GitHub Copilot

Increasing collaborative development with AI

Christopher Harrison, Senior Developer Advocate, GitHub
Kevin Alwell, Director Solutions Engineering FSI & HLC, GitHub

WRITTEN BY GITHUB WITH ♥

Table of contents

GitHub is where over 100 million developers shape the future of software, together. The platform empowers developers to contribute to the open source community, build proprietary software, secure code, ship it anywhere, and leverage generative AI to accelerate software development.

In July of 2022, GitHub made GitHub Copilot generally available: This tool is an AI pair programmer that assists developers in building performant software more quickly while optimizing for their happiness. Below, we'll discuss how generative AI tools, like GitHub Copilot, are supercharging dev teams and making collaborative practices like innersource even more effective.

Innersource and developer collaboration

A cornerstone of developer collaboration is the practice of innersource, which centers around bringing the methodologies, learnings, and collaborative spirit from open source into internal development. One of its key components is encouraging developers to explore and build upon existing solutions from within the organization rather than defaulting to building from scratch. This saves time and energy, propagates best practices, creates consistency in organization code, and ultimately allows developers to innovate and do their best work, together.

Core to innersource is developers searching for and discovering existing libraries, and contributing to projects maintained by other teams. Unfortunately, the barriers around determining what's available and how to use it, and how best to incorporate new code into unfamiliar codebases, can be difficult and time-consuming. As a result, many organizations don't realize the full set of benefits innersource has to offer. Teams remain siloed and collaboration is stifled.

We'll dive into these challenges below and discuss how implementing AI tools can help developers overcome them. Finally, we'll show you what you can do now to prepare your organization for innersourcing with GitHub Copilot and further AI-powered capabilities that will take your innersource program to the next level.

The challenges of today's innersource

To implement innersource meaningfully, two fundamental developer workflows need to be working well and in tandem: solution discovery and solution contribution. Developers need to be able to easily discover solutions to their problems within the organization and provide solutions to problems in internal projects, too.

Challenges with discovery

The traditional discovery workflow for innersource is fraught with inefficiencies. For example, a developer who is intent on solving a problem must first find the relevant code within the organization before assessing whether it's truly a solution. This is often a time-consuming and fruitless process, driving them to search Google and comb Stack Overflow for generalized solutions instead.

Historically, efforts to improve this process have largely fallen flat: internal repository catalogs, a pillar of enterprise innersource, have not proven to be a consistent accelerator. It's like trying to find a needle in a haystack. But rather than just one haystack, you have several, without any signs pointing you to the best place to start. With poorly documented repositories and obscure siloed enterprise applications, this is the difficult reality of navigating an organization's codebase.

Challenges with contribution

The traditional contribution workflow is laborious for both the repository maintainer and the contributor. The contributor needs to understand the repository they are contributing to prior to making any changes to ensure there are no regressions and that the change does what it is intended to do. The maintainer needs to review the pull request (PR), understand both its intended and actual impacts, then make a decision for whether it should be incorporated into the project. Typically, these efforts are in addition to their day jobs. The combination of these imperfect workflows slows down the innersource process, discouraging contributions from other teams within the company.

How generative AI is already making waves in the software development lifecycle

Generative AI is a form of artificial intelligence that offers suggestions for the data most likely needed by the user. GitHub Copilot is our AI-powered software development platform that has quickly become the world's most widely adopted AI developer tool, trusted by over 20,000 organizations such as Coca-Cola, Fidelity, General Motors, Shopify, Uber, and more. Its services are built upon a large language model focused on making developer workflows more efficient. GitHub Copilot is already making it easier to make the most of innersource, especially when it comes to code discovery.

As highlighted earlier, developers often reach out to search engines and external tools when seeking solutions. GitHub Copilot helps keep developers "in the flow" by offering suggestions in the IDE they're already using. In research conducted in 2022, 77% of developers using the service say they spend less time searching for solutions and 73% say they are able to stay more in their state of flow.

Our vision for innersource backed by generative AI

Today, GitHub Copilot integrates in the IDE and harnesses a model trained on billions of lines of publicly available code and text to offer solutions to problems within ~400 ms. When enabled, we see a double digit percentage of committed code originating from GitHub Copilot.

Developers are already taking advantage of having access to an AI pair programmer to offload tedious, boring and repetitive tasks. Through the use of prompt-engineering or "comment driven development," developers can describe the code they want to create and GitHub Copilot can make suggestions.

GitHub Copilot also looks at the code the developer is adding, offering snippets for the next line, block, function, or class. This is all done by using the context of the current file and those open in their IDE. So suggestions provided are customized for their environment and informed by the conventions and practices GitHub Copilot sees currently in use.

The future of GitHub Copilot

The ultimate vision for GitHub Copilot is to bring AI assistance to the entire developer workflow. With the GitHub Copilot's upcoming features, organizations will naturally start practicing innersource, collaborating more efficiently, and seeing the promised outcomes.

More context helps discovery

With today's version of GitHub Copilot, context is sent to the service to help guide suggestions. This context includes the file currently being worked on and those open in tabs in the IDE. GitHub Copilot customizes the generated code based on this context, including libraries and APIs it hasn't seen before in its training set. This allows it to offer suggestions for internal APIs and SDKs the developer is using in the current workspace.

While this is useful for situations where the developer already knows what library to use, this doesn't help with discovering any that they haven't implemented before. That's why we are building toward enabling GitHub Copilot with greater context about your organization's coding practices (where permitted) and the services that live within your application inventory. With this added pool to draw from, GitHub Copilot will be able to provide more relevant suggestions, including (where permitted) solutions originating from within your organization. All of this is achieved in the editor so developers no longer have to break their flow or manually search through your codebases.

Our early internal experiments demonstrate that contextual customization offers a double digit improvement to the generic service. It's like having a sage pair programmer with decades of experience working for your organization and coding alongside your developers.

For an idea of scale, the volume of tokens we currently support numbers in the thousands! Of course, not all context is the same, so capturing the right context is key. To learn how we ensure the context we share with zthe model is relevant to your development, check out this recent post about how GitHub Copilot is getting better at understanding your code.

More insight helps contributions

When it comes to overcoming today's challenges with innersource, both discovery and contribution workflows will benefit from having AI in pull requests. Combining knowledge from the commit messages and code updates within the pull request, GitHub Copilot will synthesize the proposed change, describe the impact of that change, and articulate why that change was necessary. This additional insight will help maintainers understand the implications of merging code with unambiguous language to support their review, leading to faster time-to-merge for contributors.

Less guesswork

Another barrier to entry when contributing to a new project is understanding the existing code and structure. Documentation is often lacking, leaving it to the developer to explore on their own and determine how the implementation works. GitHub Copilot Chat (now in beta) helps bridge the education gap for guest contributors, thanks to a user interface similar to ChatGPT. Here are a few ways this will improve code discovery and contribution:

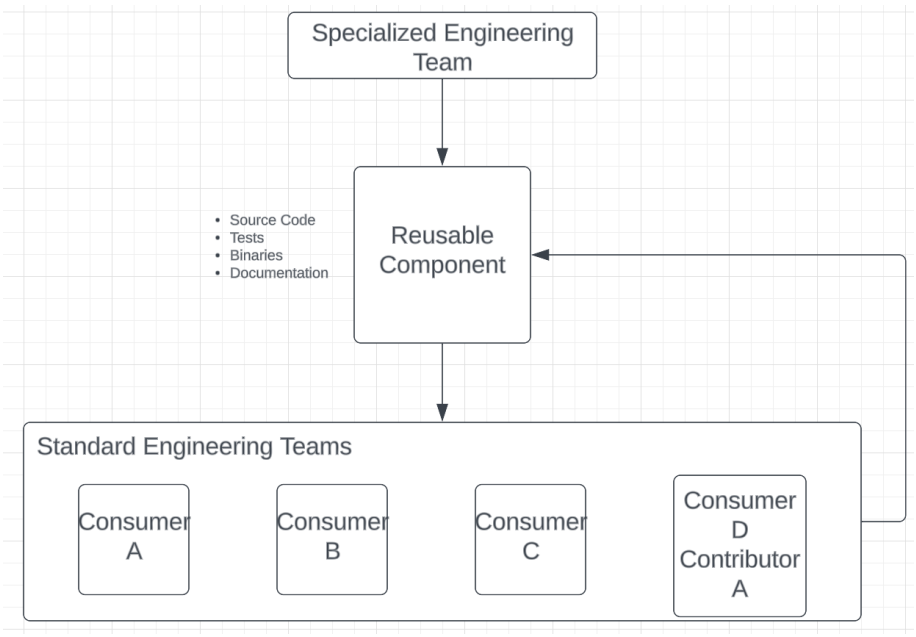
- Developers can interactively ask GitHub Copilot to explain code, decreasing the amount of time typically required to understand a solution.
- GitHub Copilot can offer suggestions for improving and enhancing code, helping with the first step toward a contribution.
- With the anticipated advancements in expanded context and model tuning, developers can prompt GitHub Copilot to ask what is happening within that service and its functions, making it easier to implement a surgical change without causing a regression.

Looking forward

Over time, organizations will increasingly have the ability to optionally augment GitHub Copilot's knowledge base for their exclusive usage. According to a recent interview with Sam Altman, founder of OpenAI, it may be possible to leverage up to a million tokens in a request by the end of 2023. This doesn't directly translate to the GitHub Copilot roadmap, but it does take the tech stack forward by orders of magnitude, which is significant considering **GitHub GitHub Copilot already has an acceptance rate of ~50% for popular languages.**

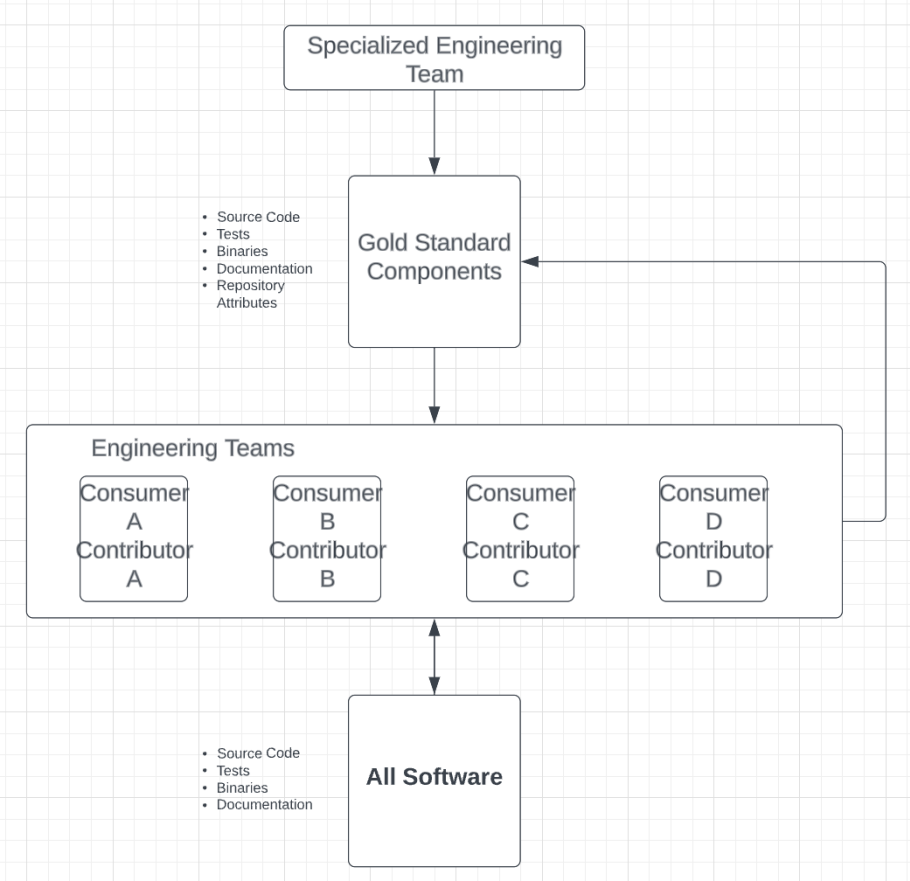
An improved innersource architecture model

As highlighted earlier, many organizations struggle with realizing the full potential of innersource due to challenges with both code discovery and contribution. This leads to a situation where only a small subset of centralized teams maintain reusable components, and the number of reusable components is in turn a small subset of what's possible. Code is duplicated, business logic implementation diverges across projects, and collaboration is limited. It can feel as though an organization has implemented innersource in name only.



The intent of innersource is for code to be shared across the entire organization whenever possible. Developers should start by searching for existing implementations and examples before building on their own. With AI-powered innersource, organizations can move closer to an ideal implementation, where all developers are empowered to utilize and contribute to projects organization-wide. This is collaboration at its finest, and a realization of the power of innersource.

GitHub Copilot can aid developers in understanding existing code and implementations. As GitHub Copilot grows and allows for increased influence of suggestions from internal code, developers will see increased suggestions using existing libraries. Plus, as the codebase grows, the quality of generated code increases, too. This helps drive a model closer to the promise of innersource.



With this new model, **every developer benefits from the flywheel of high quality code written across the organization.**

What you can do today to prepare for an AI-powered world of innersource

GitHub Copilot boasts extraordinary statistics on the positive impact it has on developer satisfaction, productivity, and overall happiness. Here's how to prepare your organization for even more effective collaboration thanks to innersource powered by AI.

Enable the service to normalize its daily usage. As GitHub and your organization make progress toward this innersource vision, updates will automatically ship to the developer's workstation, further enhancing the benefits. As developers use GitHub Copilot, they'll become more comfortable with the technology and learn where it can best support their activities.

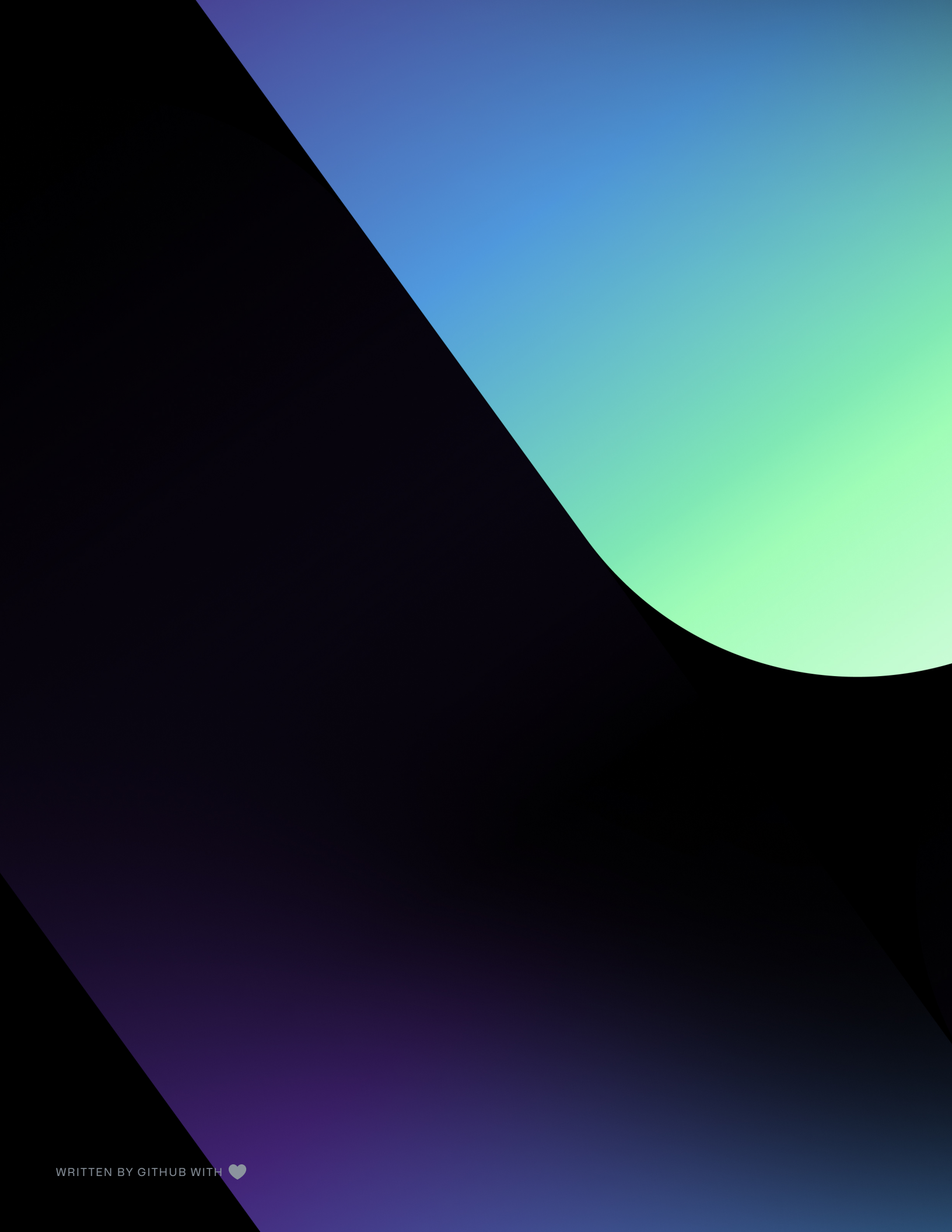
Get plugged into the InnerSource commons. The InnerSource Commons is a community project that documents and propagates best practices, including great resources like the "innersource checklist" that you can work through with your teams. This repo includes books, tools, articles, and repos to help you get started.

Architect your enterprise for collaboration. Consider adopting the Green/Red model for your organization structure, which you can read about in our getting started with GitHub Enterprise Cloud Guide. In this model, 80-90% of your repositories should live in the Green organization, where they'll be discoverable to all developers within your enterprise, regardless of their business unit. The Red organization will be for highly sensitive, private repositories that need another level of assurance layered on to their RBAC model. Any repositories added to the Red organization go through an exception process.

Understand your repository inventory. Tagging repositories using Topics or the latest repository metadata feature "Repository attributes" allows you to classify repositories, for example, as "Gold Standard" or "Business Critical". This may come in handy in the future when you are informing GitHub Copilot of what code it should consider while making suggestions.

We believe organizations are differentiated by how effectively they solve problems together at scale. The future of innersource is brighter than ever. Don't wait. Start now.

Want to learn more? Visit resources.github.com/copilot-for-business or chat with a member of our sales team.


WRITTEN BY GITHUB WITH