers. “We’re moving towards more openness, which ultimately creates a communication culture and a generative culture, rather than one that’s bureaucratic and process-based.”

“The base of queries is only expanding, not just through GitHub but through other companies’ expertise. It makes the potential for the value that we get, just at static analysis, to be much greater than it is today.” David Ross, Staff Security Engineer at Postmates

GitHub Tip: Security is everyone’s responsibility, not just your security team’s. Turn on code scanning alerts to help developers find and fix security issues whenever they open a new pull request.

Learn more
Saving time and accelerating innovation with self-service

When it comes to starting new projects, many teams still follow a request-and-wait model. They put in a request, and a central administrator spins up a new repository. This model adds two layers of complexity to what could be a simpler process: administrative maintenance and barriers to trying something new. Innersource, on the other hand, grants everyone the autonomy to get started without friction.

Allowing teams to create their own projects may sound like more work, and the upfront effort of documenting a new process and even creating governance models can seem daunting. But many teams have found that a self-service model ultimately saves time on maintenance and empowers developers to just get started.

Payne finds that self-service GitHub repositories not only remove blockers, but they also save the company time and money from an infrastructure perspective. Of switching models, she says, “We knew we were going to save money, but that savings immediately turned into more capacity for teams to do the most valuable work for our business.”

Established manufacturers like Continental and Ford have also found that removing barriers has encouraged fresh creativity, explains Continental’s Innersource Project Manager Zsuzsanna Gnandt. “With innersource, we want to enable all developers with the freedom to be creative, to drive innovation without barriers, and to be appreciated for their contributions across the company.”

With Ford’s code accessible to everyone in their organization, teams are no longer separated by their organizational structures. Instead, they can work together on new code and make the most out of existing solutions. “Our environment allows developers to find solutions that have already been developed. They can collaborate on those, and then reuse them,” says Chief Engineer Florian Frischmuth.

By leveraging GitHub’s collaborative platform and secure connection to the open source community, Spotify’s developers get the best of both worlds: innersourcing internally, and open sourcing non-IP work.
Whether outside or behind the firewall, thousands of contributors mean thousands of ideas, greater diversity of thought, and ultimately, better customer experiences. “Pull requests are welcome,” says Ploix. “Someone out there might find a better solution than I can.”

GitHub Tip: Recognize developer innovation across public and private repositories with unified contributions. By connecting developers’ GitHub Enterprise accounts with their GitHub.com accounts, they’ll get public credit for their hard work, plus collaboration opportunities and access to new talent and tools.

Learn more