



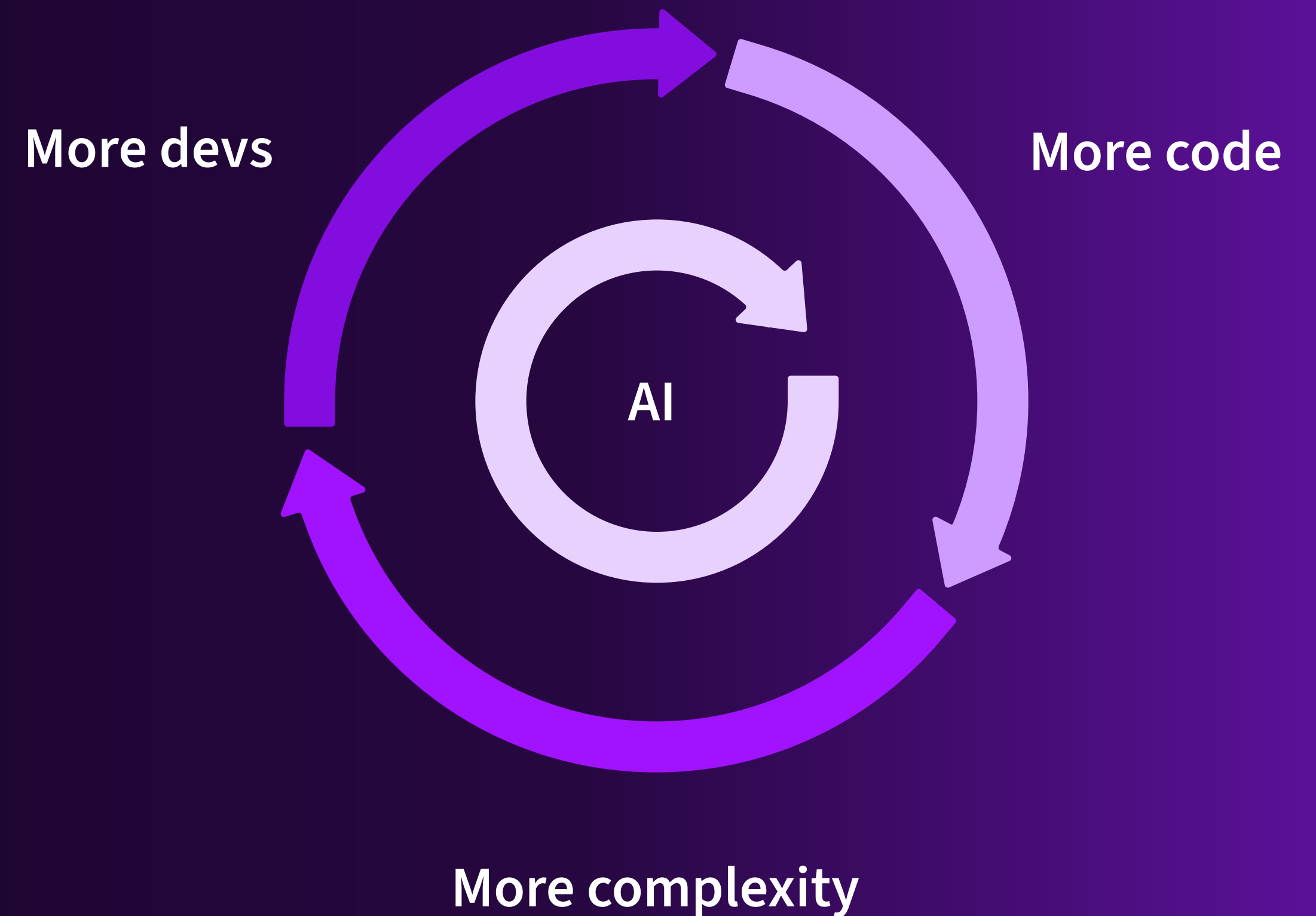
# Big Code in the AI era



# AI is the best and worst thing to happen to enterprise dev teams.

Codebases today are made up of millions (sometimes billions) of lines of code written by thousands of devs over the span of two to three decades. The complexity that comes from these massive codebases is known as “Big Code.”

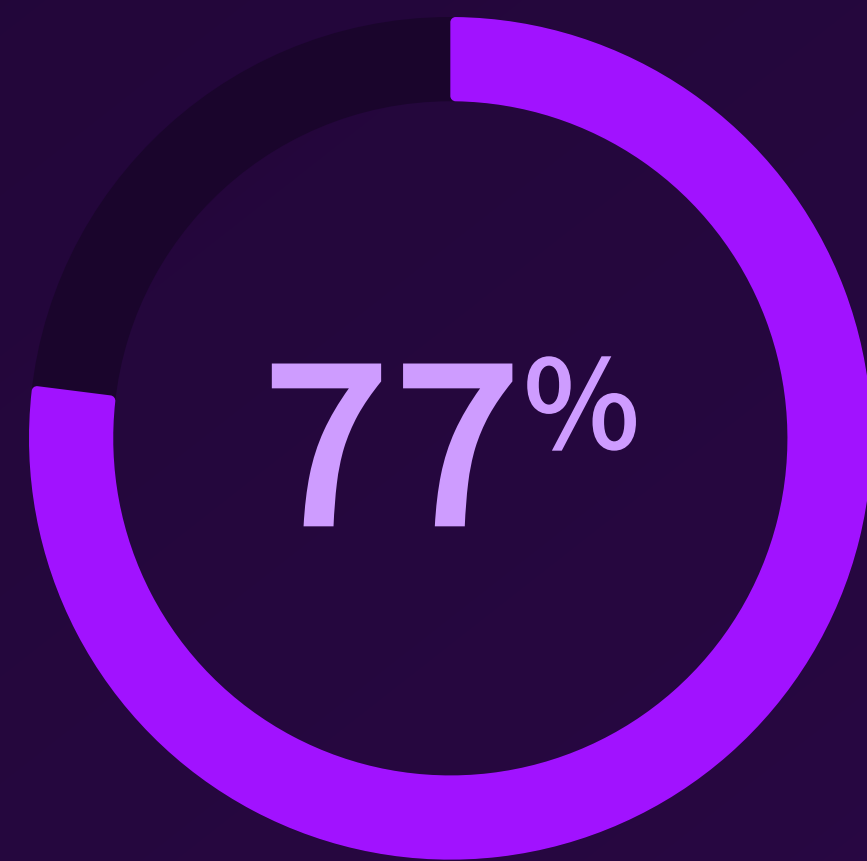
Big Code has been a problem for years and is a rapidly growing crisis for developers, engineering leaders, and companies. It not only threatens innovation, but also the stability and security of the digital products we use every day.



Developers have struggled with Big Code-related issues like tech debt, code maintenance, collaboration, and scalability for years. The overnight boom of AI in software development is boosting productivity for engineering teams while also making Big Code pains worse:

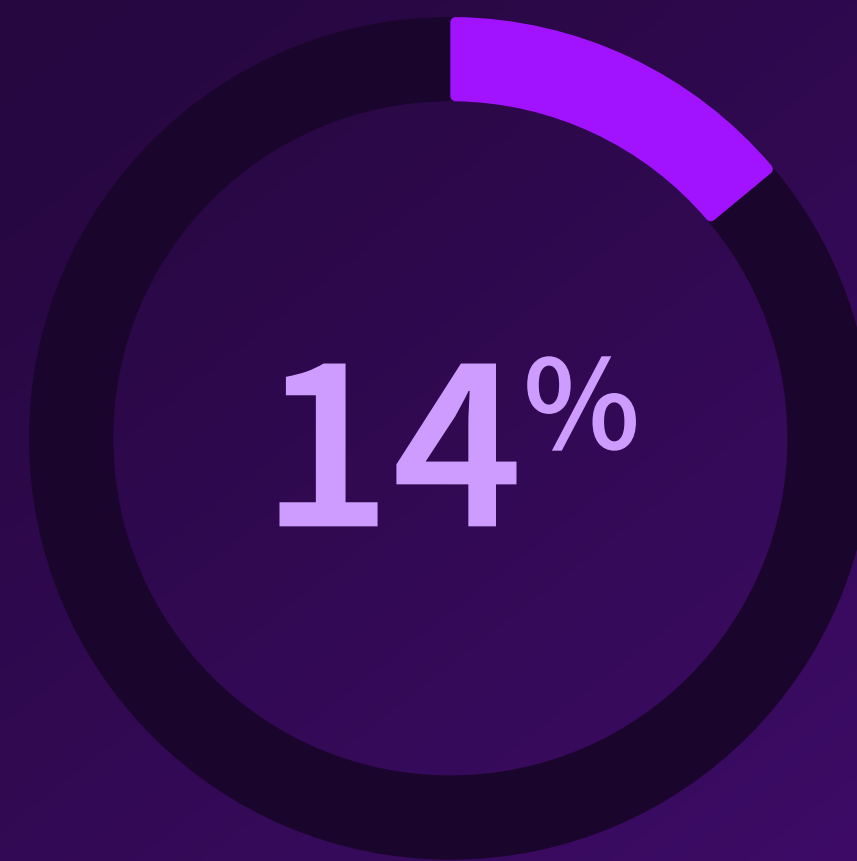
## Big Code's getting bigger.

77% of devs say their codebase grew 5x over the past three years.



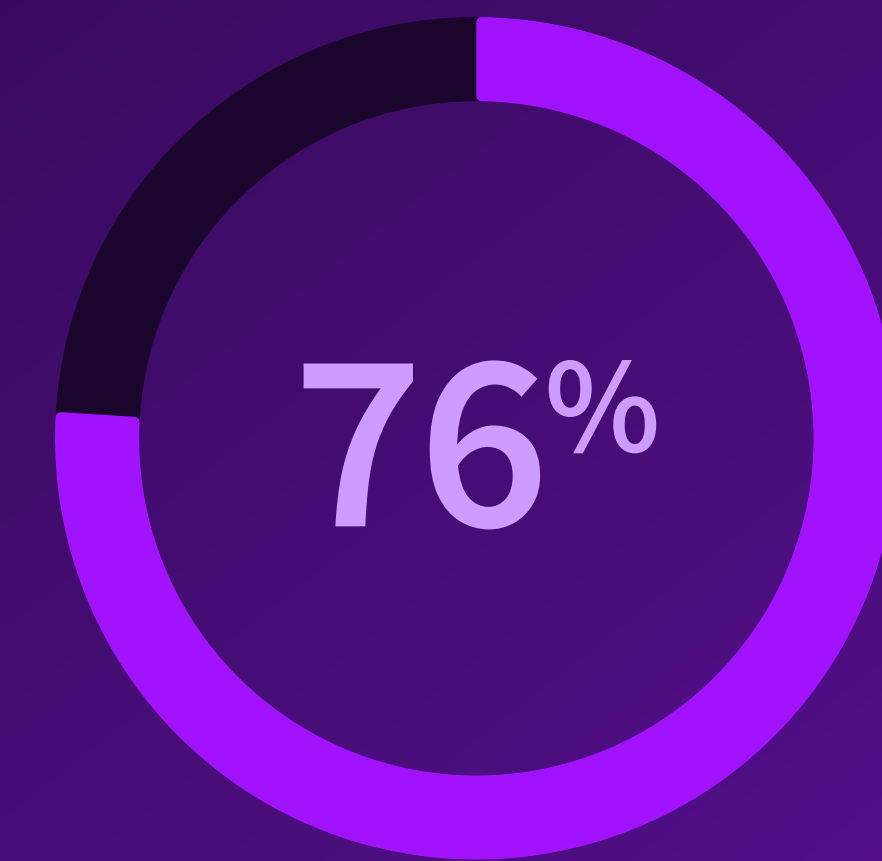
## Devs are underutilized.

Devs spend only 14% of their time in the codebase writing new code for core products. The rest is spent trying to search, understand, and fix code.



## AI is taking software by storm.

95% of devs report using AI tools to write code, but 76% of devs already struggle with code created by someone else that is confusing or difficult to understand.



The scariest part?

Only 65% of devs say their company had a plan to address Big Code prior to the rise of AI.

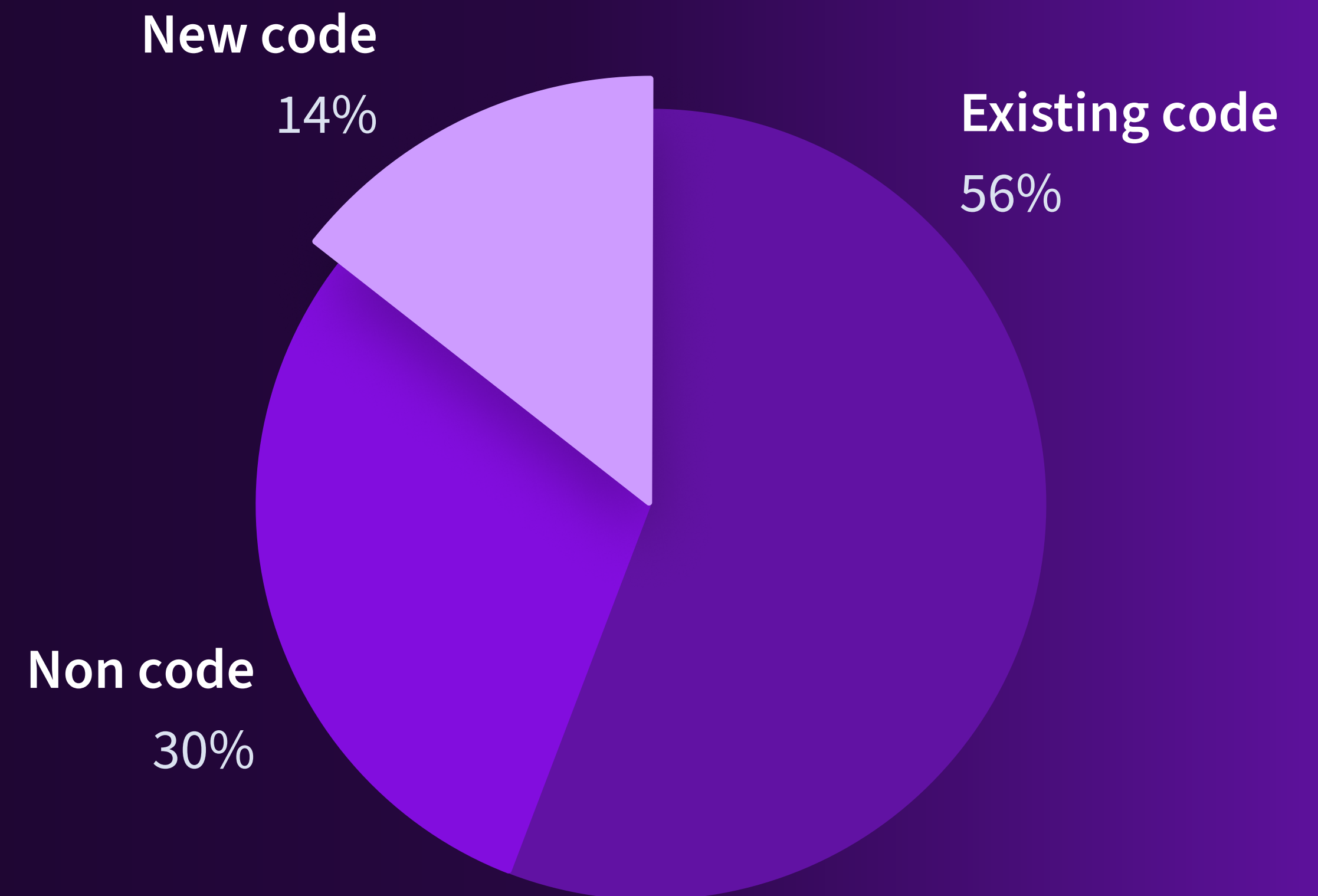


# We can't ignore the Big Code problem anymore.

Devs spend almost as much time reading, interpreting, and fixing code as they do writing it—respondents told us once you account for non-coding work, they spend only 14% of their time in the codebase writing new code for core products. That means companies are wasting precious time that could be spent innovating and growing their business on maintaining the status quo and trying to not get lost in their existing codebase.

And with 87% of devs and engineering leaders already struggling to keep their knowledge up to date on the new code, we know the problem's becoming a lot more complicated with the onslaught of new code written by AI.

## Dev time spent



**74% of developers and engineering leaders say they already rely too heavily on other people to understand their code base and product architecture in order to develop new code.**

Now imagine a world where AI-generated code exists in most companies' codebases—new code is being written faster, but questions about ownership and maintenance will become way more complicated and Big Code pains become worse.

AI and AI-powered devtools have the potential to significantly help with developer satisfaction, efficiency, and productivity, but companies have a lot of work ahead to plan for AI and make sure it's used effectively and securely.



**Not to mention, developer satisfaction is taking a big hit because of Big Code:**

**73%**

Are blocked more frequently due to the size of their codebase.

**85%**

Struggle to maintain efficiency in their day-to-day.

**82%**

Wish they could spend less time looking for information or old code and more time actually coding.

# Big Code in the AI era.

65% of devs and eng leaders said they didn't have a plan for Big Code prior to the rise of AI. And with AI already broadly adopted across companies—**95% of devs say they use GitHub Copilot, ChatGPT, Cody, or similar AI tools to write code**—the lack of a plan will have real consequences.

**76% of developers are excited about the rise of dev tools powered by AI, but there are still big concerns around AI and its impact on Big Code:**



**61% are concerned about AI's impact on tech debt.**



**67% expressed concern about code sprawl due to the growth of AI.**



**76% are worried about the amount of new code will be created that will then need to be managed.**



**Innovation velocity is more critical than ever, but 72% say that Big Code presents a real risk to their company's ability to innovate and compete.**

The rise of AI-powered dev tools brings in a new set of opportunities and challenges for today's engineering teams, and while GitHub Copilot, ChatGPT, Cody, and other AI coding assistants save devs time, the need to address growing tech debt, code sprawl, and new code creation in the rise of AI has left many teams feeling a mix of excitement and overwhelm.



**The Big Code struggle is real:**

**95%**

Need help getting up to speed and stay on top of the codebase significantly faster.

**91%**

Would save a significant amount of time if their codebase was fully searchable across all sources and repos.

**91%**

Want to be able to ID and resolve code issues more efficiently.

**88%**

Want a tool that allows them to have a greater output with fewer people and resources.



Are we headed into a future where companies declare bankruptcy on Big Code in hopes of getting back to innovation, or are there strategies and solutions dev teams can adopt to get ahead of Big Code pains in the rise of AI?

## Our advice

1

**Form an AI advisory committee** with leaders from your engineering, legal, and security teams and task them with ID-ing the risks and opportunities AI presents for your company, and planning ahead to address them.

2

**Streamline your use of AI** across teams and consider “Bring Your Own LLM”. Security standards, retention policies, and documentation for AI tools vary, so pick one tool you’re confident in and standardize on it.

3

**Put AI to work on Big Code.** Yes, AI can make Big Code pains worse, but many devtool companies are embedding AI into their workflows to help devs get back to building cool stuff. Use AI to gain clearer insight into code ownership, find vulnerabilities faster by using AI to search across your entire codebase, and rely on AI coding assistants like you would a senior dev: to onboard team members and provide valuable context that’s easily lost in growing codebases and changing teams.



# Methodology

To create our Big Code Report, we rely on survey data gathered anonymously by Ground Control Research. The respondents are developers and engineering leaders that vary in experience levels across several engineering disciplines. The respondents work at companies that span all major industries with at least 1,000 employees.

