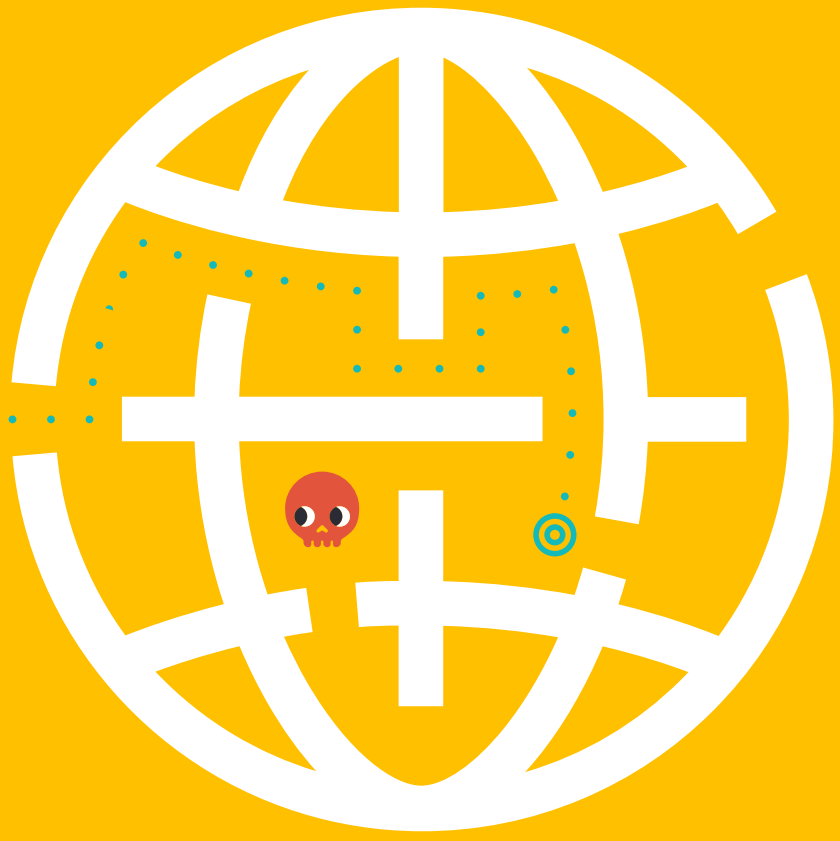
WHITEPAPER

The challenges *(and opportunities)* to improve software security



SECURE CODE WARRIOR

securecodewarrior.com

Executive summary

The increasingly complicated threat landscape is challenging organizations of all sizes to improve the security of their software and applications. One opportunity is to integrate security throughout the entire software development process.

For too many years, security has been an afterthought to the software development process. Developers were told that their primary role was to quickly build and deploy apps into a fast-paced environment where business never stops and customers never sleep. Those who could code faster, delivering innovative features and functionality were seen as stronger performers.

Meanwhile, the threat landscape has evolved to become increasingly hazardous and complex. The latest Verizon Data Breach Investigations Report makes it very clear that the threats arrayed against businesses and organizations are more dangerous and expensive today than at any other point in history.

According to the recent Cost of a Data Breach Report from IBM and the Ponemon Institute, the price that firms pay to clean up after a cybersecurity breach now tops $3.8 million per incident. And that is just the monetary costs.

Insecure code also costs thousands of hours per year in lost productivity as apps and programs are reworked to fix problems or add security. That also keeps development teams from operating at peak efficiency, slowing production at a time when demand is surging.

Movements are underway to shift more security responsibilities to earlier in the process of software development, with the ultimate goal of being able to write secure code that is free from vulnerabilities or potential exploitation right from the start. Often cited as "shifting left," this effort first gave rise to agile development and DevOps, and later to the entire DevSecOps movement, where security is a shared responsibility for everyone involved in the process of creating software from development to deployment.

The one thing that all of these efforts have in common is an evolving reliance on the developer community to help drive these much-needed changes.

From a developer's point of view, these security movements are more about "starting left" rather than shifting towards it, since the ultimate responsibility to begin the process correctly should start with them.

We hear these challenges from customers and prospects everyday at Secure Code Warrior. So we wanted to delve deeper into the developer community's attitudes, skills, and perceptions when it comes to secure coding, and how efforts were being prioritized and supported by their management and organizations. To answer those questions, we conducted a highly detailed survey to explore **The State of Developer-Driven Security**, and its related opportunities and challenges.

About the survey

The Secure Code Warrior 'State of Developer-Driven Security' survey was conducted in partnership with 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 as 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.

Here's what we found...

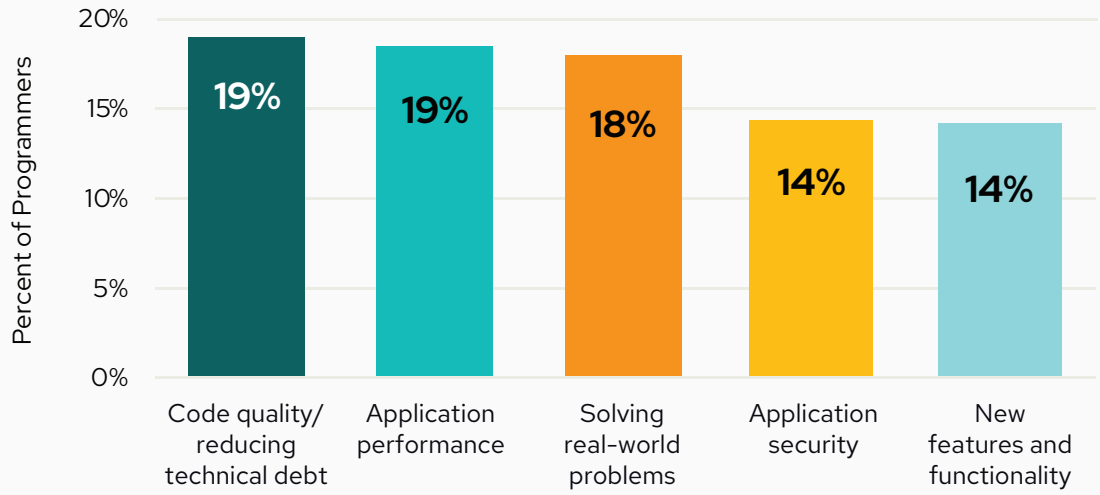
Writing secure code is not *(yet)* a top priority among developers

Only having 14% of developers put secure code as their top priority is not surprising given that they have not traditionally been graded or reviewed based on security or secure coding ability. However, the development community as a whole recognizes that this is rapidly changing. When asked what elements about their jobs will be changing over the next 12 to 18 months, the top answer (66%) was an increased emphasis on application security.

14%

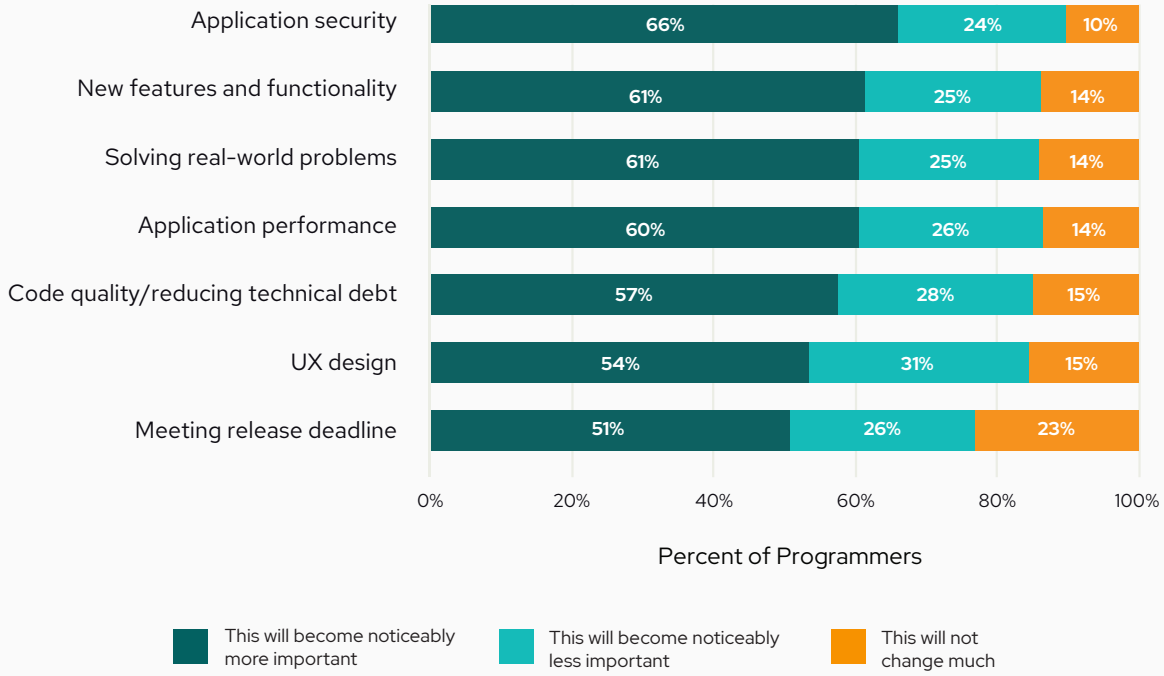
of developers list security as a top priority

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



On the management side, those who hire developers are already shifting their priorities and hiring practices. An average of 82% of managers surveyed said they were much more likely to hire a developer who could demonstrate secure coding skills versus one who could not. There was a little bit of a bias in this when broken down by company size too, with 90% of managers at larger companies looking for developers who can code securely, versus 72% for smaller firms.

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

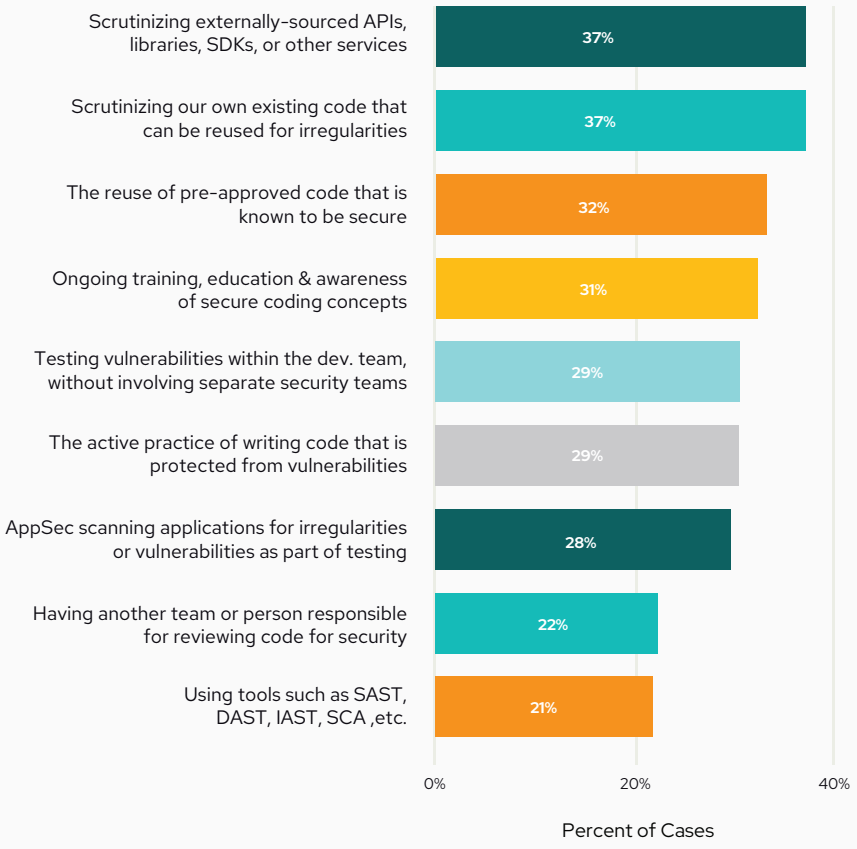


66% of developers state that application security will become more important in the next 12-18 months

Most developers are at the early stages of secure code adoption practices

When developers were asked to select up to three activities that were most helpful in creating secure code, they tended to pick activities that involved examining existing code (37%), using libraries of known safe code (37%), and reusing code that has already been deemed to be secure (32%).

Practices associated with secure coding



Actively writing original secure code themselves came in at 29%, and not thinking about security at all by letting another team or individual handle everything involving security scored 22%.

These responses demonstrate a disconnect between what developers think is a secure coding practice and what actually constitutes the creation of secure code. Many developers believe that pulling code from a supposedly secure library equates to creating secure code, even though today's secure library can become riddled with vulnerabilities over time as new exploits are discovered.

This disconnect could explain why developers feel like they are creating secure code, and yet the number of exploits and breaches continues to grow.

This problem could also be compounded by the fact that most organizations don't properly define secure code for their developers. When asked, 61% of the developers said that their organizations deem an application to be secure if the code to make it is pulled from a supposedly secure library. More disturbingly, 28% of respondents said that their applications are considered secure if no reports of breaches come in after they are deployed into a production environment.

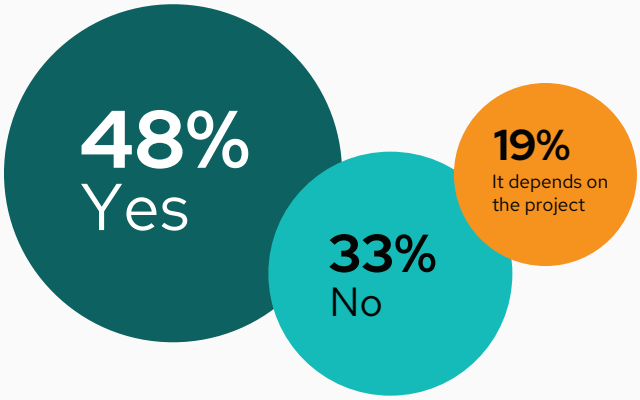
How is the code written within your organization recognized as secure?

28% of respondents said that their applications are considered secure if no reports of breaches come in after they are deployed into a production environment

Developers admit that they sometimes leave known vulnerabilities in their code

While attitudes about secure coding are changing, there is still a long way to go. To put that in perspective, when the survey respondents were asked if they sometimes leave vulnerabilities within their code, a surprisingly high number (67%) admitted that they did.

Do you think that you leave vulnerabilities in your code?



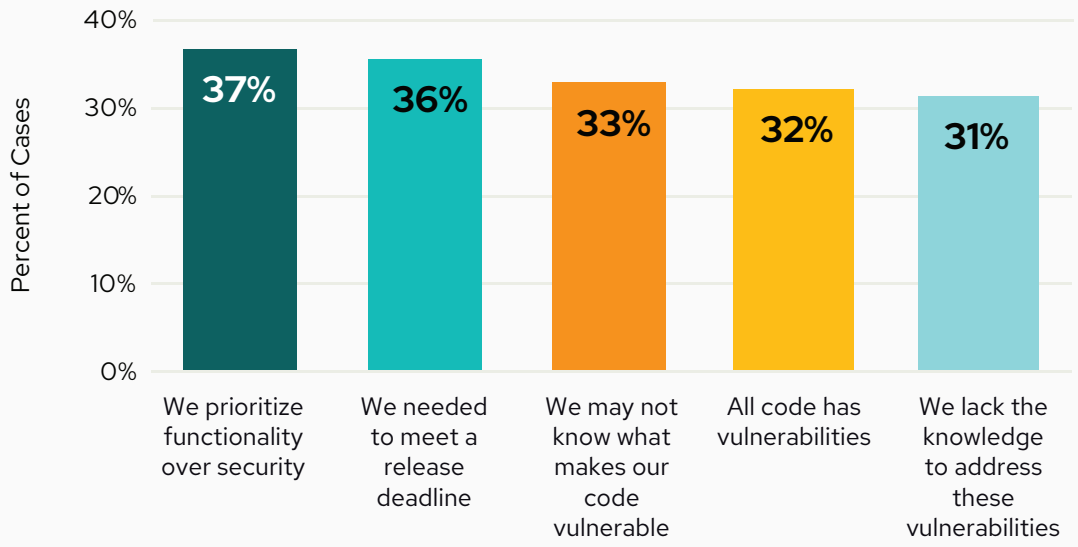
67%

of developers admit to shipping code with vulnerabilities

The blame for deploying insecure code fell in a lot of places, but the top reasons (with developers able to select more than one), were inherent security flaws in the libraries or frameworks being used (45%), prioritizing functionality over security (37%), not being able to code securely given a tight deadline (36%), not understanding how to identify or fix known vulnerabilities (33%), or the fact that finding and fixing insecure code is the responsibility of someone else (25%).

Here again, the disconnect between the perception of secure code and the actual practice of writing it is the crux of the problem.

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



The fact that developers blame vulnerabilities in the libraries that they are using (45%) for later breaches, coupled with the fact that organizations overwhelmingly say code is secure if it's pulled from those libraries (67%) **is highly problematic and constantly repeated across all sectors.**

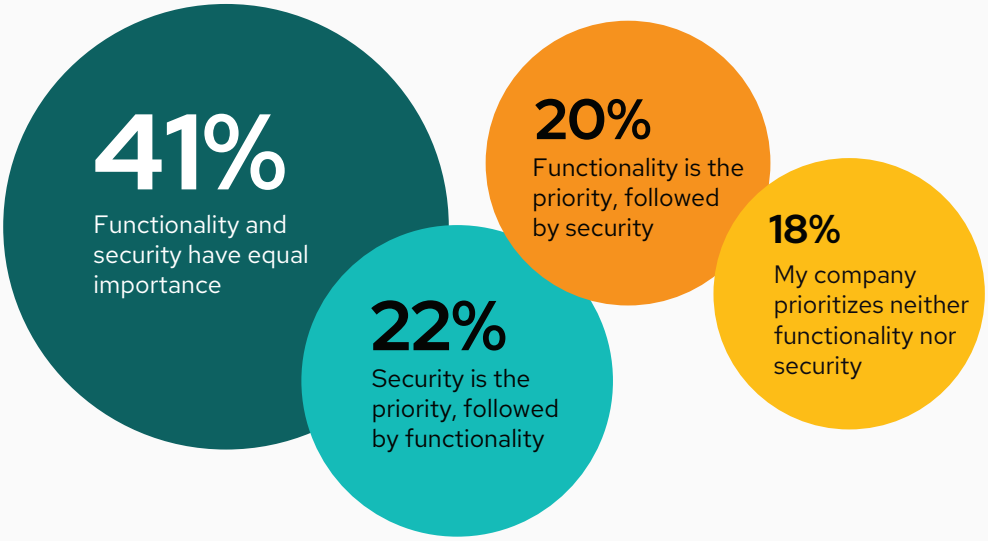
Developer attitudes towards security are improving

It's not all bad news though. Application security has traditionally been looked at as secondary by a lot of organizations, especially when compared with other factors like functionality. However, this is starting to change. In fact, 41% of those surveyed said that functionality and security are now of equal importance at their company, which was the most popular response. Beyond that, 22% said that security had actually overtaken functionality in terms of priority, while 20% said that the functionality of applications remains their organization's top concern.

41%

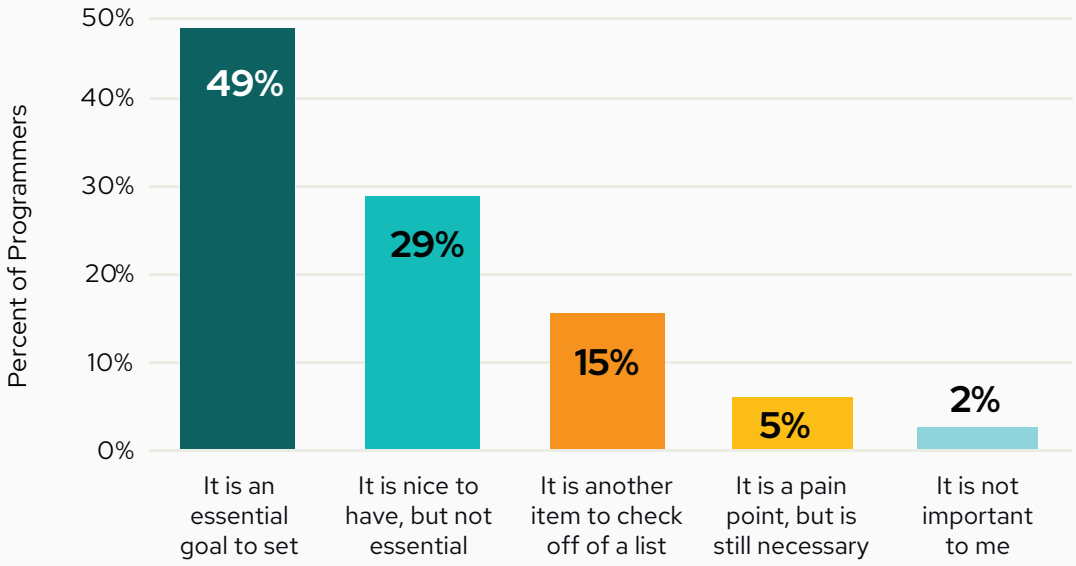
said that functionality and security are now of equal importance at their company

What does your company prioritize?



The developers themselves are also starting to see the importance of secure code regardless of their organization's priorities. When asked about their opinion of secure code, 49% said that writing secure code was an essential goal that needed to be met, while 29% stated that while secure code is not essential, it's a good thing to have. Only 2% of the respondents said that secure coding was not important to them.

In general, what is your opinion about secure code?



49% said that writing secure code was an essential goal that needed to be met

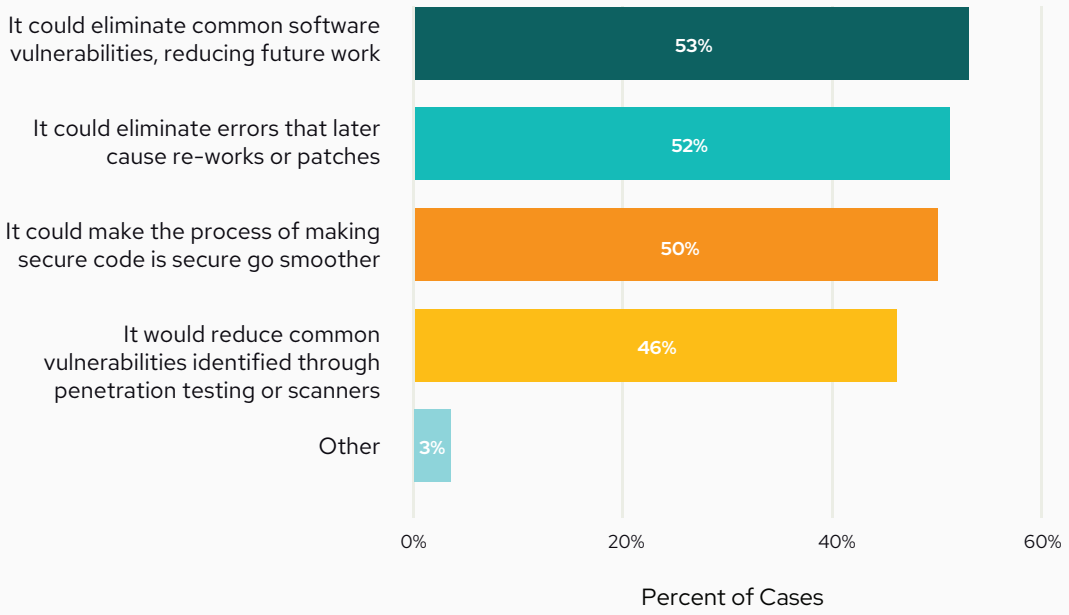
Secure coding practices save time and improve productivity

Both managers and developers agree that secure coding can be a benefit to their organization, with 76% saying that it can or already does improve productivity. The reasons for this were all about equally weighted in the responses, coming in between 46% and 53% when respondents were able to select multiple reasons for the productivity boost.

76%

said that secure coding improves productivity

In what way would secure code training MOST improve productivity?



The top reasons for secure code having a positive impact on productivity included eliminating common software vulnerabilities to reduce reworking old code, eliminating the need to deploy patches, reducing the need for penetration testing and other pre-production activities, and generally making the entire process of fielding secure applications go much more smoothly.

Challenges to secure coding remain a barrier to adoption

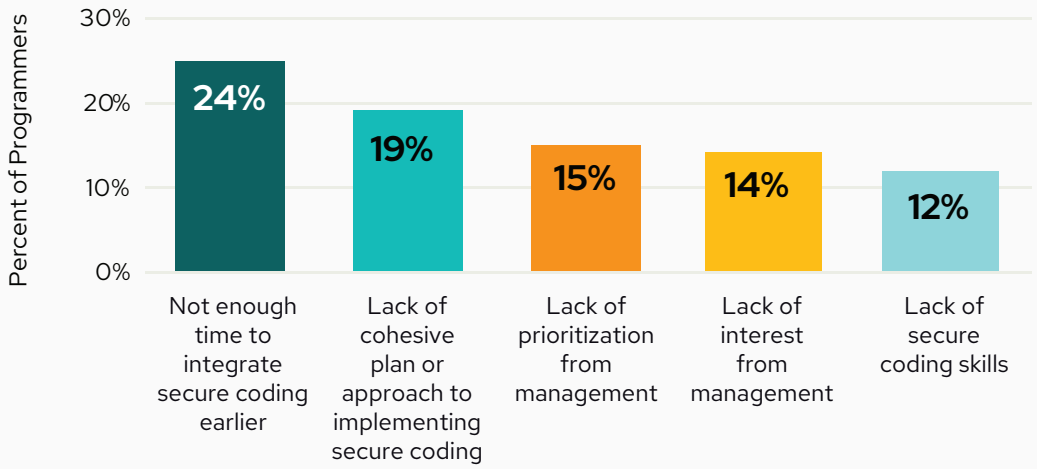
Although attitudes about the value and need for secure code are improving, developers and their organizations are running into significant challenges when trying to adopt it.

Developers said the top barrier to writing secure code was the pressure they were under to meet deadlines not giving them enough time to code securely (24%). Other challenges included not having a detailed plan about how to write secure code at their organization (19%), a lack of interest from management (14%) and not having the skills needed to properly implement secure code (12%).

63%

of developers surveyed stated the practice of writing secure code was difficult

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

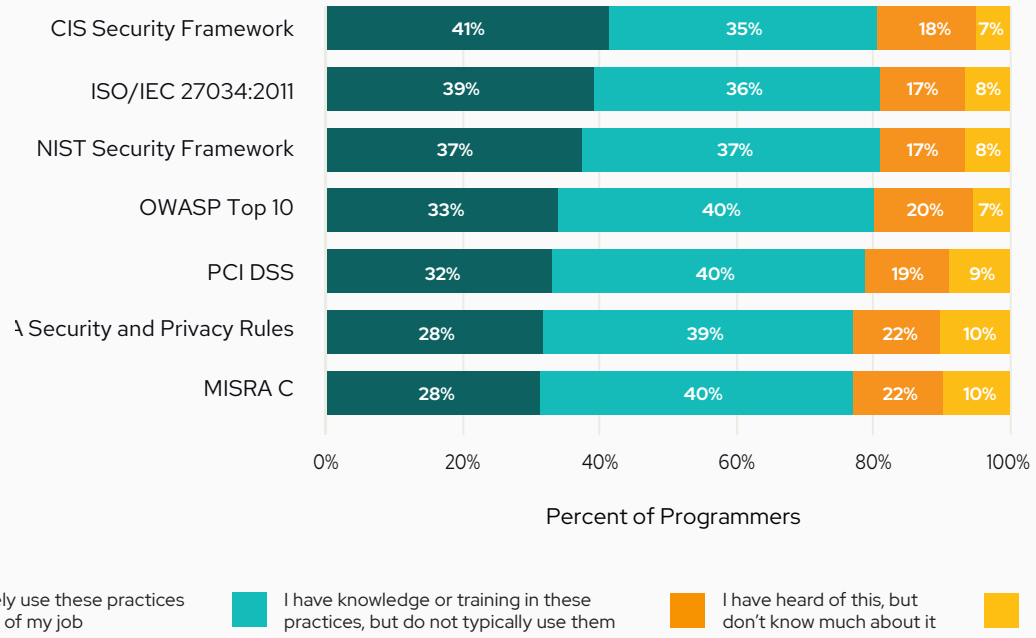


Writing secure code was also seen as very difficult (15%) or somewhat difficult (48%) by a majority of survey respondents. Only 8% of respondents said that writing secure code was very easy for them. When broken down into developers and developer managers, more of the managers ranked secure coding higher on the difficulty scale, though it's universally thought of as challenging by most of those surveyed.

Developers and their companies want (and need) more compliance training

In terms of training needs, compliance was overwhelmingly cited as the main driver of secure coding at most organizations. The rise of security frameworks created by government and private sector organizations, and the need in some cases to legally comply with them, is an increasingly critical need. Of those surveyed, 92% said they needed at least some training in security frameworks, with 50% stressing the need for significant compliance training.

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



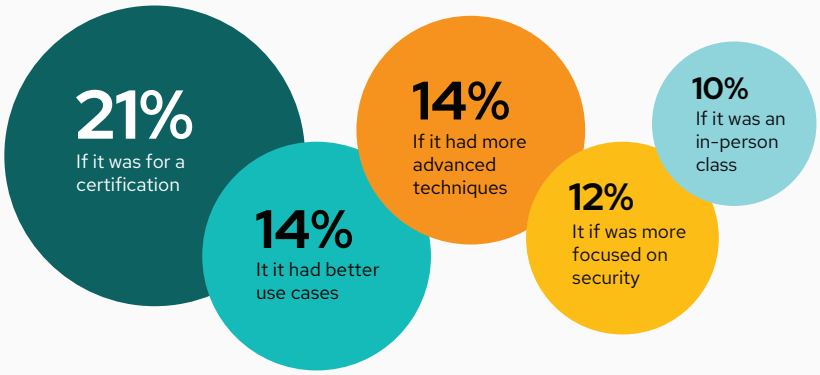
And many different frameworks or guidelines were cited as critically important. Interestingly, the security frameworks created by the National Institute of Standards and Technology (NIST) were often cited as essential, even for those who responded outside of North America. Other frameworks referenced included the CIS Security Framework, PCI DSS, the OWASP Top 10, MISRA C, ISO/IEC, and the Health Insurance Portability and Accountability Act (HIPAA). The responses clearly indicate the need for more awareness and training around common frameworks and requirements.

Problems with existing training programs

However, while developers stressed that they thought prior training was valuable, again we find a disconnect with the rising number of breaches, and the fact that developers are (purposely in some cases because of external factors like deadlines), still shipping insecure or vulnerable code. This indicates that while the concept of training is welcome, current training methods are not engaging or informative enough to make the kind of impact needed to truly help organizations evolve into implementing robust secure coding practices.

When asked about what could be done to improve the quality of the training programs at their companies, developers said they wanted to see more of an emphasis on practical training using real-world examples that were relevant to their jobs (30%). More interactivity was also seen as critical by 26% of the respondents, especially if they were able to actually practice writing secure code as part of that training. Guided training focusing on specific vulnerabilities was seen as important to 23% of the developers surveyed, while 22% wanted to see more vulnerability examples in their training courses.

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



Developers want practical training that is highly relevant to their work

In general, training where users watch a video or listen to a lecture with no opportunity for hands-on, contextual learning is not seen as effective or especially valuable.

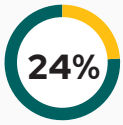
Developers want (and need) more secure code training

Given the fact that the developer community and the organizations they work for are both seeing increased demand and a need for secure coding practices, it may come as no surprise that interest in training is high. When asked about why developers wanted access to secure code training, 53% said they were motivated by both a personal desire to improve their code and by the requirements of the company where they worked. Another 23% said that their company had no interest in secure coding, but they wanted access to relevant training anyway. Only 21% said they were required to undertake training but had no personal interest in it.

76%

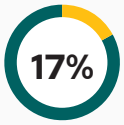
of developers are motivated to study secure coding practices

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



Desire to create top quality code

Potential career advancement



Company requires secure coding

Developers also overwhelmingly agreed that the training they have participated in has been valuable to their career advancement, with **93% saying that it has positively impacted their career growth.**

When asked to break down their personal motivation for wanting more secure code training, the most popular answer (28%) was a desire to create top-quality code. This mirrors the top reason given as to why developers say that secure code is important – personal responsibility and a desire to create the best possible code. Another 18% said that coding securely is important to their career advancement, while 16% simply want to avoid the problems caused by insecure code, like having to rewrite and patch programs after they have been deployed.

Conclusion

The need to secure code right from the beginning of the software development process is greater than ever. Only by "starting left" and incorporating security into the developer workflow can organizations protect themselves against an increasingly complex threat landscape.

The survey showed that attitudes about secure coding from within the developer community are improving, with even those who don't yet personally embrace it realizing that it's going to be a critical requirement in the very near future.

Based on the results of the survey, we've identified several ways that organizations can begin making critical improvements in their software development process, while also helping development teams secure their code. Four good paths forward include:

1. Define secure code for your organization

One of the biggest shortcomings identified by the survey is also one of the easiest to start to fix. Developers stated that their organizations did not have a clear definition of what constitutes secure code. A worrying 28% of respondents stated that code was considered secure only if no breach was reported once it got deployed into a production environment (when it comes to cybersecurity, hope is not a strategy to be endorsed). And simply pulling code from a supposedly secure library (61%) was also thought of as a fully secure practice. Organizations need to define secure code and secure coding practices, with an emphasis on writing vulnerability-free code right from the start. Only once such a standard is officially defined can the developer community work towards that goal.

2. Adopt a cohesive approach to developer-led security

Organizations also need to embrace a cohesive approach to developer-led security. Once the definition of secure code at an organization is created, the next step is to tightly plan how to implement creating it. This might involve more comprehensive, contextual training for developers to give them the skills they need to code securely. It might also involve changing the metrics by which developers are evaluated, shifting away from rewarding raw speed and instead edifying those developers who can create secure code that is free from vulnerabilities or exploits.

3. Realize that compliance and security are both important, but separate

Adopting a better, more focused view on compliance and compliance training can also help. Compliance training is important, and in many places is even mandated by law or the regulations of certain sectors. But "checking a box" on compliance training does not equal providing a foundation for the ongoing creation of secure code. Instead, approach compliance training as an opportunity to expand your developer's secure coding skills that can be repeated outside of the compliance cycle, so they can create and release secure software every day.

4. Define your organizational cybersecurity maturity

Organizations should also consider where their developer teams (and individual developers) currently sit when it comes to security maturity, or even within the context of a broader cybersecurity maturity model. This is important because organizations need to be aware of their current capabilities before they can plan what to concentrate on next. For example, if an organization is still using agile or waterfall in its software development, then jumping directly into advanced DevSecOps concepts isn't a realistic goal. Similarly, you will likely have developers within your teams with varying degrees of secure coding skills and knowledge. You need to know where you are before you can plot a course towards the desired destination.

Even with a lot of challenges to be overcome, the results from this survey demonstrate that the developer community knows that secure code is key to the future of secure software development. It will require more training, additional changes in mindsets, and a lot of support from management. The shift is already starting to happen. And while secure code isn't yet woven into the fabric of a majority of software developers or the organizations they represent, it's clear that it's a critical goal that they will soon need to achieve.

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.

Further reading

Visit www.securecodewarrior.com/cp/the-state-of-developer-security-skills-2022 to download the survey report and other assets to help your organization improve software security.

SECURE
CODE
WARRIOR

securecodewarrior.com