REPORT

The State of Developer-Driven Security Survey

2022

SECURE CODE WARRIOR

About the Survey

The Secure Code Warrior 'State of Developer-Driven Security' survey was conducted by Evans Data Corp in December of 2021.

Questions about software coding, security awareness, training, support, motivations, and other issues were asked of 1,200 active software developers working in the Asia-Pacific region, Europe and North America. The survey was given in English and translated when needed to obtain an accurate global perspective. Survey respondents included managers from within the development community as well as coders who are actively creating new applications.

The margin of error for the survey is 2.7%. Where appropriate, results from the 2021 survey have been compared with another survey that Secure Code Warrior commissioned in 2020.

Table of Contents

Introduction

Many organizations are still employing traditional software development methodologies while navigating an ever-changing landscape of cybersecurity risks and demands.

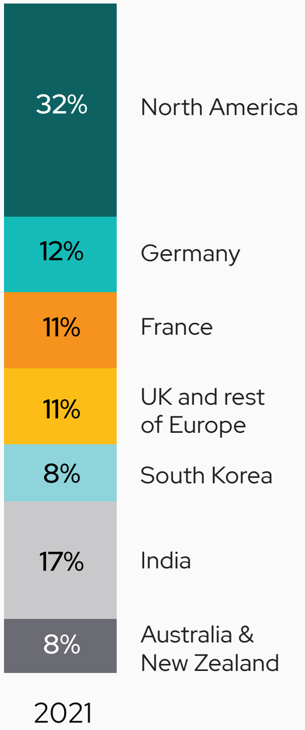
Security professionals know they must implement and maintain strategies to get closer to a DevSecOps, or even DevOps, approach if they are to defend against current threats. The coveted goal of DevSecOps considers security at the very beginning of the software development lifecycle (SDLC) and enables developers to share the responsibility without sacrificing speed. A key element of that is to shift security left - or rather start left - so that developers prioritize security alongside features and functionality. When it's done right, security-skilled developers improve productivity by reducing vulnerabilities that create rework, maintain software release velocity, and ensure quality code without compromising innovation.

But, despite the vast array of security measures adopted by organizations, we continue to feel the repercussions of exploitable software vulnerabilities.

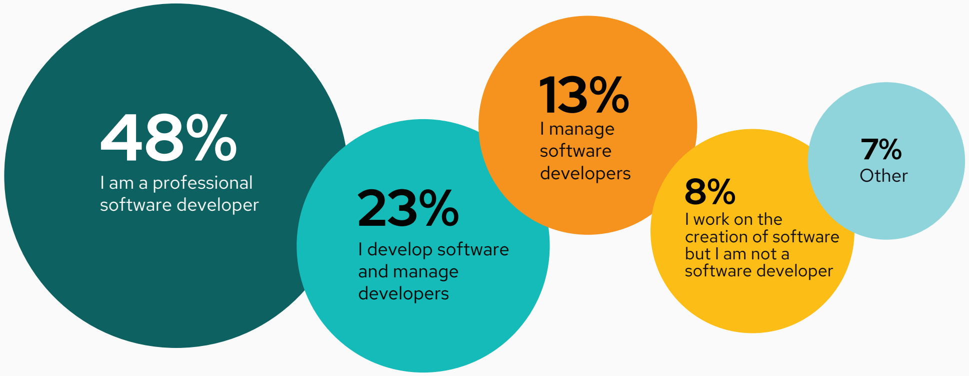
For the 2nd year, Secure Code Warrior has commissioned research with Evans Data Corp to survey 1,200 developers globally to understand the skills, perceptions, and behaviors when it comes to secure coding practices, and their impact and perceived relevancy in the SDLC.

Demographics

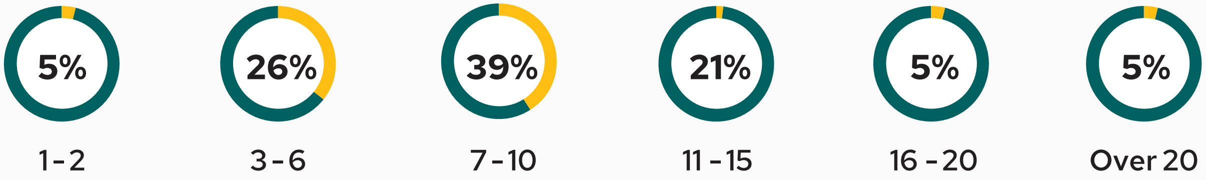
Regional split



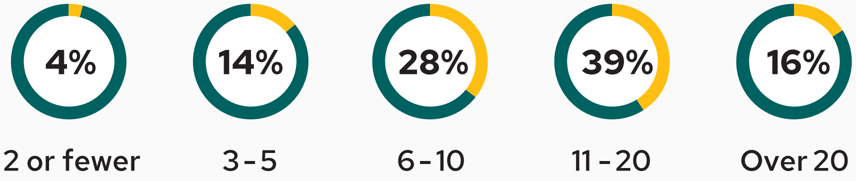
What is your role in the development of software?



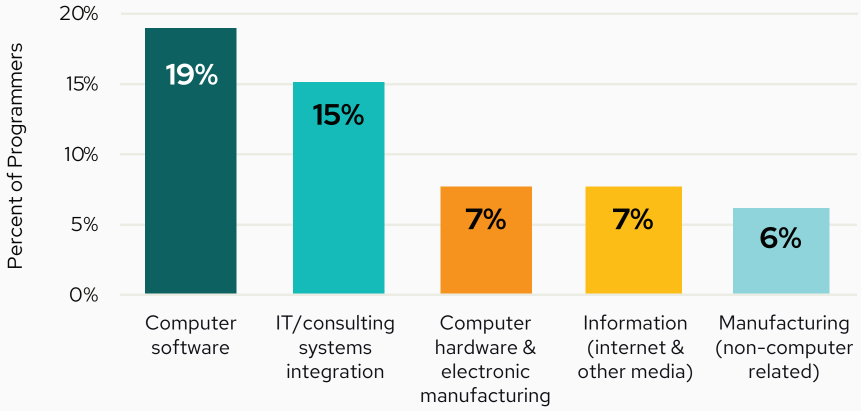
How many years have you been professionally involved with software programming?



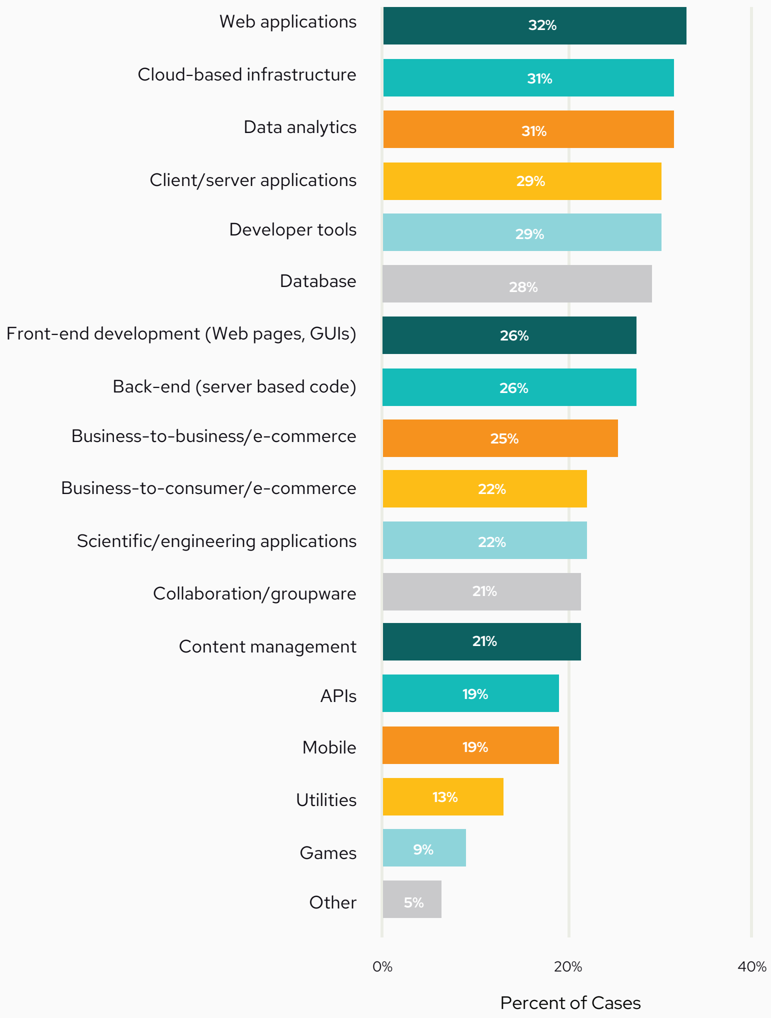
How many years has software been developed in your company?



What industry is your company in? *(Top 5 of 23)*



What types of software are you developing?



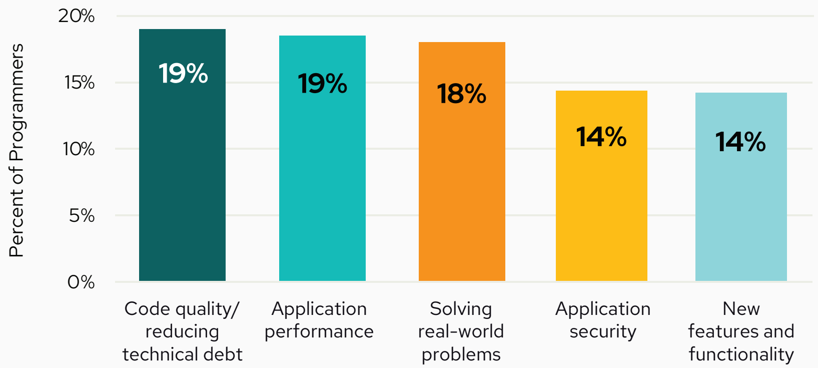
Developer priorities when writing code

Developers have a mix of priorities when writing code, but application security is deemed the priority by only 14%. However, later on 41% of respondents stated that overall, functionality and security have equal importance in their organization (page 8).

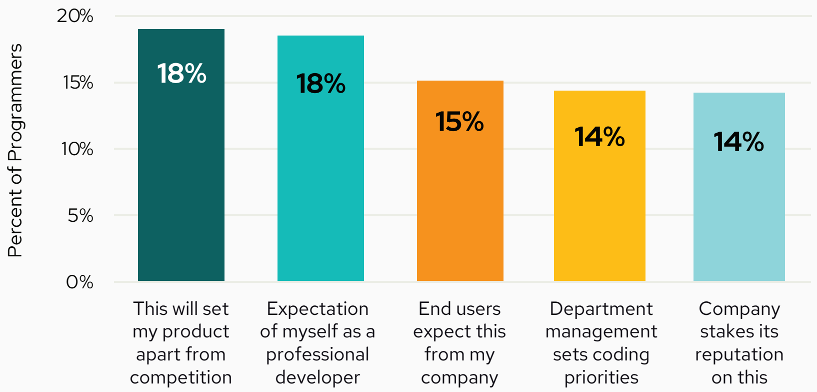
14%

of developers list application security as the top priority when writing code

What is your top priority when writing code? *(Top 5 of 8)*



Why is this your top priority? *(Top 5 of 8)*

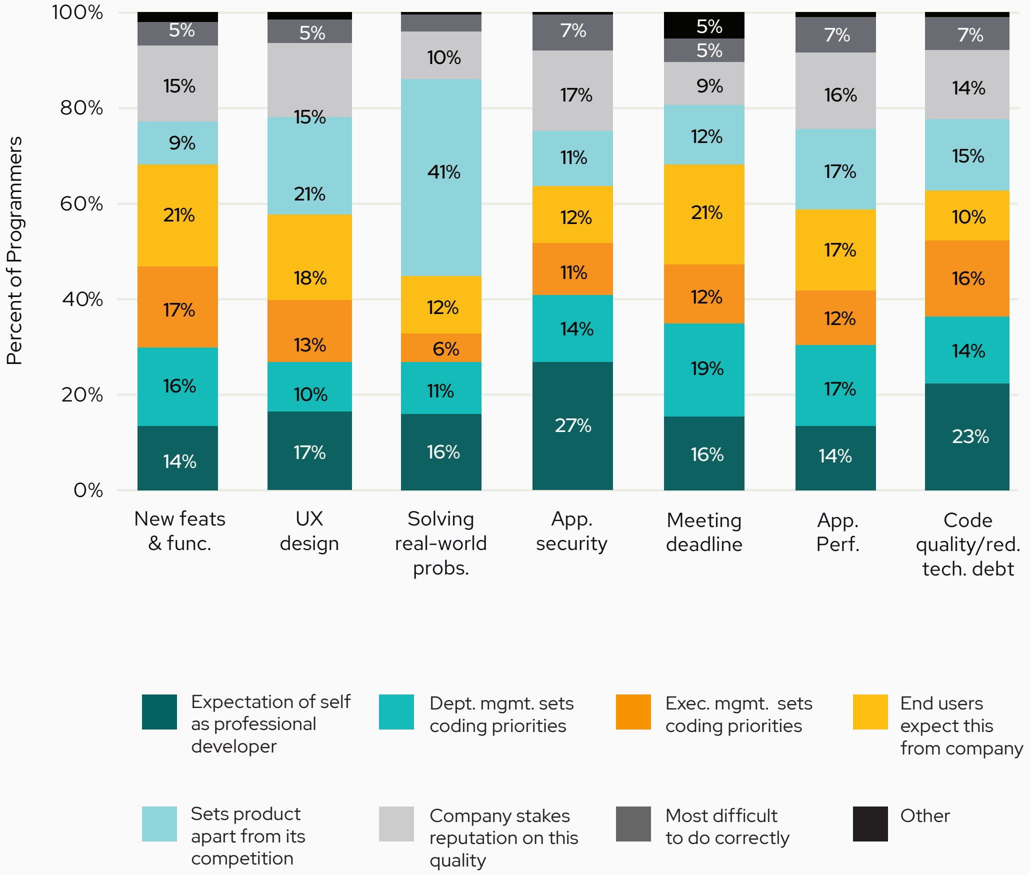


Respondents stated that management prioritizes meeting deadlines, application performance, and new features/functionality over application security.

27%

of respondents rate application security as a priority because they expect it from themselves as a professional developer

Reason for top priority by top priority

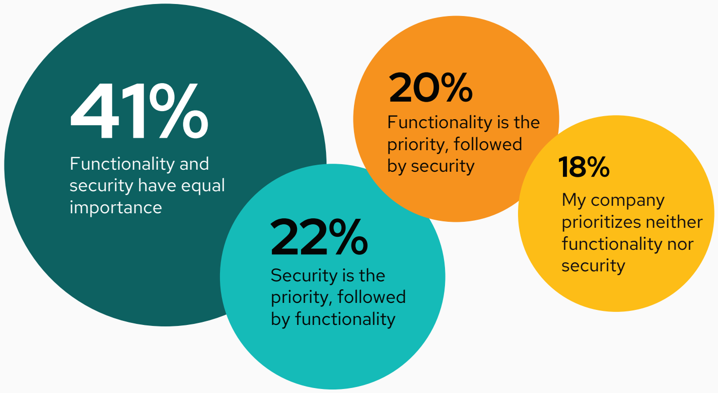


Developers are predisposed to viewing priorities' importance as growing, but app security is particularly top of mind. Interestingly, developers expect security to increase in priority, whereas managers are more inclined to expect performance to increase in priority.

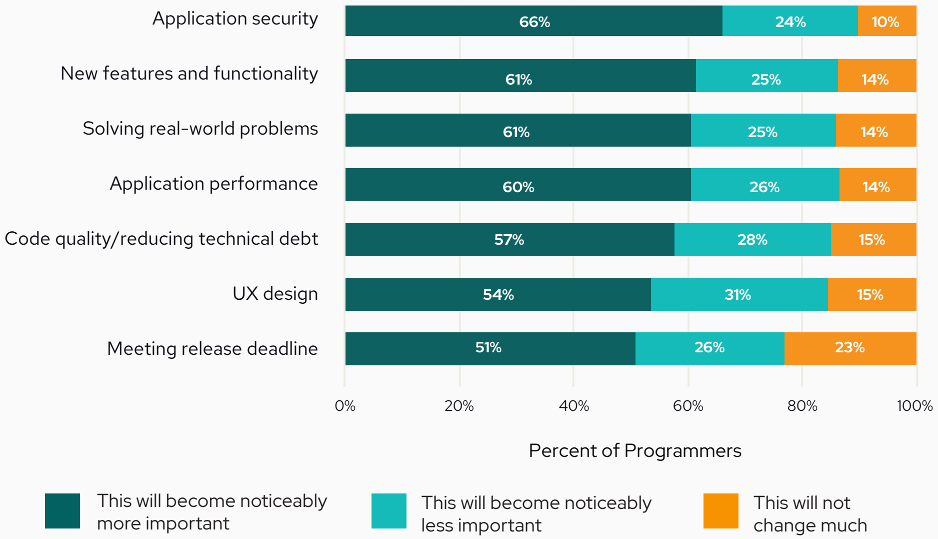
41%

of respondents state that overall, functionality and security have equal importance in their organization

What does your company prioritize?



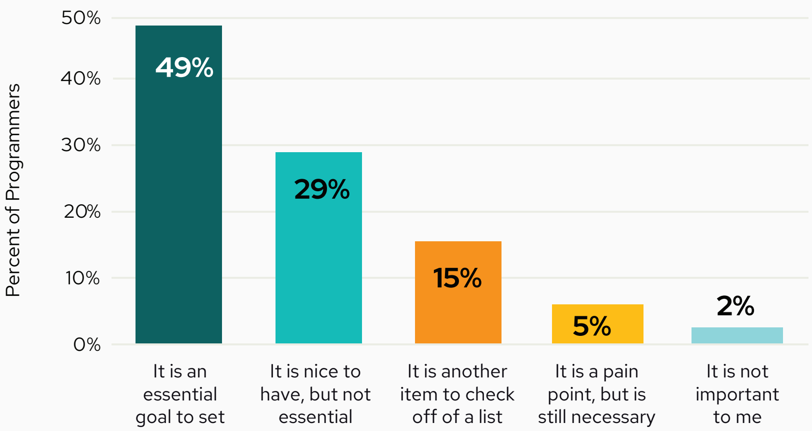
With respect to the following, how will your team's priorities change in the next 12-18 months?



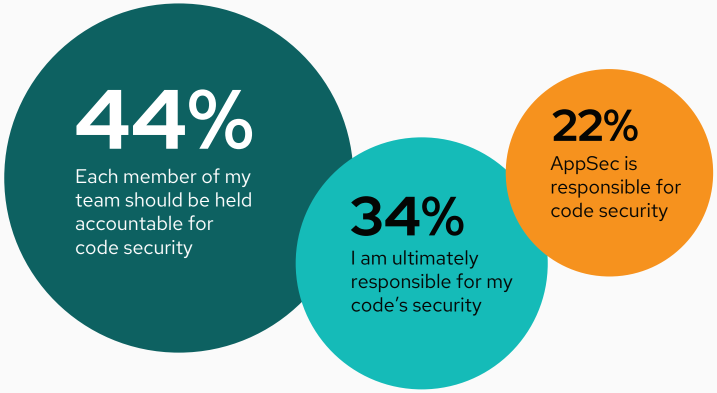
Opinions and perceptions of secure code

Just over half of all respondents state that secure code is not an essential goal to set, although 78% agree that code security is not just the responsibility of the AppSec function.

In general, what is your opinion about secure code?



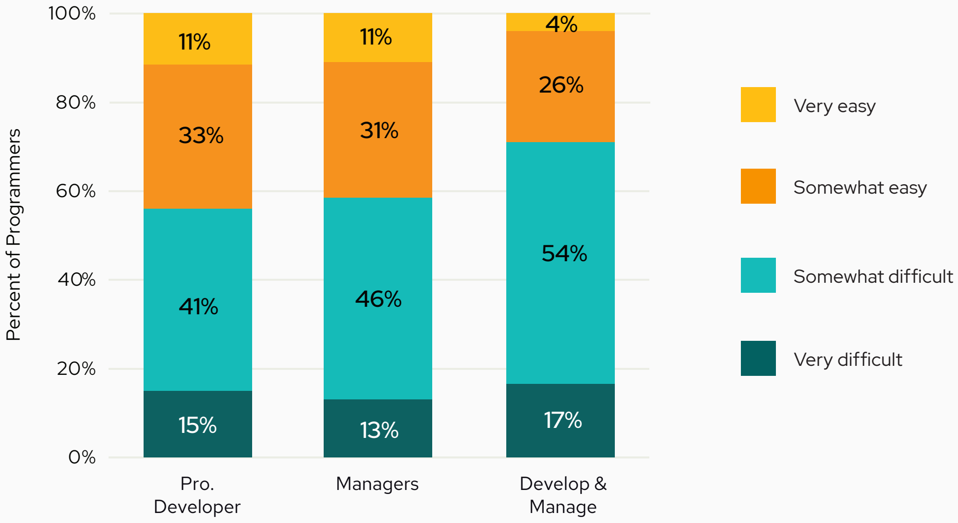
Which of these statements do you most agree with?



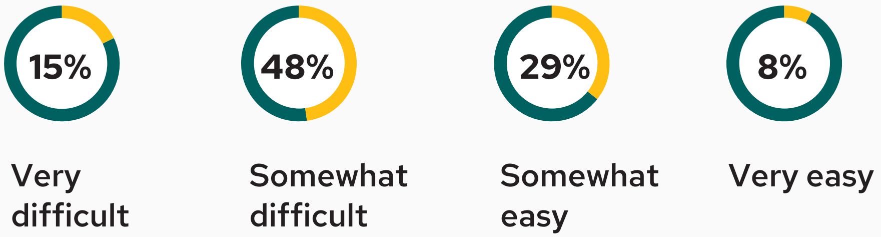
63%

of developers rate writing secure code that is free from vulnerabilities to be difficult. Developers who also manage other developers are more likely than others to perceive writing secure code as difficult

Difficulty of secure coding



How would you rate the relative ease of writing secure code that is free from vulnerabilities?

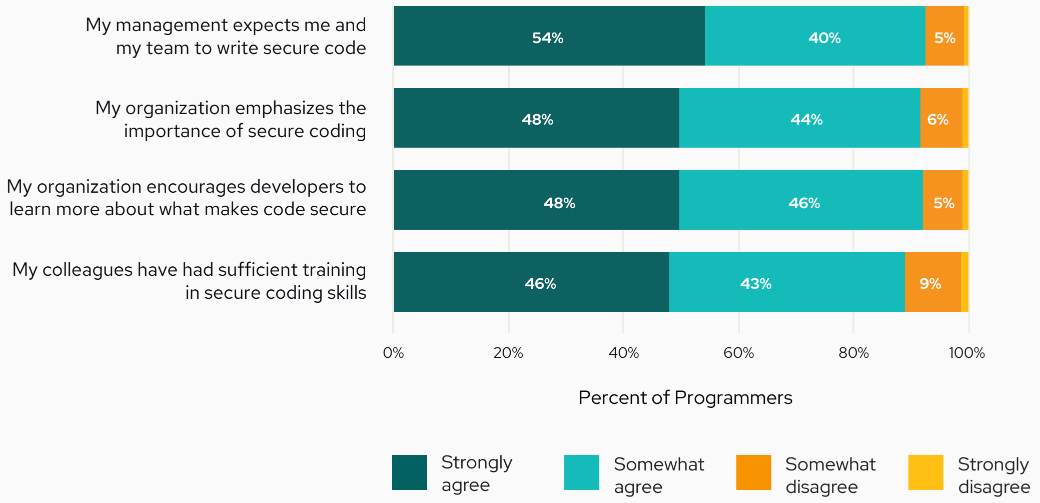


While only 49% of respondents state that secure coding is an essential goal to set, a significantly higher number of respondents agree that secure code is an expectation of their organization and management. And despite 63% of developers stating that the art of writing secure code is difficult, 87% of those surveyed state that they have received sufficient training in secure coding skills.

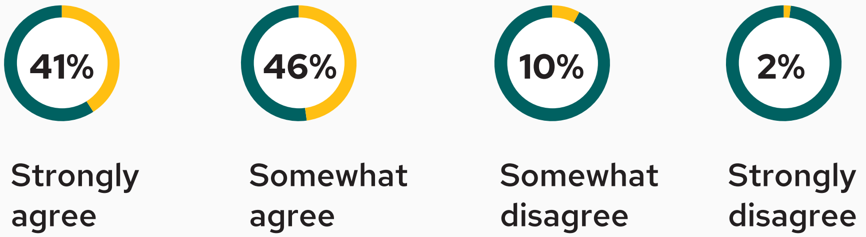
89%

of those surveyed state that they have received sufficient training in secure coding skills

Please rate your agreement with the following statements about secure coding



I have had sufficient training in secure coding skills

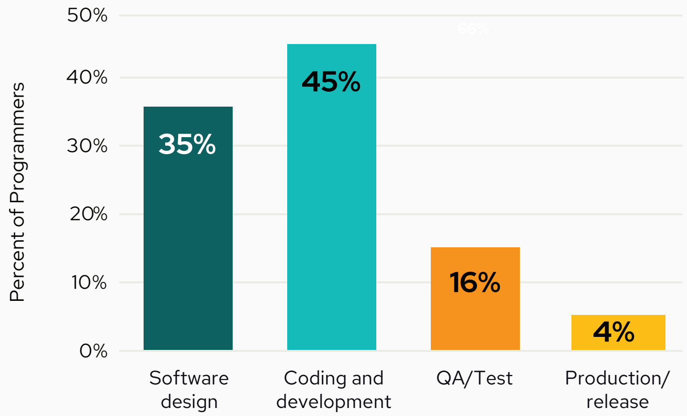


Current secure coding practices

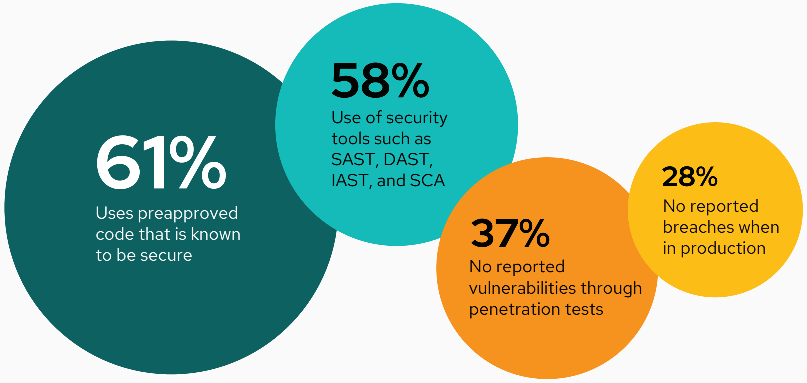
Respondents stated that secure coding is considered early in the SDLC, however, they are also relying on pre-approved code and tooling to ensure code security rather than utilizing developer skills to write code that is free from vulnerabilities. It's worth acknowledging that both pre-approved code and tools only address known vulnerabilities.

Respondents stated their organizations rely on the use of pre-approved code and tooling to ensure code security

When is secure coding primarily considered in your company's software development lifecycle?



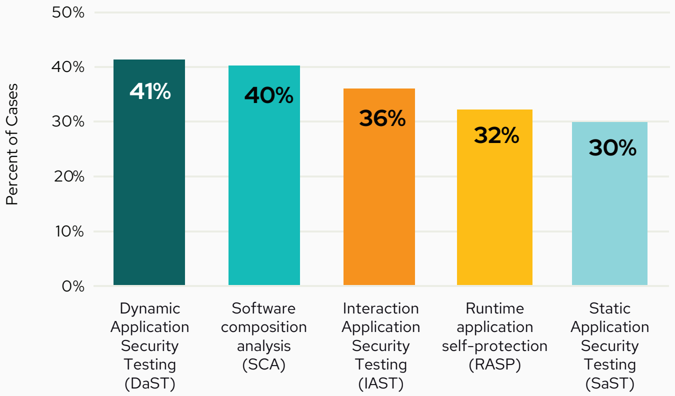
How is the code written within your organization recognized as secure?



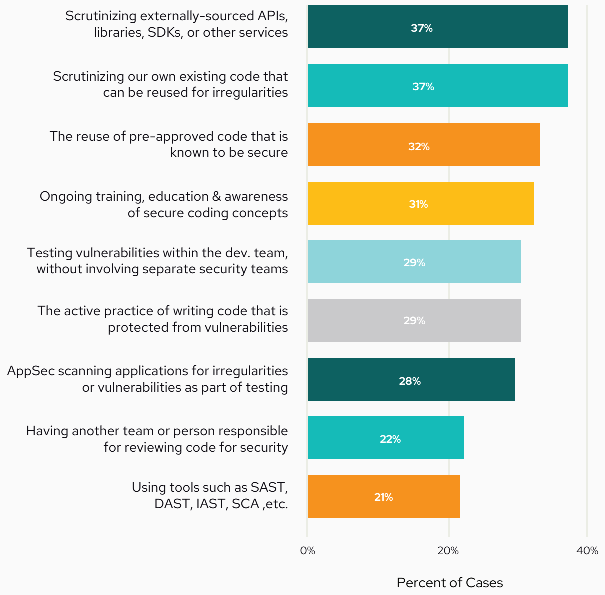
When questioned about the top 3 practices associated with secure coding, the art of writing new code that is free from vulnerabilities came in at 6th. The top 3 options cited all relate to the reuse of code (which may be secure today, but not necessarily tomorrow).

Developers rely on using existing/pre-approved code, rather than the practice of writing new code that is free from vulnerabilities.

How does your organization typically identify software security flaws? *(Top 5 of 8)*



The top practices associated with secure coding



Developers with 16+ years of experience recognize the practice of writing secure code and training/awareness of common vulnerabilities as more important than developers with less experience.

Practices associated with secure code by experience

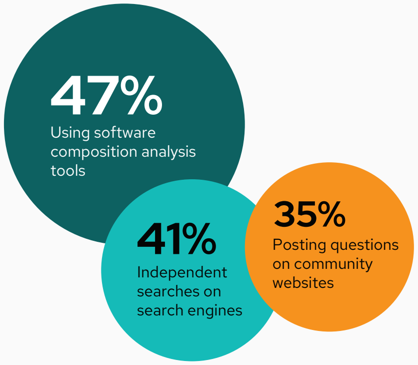
(% within row)		Which of the following are the top three practices that you associate with secure coding?								
		The active practice of writing code that is protected from vulnerabilities	Scrutinizing externally-sourced APIs, libraries, SDKs, or other services for irregularities or vulnerabilities at the start of the development lifecycle	Scrutinizing our own existing code that can be reused for irregularities or vulnerabilities at the start of each sprint	AppSec scanning applications for irregularities or vulnerabilities as part of testing	Ongoing training, education and awareness of secure coding concepts and common vulnerab-ilities	Having another team or person responsible for reviewing code for security	Testing vulnerab-ilities within the development team, without involving separate security teams	The reuse of pre-approved code that is known to be secure	Using tools such as SAST, DAST, IAST, SCA ,etc.
How many years have you been professionally involved with software programming?	1-2	29.1%	20.0%	27.3%	16.4%	34.5%	14.5%	23.6%	32.7%	9.1%
	3-6	32.4%	26.5%	31.7%	24.5%	28.8%	26.1%	30.1%	29.7%	20.6%
	7-10	26.8%	42.9%	41.6%	33.6%	31.8%	20.7%	27.5%	34.6%	23.1%
	11-15	26.4%	45.5%	42.1%	29.3%	30.2%	20.7%	32.2%	33.1%	22.3%
	16-20	34.4%	32.8%	27.9%	29.5%	34.4%	23.0%	34.4%	24.6%	23.0%
	Over 20	42.3%	30.8%	21.2%	17.3%	36.5%	36.5%	30.8%	23.1%	21.2%

Tooling, rather than information discovery, is cited as essential to creating secure code. This represents one extreme of a four-point scale, including: essential, often necessary, nice to have, and not needed.

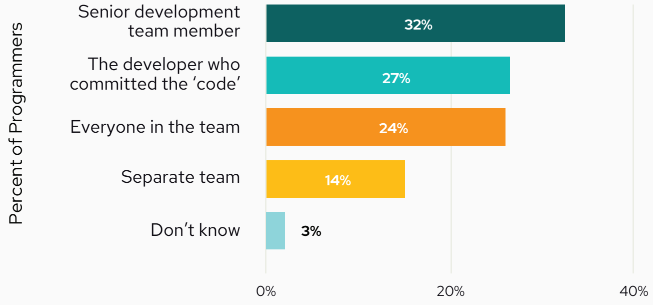
Senior development team members are the most likely to be assigned to fix security tickets, likely due to the advanced skillset necessary for fixing some security issues.

Although the majority of respondents are content with their team's proficiency in writing secure code that is free from vulnerabilities, just 35% state that they have excellent proficiency in this. And despite this, 67% of developers still think that they ship code with vulnerabilities.

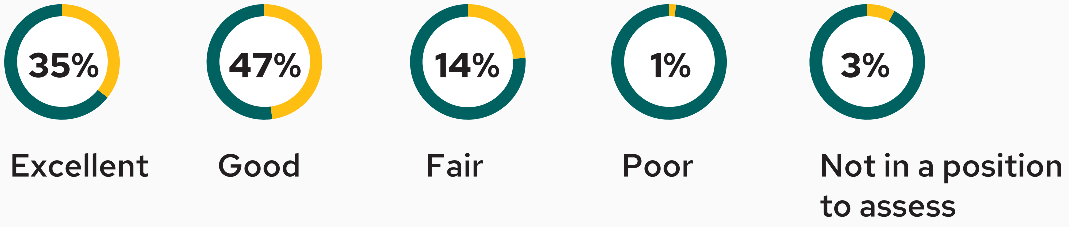
How would you rate the following in helping you fix vulnerabilities to secure your code?



Who is assigned to fix security tickets?



How would you rate your/your team's proficiency in writing secure code that is free from vulnerabilities?



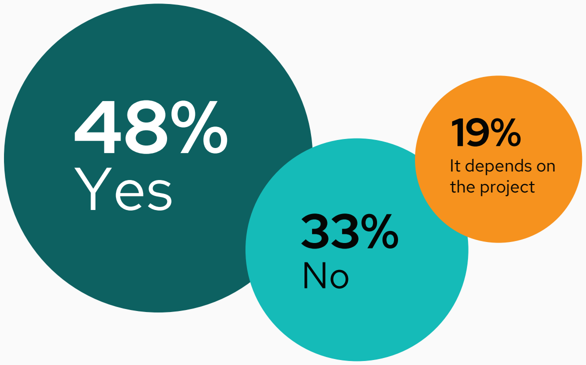
What vulnerabilities?

Of the 67% of developers who think they leave vulnerabilities in their code, 45% believe that these are inherent flaws in libraries or frameworks.

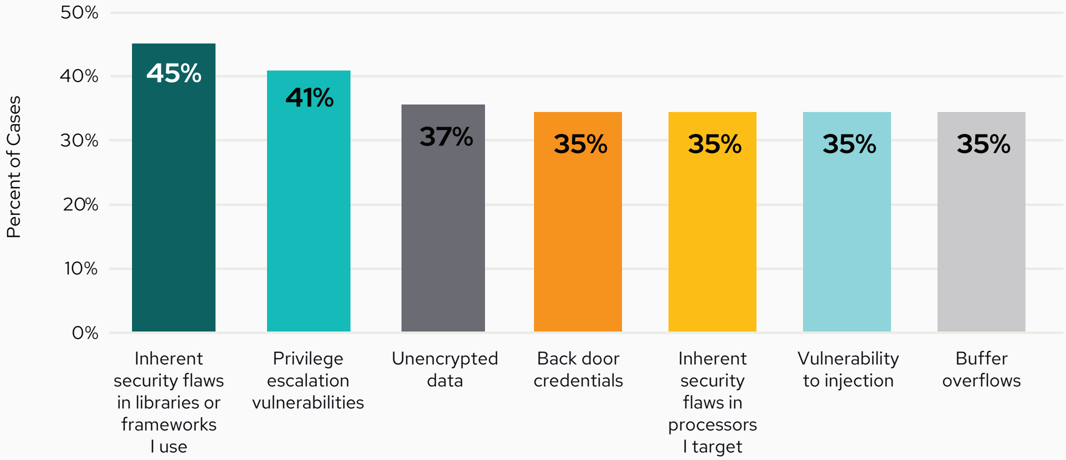
67%

of developers think that they ship code with vulnerabilities

Do you think that you leave vulnerabilities in your code?



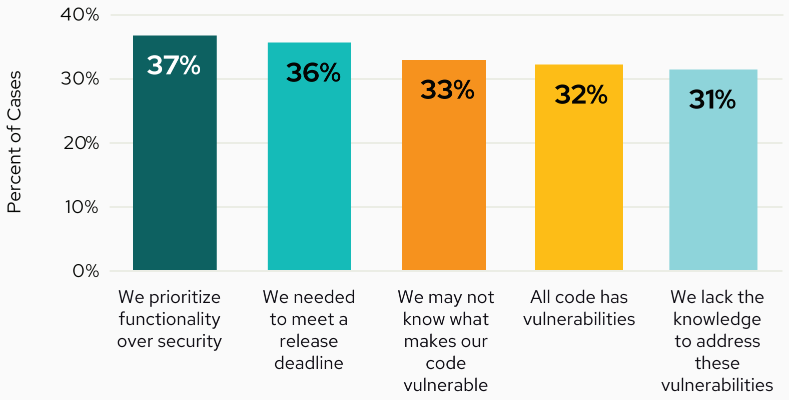
What types of vulnerabilities do you believe exist in your code?



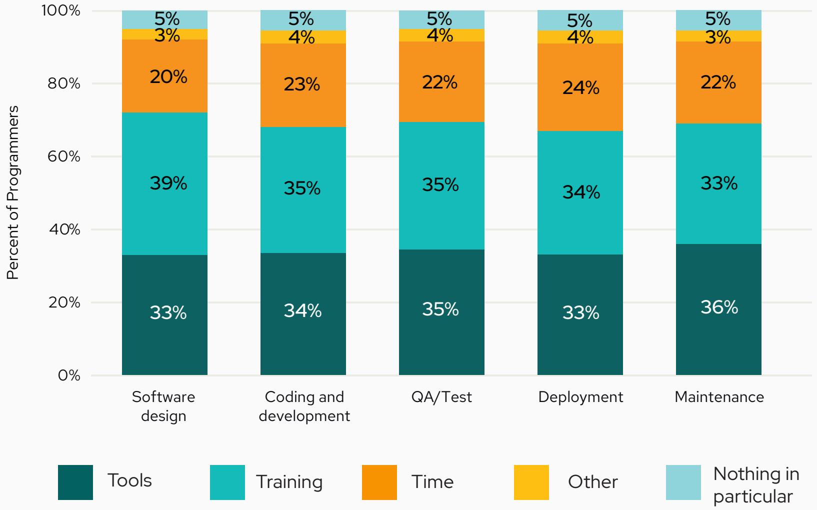
The reasons cited why vulnerabilities exist in code are concerning. Developers say they have had sufficient security training, so we must question the level of knowledge and skills if vulnerabilities continue to be shipped in code.

Training is clearly the top area of need in supporting secure coding in software design. Regardless of company size, tools and training are stated as developers' top security needs throughout the development lifecycle.

Why do these vulnerabilities exist in your code? *(Top 5 of 7)*



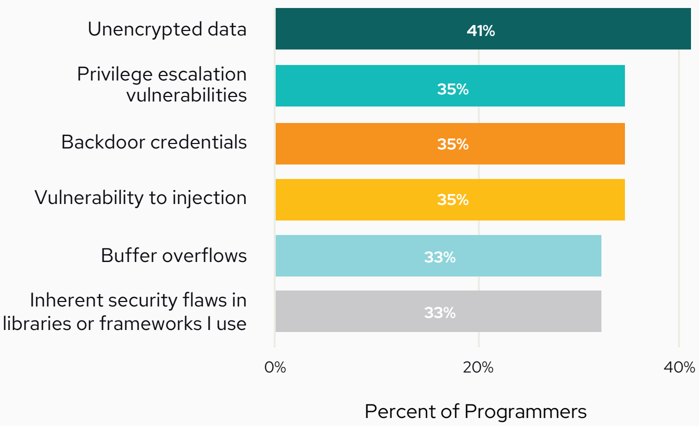
What best describes what's lacking for secure coding at each of the following stages of the software development lifecycle?



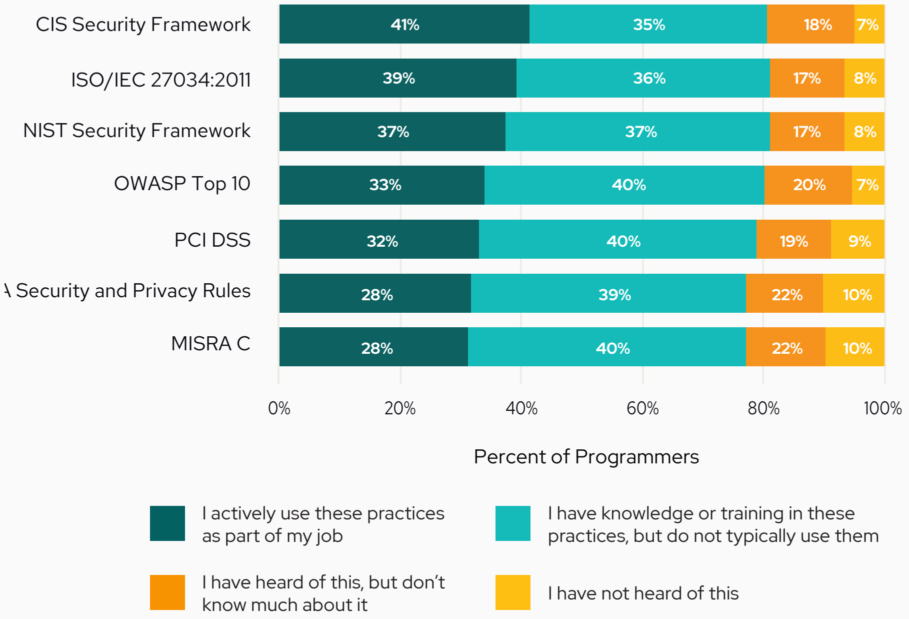
Security knowledge, understanding, and training

More than half of respondents are not familiar with common software vulnerabilities, how to avoid them and how they can be exploited. Respondents faired better with their knowledge of compliance frameworks and best practices.

How familiar are you with the following software vulnerabilities?



How familiar are you with the following compliance frameworks and best practices?



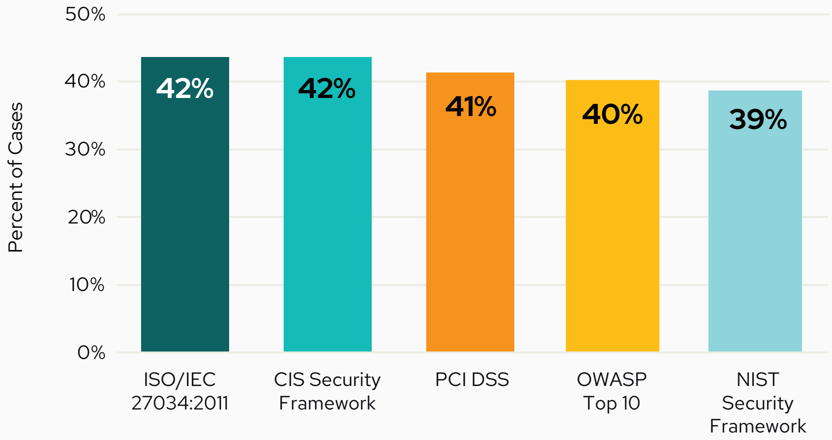
Do the developers on your team require more training in security frameworks?

50% Yes, we require significant training

42% Yes, we require some training

For which of the following compliance frameworks do you think developers on your team need better secure code training?

(Top 5 of 7)



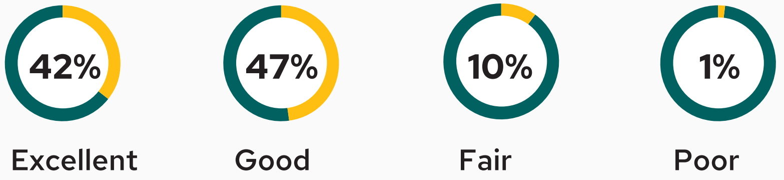
92% of respondents acknowledged that developers on their teams required additional training in security frameworks

Additional training is required even though the majority of respondents have rated their prior secure code training as good to excellent. However, only 43% of respondents felt the training received was highly relevant to their work.

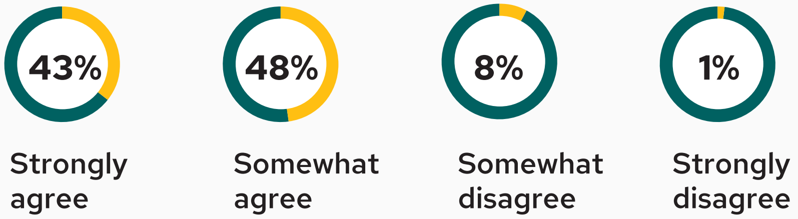
81%

of respondents stated they regularly apply what they have learned in secure code training, yet vulnerabilities are still shipped in code

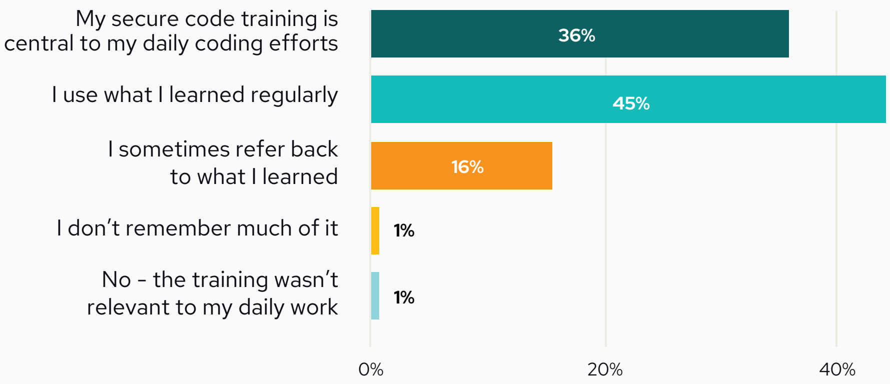
How would you rate your secure code training?



The training I have had in secure coding was relevant to my work



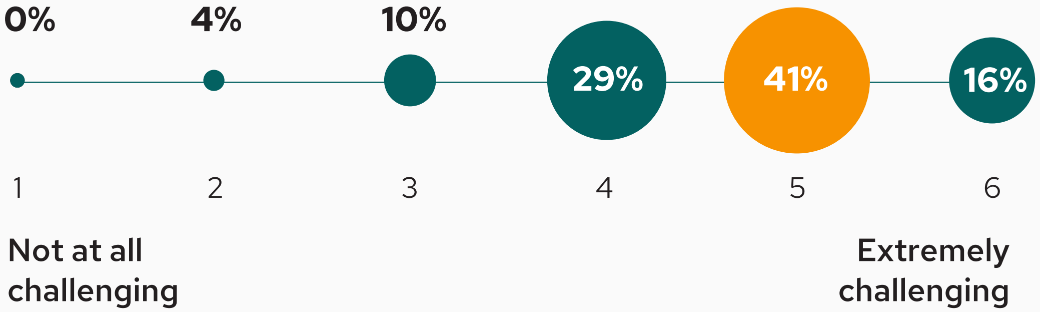
Do you use what you've learned in your secure code training?



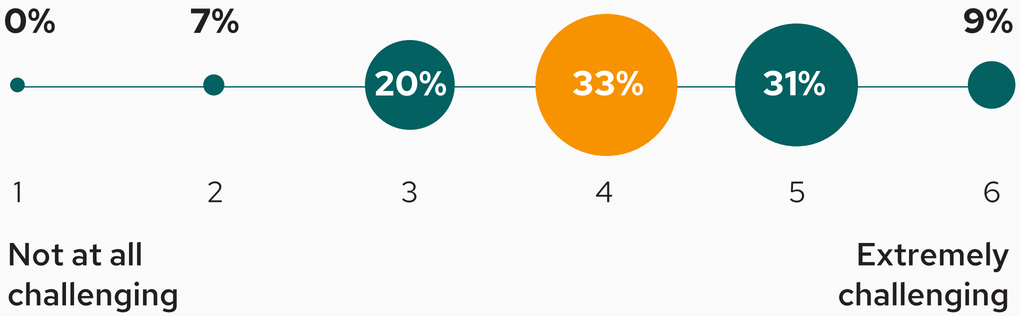
86%

of developers state they find it challenging to practice secure coding

As a developer, how challenging is it to practice secure coding?



As a manager, how challenging is it to implement secure coding practices in your organization?



Barriers to adoption

A lack of time, planning and prioritization are listed as the top barriers to shifting security left.

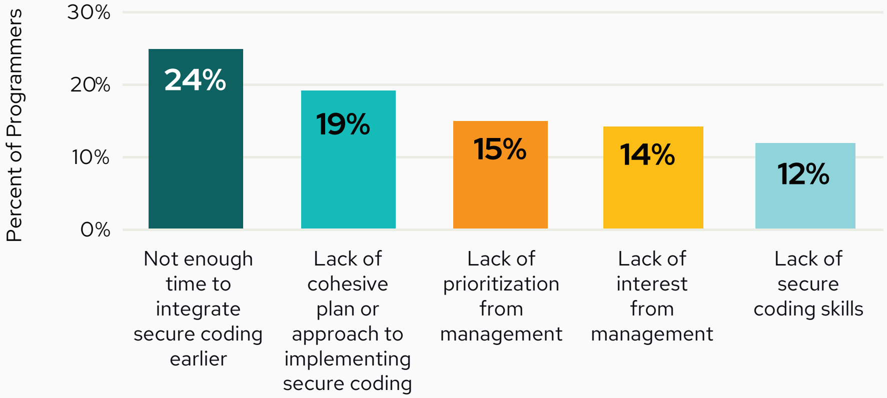
24%

of respondents say 'not enough time' is the biggest impediment to integrating secure code earlier

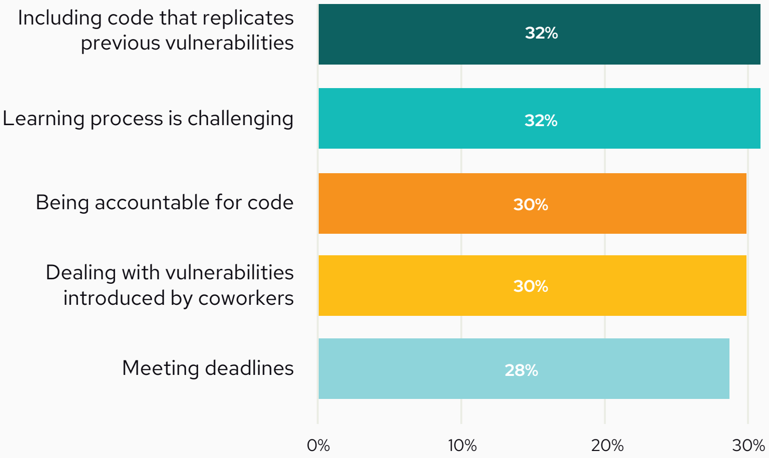
32%

say including code that replicates previous vulnerabilities is a main challenge with implementing secure code

Top impediments to shifting secure code considerations earlier in the development cycle *(Top 5 of 7)*



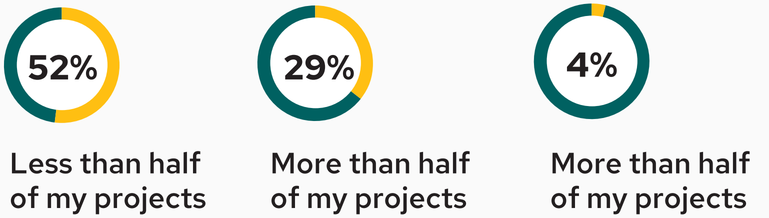
Top concerns with regards to implementation and practice of secure coding *(Top 5 of 11)*



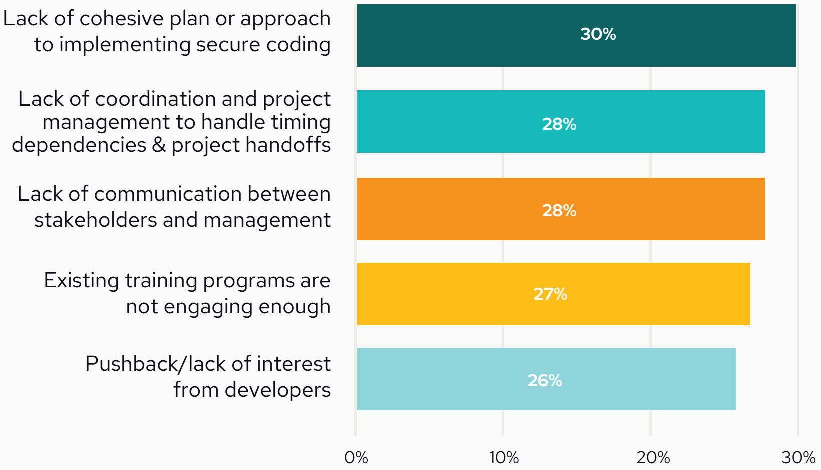
When asked, development managers cite a myriad of reasons why they regularly encounter obstacles to adopting secure code practices – from a lack of a cohesive approach, poor management, communication, issues with training and developer skills.

The main concern with the implementation of secure code points to the most relied-on practice to create secure code – the use of pre-approved code, or code from libraries that are deemed to be secure.

How often do you encounter obstacles that prevent you from adopting secure code practices?



From a management perspective, what obstacles have prevented you from adopting secure code practices? *(Top 5 of 11)*



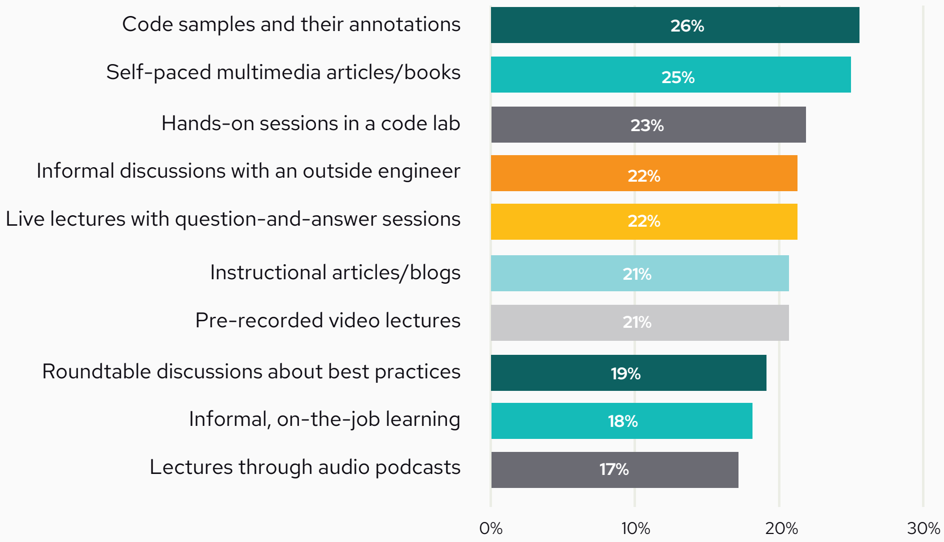
Developer needs and motivators

Self-driven learning is often the low-hanging fruit for developers; code samples and their annotations and self-paced articles/books can both be done at the developers' discretion. These top answers support a previous question where developers indicated that they do not have enough time for secure coding practices.

Developers want (and need) hands-on, interactive and contextual learning that is recognized

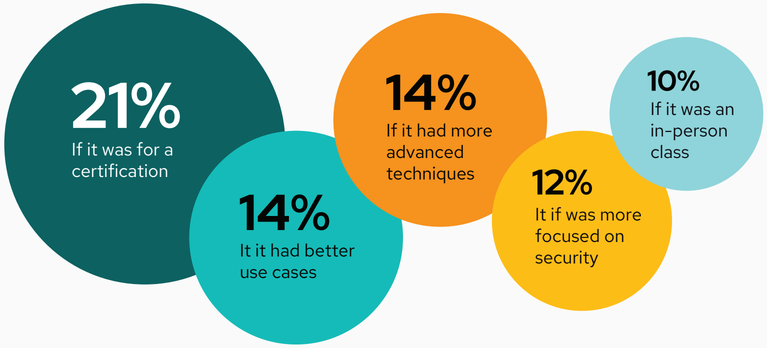
How would you prefer to train in secure coding practices?

(Top 10 of 13)



How could your secure code training be improved?

(Top 5 of 9)

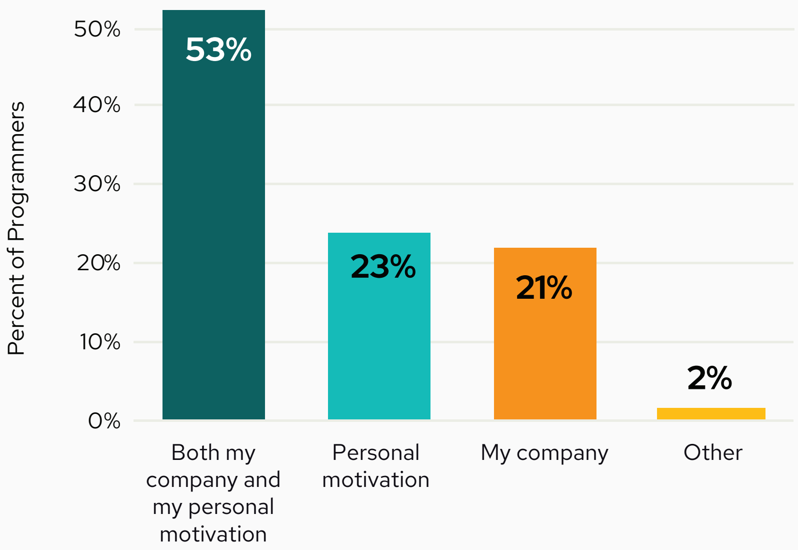


53%

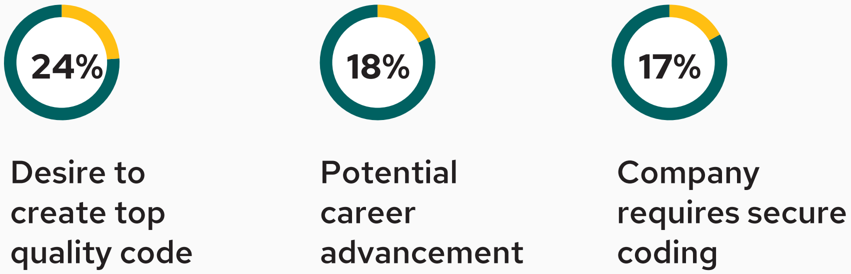
of respondents say their personal and company interests are driving their motivation to study secure coding

Yet again, developers say they are driven to create top-quality code

What is driving your interest to study secure coding practices?



What best describes your personal motivation for learning how to use secure coding practices? *(Top 3 of 6)*



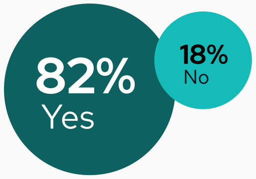
Recognized benefits of secure coding practices

There is wide agreement that secure code training improves productivity. Skilled developers reduce rework and patching, thanks to fewer vulnerabilities and coding errors. Both developers and managers agree that secure coding skills are valued and sought by employees.

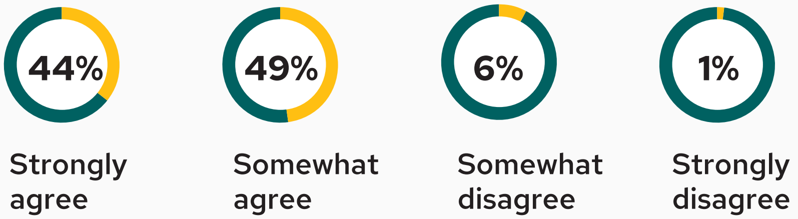
76%

of respondents agreed that good training in secure coding would improve productivity

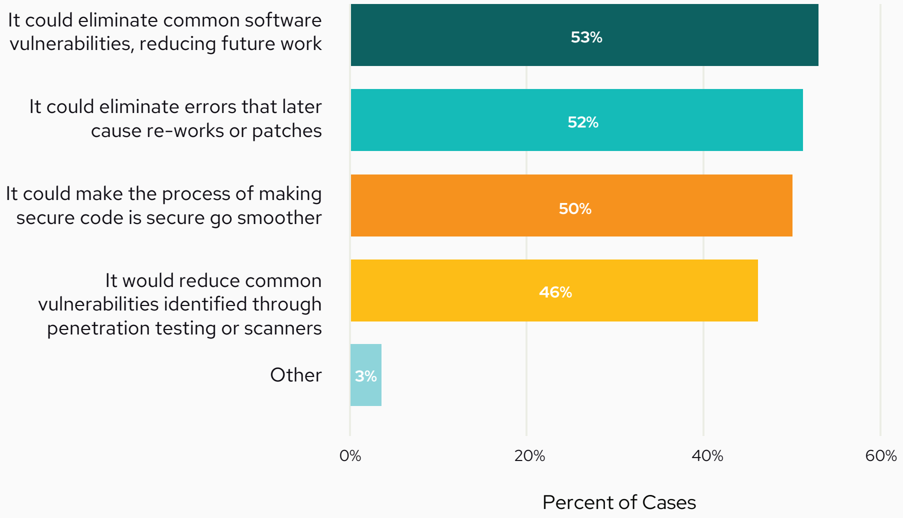
As a manager, are you more likely to hire developers who have secure coding skills?



The training I have had in secure coding has been valuable to my career



In what way would secure code training MOST improve productivity?



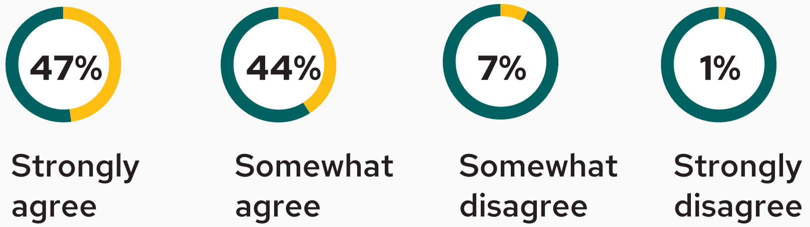
About 4 out of 5 managers place value on secure coding skills when hiring for at least some development roles.

Larger companies with more established software initiatives are particularly likely to emphasize secure coding in hiring.

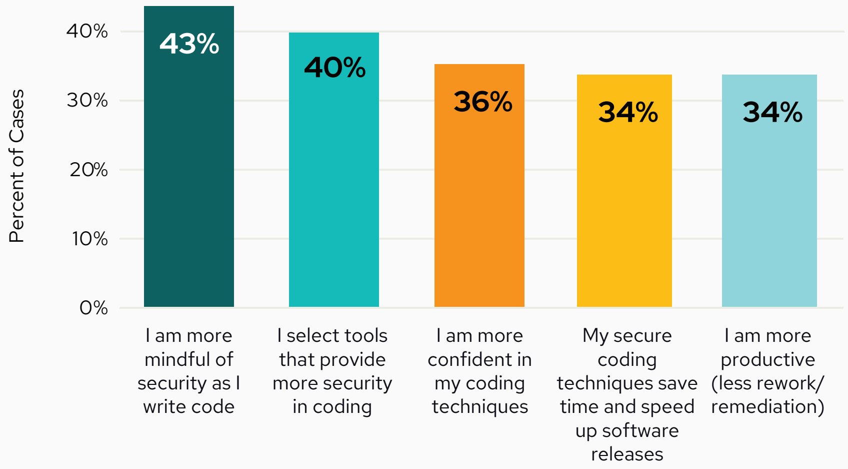
47%

of developers strongly agree that training in secure code has changed the way they write code

The training I have had in secure coding has changed the way I write code

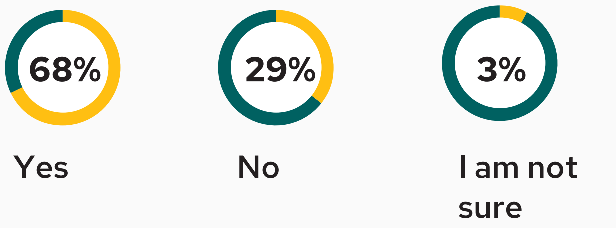


In what ways has being trained in secure coding changed the way you write code? *(Top 5 of 7)*

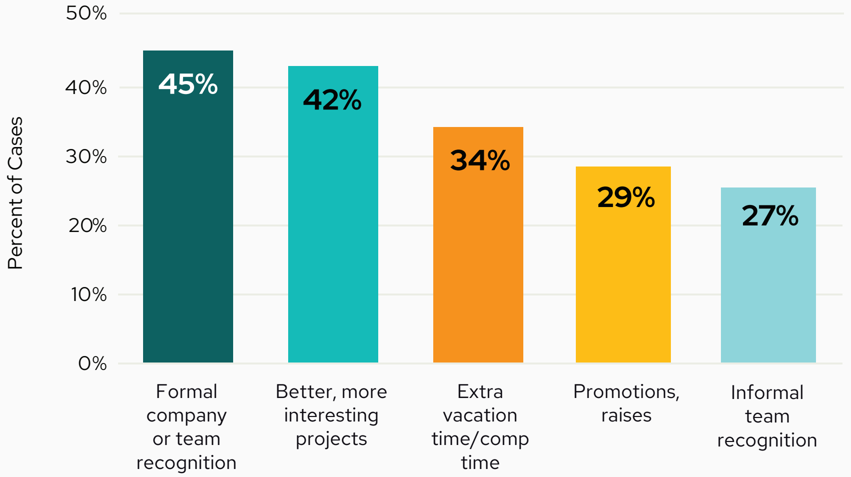


The majority of developers have been positively rewarded for writing secure code, especially in North America. Organizations use motivations such as recognition, better projects, promotions, and in some cases financial rewards to recognize secure coding.

Have you ever been rewarded for completing training in writing secure code?



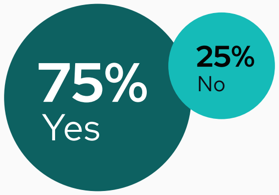
How does your company typically recognize you for your skill (competency) in writing secure code? *(Top 5 of 8)*



Manager expectations

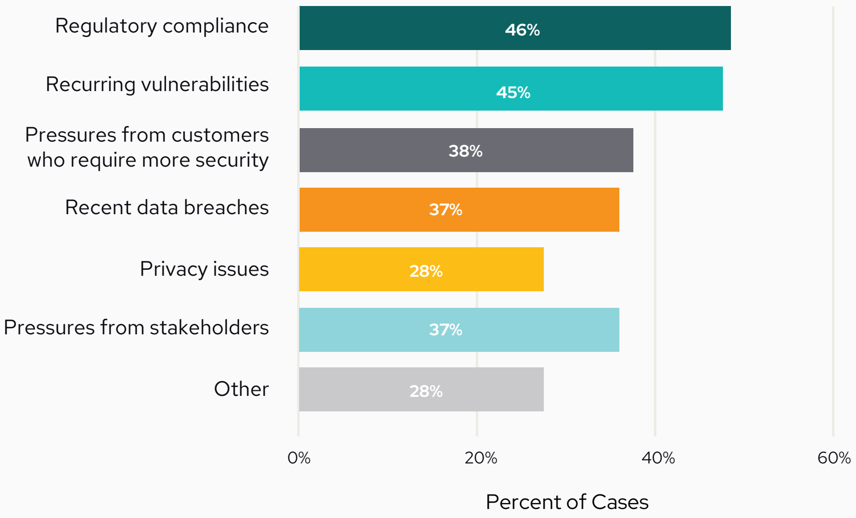
75% of managers are asking developers to learn and/or adopt secure coding practices, with regulatory compliance and recurring vulnerabilities being the top drivers.

Are you asking developers in your team to learn or adopt secure coding practices?



Managers are also assessing secure coding skills of both new hires and within their existing teams through various methods.

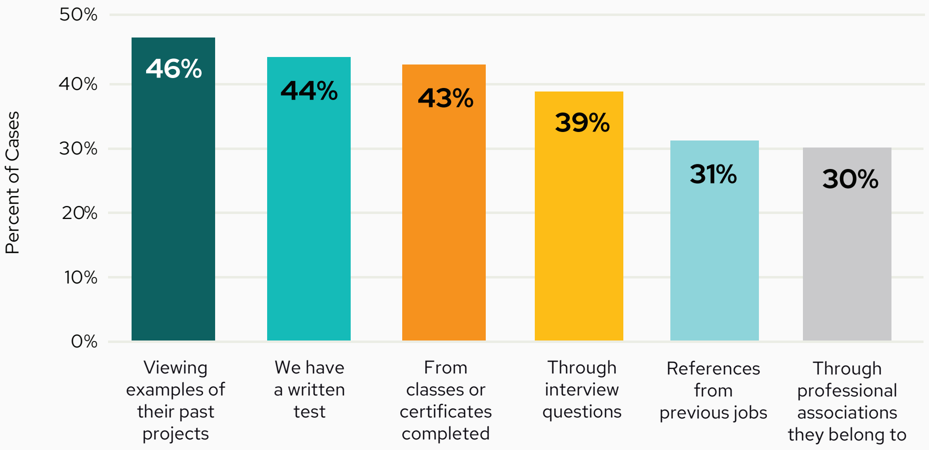
What are the organizational drivers that require you to ensure that your developers have secure code training?



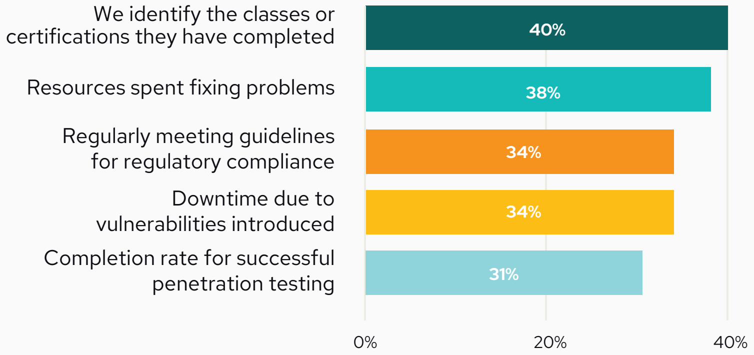
66%

of managers look at secure coding skills when assessing new hires

When hiring developers how do you assess their secure coding skills?



When evaluating your current team members' secure coding skills, how do you assess their competency? *(Top 5 of 8)*



Further reading

This report presents the findings from The State of Developer-Driven Security Survey, 2022. For our analysis and further commentary, including recommendations on what organizations can do to improve security coding practices within their developer teams please read **Whitepaper: The challenges (and opportunities) to improve software security**

Find out more about how we're helping developers ship quality code with confidence at **securecodewarrior.com**

About Secure Code Warrior

Smarter, faster, secure coding. Secure Code Warrior builds a culture of security-driven developers by giving them the skills to code securely. Our flagship Learning Platform delivers relevant skills-based pathways, hands-on missions, and contextual tools for developers to rapidly learn, build, and apply their skills to write secure code at speed.

Established in 2015, Secure Code Warrior has become a critical component for over 450 enterprises including leading financial services, retail and global technology companies across the world.

About Evans Data Corp

Evans Data Corp provides market research for the development community. Our goal is to represent the views, attitudes, desires and opinions of the community of developers to those companies who create devices, tools, operating environments, and other systems that developers use. We strive to help our clients be as successful as possible and to make the right choices regarding strategic direction and tactical product marketing. EDC offers three primary services including Multi-Client Surveys, Custom Surveys, and Custom Data Analytics. For more information, contact Evans Data Corporation at 800-831-3080 or edcsales@evansdata.com.