SNYK REPORT

AI Code, Security, and Trust:

Organizations Must Change Their Approach

56.4% say insecure AI suggestions are common — but few have changed processes to improve AI security.

AI coding assistants have achieved widespread adoption among developers across all sectors. However, many developers place far too much trust in the security of code suggestions from generative AI, despite clear evidence that these systems consistently make insecure suggestions. Unfortunately, security behaviors are not keeping up with AI code adoption.

Technology organizations need to protect themselves against AI code completion risks by automating more security processes and inserting the right guardrails to protect not only against bad AI code but also against the unproven perception that AI-generated code is always superior to novel human code.

Executive Summary

In a short period of time, AI code completion tools have gained significant market penetration. In our survey of 537 software engineering and security team members and leaders, 96% of teams use AI coding tools and over half of those teams use the tools most or all of the time. It is safe to say that AI coding tools are now part of the software supply chain at most organizations. The use of AI tools has likely accelerated the pace of software code production and sped up new code deployment. On top of that, AI coding tools are polished and convincing. Unfortunately, this polish and ease-of-use has generated misplaced confidence in AI coding assistants and have created a herd mentality that AI coding is safe. In reality, AI coding tools continue to consistently generate insecure code. Among respondents, 91.6% said that AI coding tools generated insecure code suggestions at least some of the time.

The risks of AI coding tools are magnified by the resulting accelerated pace of code development. This is particularly true in open source code, where keeping up with the latest security status of open source libraries and packages is challenging due to new insecurities and vulnerabilities landing on a seemingly daily basis. Despite these risks and challenges, our survey found that technology teams are not putting the proper measures and guardrails in place to best secure their code in this new AI coding age. Less than 10% of survey respondents have automated the majority of their security checks and scanning. 80% of respondents said that developers in their organizations bypass AI security policies. Respondents are also not taking proper measures to ensure that their open source libraries are secure, with only 25% using an automated scanning tool to check the security of open source components included in AI coding suggestions. The consequences of placing too much trust in AI coding tools are real. Among respondents who said that AI coding tools reduced productivity, the two primary reasons for this negative result were poor code quality and security problems introduced by AI.

The irony is that, while adoption and trust are high, developers are clearly aware of the risks of AI and told us so in the survey. These findings underscore why it's so important for development and security teams to adopt a responsible approach to AI. On the process and technology side, this includes stepping up security measures like automated security scans, increased code audits, and programmatic policies that prevent rapid-fire and unquestioning acceptance of AI-generated code without proper review. On the education side, this includes educating technology organizations about the inherent risks of outsourcing security to AI and why humans may be prone to the risky behaviors outlined above.

PART ONE

Risks of Outsourcing Code Security to AI

Survey responses indicate that AI code completion continues to inject security risks into the development process. What's more, developers are actively bypassing AI usage policies for coding. All of this is happening without putting in place proper guardrails, such as automated code scanning. Open source code is a particular risk as AI coding tools speed up velocity and suggest open source modules, but teams are not programmatically validating that suggested open source components are secure.

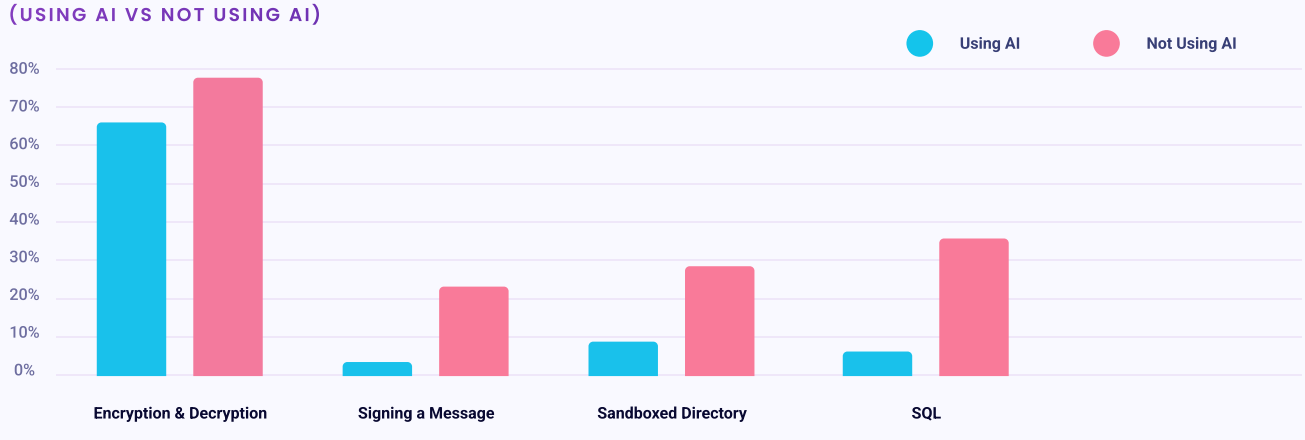
AI Coding Tools Generate Insecure Code. Developers Ignore This Fact.

In December 2022, StackOverflow banned all AI-generated submissions from ChatGPT to its coding Q&A site, stating, "The average rate of getting correct answers from ChatGPT is too low." Their assertion echoed findings from multiple respected academic studies from New York University and Stanford University finding that AI coding completion tools consistently made insecure suggestions and that coders relying heavily on the tools wrote more insecure code.

In the Stanford study, which used an AI coding model tuned specifically for computer code, for coders writing an encryption function, the AI tool consistently recommended open source libraries that explicitly stated in their own documentation they were insecure and not suitable for high security use cases. Worse, in the Stanford study, developers believed AI suggestions made their code more secure even if it actually wasn't.

In our own internal research, we have found that AI coding tools frequently make insecure code suggestions. Despite these known findings, many developers believe code suggestions from AI coding tools are secure. In our survey, 75.8% of respondents said that AI code is more secure than human code. This massive discrepancy is indicative of major problems with the way organizations are securing their development process against AI coding tools and educating their technology teams on the known risks of AI for code generation. In this report, we surveyed 537 technology and IT workers and managers to better understand this dynamic.

PERCENTAGE OF CODERS SUBMITTING SECURE ANSWERS TO CODING QUESTIONS (USING AI VS NOT USING AI)



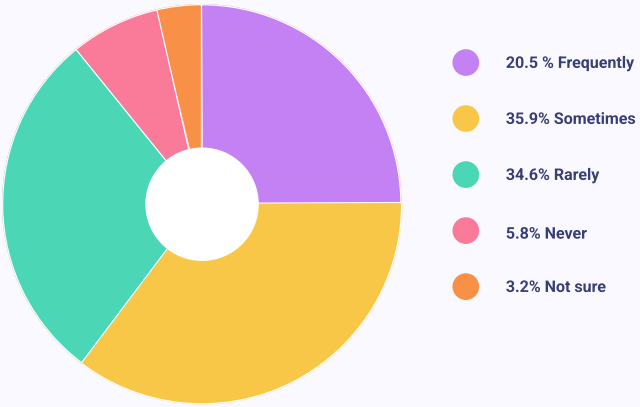
56.4% Commonly Encounter Security Issues in AI Code Suggestions

Despite voicing strong confidence in AI code completion tools and demonstrating strong adoption of the tools, respondents acknowledge that AI does introduce security issues. 56.4% admit that AI introduces coding issues sometimes or frequently.

This indicates that AI tools require verification and auditing for all suggestions due to the high rate of potentially flawed code produced. Despite the fact that respondents say that security issues with code suggestions are common, 75.4% of respondents rated the security of AI code fix suggestions as good or excellent. This mirrors the Stanford findings, where coders strongly overestimated the security of code suggested by AI that they had accepted in their work. This is indicative of a deep cognitive bias that is extremely dangerous for application security.

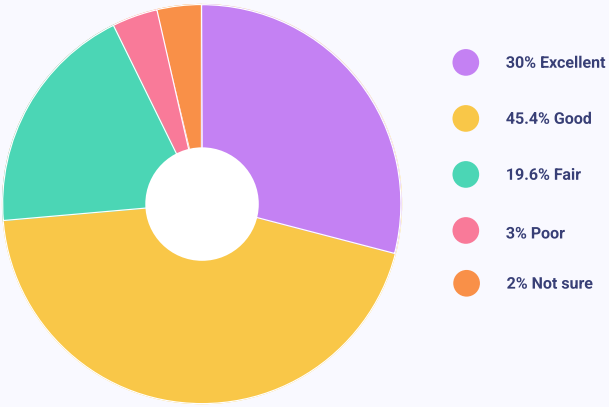
Respondents commonly found security issues with AI suggestions but...

HOW FREQUENTLY DO YOU ENCOUNTER ISSUES DUE TO CODE SUGGESTED BY AN AI TOOL?



they *also* strongly believed that AI suggestions were secure.

HOW WOULD YOU RATE THE SECURITY OF AI CODE FIX SUGGESTIONS?

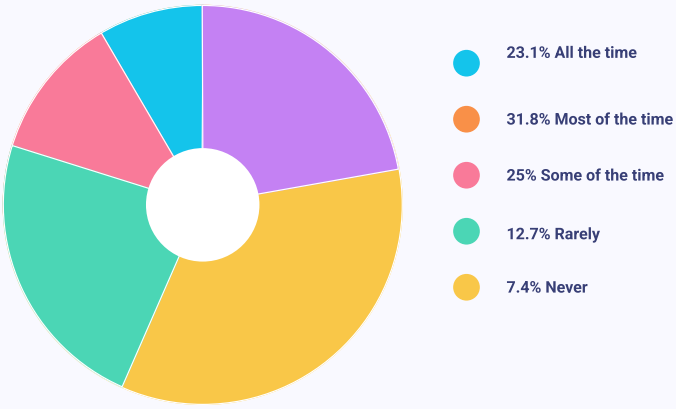


79.9% Bypass Security Policies to Use AI, but Only 10% Scan Most Code

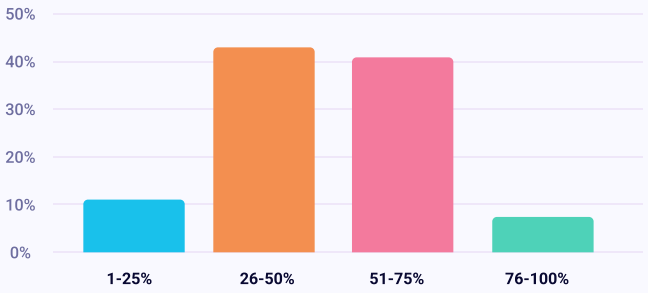
While most organizations of respondents had policies allowing at least some usage of AI tools, the overwhelming majority reported that developers bypass those policies. In other words, the trust in AI to deliver code and suggestions is greater than the trust placed in company policy over AI.

This creates tremendous risk because, even as companies are quickly adopting AI, they are not automating security processes to protect their code. Only 9.7% of respondents said their team was automating 75% or more of security scans. This lack of automation leaves a significant security gap. The gap is compounded further given that developers using AI tooling are likely producing code more quickly. 71.7% of respondents said that AI code suggestion was making them and their teams somewhat or much more productive. This lack of policy compliance plus increased code velocity makes automated security scanning even more important than ever before.

HOW OFTEN DO DEVELOPERS IN YOUR ORGANIZATION BYPASS SECURITY POLICIES IN ORDER TO USE AI CODE COMPLETION TOOLS?



WHAT PERCENTAGE OF YOUR SECURITY SCANNING IS AUTOMATED?



> "By using Snyk Code's AI static analysis and its latest innovation, DeepCode AI Fix, our development and security teams can now ensure we're both shipping software faster as well as more securely."
>
> **Steve Pugh, CISO at ICE/NYSE**

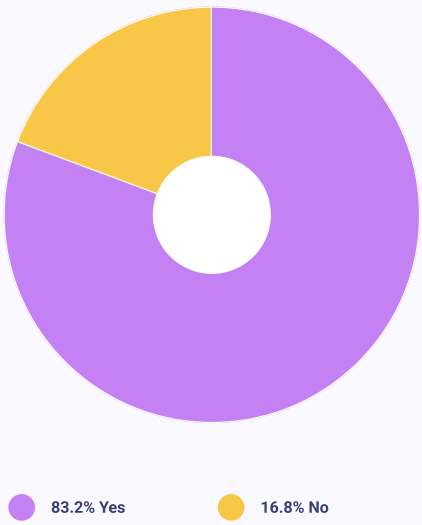
AI Further Exposes Open Source Supply Chain Security

In the survey, 73.2% of respondents said they contributed code to open source projects. So the average survey respondent is knowledgeable about open source. Despite this understanding, few use more advanced and reliable security practices to validate that code suggestions from AI coding tools are secure. Only 24.8% used software composition analysis (SCA) to verify the security of code suggestions from AI tools. Increased velocity would likely increase the speed at which unsafe open source components are accepted into code.

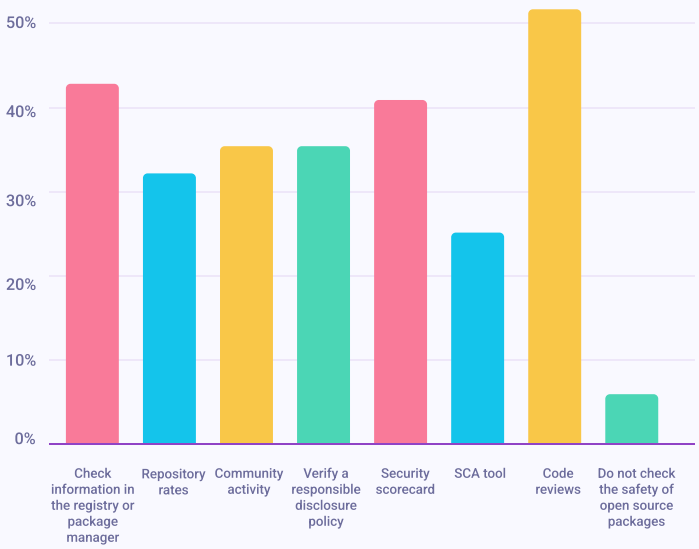
Because AI coding systems use reinforcement learning algorithms to improve and tune results, when users accept insecure open source components embedded in suggestions, the AI systems are more likely to label those components as secure even if this is not the case. This risks the creation of a feedback loop where developers accept insecure open source suggestions from AI tools and then those suggestions are not scanned, poisoning not only their organization's application code base but the recommendation systems for the AI systems themselves.

The potential for this dynamic was reinforced by the Stanford research which found that AI coding tools suggested insecure libraries that lacked the context of the application requirements. Then developers accepted the suggestions, trusting the AI tools rather than reading the documentation for the suggested components. The general, battle-tested pattern of code auditing and research appears to be breaking down in the AI coding process.

DO YOU USE AI CODE COMPLETION TOOLS FOR WORK ON OPEN SOURCE PROJECTS?



HOW DO YOU VERIFY THE SECURITY OF OPEN SOURCE PACKAGES AND LIBRARIES INCLUDED IN AI-GENERATED CODE SUGGESTIONS?



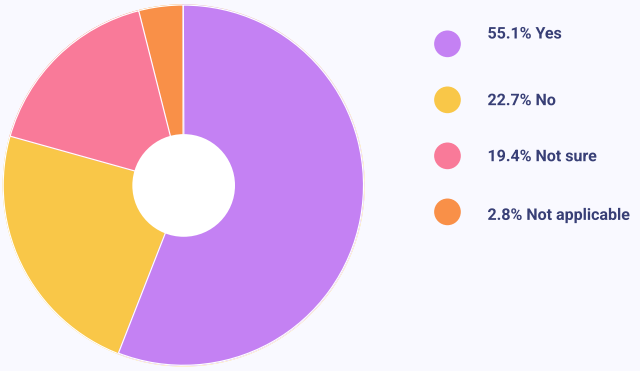
AI Considered Part of Software Supply Chain, But Few Change Practices

55.1% of respondents said that their organizations now consider AI code completion to be part of their software supply chain. This view has not resulted in correspondingly significant changes to application security processes driven by AI. While the majority of respondents said their team had made at least one change in software security practices as a result of AI code completion tools, the overall percentages on this multi-selection were on the low side.

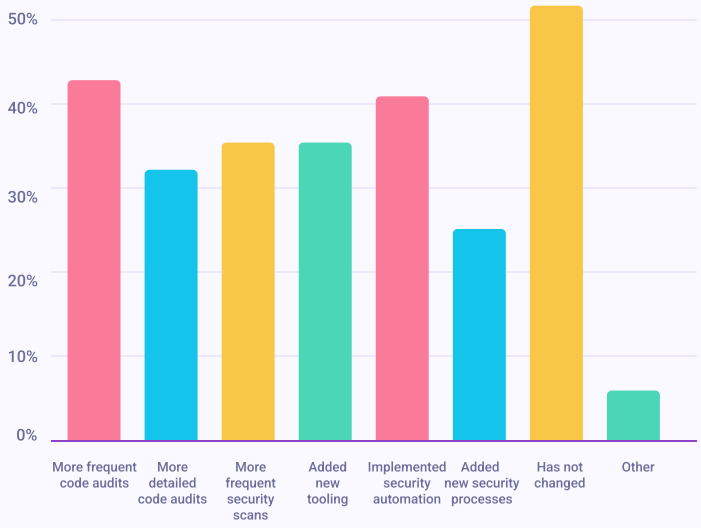
This indicates that the average organization made approximately one change. The highest percentage change was increasing security scans of 18.4% of respondents. It is possible that respondents are conflating code audits with security scans or that the audits include (or claim to include) security scans. Regardless, the relative impact of AI coding tools on security practices appears to be rather small.

This lack of change could also be attributed to the false perception that AI code suggestions are more secure than human code. Ultimately, significant changes in the way we work usually necessitate a review and corresponding change in risk management, to address new/additional risks brought about by the novel way of working. Such adjustments appear to be missing — which is reason for concern.

DOES YOUR ORGANIZATION CONSIDER AI CODE COMPLETION TO BE PART OF ITS SOFTWARE SUPPLY CHAIN?



HOW HAS YOUR ORGANIZATION CHANGED YOUR SOFTWARE SECURITY PRACTICES AS A RESULT OF AI CODE COMPLETION?



PART TWO

Developers Recognize Risks of AI Blindness, Reliance

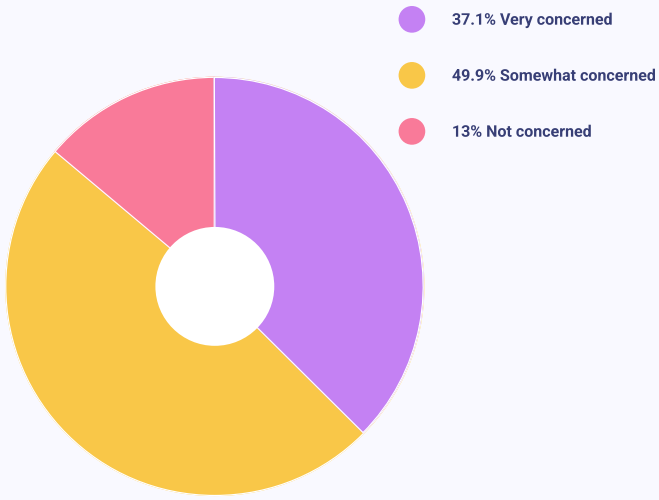
Even though developers perceive AI-written code to be secure, they overwhelmingly worry that AI code completion tools will create greater insecurity and that they will become over-reliant on the tools. In organizations that restrict AI usage, problems with code quality and security are the primary reasons for restrictions. Respondents acknowledge that a significant percentage of AppSec teams are struggling to keep pace with higher code velocity. All of this points towards a need to prioritize process and technology utilization changes – more automated security scanning – with continued education of development teams, so that they can be more aware of the real risks of AI code suggestions.

87% Are Concerned About AI Security, Indicating Cognitive Dissonance

The overwhelming majority of respondents expressed concerns about security implications of using AI code completion tools. This appears to contrast with the strong confidence in the ability of AI coding tools to generate secure code and to make code suggestions to improve security.

That cognitive dissonance is potentially a result of herd mentality, where developers believe that because everyone else is using AI coding tools, they must be trustworthy and that drives their actions. But at a more contemplative level, they understand the risks and recognize that AI may inject more insecure code than they realize or can easily see without more comprehensive security measures.

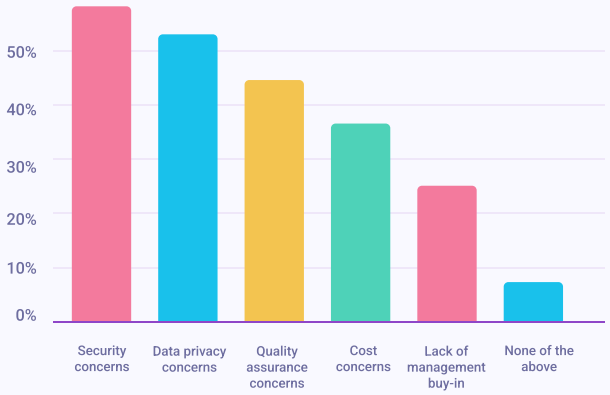
HOW CONCERNED ARE YOU ABOUT THE BROADER SECURITY IMPLICATIONS OF USING AI CODE COMPLETION TOOLS?



Security, Data Privacy Concerns Are Main Reasons for AI Code Restrictions

For the small subset of companies that restrict AI coding tools in part or in whole, the most common concern behind the restrictions was code security (57%) followed by data privacy (53.8%) and code quality (46.4%). All of the major concerns for restricting AI were related to security, reflecting leadership concerns about potential negative or unmitigated impacts of AI code completion.

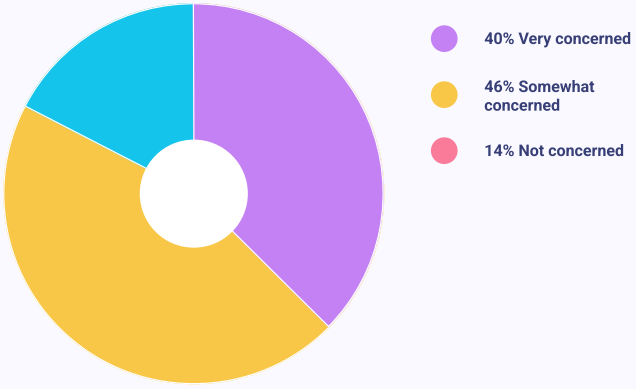
IF YOUR ORGANIZATION RESTRICTS THE USE OF AI CODING TOOLS, WHAT ARE THE REASONS FOR THE RESTRICTIONS?



Developers Concerned About AI Overreliance

A common concern is that developers using AI will become overly reliant on the coding tools and lose their ability to write code on their own or to perform key coding tasks that they commonly use AI for. In some research, knowledge workers that overly rely on strong AI become less likely to recognize good solutions, which may be atypical or out of pattern. Respondents shared this concern, with 46% saying they were somewhat concerned and 40% saying they were very concerned. In other words, they appear to be aware of the risks of outsourcing too much of their craft to AI.

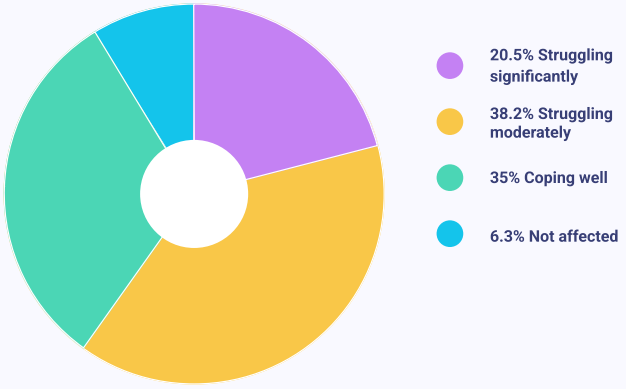
HOW CONCERNED ARE YOU THAT DEVELOPERS ARE RELYING TOO MUCH ON AI CODE COMPLETION TOOLS?



58.7% of AppSec Teams Are Struggling to Keep Up

Since AI coding tools have improved productivity and likely have increased the velocity of code production, if not the number of lines of code produced, we asked whether this was putting more pressure on AppSec teams. Respondents said that over half of all AppSec teams are struggling to some degree, with one-fifth struggling significantly to keep up with the new pace of AI-driven code completion. This is to be somewhat expected, if the productivity boost from AI code completion tools is meaningful. It also underscores the challenges created by adding more pressure to a process that even prior to AI often struggled to keep up with the pace of software development.

IS YOUR APPSEC OR SECURITY TEAM STRUGGLING TO ADAPT TO THE SPEED OF DEVELOPMENT DUE TO AI CODE COMPLETION?



CONCLUSION

To Fix the AI Infallibility Bias, Educate and Automate Security

There is an obvious contradiction between developer perception that AI coding suggestions are secure and overwhelming research that this is often not the case. The tension is underscored by seemingly contradictory responses found in this survey; most respondents (including security practitioners) believe AI code suggestions are secure while also simultaneously admitting that insecure AI code suggestions are common.

This is a perception and education problem, caused by groupthink, driven by the principle of social proof and humans' inherent trust in seemingly authoritative systems. Because the unfounded belief that AI coding tools are highly accurate and less fallible than humans is circulating, it has become accepted as fact by many. The antidote to this dangerous false perception is for organizations to double down on educating their teams about the technology they adopt, while securing their AI-generated code with industry-approved security tools that have an established history in security.

About this report

The survey contained 30 questions covering how organizations perceive and use AI code completion tools and generative coding. The survey polled 537 respondents working in technology roles. Of the panel, 45.3% were from the United States, 30.9% were from the United Kingdom, and 23.6% were from Canada. We asked respondents to self-identify their roles, choosing all titles that applied. The highest percentage selected were developer management (42.1%), developer (37.6%), IT management (30.9%), and security management (30.7%), indicating that the panel included a significant portion of respondents from management. Respondents were spread broadly across various sectors. SaaS/Technology represented the largest pool of respondents (21%) and the only sector representing greater than 20% of responses. Only two other sectors, business services (17.1%) and financial service/fintech (11.2%) represented more than 10% of respondents. The survey panel was predominantly smaller companies, with 48.6% of respondents working at companies of 500 employees or less and only 12.8% working at companies of greater than 5,000 employees. Respondents also used a wide variety of coding tools. The largest percentage cited ChatGPT (70.3%) with Amazon CodeWhisperer (47.4%), GitHub Copilot (43.7%) Microsoft Visual Studio IntelliCode (35.8%), and Tabnine (19.9%) ranked afterwards. This was a multi-select question and the high percentages across multiple responses indicates that respondents are likely using multiple AI coding tools in their jobs, potentially for different reasons or tasks.

snyk