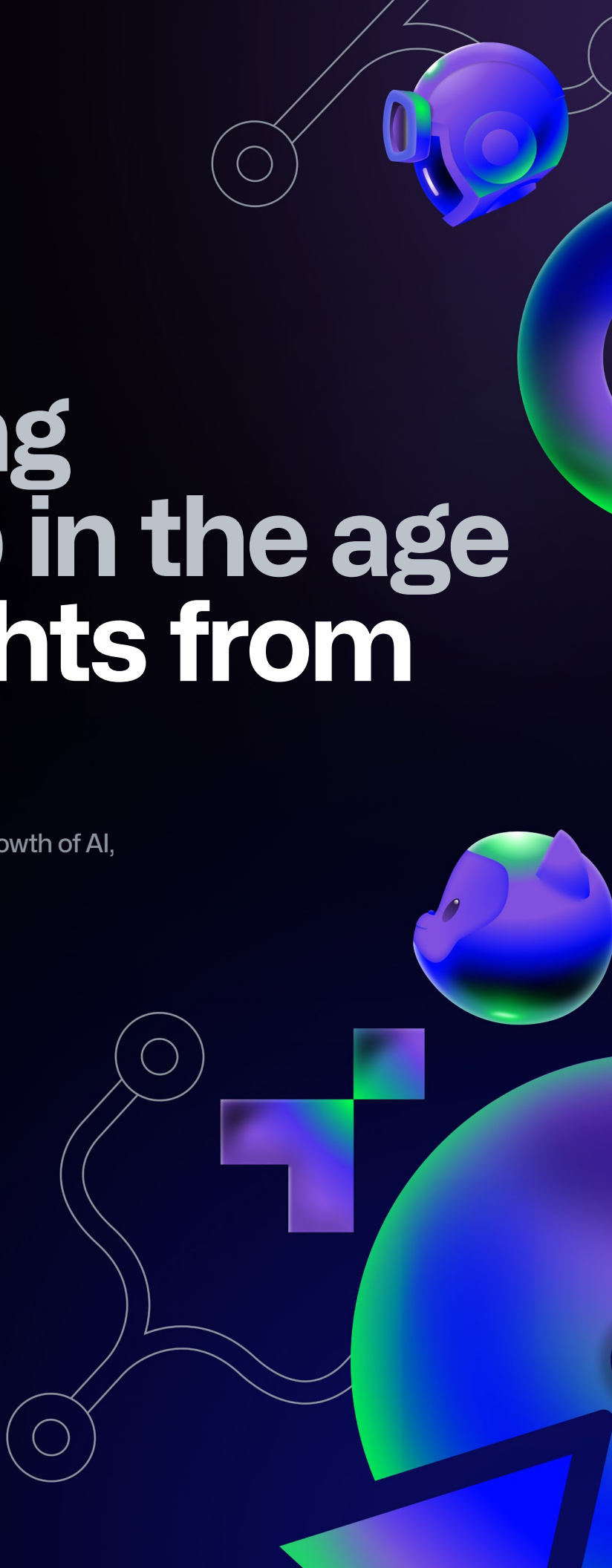
Engineering leadership in the age of AI: Insights from GitHub

What organizations need to know about the growth of AI, Python's rise, and the global developer boom.



The world of software development is experiencing a seismic shift. Not only is the developer tech stack changing, but the very notion of who a developer is and what they do is rapidly evolving.

We recently published our Octoverse report: an annual exploration of software development trends across more than 150 million global developers who use GitHub. This report is a critical resource for technology insights, highlighting key industry trends, developer priorities, and competitive opportunities for business and engineering leaders.

In this guide, we'll distill our findings, including:

- The global surge in generative AI activity across GitHub, and how developers are using AI in their workflows and building AI-powered applications with agentic AI tools
- How Python emerged as the the top language used by developers, due to its utility in AI, data science, and machine learning
- Why innovation is increasingly powered by the convergence of open source and generative AI technologies

We'll also discuss what all this means for today's organizations, as each of these insights has huge implications for how leaders should equip their teams, strategize, and invest in the months, quarters, and years ahead.

CONTINUED ON NEXT PAGE

AI is changing the development landscape

Key takeaways for business and engineering leaders

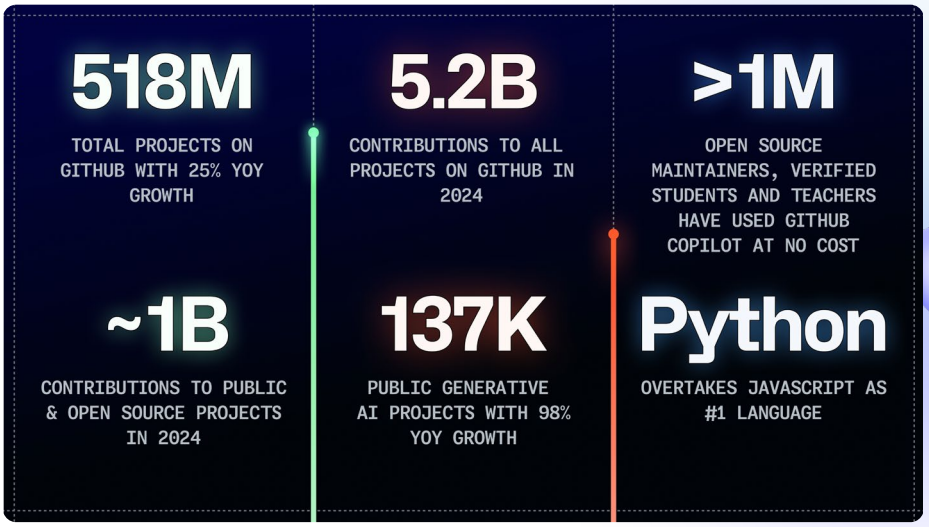
- AI coding tools are now central to software development
- AI is driving productivity (up by 55%[1]) and improving code quality
- Developers are shifting toward more efficient AI models that reduce compute and operational costs
- AI agents (e.g., GitHub Copilot Autofix and Copilot Workspace) are extending developer capabilities
- AI is attracting more developers and expanding talent pools
- AI-powered security tools are helping bridge the cybersecurity talent gap

1: GitHub, "Research: Quantifying GitHub Copilot's impact on developer productivity and happiness." September 2022.

In the 2024 Octoverse report, we found a 98% increase in the number of generative AI projects on GitHub, proving that developers across the globe are rapidly embracing and building with AI.

It's clear that a shift has occurred in the past year: Generative AI has moved from a curiosity and area of experimentation to a core reality in today's software development lifecycle. Developers aren't just using AI-powered coding tools like GitHub Copilot to code—they're using them to extend their workflows, learn, and build new, AI-powered applications.

CONTINUED ON NEXT PAGE



In the last year, developers on GitHub have:

- Contributed to AI projects that help developers run AI models locally with more efficient compute requirements
- Experimented with building AI-powered applications through playgrounds, such as GitHub Models. We've seen a number of open source projects integrating SDKs from popular AI firms, like OpenAI
- Developed AI agents to accomplish key tasks, like to control tools within their tech stacks and extend the power of LLMs beyond a chat-based interface

Early research showed us that AI-based developer tools like GitHub Copilot could help developers complete tasks up to 55% faster.[2] And our more recent research has underscored that these time savings are coupled with benefits such as higher code quality, too. [3]

In short, generative AI is increasingly being embedded in enterprise engineering workflows and applications as adoption increases—and this pattern is being compounded by a largescale interest in AI among developers globally across GitHub.

2: GitHub, "Research: Quantifying GitHub Copilot's impact on developer productivity and happiness." September 2022.
3: GitHub, "Does GitHub Copilot improve code quality? Here's what the data says." November 2024.

CONTINUED ON NEXT PAGE

AI is expanding the talent pool

Our data also shows AI isn't just a key technical area of interest for developers; it's also drawing more people into software development globally. In fact, in the past year, both open source and public AI projects on GitHub saw some of the highest activity on the platform.

In 2024, more than one million maintainers, teachers, and students used GitHub Copilot to accelerate workflows. This shows us that AI coding tools can play a key role in educating the next generation of developers. A study from the Brookings Institute found that certain groups of developers benefited more from using GitHub Copilot–and one of those groups included less-experienced developers with fewer years of professional coding experience.[4]

This underscores how young developers are turning to AI tools like GitHub Copilot as learning aids, creating a new generation of developers that is increasingly AI native. Our hypothesis? AI coding tools, including Copilot, will become even more ubiquitous as these graduates enter the workforce.

From recruiting talent to understanding what applications have the most potency for future investment, the way developers are engaging with AI on GitHub across open source projects and the tools they use has real implications for organizations today.

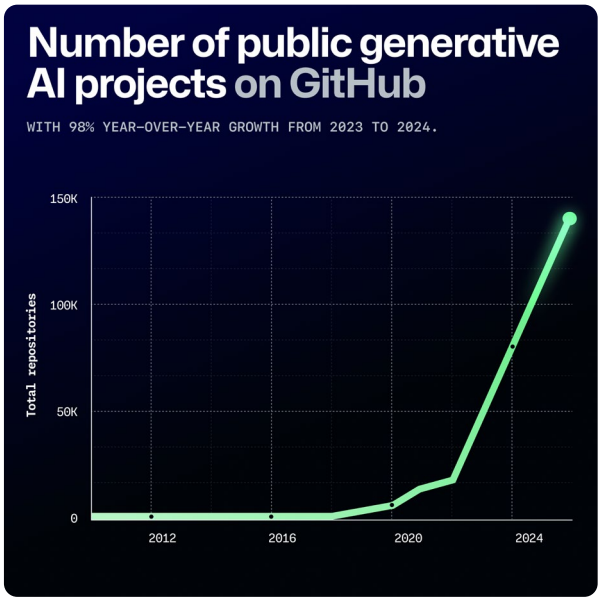
Smaller, more efficient AI models are on the rise

Well before DeepSeek became a news item, developers on GitHub were already experimenting with smaller AI models. With lower compute costs and reduced operational expenses, these models can often be run on local devices, which brings down the computational requirements to run and experiment with AI.

For organizations, this is important. These models can not only integrate with enterprise applications, but they also require less capital and operational expenditures to run. Another benefit? They can be run on local devices, making them more efficient than other models–sometimes even including cutting-edge frontier AI models.

4: Brookings, "How AI-powered software development may affect labor markets." August 2023.

CONTINUED ON NEXT PAGE



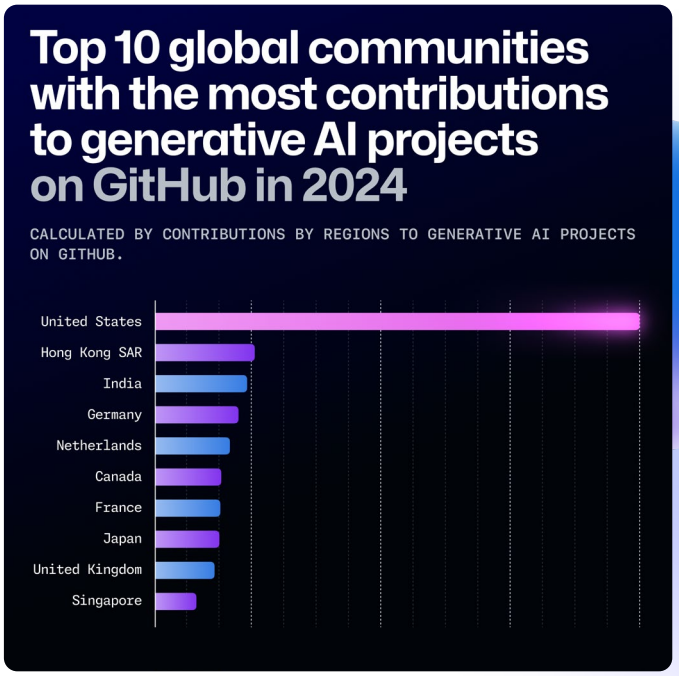
Agentic AI extends the power of LLMs

In the past year, we have seen more developers on GitHub experimenting with agentic AI applications—or applications that help extend the capabilities of an AI model past a chat-based interface to control other tools, surf the web, or even develop software prototypes.

In particular, developers have been building tools to simplify their workflows while creating experimental applications that can execute actions on their own. Notably, agentic AI tools like GitHub Copilot Autofix, a security remediation tool, aren't just allowing developers to build more secure software faster, but they're also opening up the development of new types of software for organizations across industries.

One example of agentic AI is Copilot Workspace, a Copilot-native developer environment by GitHub Next. Copilot Workspace is a multi-agent system where each agent communicates and coordinates with others to solve and carry out complex problems more efficiently than just one agent alone. For instance, a developer can ask Copilot to help create an app in Copilot Workspace, and they'll receive a software development plan and the code needed to create the app.

CONTINUED ON NEXT PAGE



Why should my organization prioritize AI tools?

- **Improved productivity:** Research at GitHub has found that developers on average complete tasks up to 55% faster with AI coding tools.[1]
- **Increased team collaboration:** 70% of developers who use AI coding tools say they offer significant advantages at work–and more than 4 in 5 developers expect AI coding tools will make their teams more collaborative.[2]
- **Higher quality code:** Tools like GitHub Copilot can help developers write more functional code, improve readability, produce a higher quality of code, and obtain a higher approval rate.[3]
- **More streamlined workflows:** Agentic developer environments such as Copilot Workspace help consolidate workflows by utilizing semi-autonomous agents that make the technical aspects of writing code and building applications even simpler.
- **Better security:** New security tools such as Copilot Autofix are bringing AI to traditional security domains to scan code, highlight vulnerabilities, and generate fixes.

1: GitHub, "Research: quantifying GitHub Copilot's impact on developer productivity and happiness." September 2022.
2: GitHub, "Survey reveals AI's impact on the developer experience." June 2023.
3: GitHub, "Does GitHub Copilot improve code quality? Here's what the data says." November 2024.

CONTINUED ON NEXT PAGE

3 strategies for building an AI-powered developer ecosystem

In the coming years, organizations will need to be strategic about how they adopt and deploy AI tools, and enforce change management to steer their use.

Train developers in key AI principles and enable experimentation

As AI models increasingly become an integral part of the developer's tech stack, engineering leaders will need to maximize the efficacy of these tools while also considering their organization's specific needs.

One way your teams can do this is by exploring prompt training, or socializing what prompts and workflows work for the broader organization. Building a culture of knowledge sharing and prioritizing upskilling amongst your teams is important. For example, you could set up a repository or Slack channel with prompt training tips.

Evaluate tools stringently for data privacy and security

Amid a rapidly evolving field of AI coding tools, it's imperative to ask vendors about their data privacy and security standards.

Start small with pilot teams and projects

At GitHub, we advise companies to begin with small-scale pilot projects and teams to test the AI tools' effectiveness and identify any challenges and benefits in advance of rolling them out to a wider organization.

CONTINUED ON NEXT PAGE

Shifts in programming language trends

Key takeaways for business and engineering leaders

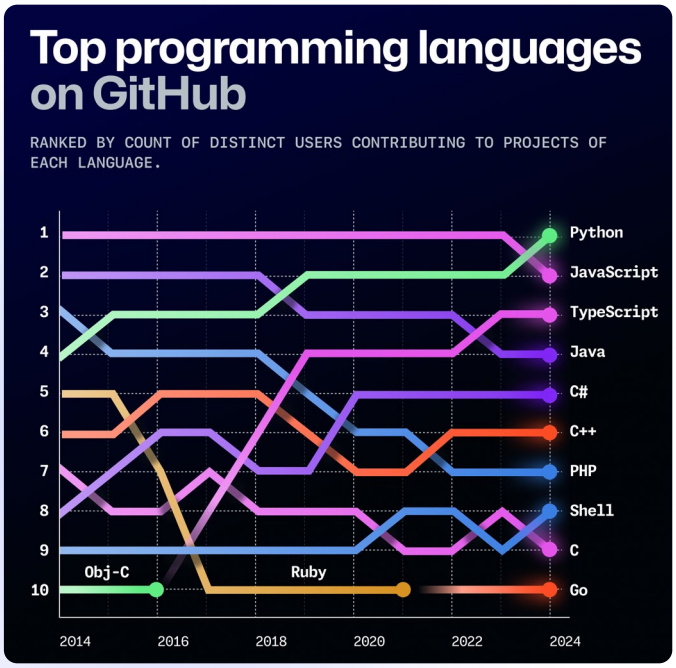
- In 2024, Python overtook JavaScript as the most-used programming language on GitHub.
- A global talent pool of developers is building expertise in Python. This rise in Python usage correlates with large communities of people joining the open source and global developer community from all STEM fields, rather than just the traditional developer community.
- TypeScript overtook Java to enter the top three programming languages on GitHub.

After 10 years, Python surpassed JavaScript as the most used programming language on GitHub in 2024. There hasn't been a shift in the top two languages since 2019, when Python overtook Java in popularity as the second most-widely used language.

We believe that this rise in Python usage stems from two areas. The first is that large communities of people are joining the open source and global developer community from all STEM fields, rather than just the traditional developer community. The other: Python is heavily used in AI, machine learning, and data science, among others.

Even still, this growth is striking. JavaScript has long reigned as the top language on GitHub, and its movement to second place signals the significant impact generative AI is having on developers and what they're building in public and open source projects on GitHub.

CONTINUED ON NEXT PAGE

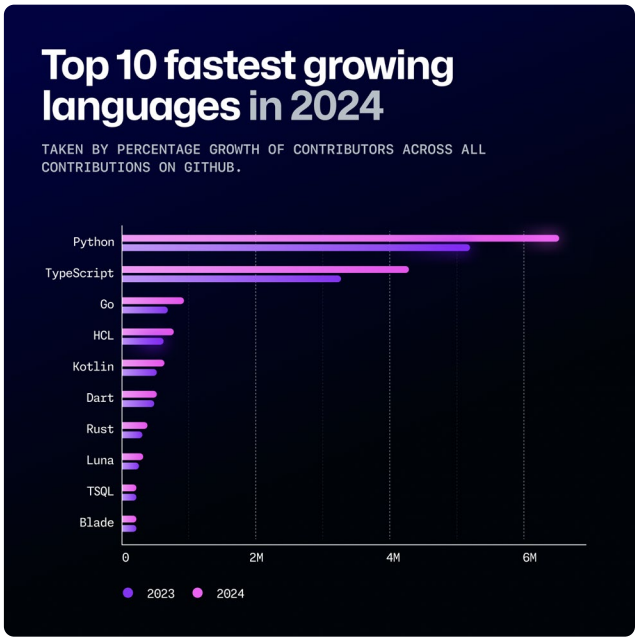


For example, when combined with a complementary tool like Jupyter Notebooks, Python can be used for developing, testing, and training AI models, along with exploratory data analysis and data visualization. In fact, in part due to an increased focus on data analysis and scientific computing on GitHub, we've seen a 92% spike in Jupyter Notebook usage on GitHub in the past year.[5] With files hosted in a version control system like GitHub, Jupyter Notebooks can be helpful for collecting, understanding, manipulating, and collaborating on data for research, reporting, and analysis.

5: GitHub, "Octoverse: AI leads Python to top language as the number of global developers surges," October 2024.

The prevalence of AI and an increased focus on data science and machine learning means that engineering leaders should consider their current developer teams' familiarity with Python and seek out developers with Python expertise—as this is now a strategic asset for large enterprises and a preferred choice for critical projects.

CONTINUED ON NEXT PAGE



Python isn't the only language making gains in popularity: Our Octoverse report shows that TypeScript overtook Java as the third most popular programming language on GitHub last year. TypeScript is a superset of JavaScript that can help to improve performance and security. Because it is a type-checked language—meaning validations are automatic—TypeScript provides developers with more immediate feedback as they're building apps. In practice, this means developers get extra assurance that they aren't breaking the project as they go, which ultimately leads to higher-quality code.

Where TypeScript can be increasingly valuable is when enterprises are looking to modernize and adopt the latest technologies without leaving their legacy systems behind. Because TypeScript extends JavaScript, developers can incrementally modernize JavaScript code while reducing technical debt in the process.

CONTINUED ON NEXT PAGE

The importance of open source

Key takeaways for business and engineering leaders

- Working in open source provides business benefits
- The uptick in AI is bringing more people into the open source community, creating a strong business need for more developers with AI expertise
- The widespread use of open source underscores the need for better supply chain security and a proactive security culture

Many developers learn by doing, and open source often acts as a training ground while helping to build a talent pool of developers versed in key technologies. For seasoned developers, continuing to work in open source environments offers a powerful platform to not only give back to key projects that are critical to their workflows and businesses, it also offers a way to further customize solutions to their needs and systems. After all, building off of something that already exists is often the most efficient way to build software.

As AI brings more and more people into software development and the open source community, we're seeing the creation of a virtuous cycle: Businesses need more developers, especially with AI expertise. With more people entering software development by way of AI *and* open source, the global talent pool is growing fast.

In the past year alone, there were nearly one billion contributions to public and open source repositories globally on GitHub.[6] That being said, when developers are working in open source environments, it's crucial to have the resources to maintain and secure all repositories, especially when generative AI is driving the activity. And because open source components appear in 96% of codebases,[7] organizations need to build the right infrastructure and policies to ensure system security and supply chain security, while also improving their understanding of the wider ecosystem that powers their applications.

6: GitHub, "Octoverse: AI leads Python to top language as the number of global developers surges." October 2024.

7: Hoffmann, Manuel, Frank Nagle, and Yanuo Zhou. "The Value of Open Source Software." Harvard Business School Working Paper, No. 24-038, January 2024.

CONTINUED ON NEXT PAGE

If your team is looking to improve how it engages with open source software and to create a more proactive culture around security, here are a few steps to take:

Use tools that help keep vulnerabilities out of code

It's more important than ever to secure your repositories, and there are native tools that do this quickly and efficiently. GitHub Advanced Security detects, prevents, and fixes vulnerabilities, while helping developers and security teams work in tandem to eliminate security debt. This allows developers to focus on building better software rather than spending time addressing security issues.

Establish an Open Source Program Office (OSPO)

When building with open source, organizations must remain compliant with license agreements and rules. That's where OSPOs come in. These teams help organizations standardize their approach to using open source properly, along with ensuring that their work follows regulations, and that security measures are implemented across dependencies.

Incorporate security best practices that support compliance

One way organizations can maintain security standards is by following best practices and using tools like OpenSSF Scorecard to ensure high-quality contributions and compliance to open source environments. Scorecard checks for vulnerabilities affecting the software supply chain, assesses the risk level, and also helps to fix the problems.

Encourage developer teams to give back to open source

Building internal programs that encourage developers to engage with open source development securely is a great way to contribute to the movement. This might look like hosting a Hackathon, a mentorship program, or even a Summer of Code. When you give developers space to devote energy to projects they're already using, it's a win-win situation—and leadership teams can drive motivation by incentivizing developers to build features that will benefit their organization.

CONTINUED ON NEXT PAGE

Take this with you

With the onset of more powerful AI models and agentic AI, the scope of what AI can help engineering teams accomplish is only increasing. This means that AI adoption is no longer optional. But integrating new technologies into your organization doesn't have to be difficult: In 2024, more than one million maintainers, teachers, and students used GitHub Copilot to accelerate their workflows, and the next generation of developers is learning with GitHub Copilot and will take their learnings with them as they enter the workforce.

Our global community of developers is growing—and the limits of what we can accomplish together with generative AI and AI agents are pushing the boundaries, too.

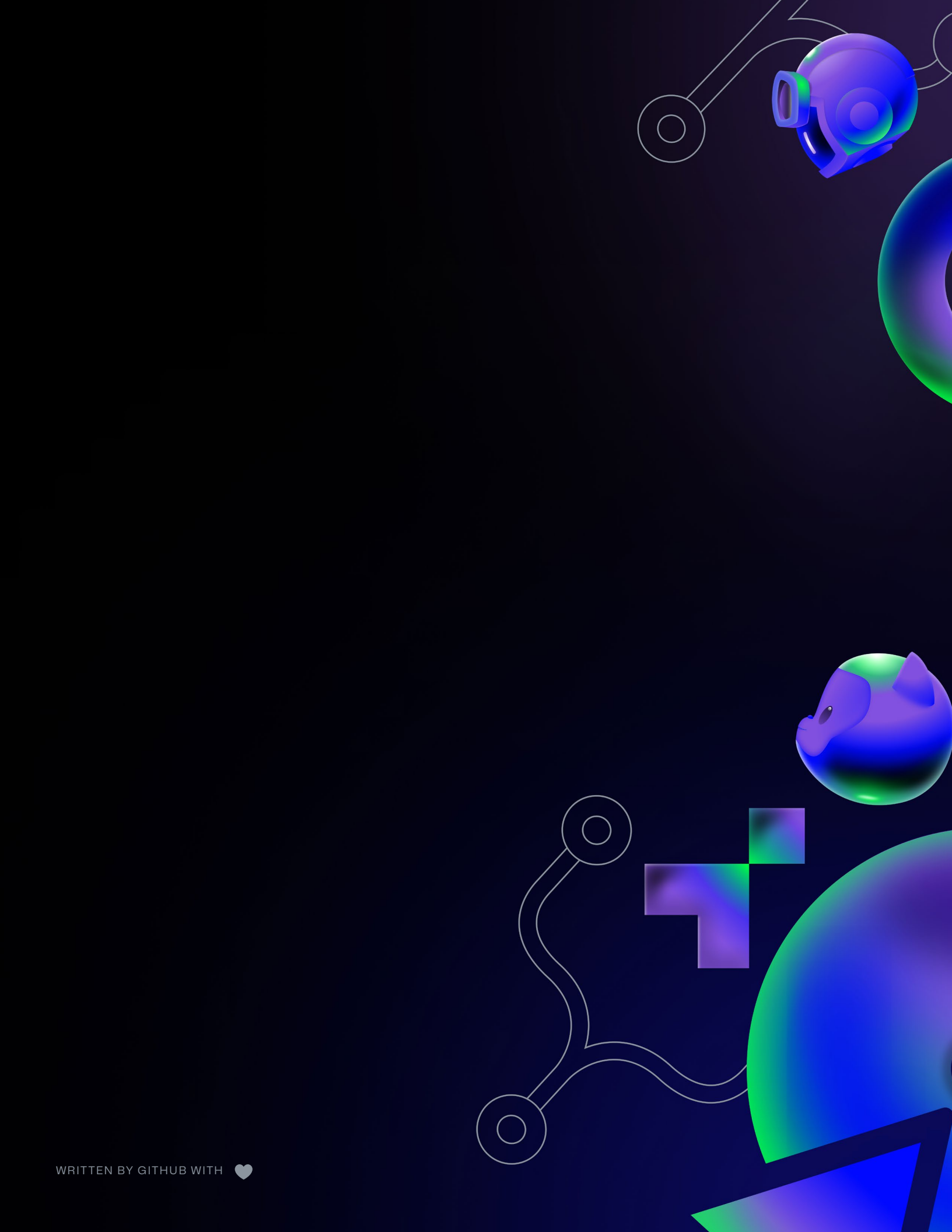
So what can you do now?

- Invest in AI coding tools to accelerate software delivery
- Experiment with and implement AI across developer workflows
- Tap into the talent pool of expert developers with knowledge in Python, who are primed for AI-driven innovation in enterprise ecosystems
- Increase supply chain security and use native security tools to catch vulnerabilities before they hit production
- Make use of open source projects and give back to the greater community to solve the industry's most complex problems

CONTINUED ON NEXT PAGE

Home to more than 100 million developers, GitHub is the world's most widely adopted AI-powered developer platform that's empowering organizations to build, secure, and ship software faster to unlock innovation at scale. Our comprehensive platform integrates enterprise-grade tools, such as CI/CD, automation, application security testing, cloud development environments, collaboration tools, and AI-powered coding tools, to facilitate the swift delivery of secure software. Plus, it is compatible with all cloud providers, so companies can confidently scale their software delivery without sacrificing familiarity.

For more trends, solutions, and strategies, check out our 2024 State of the Octoverse Report



WRITTEN BY GITHUB WITH ♥