5 strategies for navigating AI, the Cloud, and security

Your roadmap to success in 2023

WRITTEN BY GITHUB WITH ♥

Introduction

For developers, there's nothing better than making meaningful progress and feeling good at the end of a day's work. For leaders, there's nothing more important than empowering your teams with the tools they need to work smarter, faster, and more collaboratively.

Fresh off our eighth annual GitHub Universe, where we heard from experts in the industry about everything from how to implement developer-centric security to embracing the latest in AI, we uncovered five important strategies to ensure your teams have what they need to be productive, efficient, and happy at work:

1. Elevate the coding experience with AI
2. Embrace developer-centric security
3. Focus on building, not setup
4. Proactively fix security issues in minutes, not months
5. Scale with the power of the cloud

Check out the strategies below and be sure to tune in to our on-demand sessions from GitHub Universe (organized by track) to learn about ways to build a culture where developers can thrive from day one.

The power of the cloud

All about AI

Developer-centric security

What's inside

Elevate the coding experience with AI

Over the past 70-odd years, technology has changed a lot. But even after years of innovation, coding can still be tedious.

These days, we're focused on machine learning and how it can improve developer happiness and productivity all at once (since we know developer happiness can lead to accomplished business goals and vice versa).

If you're looking for ways to keep developers in the flow, try GitHub Copilot, our AI code-completion and suggestion tool. Our research shows that developers finish tasks faster, feel happier, and say they can stay in the flow longer when using an AI tool like GitHub Copilot. In fact, In a survey of 2,000 GitHub Copilot users, 75% said they feel more fulfilled in their job and less frustrated when coding."

Get started with Github Copilot

> "(With GitHub Copilot) I have to think less, and when I have to think, it's the fun stuff. It sets off a little spark that makes coding more fun and more efficient."
>
> // Senior Software Engineer

At HubSpot, GitHub Copilot was actually essential to catching a critical dormant bug in their codebase. Senior Software Engineer Claudio Wunder talks about how the tool saved his team time and energy in his GitHub Universe session. By simply reading their existing codebase, GitHub Copilot made a suggestion that alerted Claudio's team to the faulty Boolean logic that only would have been discoverable after an analyst started to notice inconsistencies.

> "I call GitHub Copilot 'an assistant' because it gives suggestions and highlights things that can be easily missed by a human. It helped my team prevent disaster by creating better code. It's the small things that create a big impact. That's the magic of GitHub Copilot."

Claudio Wunder // Senior Software Engineer, HubSpot

If you want to learn more about how GitHub Copilot can impact developer happiness and productivity at your organization, catch up on this GitHub Universe session with Seth Juarez, a Program Manager of AI at Microsoft.

Embrace developer-centric security

For a few years now, organizations have made an effort to "shift left"—introduce security testing earlier in the development lifecycle. By moving security checks from a final gate at deployment time to an earlier step, issues can be caught before they make it into production.

While ideologically, it makes sense to "shift left," the issue is that developers now need to take on a lot more responsibility: testing, infrastructure, tracking, and performing security fixes, along with their regular day-to-day work.

If you're a developer and a potential new employer says they're focused on shifting left, you might worry that you're going to be spending a lot of time context switching and trying to figure out new tools—instead of building innovative software.

That's why we at GitHub are focused on "developer-centric security." Yes, we want security integrated into the developer workflow, but we're laser focused on providing tools that empower developers—speeding them up and not slowing them down. We don't want to simply foist new tasks onto them. Developer-centric security means making sure they feel heard.

Nick Liffen, a field architect at GitHub, dives more into the consequences of shifting left in his GitHub Universe session, which you can watch on-demand here.

> "GitHub is building security tools for developers that embed security into the developer workflow. Secret scanning as a push protection is like an 'are you sure' prompt when you're pushing a secret and code scanning is like having a security expert on your team who diligently reviews every pull request. Both are built around the idea that developers are at the heart of everything in application security."

Grey Baker // Vice President of Project Management, GitHub

With GitHub tools, developers aren't alone catching and remediating bugs. It's like having your own personal GitHub tech team to help.

If you're looking for a quick rundown of the new security products from GitHub, take a look at this on-demand video from GitHub Universe.

Focus on building, not setup

One of the biggest barriers to entry for when a team is onboarding a new member or wants to start a new project is the setup.

In her presentation about GitHub Codespaces and machine learning, Tanmayee Kamath explains the difficulty of setup with a cake-baking metaphor. With some programs, you need to look up a recipe, go get ingredients, bake the cake, then decorate. Sure, it might be easier if you start with a cake mix. But the best solution would be to simply pick up a cake made to your exact specifications.

To get developers building faster, whether it's their first day or their hundredth, Kamath recommends Codespaces, a tool that is essentially a fully-baked, already-decorated cake—it allows developers to spin up an environment and start building fast.

At security company Vanta, Staff Software Engineer Robbie Ostro's main goal is to outsource the complexities of software management to a tool that can save his team time. (He's in the security business, after all, not the software business.)

That's why Vanta's 300 engineers use Codespaces for every line of code. When Ostro's onboarding a new developer, he doesn't need to waste a full day getting them set up. Creating a new Codespace takes one minute and restarting one takes just 30 seconds. This lightning speed ability to get going gives his team confidence about their environments and never disrupts their flow.

Codespaces have enabled our own GitHub developers to bring dev environment bootstrapping time down from 45 minutes to 10 seconds.

Get started with Codespaces

See how you can get your developers building faster during Robbie Ostro's Universe session right here.

But Codespaces aren't just simply used for building code quickly, they also enable creativity. Zeke Sikelianos, the founding designer of Replicate, built a stunning version of Mona Lisa Octocat using Replicate and Codespaces in his GitHub Universe session.

Not only is Zeke able to spin up a fully functional coding environment in a matter of seconds, but he takes advantage of Codespaces' new ability to select a GPUas a machine type. From there, he uploads training images to the Codespace by dragging-and-dropping, edits the training script, and soon he can see his first output. In his case, Codespaces makes making art as simple as clicking "code."

Proactively fix security issues

Whether it's intentional or by accident, credential leaks happen.

In fact, contributors leak more than 100 GitHub tokens per day. The problem with leaks is that they enable malicious actors to impersonate others and fraudulently use services, which can have huge consequences if you're, say, Solar Winds, where an exposed credential allowed hackers to inject malware into customer software updates. In this case, one exposed credential impacted 18,000 people and we still don't know the full extent of what the hacker might have stolen.

On an organization level, there's a lot you can do to support developers (who are under increased pressure to deal with security issues themselves).

- First, enable a program that will scan code for leaks.
- Then continually audit and monitor alerts, and remediate by rotating and revoking leaked secrets.

You can do this and more with GitHub secret scanning, a program that prevents unauthorized access and breaches by proactively scanning for secrets pre-commit and searching repositories for leaked secrets that may have accidentally been pushed into your code.

Because scans take less than a second, we can quickly catch leaks as they occur.

Of course, keeping track of all your alerts might sound overwhelming, but it doesn't have to be. In Product Manager Kelly Arwine's GitHub Universe talk, she talks about and demos GitHub's security overview tool, which allows teams to manage their risk and assess their security posture at any time.

In her GitHub Universe session, which you can watch here, Product Manager Mariam Sulakian covers how we as an industry can prevent leaked secrets.

Whether you have 100 or 1000 repositories, you'll be able to see all your enabled security features and alerts. You can even filter across teams, alert types, severity, and more, so developers can easily prioritize where to focus remediation efforts.

Ultimately, security overview provides a holistic view of security findings so you can make informed decisions for your business.

Don't forget about maintainers, either. Software supply chains are some of the largest attack vectors today. And since the world runs on open source, maintainers need all the insights they can get so they can stay on top of vulnerabilities in their dependencies. That's why we just launched private vulnerability reporting, which is a collaborative solution for security researchers and open source maintainers to report and fix vulnerabilities in open source repositories.

With this reporting, maintainers don't need to waste time hunting down email addresses or submitting vulnerability reports through social media. Now, community members can privately submit a report within GitHub to public repository owners, who can then take appropriate action within their GitHub workflow.

Learn about this huge step in vulnerability reporting in open source during this on-demand GitHub Universe panel.

Scale with the power of the cloud

As teams are increasingly spread across the world and remote work is becoming the norm, it's more important than ever to increase collaboration, communication, and visibility. That's why so many businesses are looking to the cloud to keep them connected.

Take Vodafone, for example. Ahmed El Sayed, Vodafone's UK CIO and European Engineering Director said that the cloud provides visibility for their 9,000 developers across multiple time zones and countries. This allows teams to maximize code reuse instead of having to have separate systems in each market—and use that saved time to invest in innovations.

At the core, Ahmed's goal is developer happiness, and providing his teams with best-in-class software like GitHub Enterprise Cloud helps him avoid attrition as he looks to grow and expand his team. He's hoping to bring on another 7,000 software developers by 2025.

The visibility and collaboration that Ahmed is after at Vodafone impacts the bottom line, too. This year's Forrester Report noted that after deploying GitHub Enterprise Cloud, an organization retired its use of legacy tools and realized a total cost of ownership savings of over $800,000 over three years.

As your organization grows, your needs are going to change as well. Solutions that used to work may not be fast or dynamic enough as you go from one developer to 10 to 100.

In this on-demand session from GitHub Universe, you'll learn how moving to the cloud impacted Vodafone, along with other big businesses, like Fidelity, Infosys, and Ford.

In his talk from GitHub Universe, Principal Engineer at Philips Niek Palm, discusses the complexities of being a health technology company (with over 8,000 software professionals) that produces everything from TVs to air fryers. With hundreds of millions of lines of code in all kinds of coding systems, and medical software regulations to follow, it can be a challenge to, well, get anything done. That's why they use GitHub and the cloud, which allow them to use modern tools in a regulated environment.

From utilizing Innersource for collaboration at scale to GitHub Actions to build, test, and deploy software, everything at Philips happens in the GitHub ecosystem. Palm is especially excited about GitHub's cloud-based solution to scale self-hosted runners up and down because it's sustainable (only using compute when you're running a job), autoscaling, and utilizes the elasticity of the cloud. This means that no matter how large his codebase is, he can build, test, and deploy in minutes.

Once your team is up and running in the cloud, you'll need to stay on track of your tasks. That's why we have GitHub Projects, a tool that helps developers project manage and plan, from idea to production. With project boards, it offers a quick glimpse of all tasks that are planned or in progress, either in a repository or across the organization.

In his GitHub Universe talk about how GitHub builds GitHub, CSO and SVP of Engineering Mike Hanley covers how we're working to help your teams find code faster and organize their work to improve collaboration within and across repositories—which is imperative as your organization grows and scales.

First, we built a search engine from scratch to help developers find results in just a few hundred milliseconds. (That means they can search, iterate, and refine their queries to find just what they're looking for.) Then we redesigned code view, so teams can quickly browse code and files across their repositories. Finally, we just launched GitHub Codespaces prebuilds, which enable fast start times for Codespaces created against large repositories and complex applications.

Conclusion

In this year's Forrester report, we found that when a composite organization implemented GitHub Enterprise Cloud, they experienced a 433% ROI over three years.

And with GitHub Enterprise Cloud, GitHub Advanced Security, GitHub Pages, GitHub Actions, and Codespaces, organizations benefit from a reduction in time spent on code remediation and improved engineer efficiency. From these stats, it's clear that there is a symbiotic relationship between financial savings, developer efficiency (and satisfaction), end-to-end security, and having best-in-class tools to get the job done.

Whether you were able to attend this year's GitHub Universe or you're still making your way through our on-demand sessions, we hope you're feeling excited to embrace AI, adopt developer-empowering security practices, and take your work to the cloud—and look to GitHub Enterprise to help you grow, scale, and take your software into the future.

- The power of the cloud
- All about AI
- Developer-centric security

Ready to try GitHub Enterprise?

Start the trial here.

WRITTEN BY GITHUB WITH ♥