GitHub Guides

How to increase developer productivity

WRITTEN BY GITHUB WITH ♥

Contents

Preface

In a 2022 survey, we interviewed more than 1,600 developers, security leads, and executive decision makers across organizations large and small. Despite differing responsibilities and company sizes, their number one goal was the same: developer productivity.[1]

But how can you really measure productivity? Certainly it's more than the number of pull requests or lines of code.

"Developer productivity has been studied extensively. Unfortunately, after decades of research and practical development experience, knowing how to measure productivity or even define developer productivity has remained elusive," according to the Association for Computing Machinery.[1]

After poring over case studies and speaking to experts, here's something we do know: The enemy of productivity in all its forms is fragmentation.

When we discuss fragmentation at GitHub, we're generally talking about the damage organizations can cause by overloading their engineering teams with tools, like extra security plug-ins and administrative management tools that run separately. While the goal might be to keep up with the ever-changing tech landscape, having large amounts of disparate tools does the opposite: slowing down developers and causing frustration while they switch between tabs, open new programs, and push through red tape with each new tool.

But fragmentation stems from more than just the pain of context switching among tools. It also comes from remote workers who don't have the means to collaborate effectively and siloed knowledge due to fast hires and attrition. All of this leads to a poor developer experience—a challenge that reverberates throughout the business, negatively impacting goals and your ROI. When your teams can't get their work done efficiently, productivity plummets.

From enabling better collaboration to adopting code reuse and automation, these tips will help you unblock developers—empowering them to get more work done and enjoy the process.

Enable collaboration the DevOps way

In a Stack Overflow survey of 73,000 software developers from more than 180 countries, 85% of developers say their organizations are at least partially remote.[3] While this means that businesses can find the very best developers for their teams no matter their time zone, it also means that effective collaboration isn't a given, and needs to be built into the developer workflow.

And distance isn't the only barrier to collaboration. Even if they are in the same office, today's organizations have teams working with different programming languages, tools, and frameworks.

73,000

Software developers surveyed from more than 180 countries.

85%

of developers say their organizations are at least partially remote.

> "If a developer within one team wants to collaborate with other parts of the organization, they would need to ramp up fast on the programming language or framework being used within that team, which they might not be familiar with. Not to mention, getting all of the needed dependencies and tooling set up on their machine."

Dhanashri Chavan // Senior Enterprise Advocate, GitHub

Developers spend a lot of time and energy in order to simply work together, which can impact business timelines and goals.

That's why we recommend implementing DevOps, a set of standards that minimizes those silos, turning the linear approach of building, testing, and shipping into a more collaborative effort. With shared tools, responsibilities, and knowledge, teams can communicate better about timelines, goals, and expectations.[6] Thanks to GitHub, teams have a centralized platform to do it all, where tools are easily integrated and easy to scale, no matter how large your organization is.

Let's take a look at Telus, a communications and information technology provider in Canada. They have a team of over 78,000 people and close to 5,000 developers across the world. Before switching over to GitHub Enterprise and adopting DevOps, they struggled with everything from managing dependencies to accessing shared packages.

“With GitHub, we now have a single platform for developer communication throughout Telus,” says Director of Engineering and Productivity, Justin Watts. “Developers can propose and discuss ideas through issues and pull requests, which has really democratized technical conversation.”

Among Telus’ favorite features is GitHub Codespaces, which allows them to create an instant, standardized developer environment for distributed teams of developers, business analysts, and technicians to better develop and collaborate on code—no matter the type of device they are using. “It’s so much easier for developers to use or even contribute to any repo when they can just spin-up a Codespace and immediately start working with the code,” explains Katie Peters, a Telus Developer Advocate. “It’s a huge win for innersource collaboration.”

Take advantage of code reuse

In GitHub’s 2021 State of the Octoverse report, we learned that frictionless code reuse makes developers more efficient and productive. In fact, developers’ performance at work can increase by up to 87% when code reuse is easy and doesn’t introduce friction.[2]

“Open source communities have solved remote, distributed collaboration at scale,” says Chris Reddington, an Enterprise Advocate here at GitHub. “By defaulting to open source principles and empowering anyone to contribute through clear documentation (READMEs and Contributor guides), we can lower barriers and foster a cross-team collaborative culture internally via pull requests.”

That's just what happened at 3M, where disparate teams, knowledge silos, and code scattered across repositories made productivity (and progress) a challenge. After moving over to GitHub Enterprise, their team adopted Innersource—secure code reuse behind their firewall with all the best practices of open source. “Code sharing is fairly large at 3M,” explains 3M Cloud and Security Architect Paul Pottorff. “We have gone from nothing to almost 3,000 innersource repos.”

Those innersource repos on GitHub make all the difference for 3M's developers. "With GitHub, we can collaborate better across our various environments," says Tina Beamer, 3M's IT Manager of Operations and Quality. "You don't have to go out to a separate project management tool. You don't have to go to a spreadsheet, or a Microsoft project, or into Jira. It's all on GitHub. It's made us more productive."

Make the most of automation

From our 2021 State of the Octoverse report, we also learned that good automation helps teams communicate better and more clearly, and it helps them go faster at scale.[2] Chris Reddington adds, "If we can automate our builds, tests and other checks (e.g. static code analysis) earlier in the development lifecycle (before our code is merged into production), then we can be sure that we're building quality software."

That's where GitHub Actions can be valuable. More than 130 million CI jobs are run every month on GitHub Actions—these jobs can help teams ship software faster, doubling the number of pull requests per day and decreasing the time to merge by 33%.[2]

When Procter & Gamble began using GitHub Actions to build automation, their teams were able to work much more efficiently, getting new developers up to speed faster and enabling their developers to reduce manual tasks and focus on innovating.

> "GitHub is an accelerator. It reduces the amount of time it takes for us to bring value to our customers and consumers."
>
> **Danilo Suntal** // DevOps Lead, Procter & Gamble

Concentrate on documentation

Documentation may not be the sexiest topic, but better documentation is a huge key to developer productivity. In fact, developers see about a 50% productivity boost when documentation is up-to-date, detailed, and available in different formats.[2]

“Think of any occasion where you’re learning something new,” says Chris Reddington. “Where do you start? Probably with a getting started document, or some form of instructions, or reference manual. This is no different in software development! Whether we’re looking to find out about the project at a glance (typically in a README.md) file, how to contribute into a project (typically a CONTRIBUTING.md file), or some deeper technical details (usually part of a set of documentation for our codebase). Concentrating on documentation is vital. After all, what good is writing software if people don’t understand how to use it?”

And documentation isn’t just about making onboarding more efficient. It’s about ensuring that developers don’t waste time searching for solutions.

> “For a team of 50 developers, the amount of time spent searching for answers and solutions add up to between 333 and 651 hours of lost time per week across the entire team.”
>
> // Stack Overflow Survey[3]

Those knowledge silos cause massive interruptions to their workflow and those hours add up. Think of what developers could do with all of that time back.

In fact, having good documentation literally pays off through stronger technical capabilities and higher software delivery and operational performance.[4]

Keep developers happy

All four of these tips contribute to the final one: Keep developers happy.

Ensure devs aren't bogged down in manual tasks, reduce friction by allowing them to use tools they know and love, and enable better collaboration. Create trust and transparency. Let them focus on doing their best work. Why? Because happy developers mean a more successful business. It's truly as simple as that.

Teams do their best work when they're personally invested in the outcome.[6]

Sources:

[1] Emerald Research Group & GitHub quantitative survey, n=1693, 2022

[2] The State of the Octoverse | The State of the Octoverse explores a year of change with new deep dives into writing code faster, creating documentation and how we build sustainable communities on GitHub. Accessed September 20, 2022. https://octoverse.github.com/2021.

[3] Stack Overflow Developer Survey 2022." n.d. Stack Overflow Annual Developer Survey. Accessed September 20, 2022. https://survey.stackoverflow.co/2022/#overview.

[4] State of DevOps 2021 - Accelerate." n.d. Google. Accessed September 20, 2022. https://services.google.com/fh/files/misc/state-of-devops-2021.pdf.

[5] University of Victoria, Microsoft Research, and GitHub. 2021. "The SPACE of Developer Productivity." ACM Queue. https://queue.acm.org/detail.cfm?id=3454124.

[6] What is DevOps? DevOps, CI/CD and DevSecOps Defined." n.d. GitHub Resources. Accessed September 20, 2022. https://resources.github.com/devops/.

Got questions about GitHub Enterprise?

We can help.

Visit our **GitHub Enterprise** page
or connect with **our sales team**.

WRITTEN BY GITHUB WITH ♥