

SNYK REPORT

Secure Adoption in the GenAI Era

In a survey of a broad selection of technology team members ranging from top management down to application developers, we found that most believed their organizations were ready for AI coding tools but also worried those tools presented a major security risk. Further, we found that organizations were failing to adopt basic preparedness steps, such as running a proof-of-concept prior to adoption and providing widespread training to developers. Lastly, we also found that respondents who are more likely to be directly exposed to AI coding tools and AI-generated code in their daily workflows were concerned about AI code quality and were comparatively more cautious about rapid AI adoption and security.



Introduction

- **Less Than 20% of Organizations Did AI Tool POCs**
- **AppSec 2X More to Rate Gen-AI Code Security as “Bad,” Devs Less Likely to Rate as “Excellent”**
- **C-Suite 2x to 5x Less Likely to See Security Risk from AI Coding Tools**
- **AppSec Practitioners 3X More Likely to Say AI Security Policies Insufficient**
- **Less Than 20% of Organizations Did AI Tool POCs**
- **CTOs and CISOs More Strongly Favor ASAP AI Coding Tool Adoption**
- **Perception Gap on Developers Using Unauthorized AI Code Tools**
- **C-Suite is More Confident That Their Organization is Ready for AI Coding Tools**
- **Security Fears Remain the Biggest AI Coding Tool Barriers**

Conclusion

Methodology

Introduction

Developers are using AI coding tools at work. According to a survey of over 500 developers from [Snyk's 2023 AI Code Security Report](#), 96% of coders use generative AI tools in their workflows. They use these systems for various tasks, from basic code completion and correction to writing unit tests to code QA and even doing security scans. There is no going back. Organizations that build software understand they must adopt these tools to keep up with competition and to attract and retain top talent. Bringing AI coding tools into the software development lifecycle introduces various security and operational challenges. The challenges are novel, and the pressure to adopt AI is intense. All these beg the question: How ready are technology leaders and their teams for the new era of AI coding tools? And how are they preparing for this significant shift in how software is written?

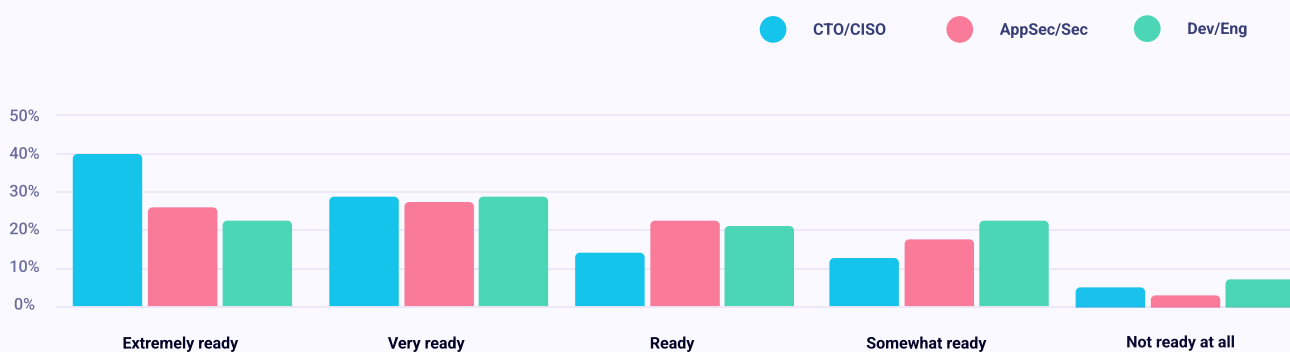
To gain insights into this topic, Snyk asked over 400 technologists a range of questions designed to gauge the AI readiness of their organizations and to measure their perceptions of AI coding tools. The questions specifically focused on security topics. The survey covered three groups: C-suite technology executives, application security teams, and developers/engineers. We found considerable differences in how these groups viewed the security of AI coding tools and code, the efficacy of AI code security policies, and organizations' general level of preparedness for AI coding. In this report, we outline the most notable findings from our research.

Section 1

Orgs are confident in their AI readiness, particularly leadership

Organizations generally feel confident that they are ready and prepared to adopt AI. In response to questions that either directly or indirectly question AI readiness, the majority of organizations are moving quickly to adopt AI to the point of short-circuiting standard use case analysis and product testing before deployment. For their part, C-suite respondents are both more sure their organizations are prepared to adopt AI and more sure that their AI tools are secure.

HOW READY IS YOUR ORGANIZATION FOR AI CODING TOOLS



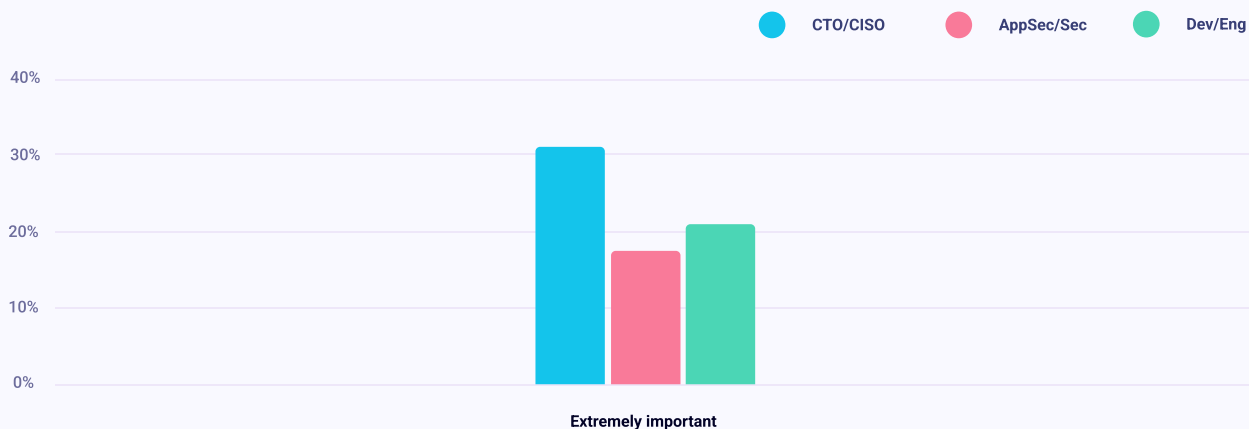


Across all three role types, a majority of respondents said that their organization was “extremely ready” or “very ready” for AI coding tool adoption. Less than 4% said their organizations were not ready. However, C-suite respondents are more confident than other groups of respondents that their organization is primed and ready for AI coding tool deployment and adoption. 40.3% of that group rated their organization as “extremely ready” compared to only 26% of AppSec team members and 22.4% of developers. There was not a significant difference between CISOs and CTOs, which seems counterintuitive given the security and risk focus of CISOs. This could be due to the intense pressure on technology leadership to quickly roll out AI coding tools and accelerate software development processes. Other groups’ reluctance likely reflects on-the-ground concerns about specific readiness issues around security, training, code quality, and other implementation-layer details.

CTOs and CISOs more strongly favor ASAP AI coding tool adoption

Among C-Suite respondents, 32.5% felt the rapid adoption of AI coding tools is “critical.” This means they are almost twice as likely to see adoption as urgent compared to AppSec respondents. Developers were more enthusiastic than AppSec but still lacked C-Suite enthusiasm levels. This intensity likely reflects strong demands from the Board of Directors and CEOs that CTOs move quickly to embrace AI.

HOW IMPORTANT IS IT FOR YOUR ORGANIZATION TO ADOPT AI CODING TOOLS ASAP?



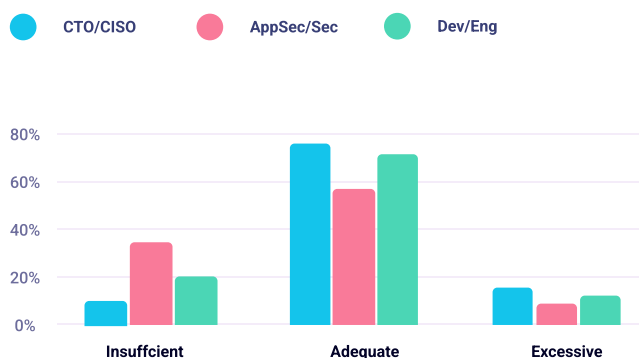
Most respondents believe AI coding tool security policies are good

Across all three response groups, the majority of respondents, including more than two-thirds of C-Suite respondents and developers, found their organization's AI coding tool policies to be adequate. Only a very small percentage found the policies to be overly restrictive. However, a far greater percentage of security practitioners found the policies to be insufficient, indicating that AppSec and security respondents still see risks in AI code security practices at their organizations.

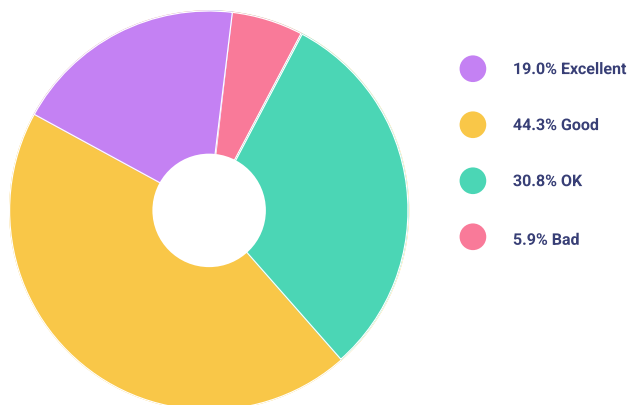
63.3% rate AI-generated code security highly

Roughly two-thirds of respondents rated the security of AI-generated code as either "excellent" or "good". Only 5.9% rated it as "bad". The sentiment towards AI-generated code is positive among the entire sample, echoing positive sentiments about policies governing AI coding tool use and adoption.

HOW WOULD YOU DESCRIBE YOUR ORGANIZATION'S SECURITY POLICIES FOR AI CODING TOOLS (E.G., COPILOT, CODEWHISPERED, GEMINI, TA...)



HOW WOULD YOU RATE THE SECURITY OF AI GENERATED CODE?



Section 2

Organizations fear AI coding security but aren't taking proper preparations

Despite strong positive responses about organizational readiness, security policies, AI code quality, and risk, respondents still cite security as the biggest barrier to AI coding tool adoption. In a seeming contradiction of this sentiment, they also are failing to take basic steps to minimize risk and prepare their organizations, such as running POCs and training developers on AI coding tools.

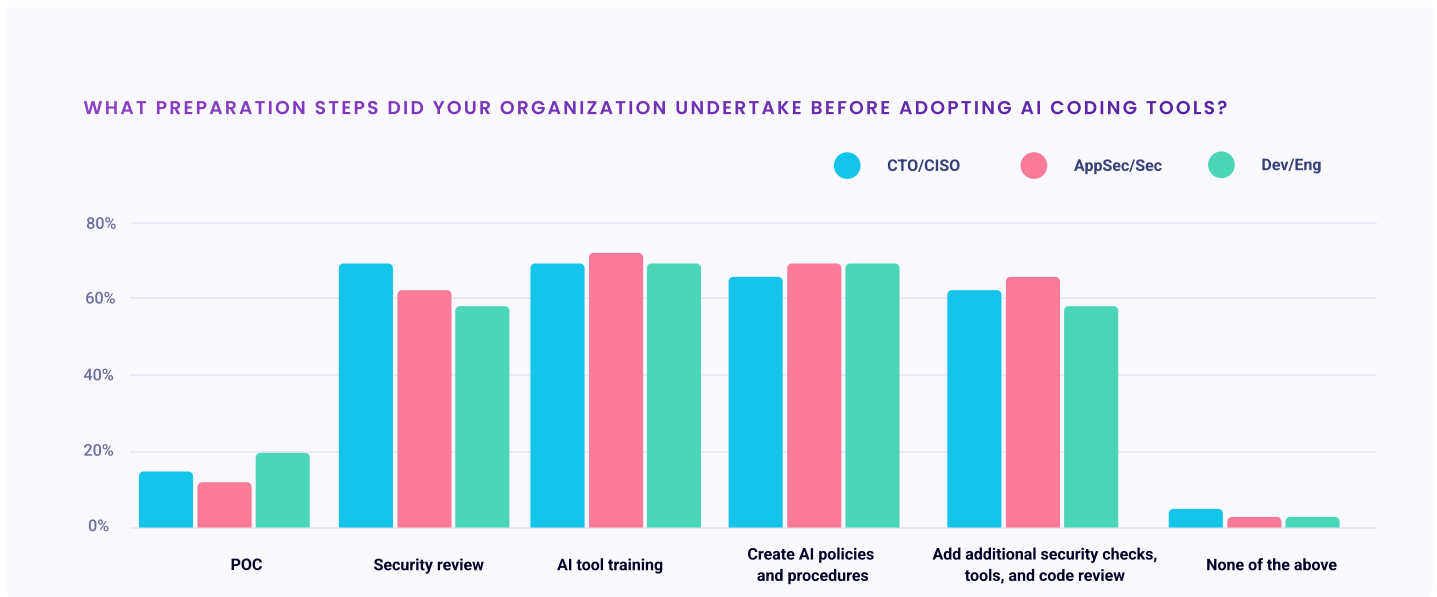
Security fears remain the biggest AI coding tool barriers

All three types of respondents agreed that security fears are the biggest concern in their organization about adopting AI coding tools, with roughly 58% across all types of respondents. Conversely, under half of respondents viewed lack of executive buy-in as a barrier. This finding matches the general viewpoints of AppSec practitioners and, to a lesser degree, developers. Still, it contradicts the generally positive view of AI coding tools and AI coding tool readiness expressed by the majority of respondents.

WHAT BARRIERS HAS YOUR ORGANIZATION FACED IN ADOPTING AI CODING TOOLS?



Less than 20% of organizations did AI tool POCs



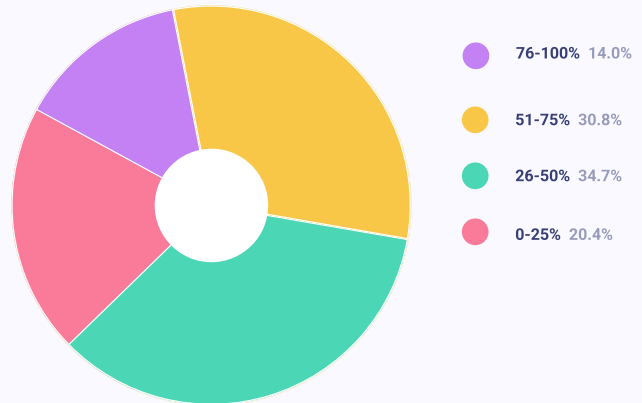
The standard process of introducing new technologies and tools into an organization is to do a feature and cost analysis and then run a “proof of concept” exercise with a small subsection of the team. This is how [Pinterest’s platform engineering team](#) addressed AI coding tool adoption. Our survey found that less than 20% of organizations undertook POCs as part of their preparation steps for adopting AI coding tools. Among all the preparation steps, POCs were by far the least utilized. Organizations were roughly one-third as likely to use a POC as other methods. Why AI coding tools managed to get over the wall into so many technology organizations without a POC is an excellent question. One can speculate that the broad availability of these tools in popular IDEs, code repositories, and online accelerated ad hoc adoption.

Potentially, organizations viewed POCs as superfluous. Moreover, this finding applied equally to AppSec, CTO/CISO, and Dev/Eng respondents. While the majority of respondents indicated that their organization added more security tools and checks to prepare for AI coding tools, over one-third of organizations did not take this precaution. This implies that they either felt sufficiently secure in their existing software development practices to cover any new challenges brought by AI or that AI coding tools don’t necessarily add more risk to the software development lifecycle.

Only 44.8% of organizations gave the majority of developers AI coding tool training

Proper training is an essential component in the adoption of any new technology that could inject considerable security risk. However, considerably less than half of all respondents said their organizations provided AI coding tool training to the majority of their developers. This may reflect the ease of use of the tools or that many of the tools actually include security scanning as part of the workflow. That said, coding tools do not offer training on how users can spot mistakes that the tools have made, even though such security mistakes are common and well-documented.

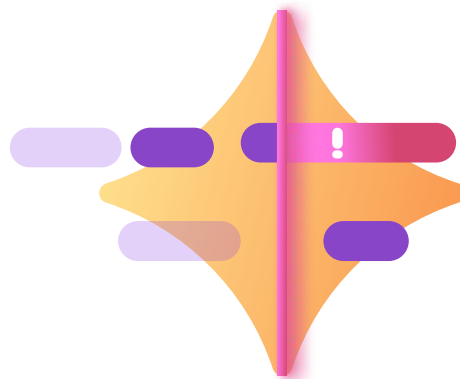
PERCENTAGE OF DEVELOPERS RECEIVING AI CODING TOOL TRAINING



PART TWO

Those who work more closely with code have greater doubts about security issues

AppSec teams tended to have a more negative view of the security risks of AI and how their organization was handling those risks. This included a lower opinion of AI-generated code security, a greater perceived risk from AI tools, and a dimmer view of the sufficiency of their organization's AI security policies.



AppSec team 3x more likely to rate gen-AI code security as "bad"

While representing a small percentage of total responses, AppSec, and security practitioners were 3x more likely than C-Suite respondents and significantly more likely than developers to state that AI-Generated code was “bad”. This divergence implies that those tasked with fixing and securing code may be alerted to the failures of AI tools more frequently than developers, who may not see the vulnerabilities and code errors, and C-suite members, who rarely touch code. On the opposite end of the spectrum, CTOs and CISOs were considerably more likely than developers working with AI-generated code on a daily basis to believe that the quality of generated code is “excellent”. This likely implies developers are more realistic about the actual quality of AI-generated code and are more exposed to flaws and problems that are common in AI-created code, according to [Snyk’s own findings](#) and [academic research](#).

HOW WOULD YOU RATE THE SECURITY OF AI-GENERATED CODE?

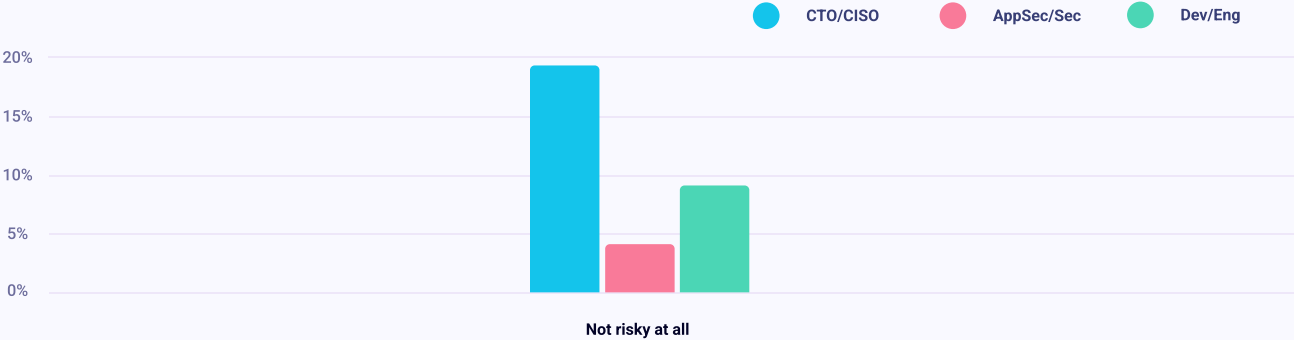


These findings raise several questions. First, are organizations broadly underestimating risk from AI coding tools? Respondents across all roles, on average, rated AI code quality with high marks. This is despite the fact that multiple academic research papers have found that AI-generated code consistently injects security risk and requires additional code reviews and remediation. ([Link to Snyk Webinar on the topic](#)). Second, if CTOs and CISOs are overestimating the quality of AI-generated code, is this because they are receiving imperfect information or have little direct contact with those working with the tools? And why are they not on the same page as developers?

C-Suite 2x to 5x less likely to see security risk from AI coding tools

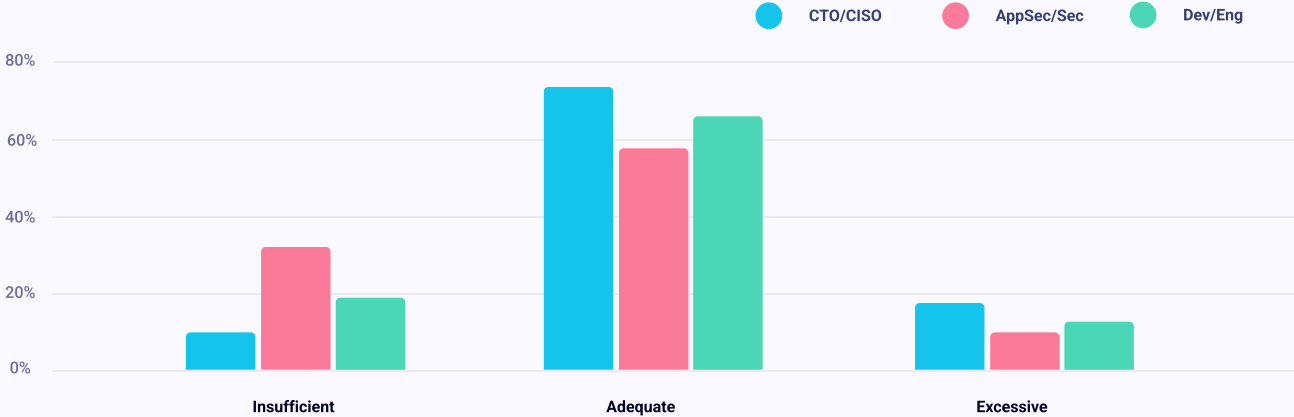
While respondents largely agreed that AI coding tools did not create an extensive risk, there was a large disparity among those who felt that AI is not risky at all. In our survey, 19.4% of C-Suite respondents said AI coding tools are “not risky at all,” while only 4.1% of AppSec team members agreed. Developers were closer to AppSec views, with 8.8% of Dev/Eng respondents saying that AI coding tools are minimally risky. Conversely, 38.3% of AppSec practitioners felt AI coding tools were “very risky” or worse, while only 29.8% of C-suite respondents agreed. One potential interpretation of this finding is that AppSec teams, which are much closer to daily remediation of flawed code and vulnerabilities, are seeing many more security issues emanating from AI tools than the C-suite, which tends to be more removed from daily security and coding activities.

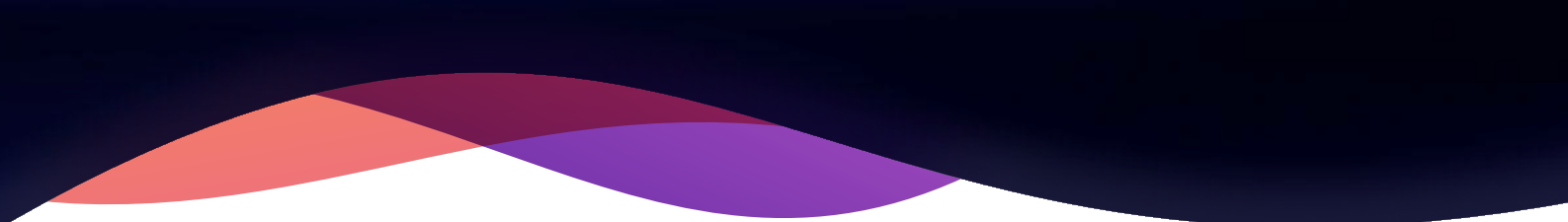
HOW WOULD YOU RATE YOUR ORGANIZATION'S SECURITY RISK FROM THE USE OF AI CODING TOOLS?



AppSec practitioners 3x more likely to say AI security policies are insufficient

HOW WOULD YOU DESCRIBE YOUR ORGANIZATION'S SECURITY POLICIES FOR AI CODING TOOLS (E.G., COPILOT, CODEWHISPERER, GEMINI, TA...)





AppSec practitioners doubt their organization's security policies for AI coding tools. Nearly three times as many respondents from AppSec roles described their AI coding tool policies as "insufficient" compared to the number of CTO and CISO respondents making the same observation. In comparison, developers and engineers feel in the middle, with only 19% of respondents saying their org's AI policies are insufficient versus 30.1% of AppSec members. In other words, the further away someone in the technology organization is from security processes, the less likely they are to approve of AI security policies. This could be an indication that AppSec teams are seeing more risks. It also might mean they feel that AI security policies need to be constructed in a logical way that works with application security requirements. C-Suite respondents were the most likely to think these policies were excessive. This thinking may reflect their strong desire to accelerate AI coding tool adoption, as expressed in other questions on this survey.

CONCLUSION

Organizations remain conflicted on their state of AI readiness, and fail to take basic steps toward AI readiness and preparedness

Ready or not? Respondents are generally positive about the state of AI coding tool readiness in their organizations. They generally think their security policies are sufficient and that AI-generated code is secure. In the main, they believe they are ready for AI adoption. However, they remain conflicted on AI coding tool security. Across all roles, security fears are perceived as the biggest barrier to entry of AI coding tools. In terms of practical processes to prepare, less than one-fifth of respondents said their organizations ran PoCs, a basic step that is fundamental to new technology adoption. And less than half of respondents said that the majority of their developers had received AI coding tool training. These contradictions may indicate a lack of planning and strategy, as well as a lack of structure around AI adoption.

Diving deeper, survey respondents demonstrated a consistent divergence by role in their perceptions of code quality, tool safety, and general organizational preparedness. The C-suite held a more positive view of AI coding tools and preparedness than respondents who work closer to the code or security processes and policies. In particular, security team members held a dimmer view of AI coding tool security, implying that this influential group is exposed to more problems generated by AI coding and is reacting accordingly.

The above contradictions imply insufficient planning or cohesive strategy around AI coding tool adoption, as well as a lack of structure in determining and fulfilling necessary pre-conditions, potentially because of a lack of consistent cross-organizational visibility. This may have happened because, like with smartphones and certain consumer software products, adoption was initially rapid and uncontrolled before being institutionalized by IT organizations. In that sense, rollouts might have been chaotic in the beginning and challenging to control later on. The bottom line, however, is that organizations should consider a more structured approach to AI coding tool adoption and security that is closer to the adoption processes of other types of enterprise software. Taking this approach should also resolve security fears and also address outsized concerns of developers and security teams. It will do this by putting better checks and balances in place and providing a more holistic, methodical, and programmatic approach to deploying a fundamental shift in the software development process.

Technology leaders listening to the signals from this survey could benefit from the following actions:

- Set up a formal POC process as part of the adoption of any AI tools. And in this process, consider data quality and data controls; whether your AI model is multimodal or whether everything rests on a single type of AI model; how much accuracy or reliability is required from your AI tool, and if a high level is required, whether there is extensive expert tweaking throughout the AI loop.
- Give more weight to recommendations from AppSec teams that are most directly exposed to code security issues and tool risks.
- Document and audit all instances of AI code generation tools to better inform security and QA processes.
- Take regular pulse surveys of all three groups to ensure greater future alignment of views on AI readiness, preparedness, and security.
- Consider engaging expert guidance on AI best practices for a structured approach to AI adoption. \
- Proactively drive executive buy-in by examining tools that give visibility, reporting, and control over your security posture, demonstrating ROI for expenditure on AI security tools.
- To fulfill the desire for swift adoption of AI and overcome the barrier of security concerns — alongside safeguards like governance, education, and training — consider adopting purpose-built security tools that proactively prevent and fix security incidents early in the development cycle that can be implemented quickly and scale consistently.
- If developer adoption is a barrier to the adoption of security tools that can help manage AI risk, consider security tools that can fit more seamlessly into the developer workflow and ask questions such as: Does this security tool genuinely work directly within the IDE? Does it automate vulnerability fixing for busy developers? Is it fast enough to keep up with AI-generated code?
- Ramp up education and training around AI, e.g., the benefits and risks of various types of AI, what roles they can play in software development, and how to use AI tools appropriately. Snyk's [blog](#), and free educational tool, [Snyk Learn](#), are rich educational resources and good places to start. Another way to up-skill frontline AI-facing teams frictionlessly is to build continuous professional development into their daily tools. Google Gemini is an AI coding assistant that is integrated with Snyk, allowing developers to query Snyk's extensive security knowledge base directly with natural language interactions.

Methodology

For this report, we surveyed 406 IT professionals from around the world. Snyk limited the survey to respondents who described their roles as "CTO", "CISO", "developer", "engineer", "security", or "appsec". Snyk intends to continue collecting data for this survey at online and offline events throughout 2024 to paint an even broader picture of enterprise AI readiness and differences in perceptions of AI risks, preparedness, and challenges.

