



GitHub Universe 2023



[Dan Meineck](#)

[GitHub](#) recently held their [GitHub Universe](#) conference in San Francisco where they talked about what they are doing to make developer's lives easier and, as a GitHub Technology Partner and, we were again invited to hear about the future of development.

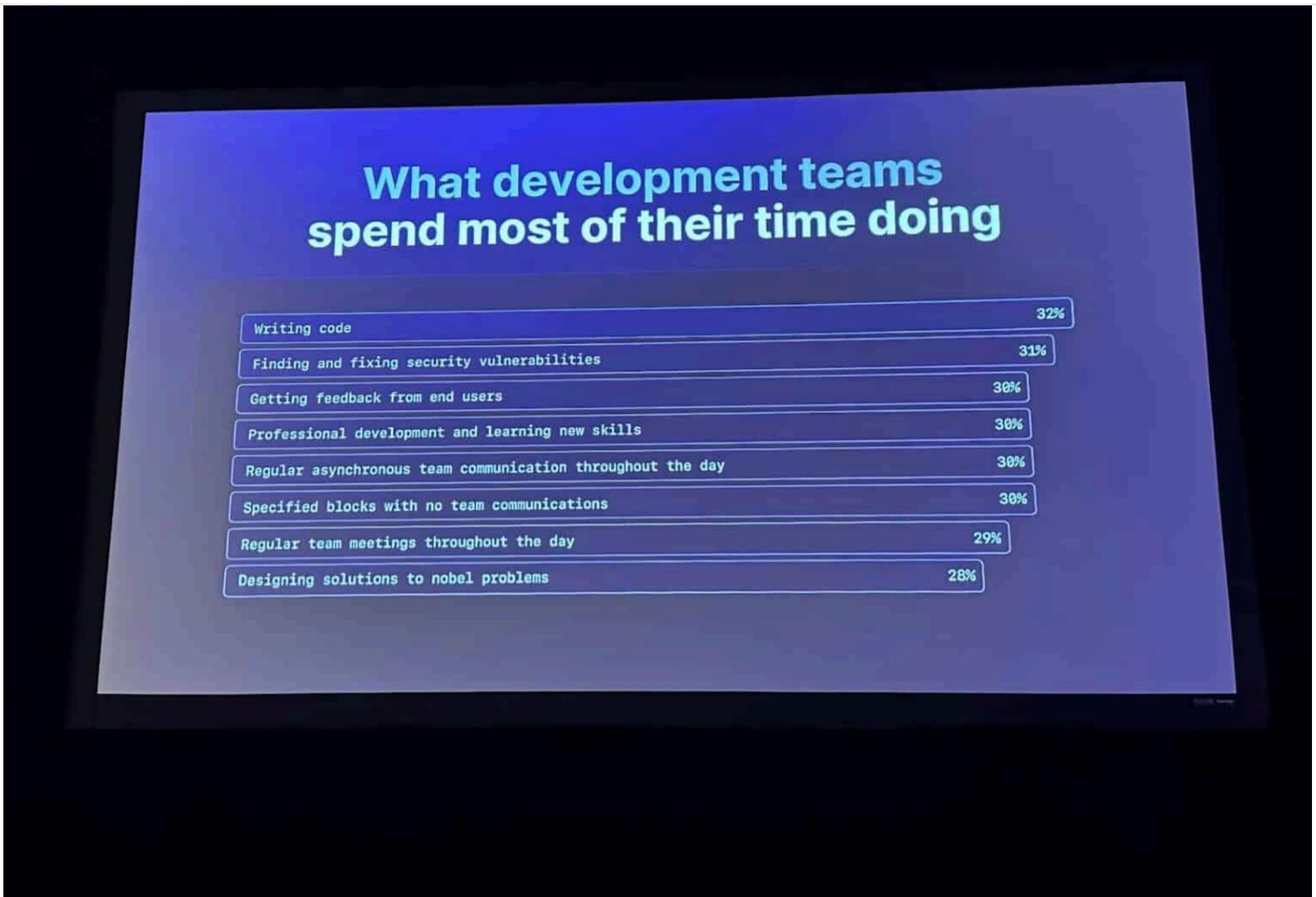
In a year that has seen generative AI and Large Language Models hit mainstream audiences, it was exciting to see how GitHub have evolved their AI offerings. GitHub recognised the opportunity that gen AI presented back in 2020 from a whitepaper written by Oracle. They started developing against this with what people now know of as GitHub Copilot. Since then, over a million paying developers actively use Copilot, which is now the world's most widely adopted developer tool in history. Remarkable.



Fast forward to today, and GitHub consider AI to be “foundational and fundamental across the software development lifecycle”. In fact, according to GitHub’s CEO Thomas Dohmke, *“GitHub is re-founded on Copilot. AI is now the electricity powering the home for the world’s developers”*. Considering that GitHub was originally founded on top of Git, this is a massive statement and signals how fundamentally important AI is to the future of software development. Their announcements at the conference take the evolution of their AI offerings to an entirely new level.

Flow State

GitHub claim that it takes an average of 23 minutes of uninterrupted focus time for a developer to enter the “flow state” – a state of hyper productivity where work gets churned out at a high rate and time seems to disappear without the developer even noticing. It’s very easy for this state to be interrupted, e.g., meetings, messages, emails, calls etc. Even aspects of coding itself such as having to search online for documentation, error codes, best practice, or supporting team members with reviewing pull requests – will tear a developer out of that flow state at which point that 23-minute warm up period needs to repeat to allow flow state to be resumed.



GitHub created [Copilot](#) to reduce the distractions a developer faces and allow them to maintain their flow state – Copilot will suggest code snippets for the developer without them having to jump out of their IDE and into their search engine of choice.

Not only does the developer avoid exiting their IDE into a search engine, but even if they are using ChatGPT, the context which Copilot has is immediately far greater than the prompts they could craft. Copilot will look at the open tabs of their IDE, the existing code and way in which patterns have already been implemented and use this to provide them with the code snippet they need, all without that developer having to context switch out from one environment into another to achieve their goal.

So, all of this is not new – Copilot was announced last year with these promises, and since then 1 million of us have been happily enjoying the benefits of it long before ChatGPT had entered the public lexicon. In fact, 1 in 3 Fortune 500 companies use GitHub Copilot. What GitHub have done since then is evolve the Copilot experience and splice it into many more aspects of the software developer's world, to further improve the developer experience.

What was announced?

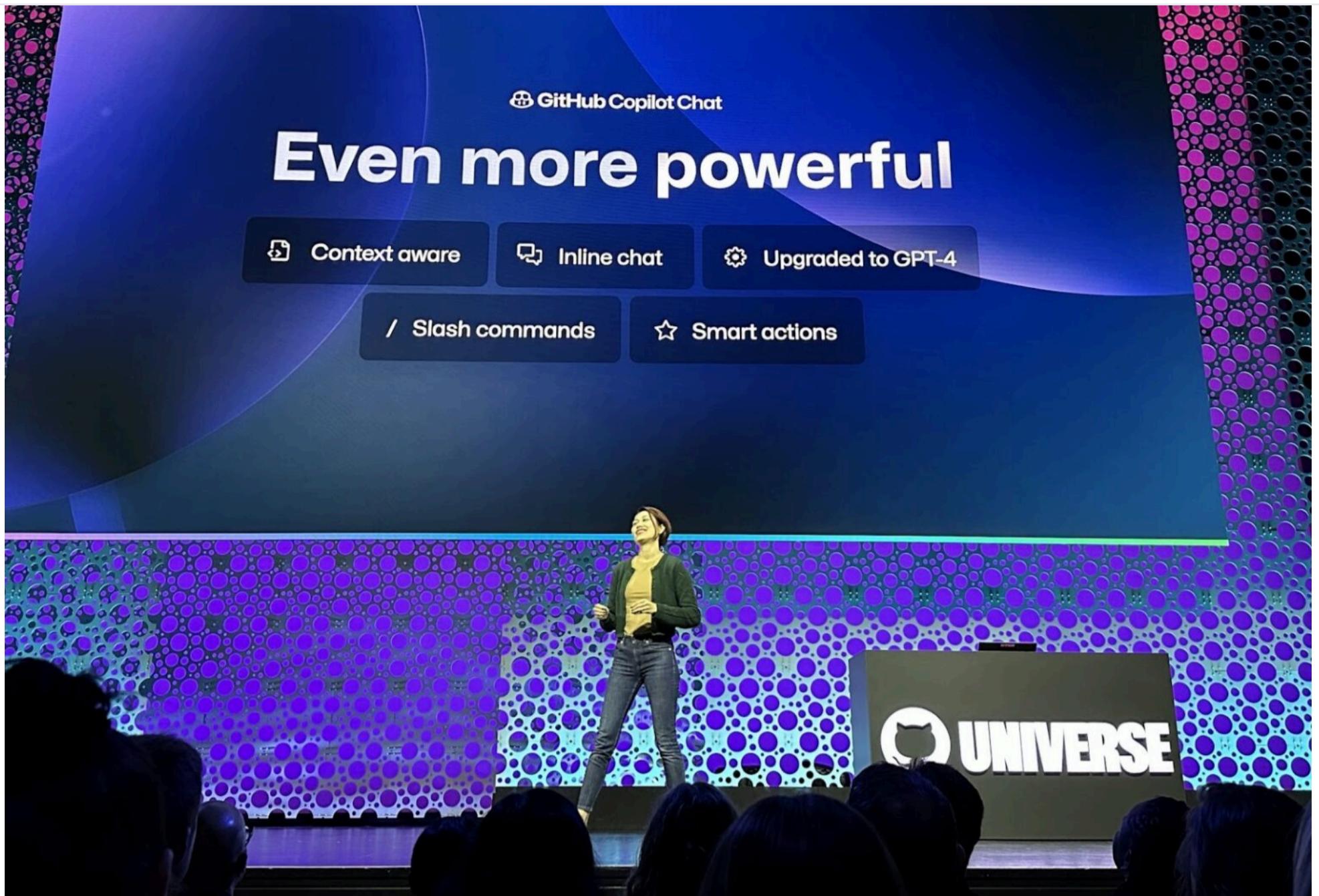
GitHub made several announcements at their conference, below are the key ones along with their descriptions of them:

Copilot Chat – based on GPT-4

For those of us already familiar with GitHub Copilot, you'd have been impressed with the automatic code suggestions available to you in both Visual Studio Code and full Visual Studio – based on public API documentation and the context of the code already written within your project. Whether that is automatic single lines of code which follow a predictable pattern to what's already written above, to full snippets based on a descriptive comment prompt you'd written.

GitHub Copilot Chat takes this to the next level entirely. Not only is the existing functionality now based on GPT-4 bringing more accurate code suggestions, but the Copilot's help spans beyond the code editor. Within Visual Studio Code, and soon full Visual Studio and multiple JetBrains IDEs, there is a dedicated chat interface where you can fully interact with Copilot. Highlights of this include:

- Slash commands – the ability to execute functions, the most obviously impressive was the '/new' command where an entirely new boilerplate project can be created by entering this command followed by a natural language description of exactly what you want your project to consist of – for example: /new Create me a NextJS project with Jest and Cypress unit and integration tests – Copilot will respond with a folder preview of what the project will look like where you can even drill down to see the contents of the individual files that will get created.
- Asking questions of the codebase – you can now use the Chat interface to simply ask questions such as "tell me what this section of code is doing" or "explain what this project does".
- Integration into the GitHub web portal and mobile app – Copilot will also integrate into the web portal, allowing automated and rich Pull Request descriptions based on the files and commit details that have been made – including links to documentation, generally way beyond the level of detail a developer would normally provide in a PR.



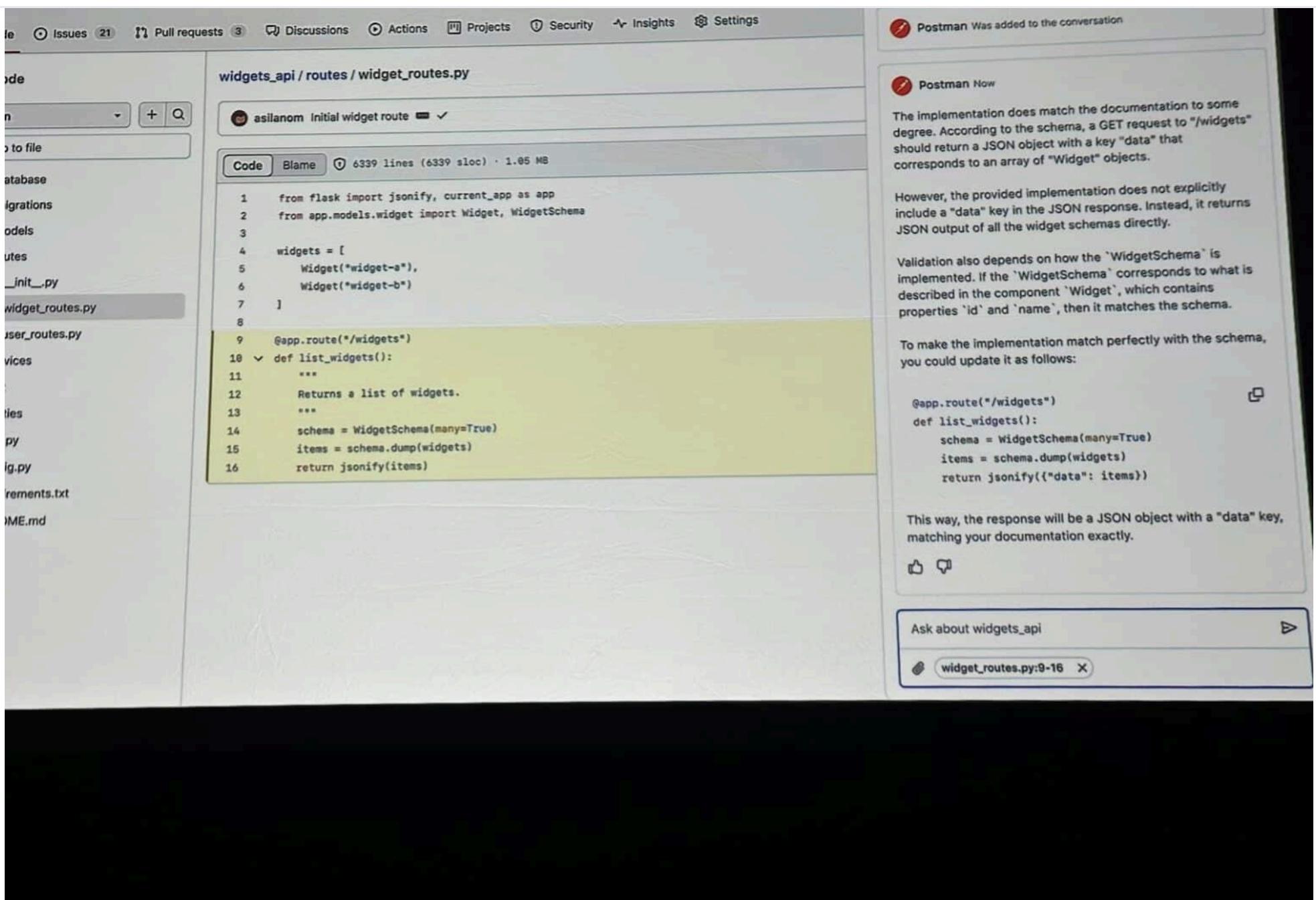
Copilot Enterprise

GitHub Copilot Enterprise is their Copilot offering for organisations using GitHub Enterprise. It builds on GitHub Copilot Chat, offering the same functionality but enhanced with interactions that are personalised to the codebases within the Enterprise. Copilot can now integrate with docsets and the code from private repos within the enterprise allowing private APIs that are not available for public indexing to be used as part of its code suggestions and output. For example, say you have a private repo in your enterprise that describes an internally used API to deploy to Azure Kubernetes Service, that you prefer your teams to use compared to publicly available methods. With GitHub enterprise, the docsets and codebase that describe this API will cause it to provide code snippets and explanations of it. Functionality like this not only will improve the flow state for existing engineers, it will also improve the onboarding experience for new engineers joining your team. GitHub also promise to support enterprises with unlimited liability protection to ensure that any code suggested by GitHub does not contravene copywrite or break licence adherence.



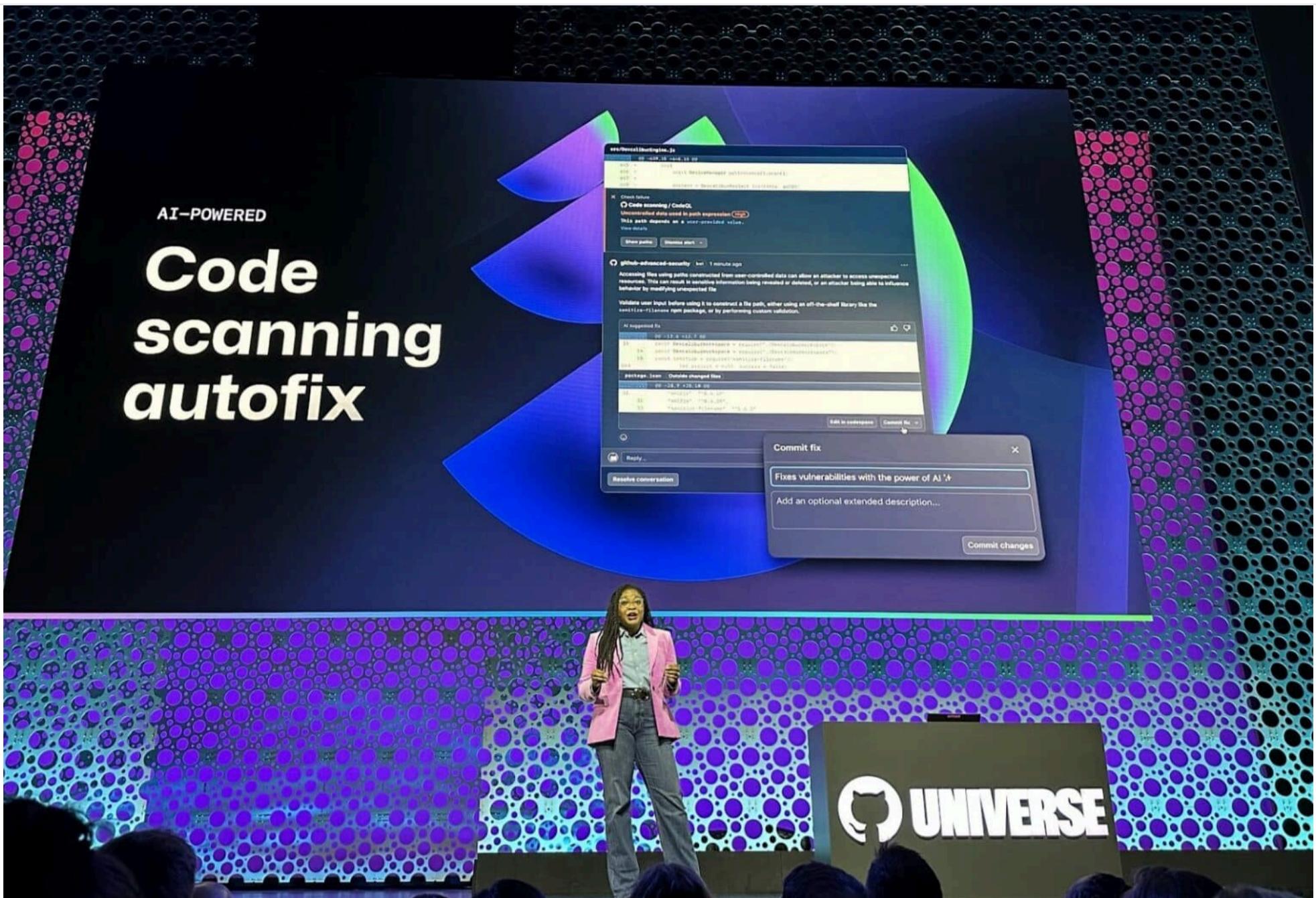
Copilot Partner Programme

The Copilot partner programme is great news for both developers who use GitHub Copilot and companies who want to integrate with it. Using the same interface that developers will be using everyday to chat to Copilot, GitHub are allowing partners to develop specific Copilot Apps that allow natural language questions. They provided a great demo where a section of code was highlighted, and then using the LaunchDarkly CoPilot app, the developer was able to ask if the code was enabled in production – with a rich response provided. Other apps detailed included Postman and Datadog – you can imagine the possibilities for this are endless – with the key being to remove distractions and keep developers in the flow state within their IDE. Partners can [apply to be part of the partner program here](#). Here at Talk Think Do we've already applied as we see strong potential of how this can benefit projects we are working on.



AI-powered code security

We've been using GitHub Advanced Security (GHAS) for a while within the engineering teams here at Talk Think Do, having setup and configured CodeQL, Secret Scanning and Dependabot across a number of repos for our clients. Shifting left on security as a DevOps best practice is a great way of reducing the chances of vulnerabilities reaching production and decreases the time taken to fix any vulnerabilities that do come up. Overall the GHAS suite will improve code quality across any team that have this configured. GitHub have further enhanced the code scanning aspects of this by providing autofix – a way of automatically suggesting through AI what the fix should be for detected vulnerabilities. Currently this is limited to just JavaScript and TypeScript, but the examples shown were sophisticated, with the fix demonstrated including a new library to escape input before it was then used as output on a page. The descriptions of why a certain area of code is vulnerable are made much clearer for a developer to understand when an autofix example of how to remediate it are shown – this will radically transform the concept of shifting left with security.



Natural Language *is* the Programming Language

So, with all this tooling both available now and improving in the future, what does it mean to be a software developer now and going forward?

Adaptability

One key aspect I've found which elevates a great developer from an average developer, is their ability and willingness to adapt to change. Most developers have seen quite a lot of change over their careers, but this has accelerated in recent years and in 2023 with generative AI the pace of change has seemingly exponentially increased. Adaptability has become the skillset required to be great, and in fact going forward it will become the mandatory skillset to simply exist within the role.

Accessibility

With the barrier to entry being lowered by tools such as Copilot, software development is now a much more accessible field of work. Developers who are adept at knowing what they want to accomplish, and skilled enough with prompt engineering, will be able to get a head start on achieving their aims. Whether that is creating a new project from scratch using latest best practices or writing code to invoke various APIs with ease – natural language is becoming the programming language.

Accountability

AI powered tools such as Copilot can suggest code snippets – these are no different than copy pasting code that's been found on stack overflow or another blog post. Regardless of where that code has come from, the developer needs to analyse it and decide if it is appropriate for their needs. Further than that, they need to understand it and take responsibility for it, as at some point they will have to support it. One can picture multiple scenarios where junior developers have happily taken code suggested to them, got their app to work, but not entirely understood why it works – this is dangerous.

Ingenuity

We know that reading through pages and pages of documentation is less necessary now that Copilot can automatically produce code snippets based on this same documentation tailored to our needs – so our approach has already changed in this regard. You can even use Copilot to simply help describe an existing repo to you, making onboarding new developers easier. But let's move on to one of the tenets of software development – [test driven development](#).

Previously a developer would write all the unit tests for some functionality that needed development, then gradually implement the functionality, and get those new and previously failing tests to pass. This was based on the premise of translating requirements typically given in user story form from stakeholders into test cases. Copilot allows the developer to generate unit tests based on the functionality that has been written – often creating more

Takeaway

Ultimately, this last year has been a step change in terms of the role a developer performs and the way they will perform that role. GitHub is at the forefront of modern software development, and this conference brought home to me what a fundamental advantage in terms of both experience and productivity that a developer working within that ecosystem enjoys, compared to those who don't.

Adopt GitHub with Talk Think Do

We hope this article has helped you understand how GitHub can help manage your code and improve productivity and collaboration for developers, as well as help make your systems more secure.

If you are interested in adopting GitHub for new projects or modernising your existing DevOps processes, you'll benefit from the support of a trusted GitHub partner, like Talk Think Do. Our end-to-end support ensures that your unique needs are met from start to finish.

If you'd like to learn more about how Talk Think Do could help you maximise the potential of GitHub, [get in touch](#) with one of our experts today.



FAQs

What is GitHub Universe?

GitHub holds their annual GitHub Universe conference for customers and employees to meet up from their dispersed locations to discuss valuable product announcements and reports.

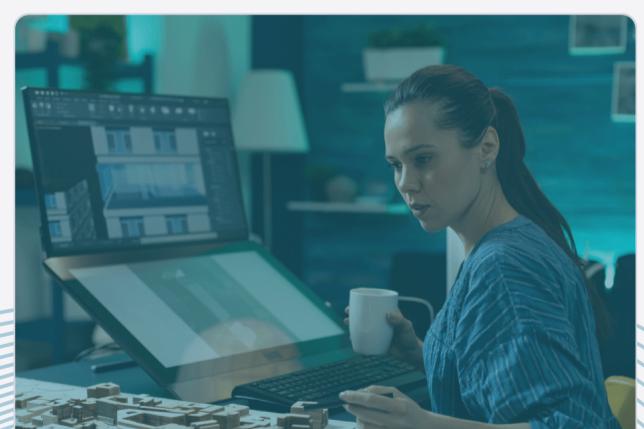
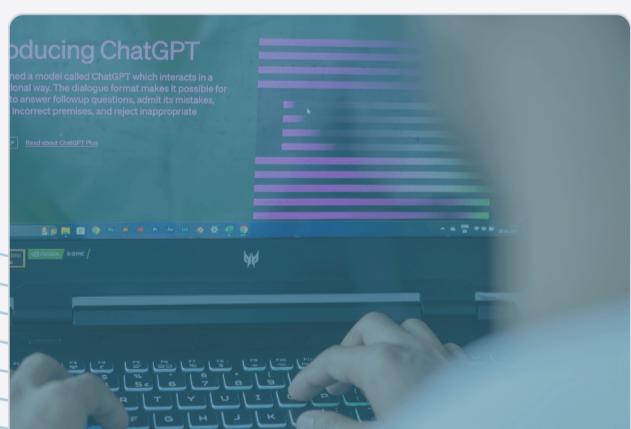
How is GitHub reinforcing collaboration?

Supporting open-source projects with dispersed teams, GitHub's collaboration features allow projects to be effectively pushed forward, and have been widely adopted across the industry.

How does GitHub increase application security?

GitHub's Advanced Security is a developer-centric suite of security tools that directly assess the security risk of code as it is being created.

See our Latest Insights



[End of Human-Lead Thinking](#)

[Software & Product Delivery](#)

[Common Risks and How to](#)

Legacy systems are costing your business growth.

Get your free guide to adopting cloud software to drive business growth.

[Download Your eBook](#)

Talk | Think | Do

Get the [latest news and insights](#)
sent straight to your inbox

Enter Your Work Email *

[Subscribe Today](#)

Resources

[Blog](#)

[Legacy Systems eBook](#)

Company

[Careers](#)

[About Us](#)

[Contact Us](#)

[01202 006729](#)

hello@talkthinkdo.com

[Visit Us](#)

Crown
Commercial
Service
Supplier



 Microsoft
Solutions Partner

Talk Think Do Limited

[Privacy Policy](#)

[Cookie Policy](#)

[Terms and Conditions](#)

[Information Security Policy](#)

All rights reserved © 2024