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.

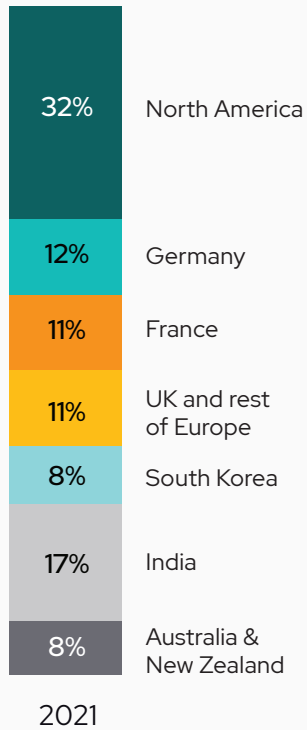But, despite the vast array of security measures adopted by organizations, we continue to feel the repercussions of exploitable software vulnerabilities.

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

32%	North America
12%	Germany
11%	France
11%	UK and rest of Europe
8%	South Korea
17%	India
8%	Australia & New Zealand

2021

What is your role in the development of software?
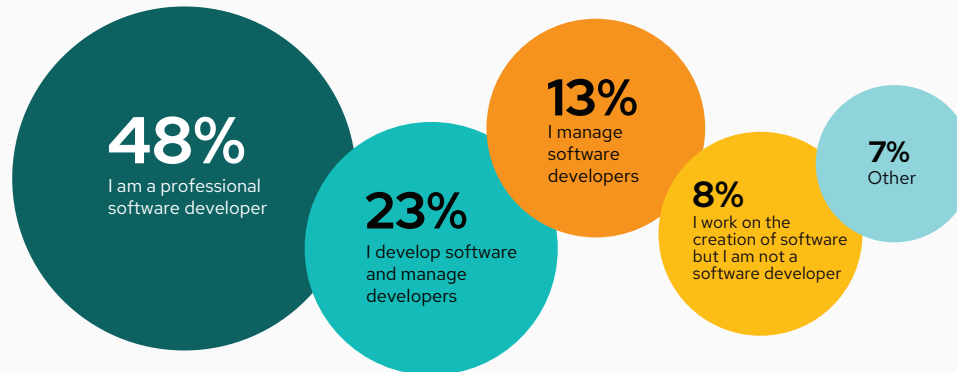
48%
I am a professional software developer

23%
I develop software and manage developers

13%
I manage software developers

8%
I work on the creation of software but I am not a software developer

7%
Other

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

5%	26%	39%	21%	5%	5%
1 – 2	3 – 6	7 – 10	11 – 15	16 – 20	Over 20

SECURE CODE WARRIOR
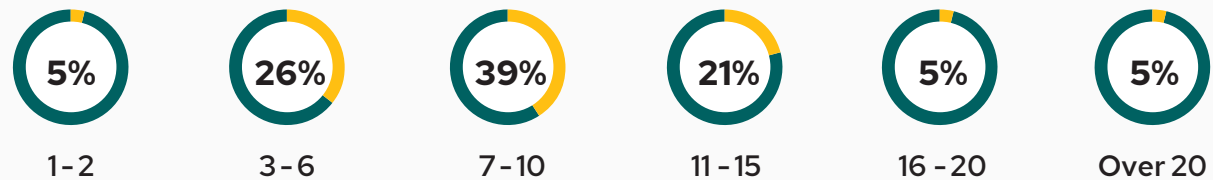
How many years has software been developed in your company?

4% — 2 or fewer

14% — 3 – 5

28% — 6 – 10

39% — 11 – 20

16% — Over 20
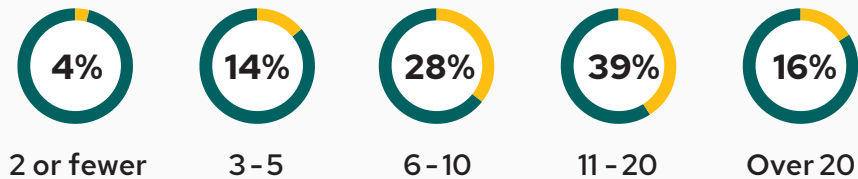
What industry is your company in? *(Top 5 of 23)*

Percent of Programmers

Industry	Percent
Computer software	19%
IT/consulting systems integration	15%
Computer hardware & electronic manufacturing	7%
Information (internet & other media)	7%
Manufacturing (non-computer related)	6%
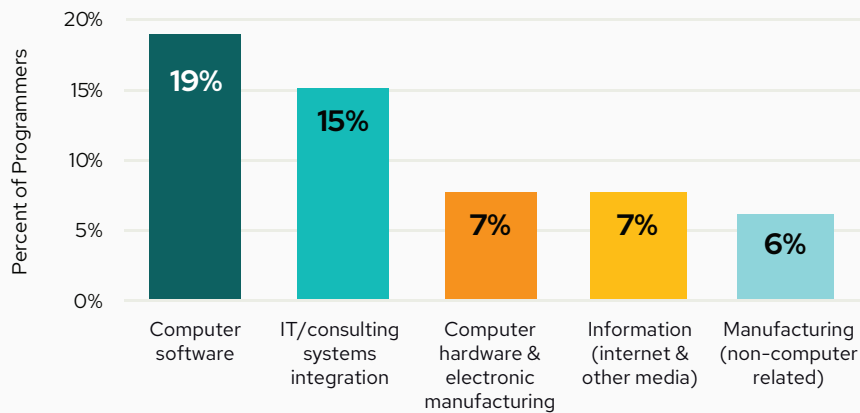
What types of software are you developing?

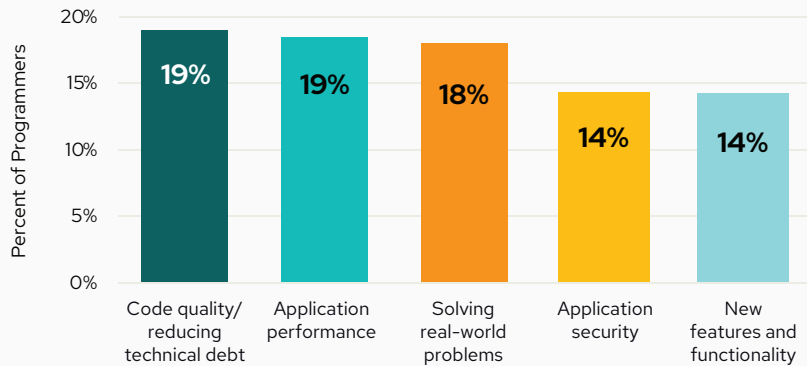| Type	Percent of Cases
Web applications	32%
Cloud-based infrastructure	31%
Data analytics	31%
Client/server applications	29%
Developer tools	29%
Database	28%
Front-end development (Web pages, GUIs)	26%
Back-end (server based code)	26%
Business-to-business/e-commerce	25%
Business-to-consumer/e-commerce	22%
Scientific/engineering applications	22%
Collaboration/groupware	21%
Content management	21%
APIs	19%
Mobile	19%
Utilities	13%
Games	9%
Other	5%

SECURE CODE WARRIOR

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).
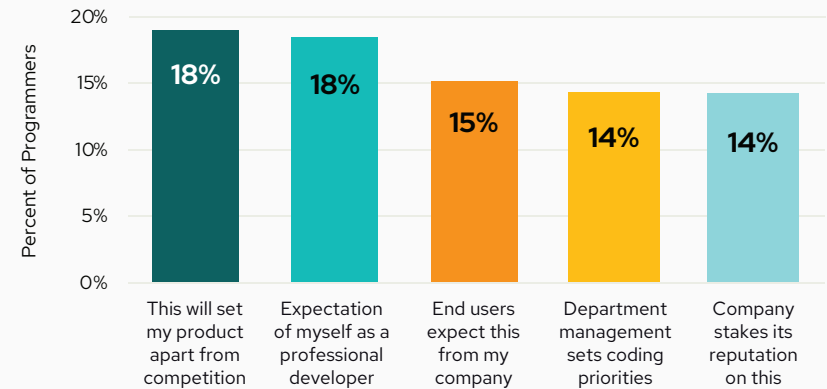
14%

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

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

Category	Percent
Code quality/reducing technical debt	19%
Application performance	19%
Solving real-world problems	18%
Application security	14%
New features and functionality	14%

Percent of Programmers

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

Category	Percent
This will set my product apart from competition	18%
Expectation of myself as a professional developer	18%
End users expect this from my company	15%
Department management sets coding priorities	14%
Company stakes its reputation on this	14%

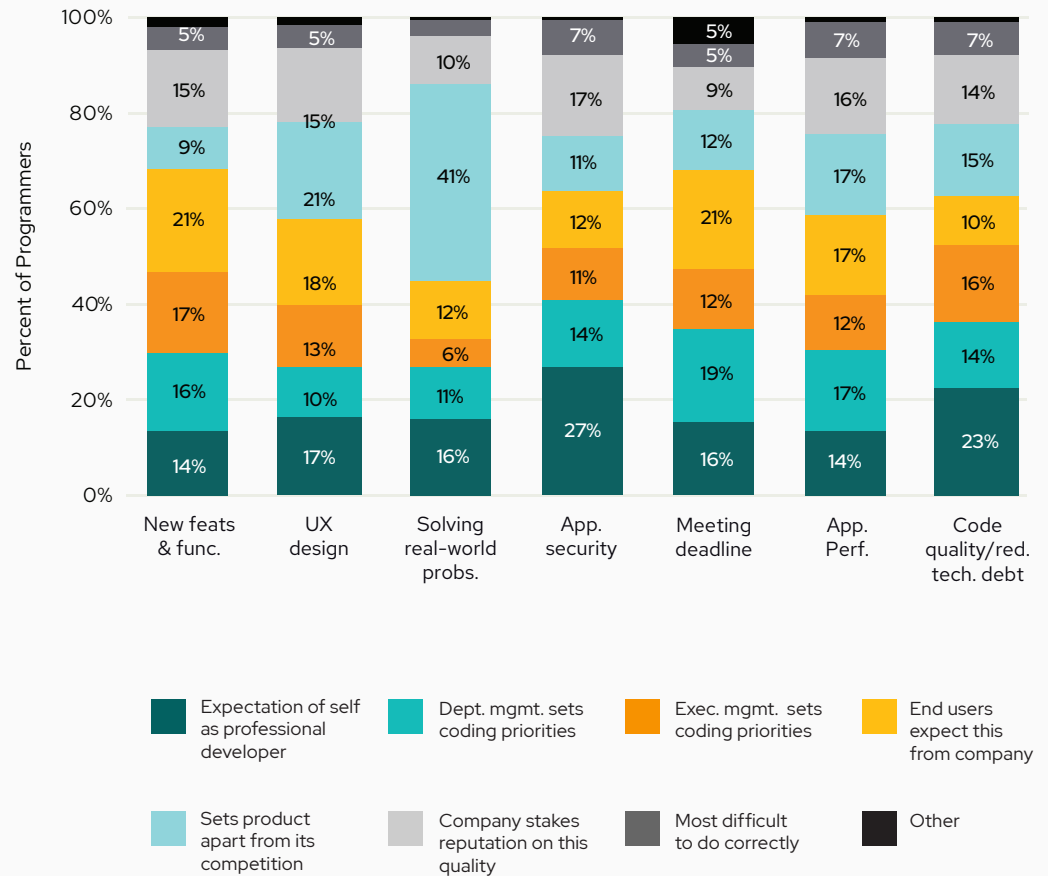*Percent of Programmers*

SECURE CODE WARRIOR

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

Chart: "Reason for top priority by top priority"

Y-axis: Percent of Programmers (0% to 100%)

X-axis categories: New feats & func., UX design, Solving real-world probs., App. security, Meeting deadline, App. Perf., Code quality/red. tech. debt

New feats & func.: 14%, 16%, 17%, 21%, 9%, 15%, 5%

UX design: 17%, 10%, 13%, 18%, 21%, 15%, 5%

Solving real-world probs.: 16%, 11%, 6%, 12%, 41%, 10%

App. security: 27%, 14%, 11%, 12%, 11%, 17%, 7%

Meeting deadline: 16%, 19%, 12%, 21%, 12%, 9%, 5%, 5%

App. Perf.: 14%, 17%, 12%, 17%, 17%, 16%, 7%

Code quality/red. tech. debt: 23%, 14%, 16%, 10%, 15%, 14%, 7%

Legend:
- Expectation of self as professional developer
- Dept. mgmt. sets coding priorities
- Exec. mgmt. sets coding priorities
- End users expect this from company
- Sets product apart from its competition
- Company stakes reputation on this quality
- Most difficult to do correctly
- Other

SECURE CODE WARRIOR

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.
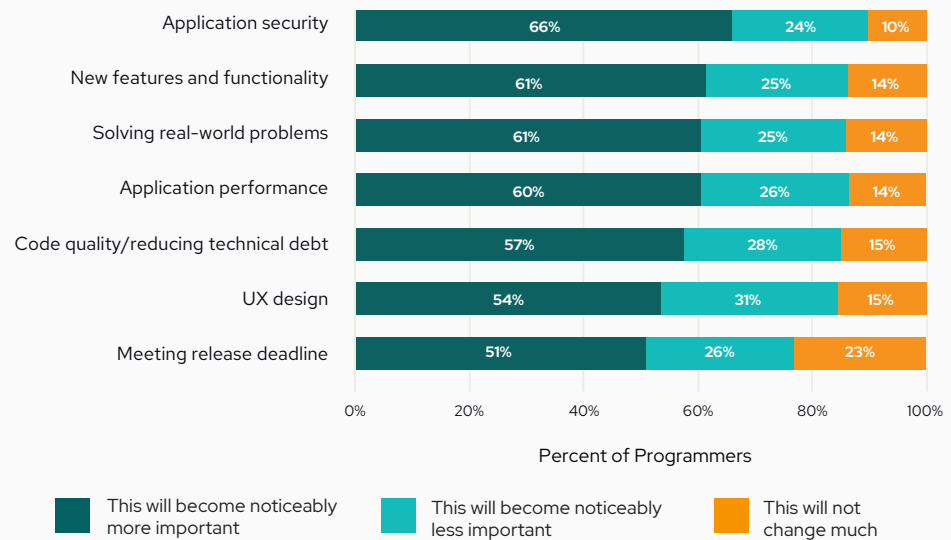
41%

of respondents state that overall, functionality and security have equal importance in their organization
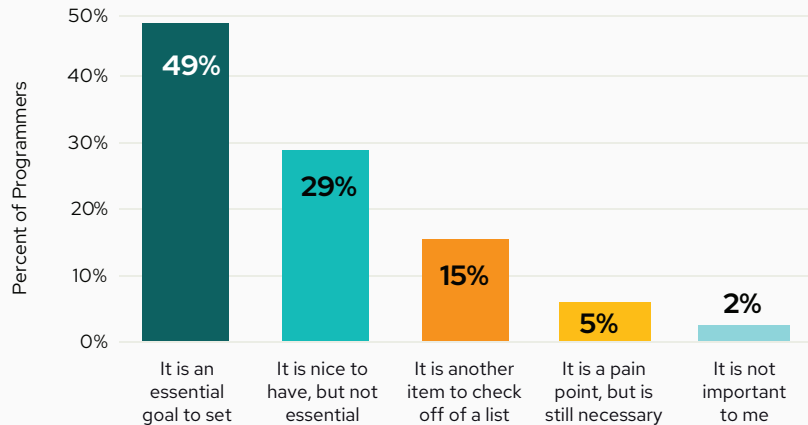
What does your company prioritize?

41%
Functionality and security have equal importance

20%
Functionality is the priority, followed by security

18%
My company prioritizes neither functionality nor security

22%
Security is the priority, followed by functionality

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

	More important	Less important	Not change much
Application security	66%	24%	10%
New features and functionality	61%	25%	14%
Solving real-world problems	61%	25%	14%
Application performance	60%	26%	14%
Code quality/reducing technical debt	57%	28%	15%
UX design	54%	31%	15%
Meeting release deadline	51%	26%	23%

Percent of Programmers

■ This will become noticeably more important
■ This will become noticeably less important
■ This will not change much
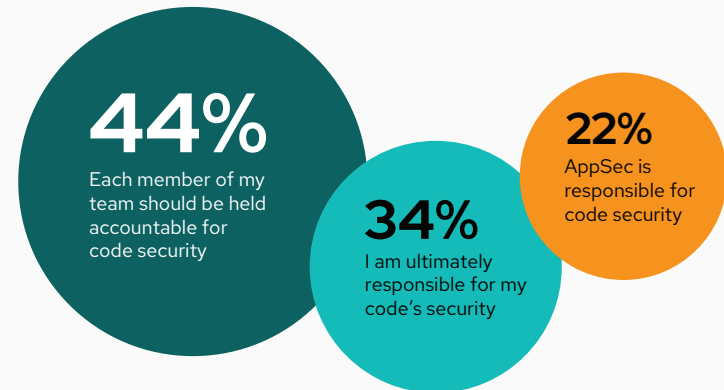
SECURE CODE WARRIOR

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.

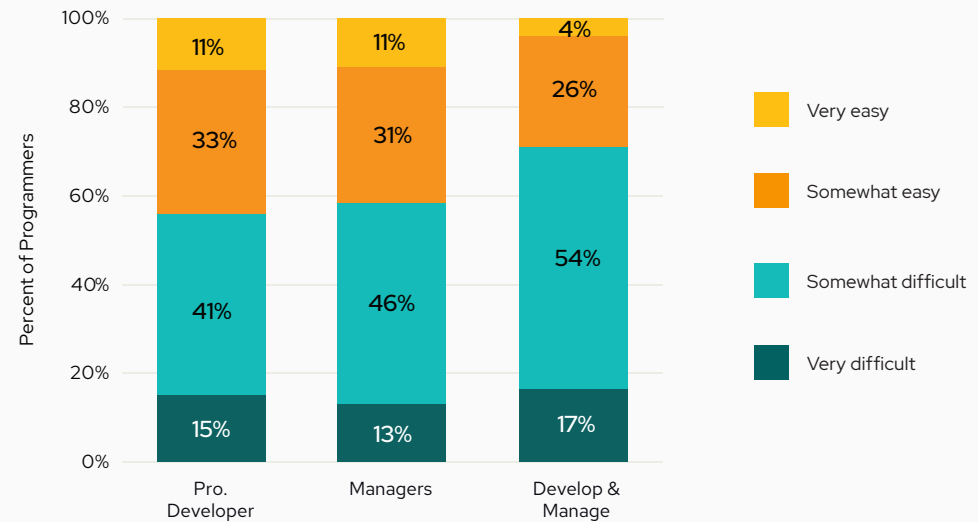## In general, what is your opinion about secure code?

Percent of Programmers

49%	It is an essential goal to set
29%	It is nice to have, but not essential
15%	It is another item to check off of a list
5%	It is a pain point, but is still necessary
2%	It is not important to me

Which of these statements do you most agree with?

44% Each member of my team should be held accountable for code security

34% I am ultimately responsible for my code's security

22% AppSec is responsible for code security

SECURE CODE WARRIOR

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
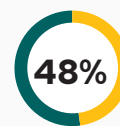
Difficulty of secure coding

	Pro. Developer	Managers	Develop & Manage
Very easy	11%	11%	4%
Somewhat easy	33%	31%	26%
Somewhat difficult	41%	46%	54%
Very difficult	15%	13%	17%

Percent of Programmers

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

15% Very difficult

48% Somewhat difficult

29% Somewhat easy

8% Very easy

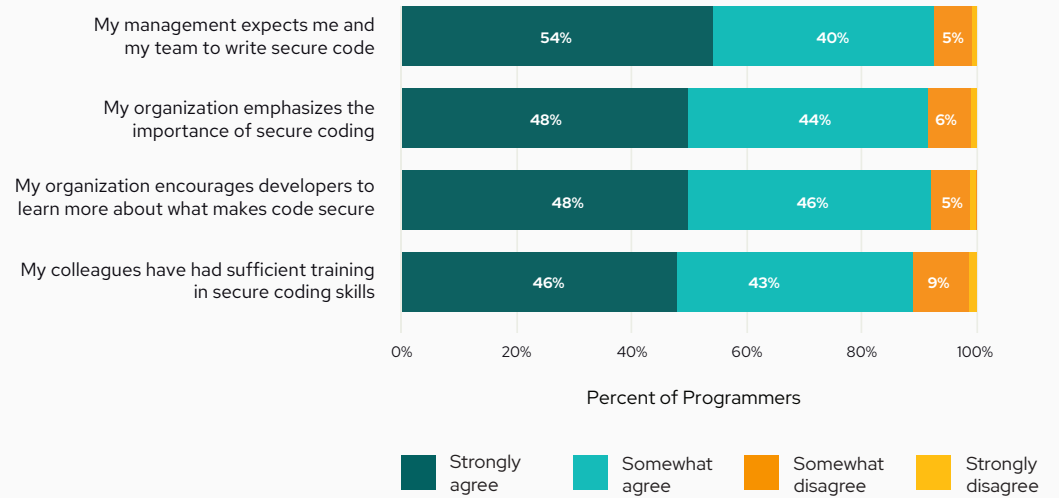SECURE CODE WARRIOR
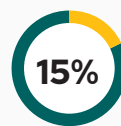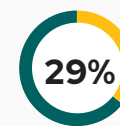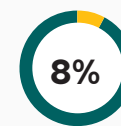
While only 49% of respondents state that secure coding is an essential goal to set, a significantly higher number of respondents agree that secure code is an expectation of their organization and management. And despite 63% of developers stating that the art of writing secure code is difficult, 87% of those surveyed state that they have received sufficient training in secure coding skills.
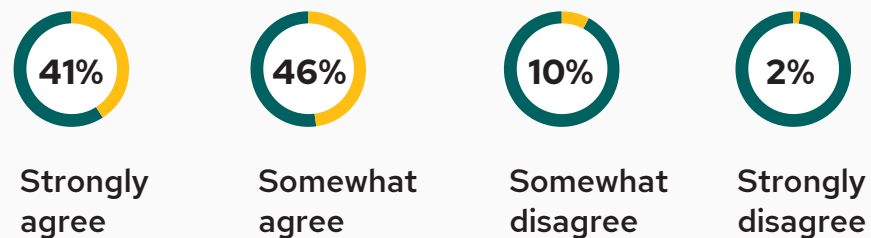
89%

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

Statement	Strongly agree	Somewhat agree	Somewhat disagree	Strongly disagree
My management expects me and my team to write secure code	54%	40%	5%	
My organization emphasizes the importance of secure coding	48%	44%	6%	
My organization encourages developers to learn more about what makes code secure	48%	46%	5%	
My colleagues have had sufficient training in secure coding skills	46%	43%	9%	

Percent of Programmers

■ Strongly agree   ■ Somewhat agree   ■ Somewhat disagree   ■ Strongly disagree

I have had sufficient training in secure coding skills

41% Strongly agree

46% Somewhat agree

10% Somewhat disagree
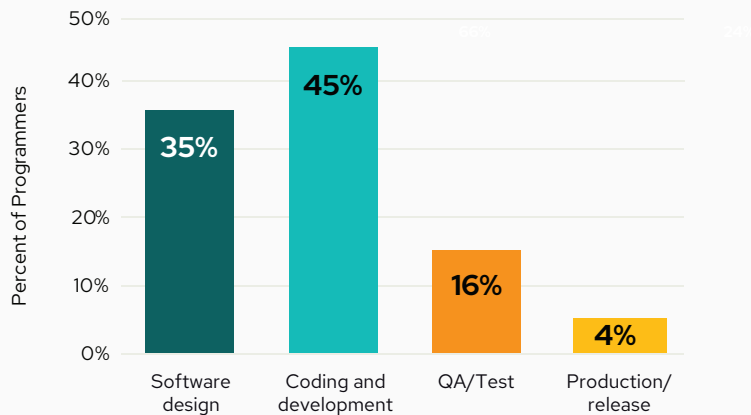
2% Strongly disagree

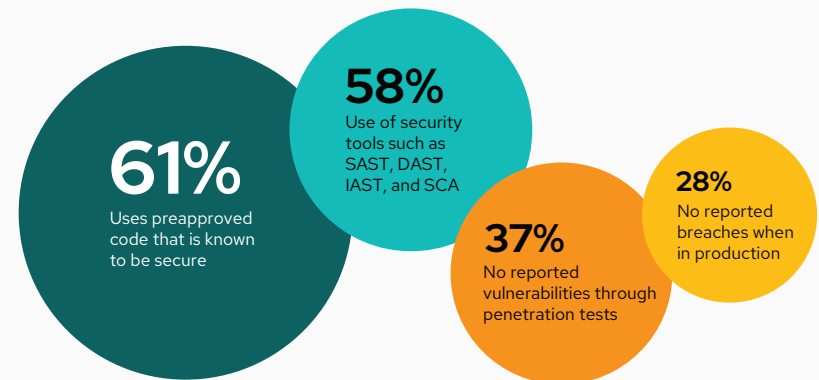SECURE CODE WARRIOR

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

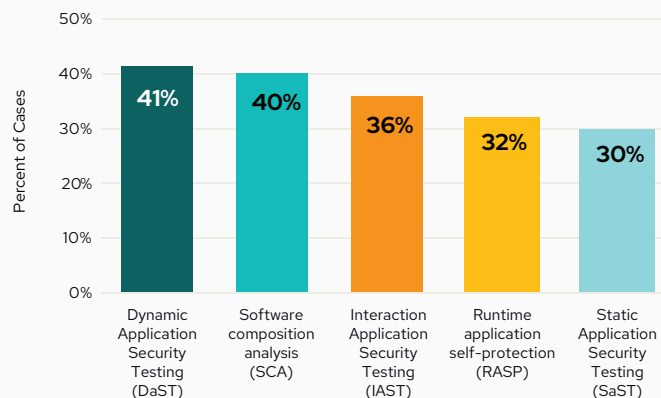## When is secure coding primarily considered in your company's software development lifecycle?

Bar chart — Percent of Programmers:
- Software design: 35%
- Coding and development: 45%
- QA/Test: 16%
- Production/release: 4%

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

- **61%** Uses preapproved code that is known to be secure
- **58%** Use of security tools such as SAST, DAST, IAST, and SCA
- **37%** No reported vulnerabilities through penetration tests
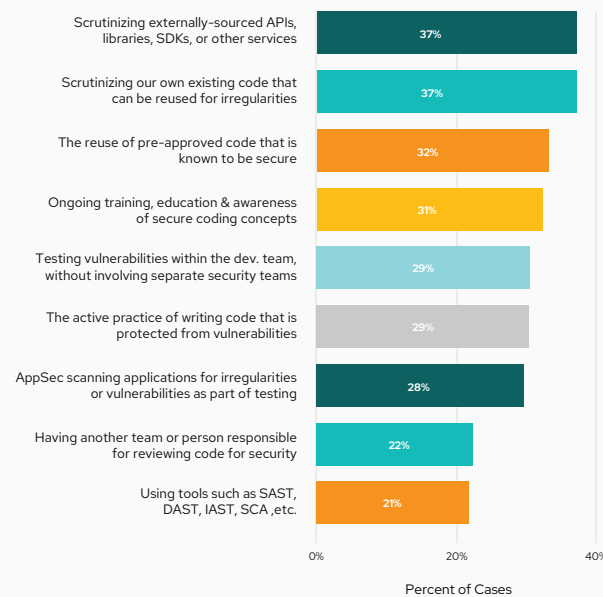- **28%** No reported breaches when in production

SECURE CODE WARRIOR

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)*

Percent of Cases

41%	**40%**	**36%**	**32%**	**30%**

Dynamic Application Security Testing (DaST) · Software composition analysis (SCA) · Interaction Application Security Testing (IAST) · Runtime application self-protection (RASP) · Static Application Security Testing (SaST)

The top practices associated with secure coding

Practice	Percent
Scrutinizing externally-sourced APIs, libraries, SDKs, or other services	37%
Scrutinizing our own existing code that can be reused for irregularities	37%
The reuse of pre-approved code that is known to be secure	32%
Ongoing training, education & awareness of secure coding concepts	31%
Testing vulnerabilities within the dev. team, without involving separate security teams	29%
The active practice of writing code that is protected from vulnerabilities	29%
AppSec scanning applications for irregularities or vulnerabilities as part of testing	28%
Having another team or person responsible for reviewing code for security	22%
Using tools such as SAST, DAST, IAST, SCA ,etc.	21%

Percent of Cases

SECURE CODE WARRIOR

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.

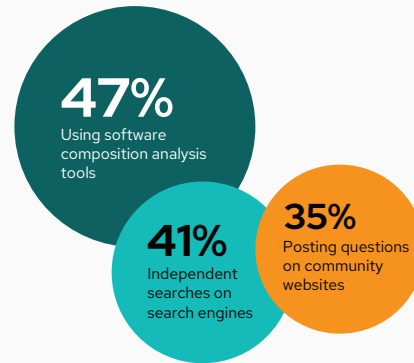## Practices associated with secure code by experience

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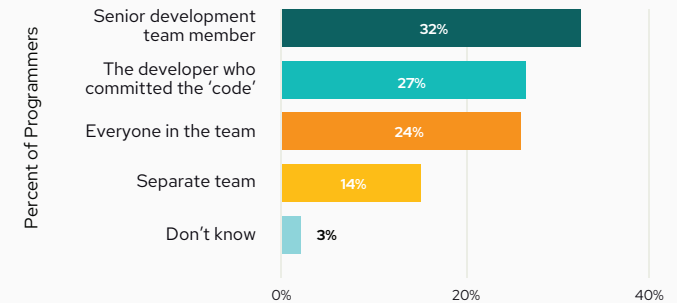
Senior development team members are the most likely to be assigned to fix security tickets, likely due to the advanced skillset necessary for fixing some security issues.

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?

47%
Using software composition analysis tools

41%
Independent searches on search engines

35%
Posting questions on community websites

Who is assigned to fix security tickets?

Percent of Programmers

Senior development team member	32%
The developer who committed the 'code'	27%
Everyone in the team	24%
Separate team	14%
Don't know	3%

0%   20%   40%

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

35%
Excellent

47%
Good

14%
Fair

1%
Poor

3%
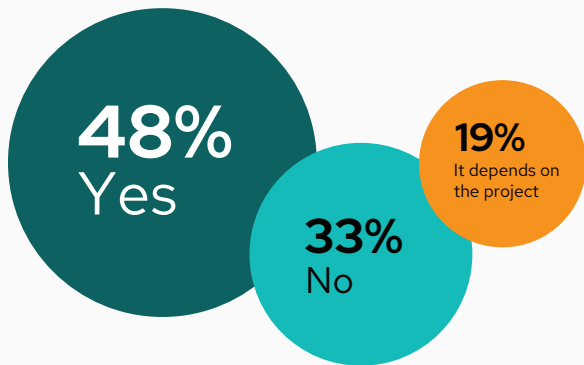Not in a position to assess

SECURE CODE WARRIOR
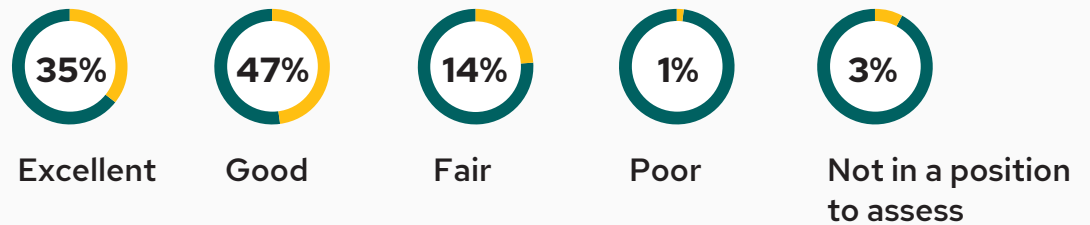
What vulnerabilities?

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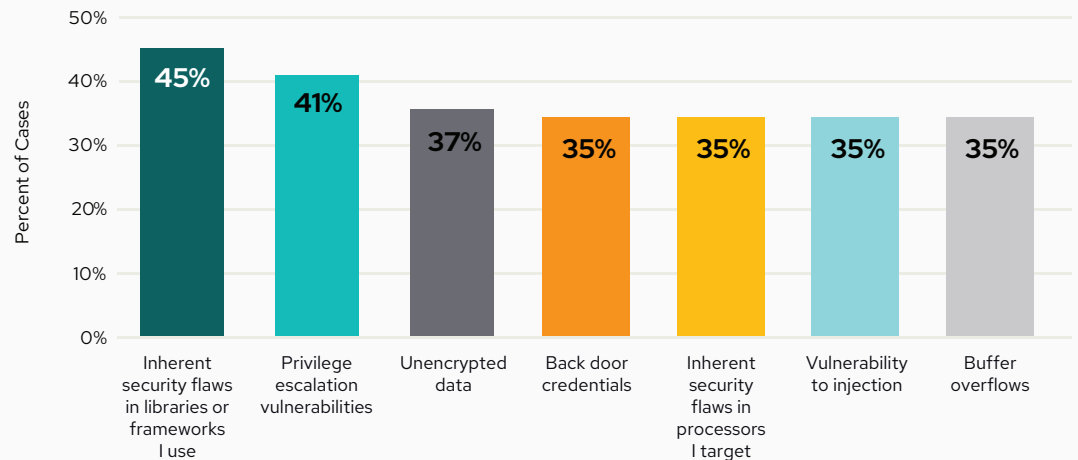
67%

of developers think that they ship code with vulnerabilities

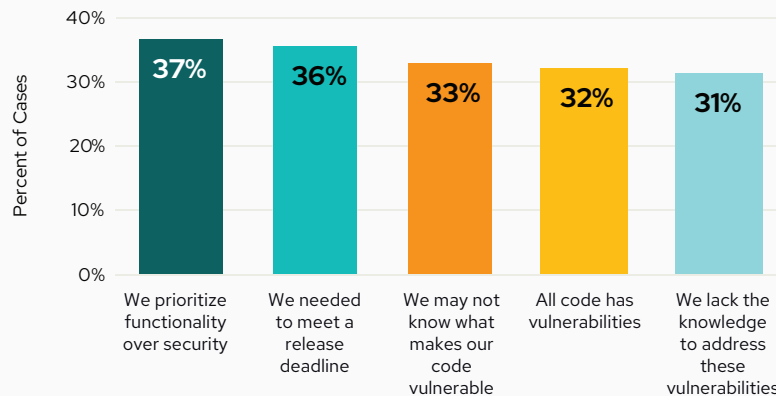## Do you think that you leave vulnerabilities in your code?

48%
Yes

33%
No

19%
It depends on the project

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

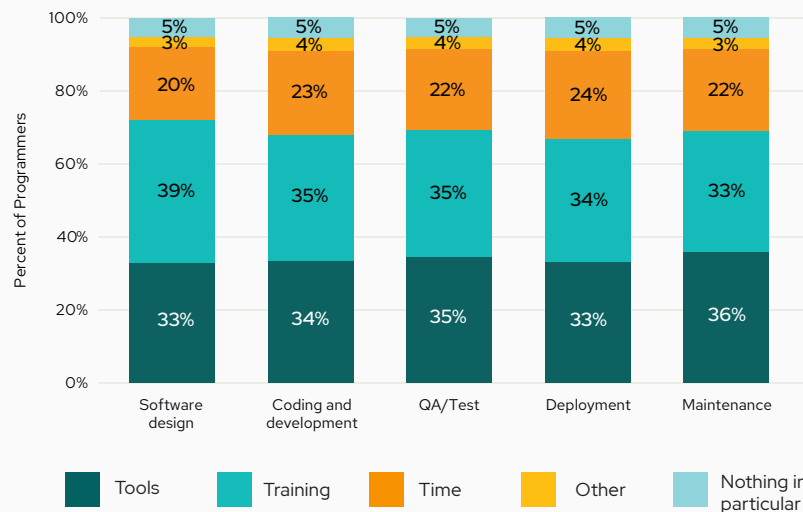| Vulnerability type	Percent
Inherent security flaws in libraries or frameworks I use	45%
Privilege escalation vulnerabilities	41%
Unencrypted data	37%
Back door credentials	35%
Inherent security flaws in processors I target	35%
Vulnerability to injection	35%
Buffer overflows	35%

Percent of Cases

SECURE CODE WARRIOR

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.

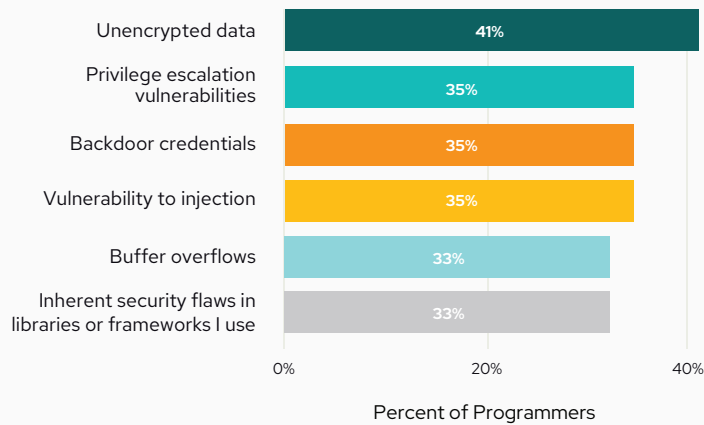## Why do these vulnerabilities exist in your code? *(Top 5 of 7)*

Chart — Percent of Cases:
- We prioritize functionality over security: 37%
- We needed to meet a release deadline: 36%
- We may not know what makes our code vulnerable: 33%
- All code has vulnerabilities: 32%
- We lack the knowledge to address these vulnerabilities: 31%

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

Stacked bar chart — Percent of Programmers:

	Software design	Coding and development	QA/Test	Deployment	Maintenance
Nothing in particular	5%	5%	5%	5%	5%
Other	3%	4%	4%	4%	3%
Time	20%	23%	22%	24%	22%
Training	39%	35%	35%	34%	33%
Tools	33%	34%	35%	33%	36%

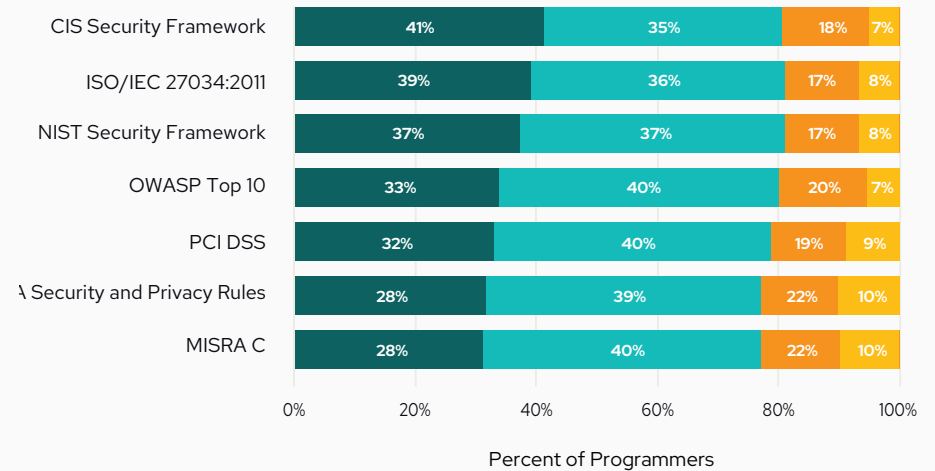Legend: Tools, Training, Time, Other, Nothing in particular

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?

Category	Percent
Unencrypted data	41%
Privilege escalation vulnerabilities	35%
Backdoor credentials	35%
Vulnerability to injection	35%
Buffer overflows	33%
Inherent security flaws in libraries or frameworks I use	33%

Percent of Programmers

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

Framework	I actively use these practices as part of my job	I have knowledge or training in these practices, but do not typically use them	I have heard of this, but don't know much about it	I have not heard of this
CIS Security Framework	41%	35%	18%	7%
ISO/IEC 27034:2011	39%	36%	17%	8%
NIST Security Framework	37%	37%	17%	8%
OWASP Top 10	33%	40%	20%	7%
PCI DSS	32%	40%	19%	9%
Security and Privacy Rules	28%	39%	22%	10%
MISRA C	28%	40%	22%	10%

Percent of Programmers

- I actively use these practices as part of my job
- I have knowledge or training in these practices, but do not typically use them
- I have heard of this, but don't know much about it
- I have not heard of this

SECURE CODE WARRIOR

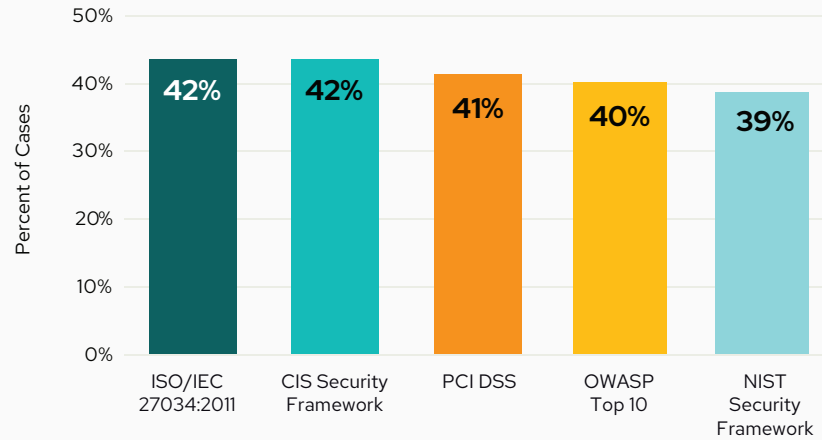**Do the developers on your team require more training in security frameworks?**

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)*

Percent of Cases

ISO/IEC 27034:2011	42%
CIS Security Framework	42%
PCI DSS	41%
OWASP Top 10	40%
NIST Security Framework	39%

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
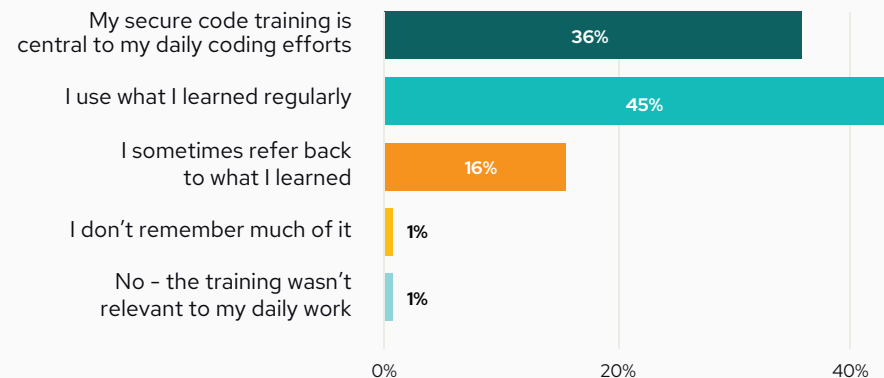
How would you rate your secure code training?

42% Excellent

47% Good

10% Fair

1% Poor
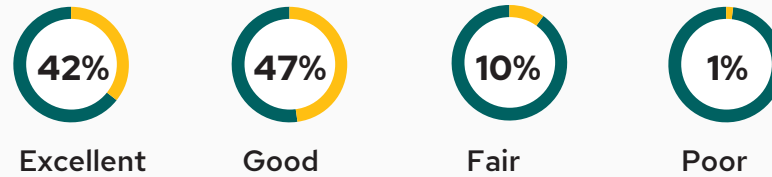
The training I have had in secure coding was relevant to my work

43% Strongly agree

48% Somewhat agree

8% Somewhat disagree

1% Strongly disagree
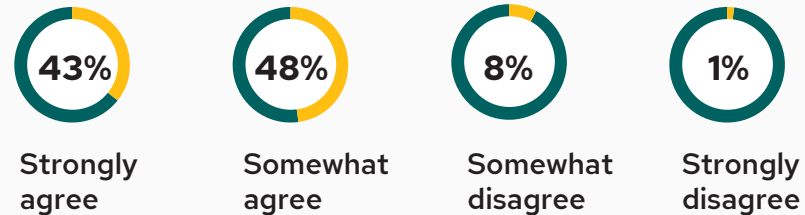
Do you use what you've learned in your secure code training?

|	
My secure code training is central to my daily coding efforts	36%
I use what I learned regularly	45%
I sometimes refer back to what I learned	16%
I don't remember much of it	1%
No – the training wasn't relevant to my daily work	1%

0%   20%   40%

86%

of developers state they find it challenging to practice secure coding

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

0%   **4%**   **10%**   **29%**   **41%**   **16%**

1   2   3   4   5   6

Not at all challenging

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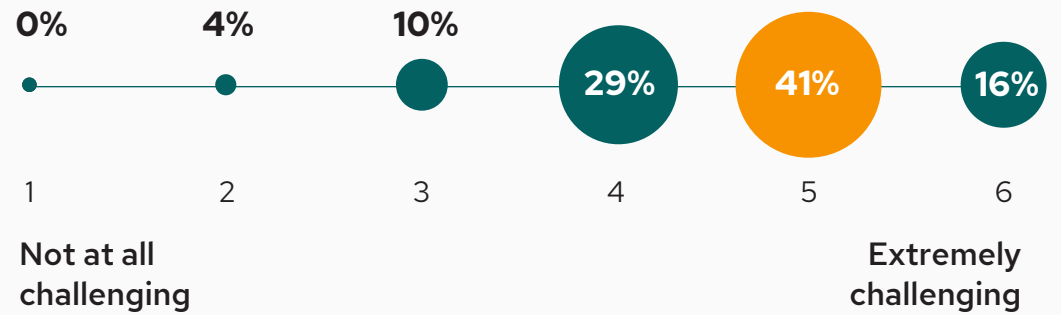
As a manager, how challenging is it to implement secure coding practices in your organization?

0%   **7%**   **20%**   **33%**   **31%**   **9%**

1   2   3   4   5   6

Not at all challenging

Extremely challenging
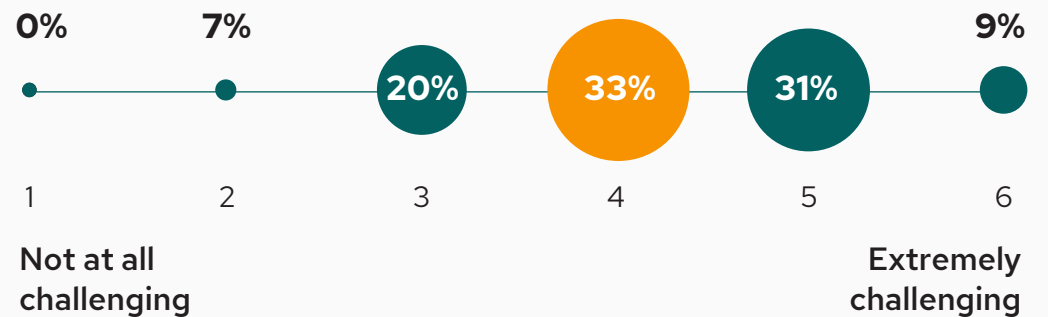
Barriers to adoption

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
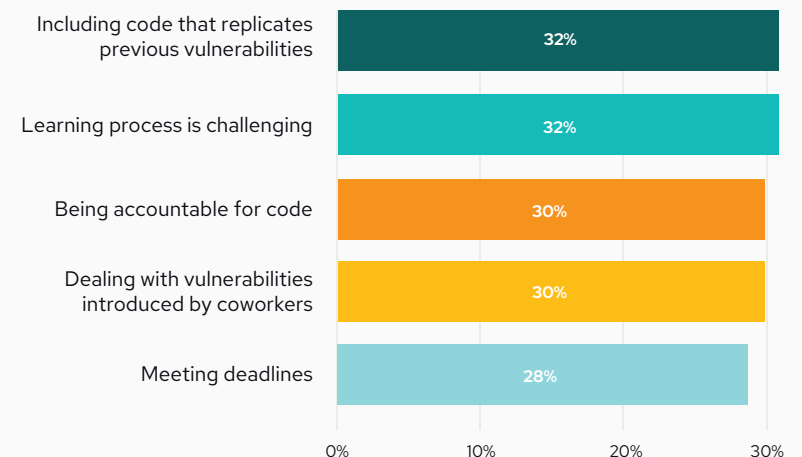
32%

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)*

Percent of Programmers

24%	Not enough time to integrate secure coding earlier
19%	Lack of cohesive plan or approach to implementing secure coding
15%	Lack of prioritization from management
14%	Lack of interest from management
12%	Lack of secure coding skills
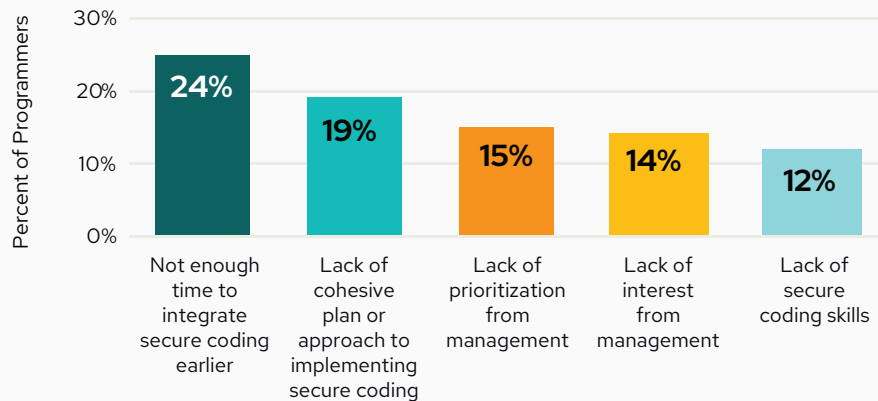
Top concerns with regards to implementation and practice of secure coding *(Top 5 of 11)*

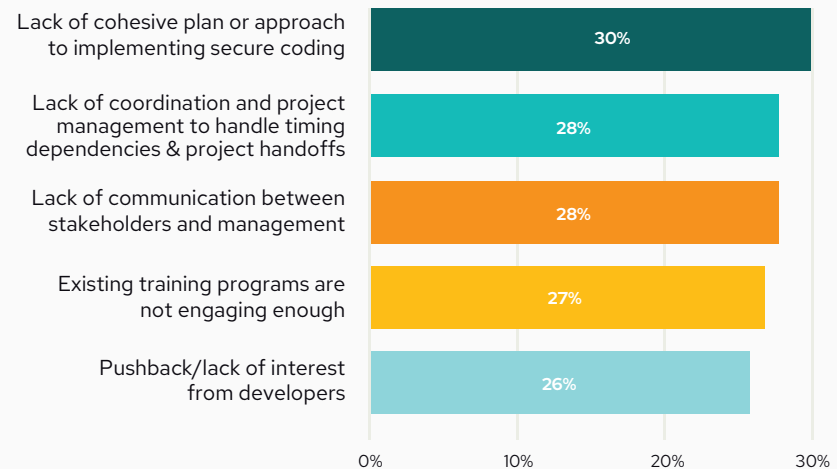|	
Including code that replicates previous vulnerabilities	32%
Learning process is challenging	32%
Being accountable for code	30%
Dealing with vulnerabilities introduced by coworkers	30%
Meeting deadlines	28%

SECURE CODE WARRIOR

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?

52%
Less than half of my projects

29%
More than half of my projects

4%
All of my projects
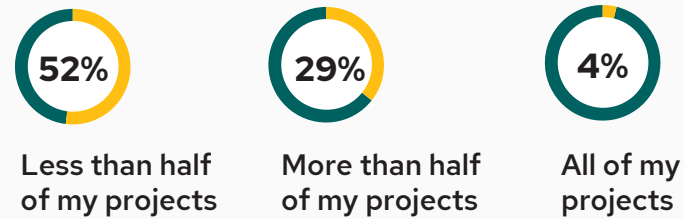
From a management perspective, what obstacles have prevented you from adopting secure code practices? *(Top 5 of 11)*

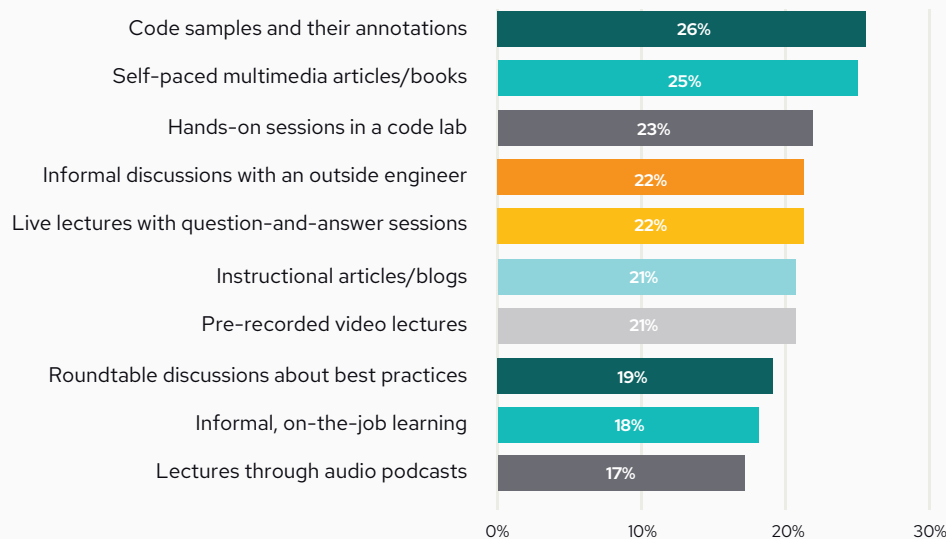| Obstacle	Percentage
Lack of cohesive plan or approach to implementing secure coding	30%
Lack of coordination and project management to handle timing dependencies & project handoffs	28%
Lack of communication between stakeholders and management	28%
Existing training programs are not engaging enough	27%
Pushback/lack of interest from developers	26%

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.

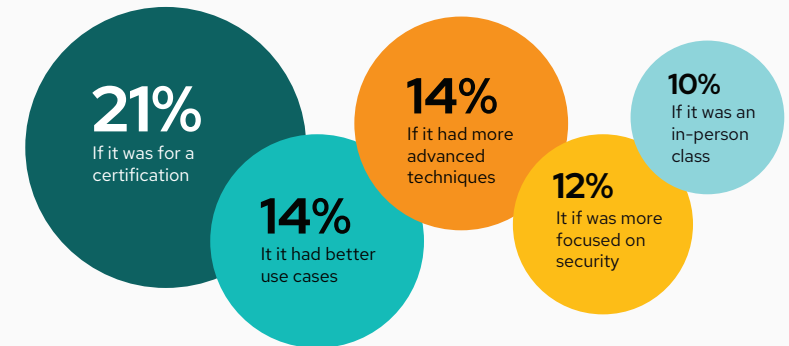Developers want (and need) hands-on, interactive and contextual learning that is recognized

How would you prefer to train in secure coding practices?
(Top 10 of 13)

Option	Percentage
Code samples and their annotations	26%
Self-paced multimedia articles/books	25%
Hands-on sessions in a code lab	23%
Informal discussions with an outside engineer	22%
Live lectures with question-and-answer sessions	22%
Instructional articles/blogs	21%
Pre-recorded video lectures	21%
Roundtable discussions about best practices	19%
Informal, on-the-job learning	18%
Lectures through audio podcasts	17%

How could your secure code training be improved?
(Top 5 of 9)

21% If it was for a certification

14% It it had better use cases

14% If it had more advanced techniques

12% It if was more focused on security

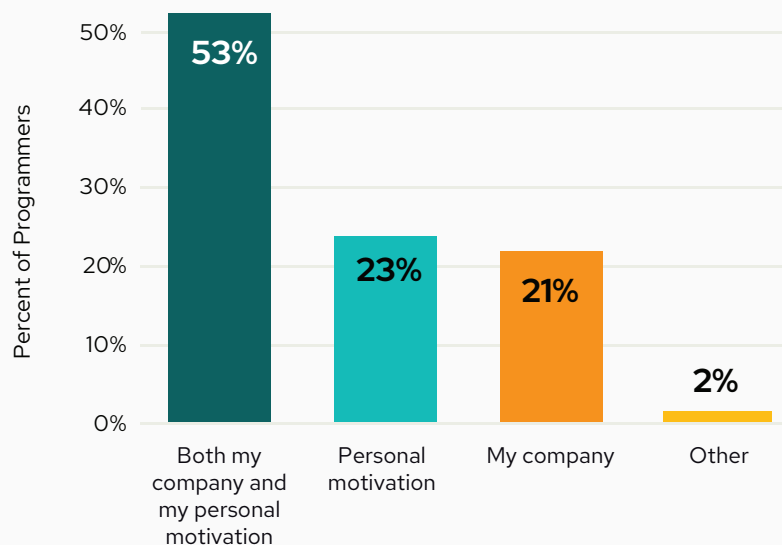**10%** If it was an in-person class

SECURE CODE WARRIOR

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

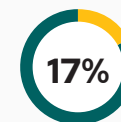## What is driving your interest to study secure coding practices?

Percent of Programmers

53%	23%	21%	2%
Both my company and my personal motivation	Personal motivation	My company	Other

What best describes your personal motivation for learning how to use secure coding practices? *(Top 3 of 6)*
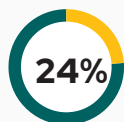
24%
Desire to create top quality code
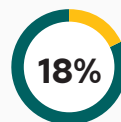
18%
Potential career advancement

17%
Company requires secure coding

SECURE CODE WARRIOR

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.
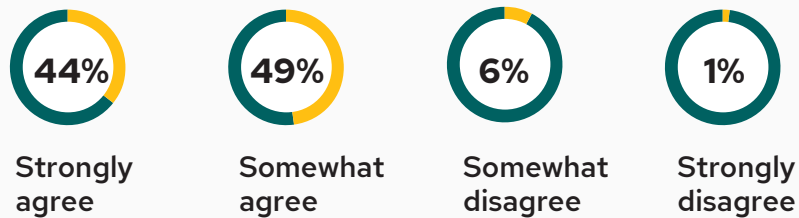
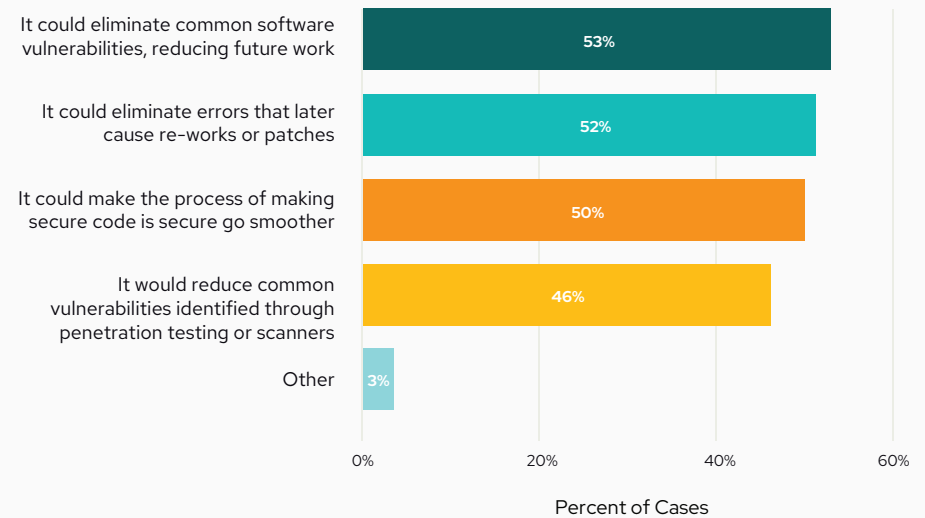**76%** of respondents agreed that good training in secure coding would improve productivity

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

82% Yes

18% No
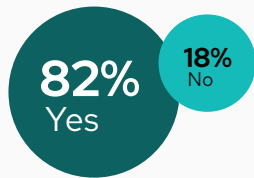
The training I have had in secure coding has been valuable to my career

44% Strongly agree

49% Somewhat agree

6% Somewhat disagree

1% Strongly disagree

In what way would secure code training MOST improve productivity?

|	Percent of Cases
It could eliminate common software vulnerabilities, reducing future work	53%
It could eliminate errors that later cause re-works or patches	52%
It could make the process of making secure code is secure go smoother	50%
It would reduce common vulnerabilities identified through penetration testing or scanners	46%
Other	3%

Percent of Cases

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.
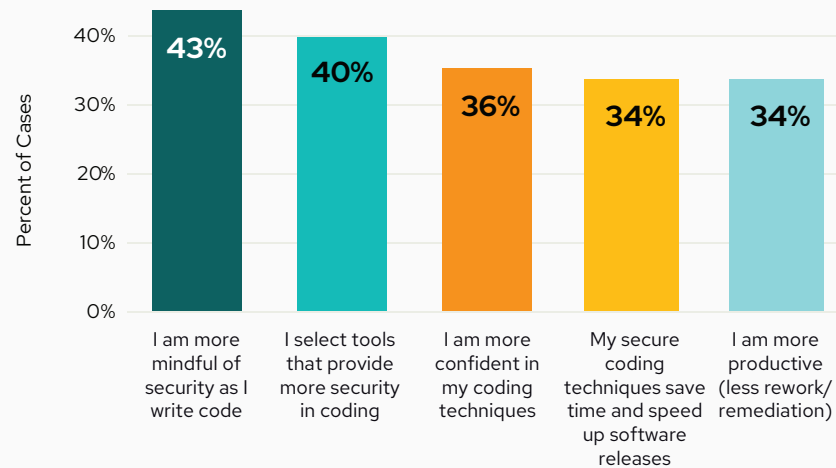
47%

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

47%
Strongly agree

44%
Somewhat agree

7%
Somewhat disagree

1%
Strongly disagree
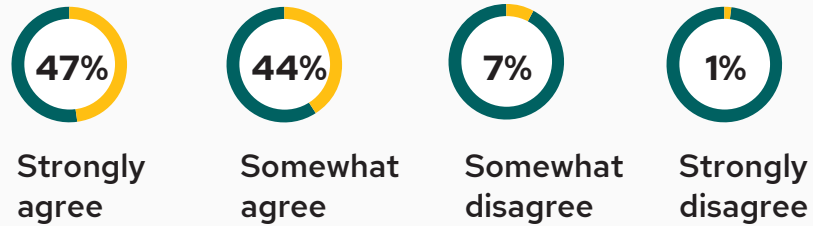
In what ways has being trained in secure coding changed the way you write code? *(Top 5 of 7)*

Percent of Cases

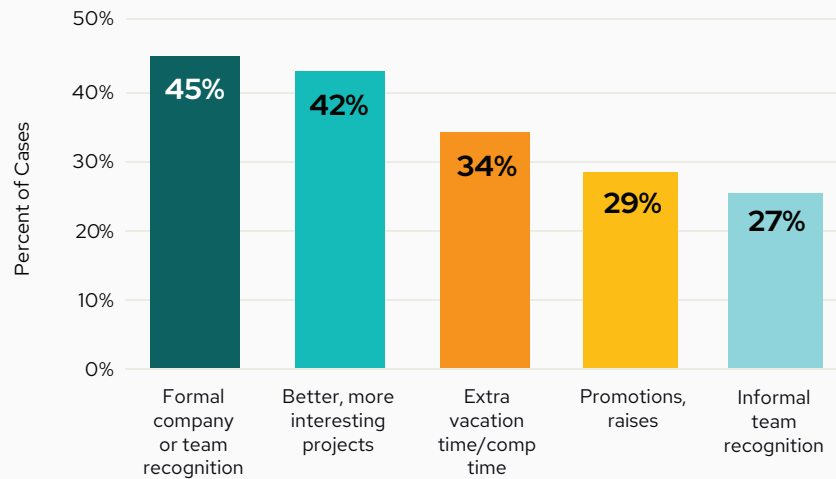|				
43%	**40%**	**36%**	**34%**	**34%**
I am more mindful of security as I write code	I select tools that provide more security in coding	I am more confident in my coding techniques	My secure coding techniques save time and speed up software releases	I am more productive (less rework/ remediation)

SECURE CODE WARRIOR

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?

68% Yes

29% No

3% I am not sure
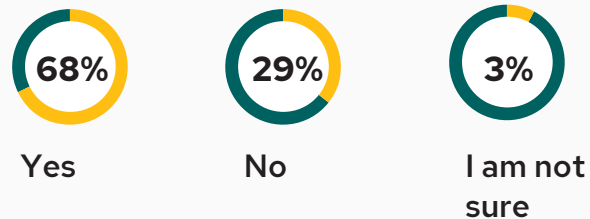
How does your company typically recognize you for your skill (competency) in writing secure code? *(Top 5 of 8)*

Percent of Cases

- Formal company or team recognition: **45%**
- Better, more interesting projects: **42%**
- Extra vacation time/comp time: **34%**
- Promotions, raises: **29%**
- Informal team recognition: **27%**
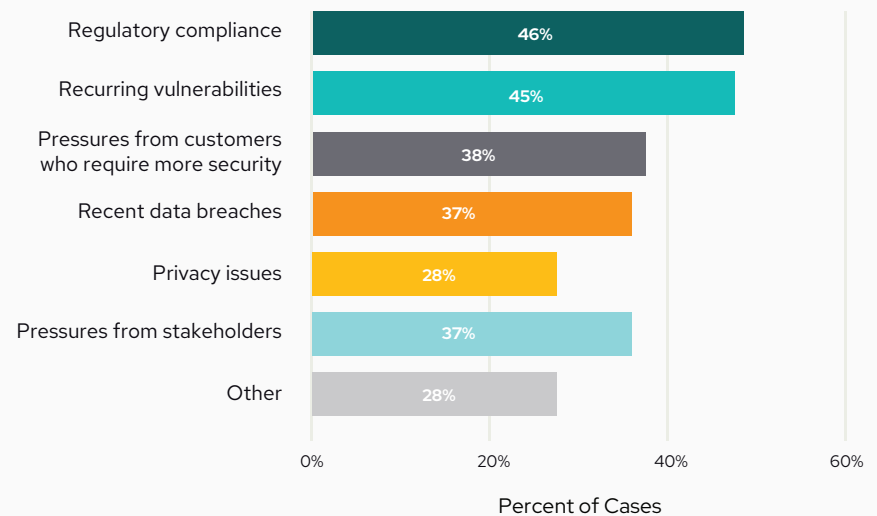
Manager expectations

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?

75% Yes

25% No
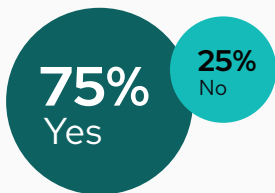
Managers are also assessing secure coding skills of both new hires and within their existing teams through various methods.

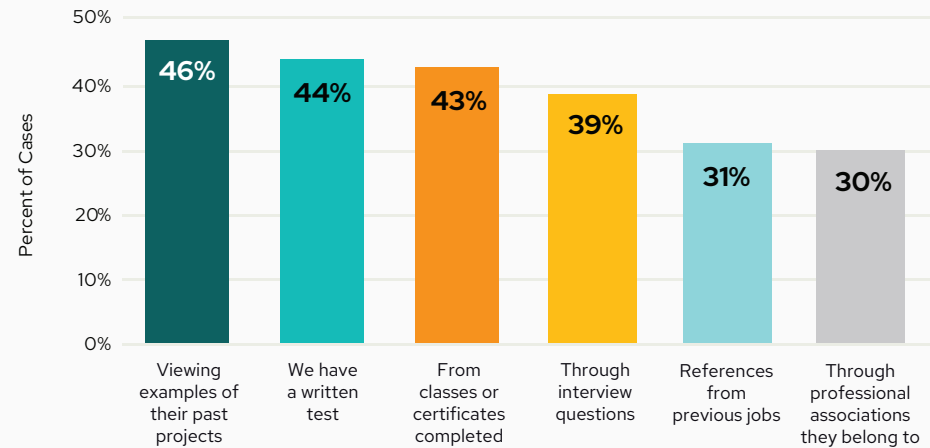## What are the organizational drivers that require you to ensure that your developers have secure code training?

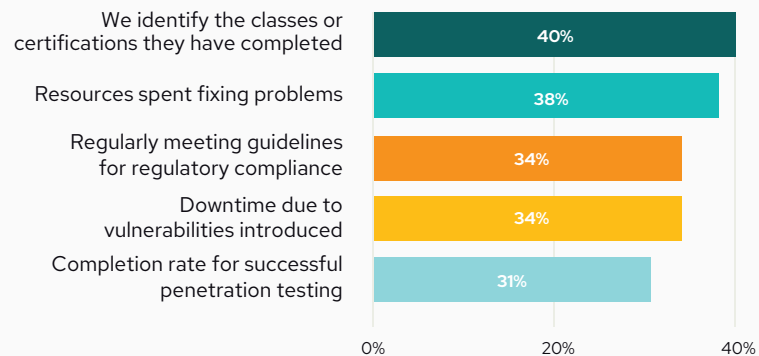| Driver	Percent
Regulatory compliance	46%
Recurring vulnerabilities	45%
Pressures from customers who require more security	38%
Recent data breaches	37%
Privacy issues	28%
Pressures from stakeholders	37%
Other	28%

Percent of Cases

SECURE CODE WARRIOR

66%

of managers look at secure coding skills when assessing new hires

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

Percent of Cases

- Viewing examples of their past projects: 46%
- We have a written test: 44%
- From classes or certificates completed: 43%
- Through interview questions: 39%
- References from previous jobs: 31%
- Through professional associations they belong to: 30%

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

- We identify the classes or certifications they have completed: 40%
- Resources spent fixing problems: 38%
- Regularly meeting guidelines for regulatory compliance: 34%
- Downtime due to vulnerabilities introduced: 34%
- Completion rate for successful penetration testing: 31%

SECURE CODE WARRIOR

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.