Guide

The ultimate guide to security trends in financial services

SECURE CODE WARRIOR

The ultimate guide to security trends in financial services

Introduction

How secure code training supports success in financial services

Financial services institutions face an array of challenges that hinge on their ability to make efficient, effective use of technology in a fast-evolving financial world.

Organizations are operating in a time of rapid changes—both internally and across the industry—in a highly competitive, cloud-based business environment. In pursuing their ongoing digital transformations, for example, organizations are working to get around the organizational friction that hinders investments into new technologies, such as artificial intelligence, that could accelerate payment processes and other procedures.

Never far from their minds, of course, is the need to stay compliant with a gamut of regulatory requirements, ranging from the Sarbanes-Oxely Act requirements on financial records management and Payment Card Industry Data Security Standard (PCI DSS) rules on protecting cardholder data to personal information protections contained in the California Consumer Privacy Act (CCPA) and the EU's General Data Protection Regulation (GDPR).

Fines and the other costs of non-compliance can quickly add up. IBM's Cost of a Data Breach Report 2023 found that the average costs for organizations with a "high level of non-compliance" totaled $5.05 million—which is more than a half-million dollars over the average cost of an actual data breach.

And speaking of data breaches, cyberattacks continue to be a scourge of financial institutions, which are the second-most targeted industry for cyberattacks. Ransomware attacks alone have increased dramatically, going from 34% in 2021 to 55% in 2022 and up to 64% in 2023, according to Sophos' State of Ransomware in Financial Services 2023 report. The number of data compromises (including data breaches and exposures of private data) in the U.S. financial services industry increased from 138 in 2020 to 744 in 2023, according to Statista.

> **Failure in any of these areas—be it internal efficiency, compliance, or security—threatens an organization's reputation and, by extension, customer loyalty, which is the lifeblood of financial services. To remain competitive and successful, organizations need to foster trust among their clients, and that starts by ensuring that their software, systems and processes are efficient and effective. At the core of all of that is secure software code.**

Developers are under mounting pressure to create, update and deploy applications and services at a faster pace than ever. And that means developing and working with more software code, whether written by in-house developers themselves, produced by AI, gleaned from open-source repositories or delivered by third parties.

The quality and security of that code is paramount to ensuring effective operations, but that's one area where organizations too often fall short. And as the volume of code increases, the number of errors, flaws and vulnerabilities will only grow unless corrected early in the software development lifecycle (SDLC).

Organizations need to start left to create more secure code and correct errors early. Agile training in secure coding best practices can provide the foundation of secure, trustworthy applications, which not only reduces an organization's risk, but helps feed business success.

Trend 1 AI development tools

Innovation with AI depends on secure code

Artificial intelligence, especially generative AI, is quickly becoming pervasive in commerce, education and everyday life. Among the myriad uses generative AI has been applied to, one notable function has been writing code. This has generally proved to be beneficial, but it also raises another issue tied tightly to AI use—security. As early adopters of new technologies, financial institutions need to ensure a balance between productivity gains and the secure and responsible use of AI.

To date, AI has been used mostly for assisting in code development, rather than, say, automating the remediation process, and its impact has been mostly positive. Respondents to a survey by GitHub, which found that more than nine out of 10 U.S.-based developers were using AI coding tools, said the advantages included productivity gains (53%), giving developers the freedom to focus on creative rather than repetitive tasks (51%) and preventing burnout (41%).

> **Large banks and other financial services organizations are more likely than those in other industries to be early adopters of AI. After all, financial institutions are essentially tech companies because they have the money to invest in new technology at scale, and are always looking for a competitive edge.**

While some have expressed fear that AI will supplant human workers, the technology actually works best when paired with human counterparts. But developers need more than a bare-minimum checkbox approach to learn how to use it. They require precision training to truly grasp security best practices in real-world settings, so they can not only write secure code themselves but can also ably supervise the work of their code-writing AI assistants.

For example, one exercise in SCW's training prompts an LLM model to change the content of a real code snippet in order to modify the code's function. The AI responds by producing a code block—but that block is susceptible to cross-site scripting (XXS). The training ensures that the developer can recognize that vulnerability.

AI and developers can work together very productively, but only if developers are trained well enough to ensure that AI is generating secure code.

AI's coding mistakes can spread quickly

In being trained to write code, an AI model will ingest thousands of examples of code writing, just as another model ingests thousands of examples of prose or poetry before it can write a story or poem for a user. But there is no guarantee that the AI model isn't drawing on an example that contains errors. Because AI models aren't transparent about their processes, errors won't show up until after the fact. And the AI will repeat those errors until they're corrected.

An early study by AI researchers found that AI-generated code had introduced significant flaws, including cross-site scripting (XSS) vulnerabilities, susceptibility to code injection attacks and new vulnerabilities specific to AI-generated code, such as those associated with prompt injection. If AI tools were used unchecked to write code, bad code could spread quickly, resulting in a world where software, which already has plenty of vulnerabilities, would be less secure than ever.

> **It is imperative that human developers and AI models work together in developing code, ensuring that secure coding best practices are followed, allowing financial institutions to gain the benefits of increased speed and efficiency while limiting the risk that, without human involvement, could potentially be catastrophic.**

Financial institutions can lead the way on secure development

The rapid growth of AI, particularly those that use large language models (LLMs) such as ChatGPT and the many other generative AI implementations that are capable of creating their own content, has had its missteps. AI models have generated errors, biased findings and AI hallucinations, resulting in increased calls for regulation. The White House, for example, has issued an Executive Order on the development and use of AI. It also proposed an AI Bill of Rights intended to help protect the public from AI-related risks. Any government initiatives, however, will rely on cooperation and collaboration with the technology companies developing AI, many of whom have pledged to uphold ethical standards.

The financial services industry is also likely to implement strong internal controls. Organizations may be always looking for a competitive edge, but they know that the security of their information—both internal data and that of their customers—is paramount. They also must be sure to satisfy the requirements of regulations, such as those of the Office of the Comptroller of the Currency (OCC) in the United States or the European Central Bank (ECB) for European operations.

As early adopters of AI, banks and financial institutions will be interested in seeing what AI can do to improve efficiencies, sponsoring innovation hubs and other efforts to explore AI's capabilities. But organizations also need controls to ensure security. Early adoption always carries inherent risk, and as the use of AI scales, the risks and rewards need to be balanced. Organizations, for example, would benefit from performing a strengths, weaknesses, opportunities, and threats (SWOT) analysis in the early stages of using AI.

Effective use of AI starts with secure code

The financial industry's ability to make effective use of AI will depend on security, and that relies on ensuring that the code AI creates is secure from the outset. Organizations need to ensure they have highly trained engineers who will closely oversee AI code writing, identify errors and quickly correct them. Partnering with companies that provide agile-based training and other services that ensure security and compliance is a good first step towards fostering a strong security posture.

The risk landscape is continually changing, especially within the software development cycle. And as early adopters of AI, financial institutions must act as leaders in the secure, responsible use of AI.

Some financial institutions are large enough that they can affect government policies. By taking steps to ensure secure code by enabling AI models and developers to work together, institutions can establish best practices for other industries to follow.

You can learn more from SCW's whitepaper explaining the role AI can play in code writing and why it's essential that developers are thoroughly trained in recognizing secure coding patterns and the vulnerabilities that result from unsecure code.

Learn more

Trend 2 Increased regulation

Code sits at the heart of regulatory compliance

An important driver for implementing secure coding practices is the need for financial institutions to ensure compliance with regulations governing their business. Institutions have an array of applicable regulations, which vary depending on the kind of transactions they handle.

For example, PCI DSS 4.0, the most recent iteration of the Payment Industry Data Security Standard, is a global standard designed to protect credit and payment card data and transactions. Intended to prevent fraud and other misuse, it applies to any organization that stores, processes or transmits cardholder data. The standard is not a law, but it is enforced through contracts, and it can help prevent data breaches, which are violations of other regulations such as GDPR.

Another regulation, the Digital Operational Resilience Act (DORA), is a binding EU risk management framework for the financial services sector, covering both financial institutions and their third-party providers. DORA sets technical standards designed to unify risk management practices with the EU, creating a universal standard. The Society for Worldwide Interbank Financial Telecommunication, known as Swift, is a cooperative effort that has set the standard for fund transfers across the global financial sector. Swift's Customer Security Program (CSP) has developed the Customer Security Controls Framework (CSCF), which is updated annually. Swift's 11,000-plus members in more than 200 countries use the CSCF to plan their own security controls and attest to their level of compliance.

> **What these regulations have in common is that they attempt to set high standards for protecting data and transactions in the financial services industry. Compliance not only protects customer data and money, it also helps protect institutions from the consequences of inadequate security, which can include fines and penalties for non-compliance, damaged reputations and the loss of trust from investors in the event of a breach.**

Developers have to keep pace with a shifting regulatory landscape

Secure coding provides a solid cornerstone for organizations looking to satisfy the expectations of applicable regulations and the requirements of the Office of the Comptroller of the Currency (OCC) in the United States or the European Central Bank (ECB) in Europe. One of the fundamental drivers for adopting Secure Code Warrior's platform, or that of another provider, is to teach developers secure coding in a practical, engaging environment.

That kind of training makes a difference when dealing with sometimes complex regulatory requirements, particularly because they don't remain static. Regulations are always evolving, adding new, often more sophisticated requirements. Depending on the requirement, a regulation might not change dramatically in any given year, but organizations can expect significant changes at least every couple of years.

For example, PCI DSS 4.0, mandatory as of April 1, 2024, updates PCI DSS 3.2.1 (released in 2022) in several significant ways. It implements a customized approach that gives organizations more flexibility in meeting requirements, focusing on outcomes more than checkbox procedures. But it also adds new requirements for authentication controls, password lengths and shared accounts, to name a few of many updates. It also requires organizations to clearly define roles and responsibilities for meeting each requirement.

Significantly, Version 4.0 also requires that developers working on bespoke and custom software are trained at least once every 12 months on software security, including secure design and coding techniques and, if testing tools are used, on how to use the tools to detect vulnerabilities. It also stipulates that every vulnerability identified during penetration testing be remediated, regardless of its severity, and that teams follow up with a second penetration test to confirm successful fixes.

> **Although changes are typically made gradually, regulations are also event-driven—a major breach can prompt sudden, extensive changes. A breach of 87 million JPMorgan Chase accounts in the mid-1990s, for example, shook the regulatory landscape, with regulators raising the expectations for the developer/tech community, and requiring that banks provide proof of the steps they were taking and how they were applying the lessons learned from the breach.**

Compliance banks on the quality of the underlying code

The quality of code used in meeting these requirements has a significant impact on how well new functions perform and can come into play when companies are completing the lengthy and detailed PCI DSS Compliance Report, which is required annually. As regulations become more complex, the impact of secure coding becomes commensurably larger, making a substantial difference in reducing risk and increasing control over an organization's software.

Compliance is essential for financial institutions because of the importance of establishing trust with customers. Secure coding can also help improve the customer's experience because so much depends on the seamless interaction with reliable software, which helps build customer loyalty.

The coding that goes into creating new applications and services has an impact across the enterprise. It's essential for increasing efficiencies, managing cloud migrations and enabling other capabilities in a fast-moving business environment. But the code absolutely must be secure, and it must meet compliance requirements for the business to succeed.

Read the no-nonsense guide to getting your development team on board with PCI DSS compliance, including how security professionals and development managers can work together to build powerful security programs.

Learn more

Trend 3 Agile learning

Agile training and human-AI teams ensure secure code

The proliferation of AI models has revived fears that artificial intelligence will take away a lot of jobs from people. While people in some occupations may have cause to worry—ranging from bookkeeping and customer service to law clerks and content creators—software developers are more likely to welcome AI as helpful assistants that won't take their jobs but will take some time-consuming or repetitive tasks, such as writing code, off their plates.

Writing code, in fact, is only part of a developer's job. In GitLab's 2023 Global DevSecOps Report, most developers said they spend about a quarter of their time writing code. The rest is divided among other tasks, such as improving code (17%), testing (11%) and sitting through meetings or performing other administrative tasks (also 17%).

> **Improving code is one aspect of the job that will likely become more prominent when AI models are generating code. AI adds speed and efficiency to creating code, but that code can't be entirely trusted. There are countless examples of errors, bias and vulnerable code generated by AI models. Security-skilled developers must be closely involved in checking AI-generated code to fix vulnerabilities and ensure that their software adheres to development standards.**

Developers need hands-on, agile training

For developers themselves, working with AI-generated code requires that they sharpen their existing skills—and acquire some new ones—regarding security best practices and the ability to spot poor coding patterns. Developers who are properly trained will be able to spot an AI model's missteps before deployment and enhance the advantages of using AI to accelerate development.

The skills required are complex, however, and can't be learned or strengthened by simply employing standard, static training methods. Developers aren't known for having spare time on the job—they're under more pressure than ever to develop and deploy code quickly. They need to be able to increase their skills in a way that suits the job they're already doing.

Organizations need to offer developers a complete program of agile-based training that takes a hands-on approach to secure coding and that has been shown to significantly reduce the number of vulnerabilities in software.

Agile training can be tailored to include the programming languages developers will come across, from ancient-but-still used COBOL to advanced new tools written in Google Go. It can be designed to deliver advanced content in formats suited to the developer's preferred learning methods—such as visually, auditorily or verbally, as well as directly hands-on—and delivered at a pace that works best with individual developers and their work schedules.

Training can also be tailored to the specific roles and needs of employees. A platform can make use of a feedback loop to improve content and recognize when a developer is weak in a certain area, so the content can be automatically targeted to address that area.

> **And instead of delivering security training in a drawn-out and dry classroom-style training course, learning programs such as those employed by Secure Code Warrior deliver optimal, customized training by breaking it up into interactive microbursts that engage developers within the context of real-world problems. Microlearning is also fully accessible to employees whenever they need it.**

An agile approach gets results

Agile-based training has proved to be effective in reducing coding errors. According to research by Secure Code Warrior, developers already rework about 26% of their code before it goes into production. That adds up to about 13.5 hours per week per developer (about 700 hours a year) spent on cleaning up technical debt. The hours spent fixing code hinders productivity and slows down development cycles.

And not all of those mistakes are caught, with 67% of developers admitting to having shipped code with vulnerabilities. Organizations also are using code from other sources, such as AI, open-source repositories and third parties. Those sources help increase productivity at a time when organizations need a higher volume of code than ever, but they also increase the risk of vulnerabilities and errors in the code base.

Agile-based training can help stem that trend. It bolsters the first line of defense, with developers being more skilled at catching flaws in code, whether it's been created by themselves, AI or third parties. As part of its research, Secure Code Warrior examined data from 75,000 developers (about 30% of its base) and found that developers who had studied security using its agile-based training introduced 53% fewer vulnerabilities than their peers. Developers applying those skills to checking AI-generated code can more quickly clean up their AI partner's mistakes before any software goes into production.

> **With Secure Code Warrior's agile-based training, the developers at Workday, which provides financial, HR and student/faculty lifecycle management cloud applications, gained a clear idea of what the training was designed to accomplish, and quickly learned how to identify and act on problems within their codebase and software lifecycle.**

Workday's experience offers a clear example of what agile, hands-on training can do. Before partnering with Secure Code Warrior, Workday saw that their developers were disenchanted with the company's slideshow-based security training. But the entire developer community responded well to the Secure Code Warrior training, which was tailored to both their needs and learning preferences. And the results speak for themselves. In just one example, a team in Dublin went from experiencing 4,662 yearly security issues to having none at all in just 18 months.

In a threat landscape rife with increasingly sophisticated attacks, financial services companies need to do everything they can to ensure secure data and applications. Creating secure code at the beginning of the software development lifecycle (SDLC) is a critical component of security. Developers with the right kind of agile training can do a lot to ensure the security of software while reducing the overall risks to their company.

Learn how Workday focused on creating secure agile learning that drove developer engagement and helped achieve their goals of improving Workday's overall security posture.

Learn more

Trend 4 Growth of third-party apps and APIs

Responsibility for security applies to third-party apps, too

In today's hyperconnected business environment, no company operates in a vacuum. Financial institutions enter into partnerships with other companies to provide certain services, make use of third-party apps in daily communications and transactions, and in many cases have outside contractors writing software code. Developers within companies also make use of open-source software repositories and, increasingly, AI-generated code in developing applications.

> **Regardless of the source of software, people who engage with a financial services company have the expectation that every application they use will have the same high level of security. In any transaction or exchange of information, the host company still has responsibility for the customer's data. And regulators won't allow an institution to pass the buck on non-compliance to a third party.**

How does a financial institution ensure that every application is secure and reliable? It starts with secure code and with giving developers the skills they need to create secure software code at the beginning of the development process, while also identifying when the third-party code they're using falls short.

Make training available to contractor teams

Companies would benefit from implementing an agile, hands-on training program to teach developers about secure code. The results of that kind of training are clear: Developers with training in secure coding produce code with 53% fewer vulnerabilities than those who don't have the training. And they're also going to catch more coding mistakes in the code generated by third parties.

Even if a company's own developers are trained in writing secure code, the potential flaws in third-party code need to be addressed. Many developer teams are comprised of both full-time employees (FTEs) and contractors, particularly within large institutions with far-flung locations around the country or around the world, including primary, secondary and tertiary locations.

A couple of decades ago, there was a big push in the industry to outsource software development so that companies could keep up with the demand for new applications. That trend lasted five or 10 years before starting to flip back, but combined teams of FTEs and contractors still remain at some locations within financial institutions. With so many applications continually being developed, some of them are going to be outsourced. Regardless of whether code is developed by FTEs or third parties, however, the expectations of customers and regulators remain. All the software code a company uses must meet the same standards, which means that all developers have the same level of competency.

Companies may have contractual limits on requiring training for contractors, though it is important to at least make training available. Some financial companies have created their own training programs, such as Capital One, which launched its Tech College in 2016. A number of other banks and financial companies, such as Synchrony Bank and North American Bancard, are embracing the idea of agile continuous learning to increase the level of talent they have.

> **It's possible that companies may even take a "license to code" approach, requiring certain certifications before allowing developers access to specific systems.**

Amid an ongoing IT skills shortage, companies have opted to try upskilling current employees rather than competing for talent in an undersupplied market. Providing training benefits both the employees, who can further their own careers, and the company, which gets employees with the skills the company needs. Training programs can also improve retention by improving the employee experience.

Continuous learning is important in the current environment. The cybersecurity landscape is continually evolving and becoming more sophisticated. And regulators' requirements also change yearly, increasing the complexity involved in staying compliant. Failure to meet those requirements can result in fines, other costs and reputational damage that materially affect the company.

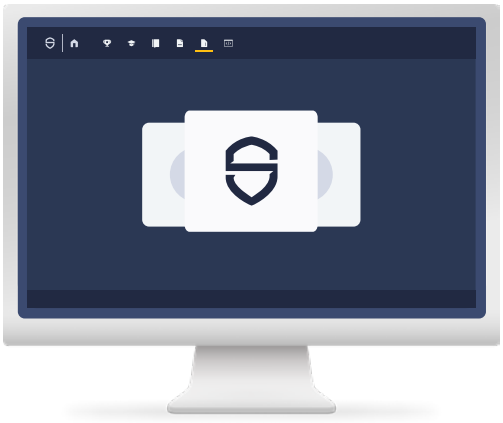
The key is to engage developers

Companies are moving away from compliance training as a burdensome, checkbox exercise with once-a-year training sessions and toward year-round training designed to increase the level of engagement developers and other employees have with security. The key is in having an agile platform that gives employees the training they need when they need it, and in a format that suits their working environment.

For example, the UK-based software solutions company Sage, which adopted Secure Code Warrior's platform, does quarterly training that is targeted to the technology that developers are working with at that time. The training is personalized as much as possible and also tailored to how the developers like to work, said Mads Howard, a security specialist at Sage, during a recent webinar with SCW. Sage also encourages two-way communication via a feedback loop. The training is focused more on engagement than on check-box compliance, with regularly scheduled tournaments and efforts to pair the training program with other company initiatives designed to increase employee engagement with security issues.

> **One result: in the past year, Sage has seen an 82% reduction in the average time it takes to fix software issues, Howard said. The company has also seen a greater level of engagement from employees.**

Coding is part of building a security culture, Howard said, with the ultimate goal being to build trust with customers. A security culture revolves around people's attitudes, perceptions and beliefs, and it also involves people throughout the company, including executive leadership. A flexible secure code training program that delivers targeted training in easily consumable microbursts can be an essential part of that culture.



Discover how Sage implemented best practices in building a secure code learning program, learn the do's and don'ts of a secure code learning program, and see the business and financial impacts including productivity gained and time saved.

Learn more

Trend 5 Enhanced focus on ROI across teams/vendors

How secure coding directly benefits return on investment

The financial services industry covers a broad swath of sectors, ranging from banking and financial management to credit card and digital payment services. Even insurance often falls under that banner. Companies in each sector face different, although often overlapping, compliance requirements, depending on the transactions they handle and whether they are domestic or multinational companies.

But regardless of the field the companies work in, economic considerations have an effect on business strategies. The stock markets have been doing well, but stock markets aren't always indicative of future success. Meanwhile, there is some trepidation in the financial sector over the possibility of a recession and other critical challenges.

> **As a result, many organizations are tightening their belts, looking for greater efficiencies, and emphasizing the importance of having a good return on investment (ROI) for any new expense. One way to increase ROI is by investing in secure code learning. In the software development lifecycle (SDLC), where engineering and security meet, making sure to create secure code at the beginning of the process and fixing flaws as early as possible will deliver clear, quantifiable gains for the business.**

The high costs of unsecure software

Data is at the core of any financial services institution, and the costs of handling that data with unsecure, inefficient software can mount quickly. Secure Code Warrior's research found that software quality issues cost U.S. businesses a total of $2.41 trillion in 2022. And the costs extend all the way to developers, who spend an increasing amount of time maintaining and remediating technical debt. Developers currently spend about a third of their time maintaining technical debt, but that's projected to hit 40% by 2025.

Agile-based training on secure coding practices can put a significant dent in those negative numbers. Developers training with Secure Code Warrior have been found to introduce 53% fewer vulnerabilities than colleagues that haven't had the training, and remediation times are reduced by half.

A large global bank saw a 50% reduction in vulnerabilities with SCW training, which virtually eliminated SQL injection flaws and Cross-Site Scripting (XSS).

The benefits of agile secure code training are also considerably greater the further an organization shifts left. The costs of technical debt, for example, are cut in half when addressed during testing, but they are reduced 14 times more if addressed during the coding phase.

Case in point: How Envestnet engaged its developers

The impact of secure coding can result in measurable gains in ROI. In one example, the large financial technology company Envestnet wanted to go beyond its passive, "death by PowerPoint" security and compliance training, which was focused mostly on the Open Worldwide Application Security Project (OWASP) top ten web application risks.

Envestnet adopted a shift-left strategy focused on writing secure code and addressing any problems early in the SDLC, but first needed to address developers' lack of engagement with the company's existing security training. With SCW, they implemented a four-level, hands-on program that involved training on real-world scenarios and awarded certificates to developers for each level of achievement—not only improving application security, but helping developers advance their careers.

The first two levels focused on security awareness, and levels three and four recognized security champions. The training program included tournaments, which also increased the level of engagement from developers. Between the first two tournaments, spaced about six months apart, participation doubled.

> **The results: Developers with SCW training fixed 2.7 times more vulnerabilities than their peers, and they increased their remediation rates by 120%. SCW-educated developers also closed vulnerability issues at a rate of 4.5 per developer, compared with a rate of 1.82 per developer for their peers who did not have the benefit of the training.**

2.7x

Starting left improves the bottom line

Agile secure coding programs can be tailored for each company, depending on the type of financial services they provide, the size of their systems and their individual requirements.

Companies need to maintain compliance, for instance, whether with Payment Card Industry Data Security Standard (PCI DSS), the OWASP Application Security Verification Standard (ASVS) or other requirements. And they need to ensure that they are always writing secure code. But they need more than a training program that merely checks a box, which won't do enough to teach secure coding methods or to reach the increased efficiency levels that come with having highly-trained, security-aware developers.

Hands-on training that supports a strategy of introducing secure code early in the SDLC and maintaining it throughout the product lifecycle will reduce risk and speed time to market, inevitably increasing ROI. As systems, processes and even cyberattacks grow increasingly sophisticated, secure coding is sorely needed. It has the power to make a critical difference not only in security, but in benefiting any organization's bottom line.

Explore how Envestnet adopted an agile secure code learning platform and tripled developer effectiveness in vulnerability reduction.

Learn more

Conclusion

Secure coding enhances security and productivity while building trust

Financial services institutions work with very valuable commodities, namely people's money and highly sensitive personal information, but in some ways the most precious thing organizations can have is trust. It's essential to attracting and maintaining loyal customers. And because so many financial transactions are handled digitally, that reliability and the security of software depends on secure software code.

In complex hybrid cloud environments, financial organizations must shift left, introducing security at the beginning of the software development lifecycle (SDLC). That means training developers to write secure code and being able to identify vulnerabilities in AI-generated or third-party code.

It's a cultural shift for many companies, where developers are used to working a mile a minute and spinning up new applications and services to meet an ever-increasing demand. The key is to create a security culture within a company and to engage developers with agile, hands-on secure code training. The benefits of this approach are clear.

Fostering a win-win cultural change

Secure Code Warrior's research has found that developers who learn secure coding practices with SCW produce 53% fewer vulnerabilities than those without the training, which adds up to a lot of time and money saved.

> **Developers currently waste about a third of their time reworking software code, with 67% of them admitting to shipping code they knew had vulnerabilities. SCW customers have seen 2x to 3x gains in both risk reduction and developer productivity as a result of agile learning.**

And the further left an organization starts in the SDLC, the greater the savings. Costs can be cut in half if addressed during testing, but they can be reduced 14-fold if addressed during the coding phase.

An agile training platform benefits both the company and developers. Developers can further their careers by gaining new skills, and the company benefits because skilled developers are more likely to stay with a firm that provides effective training and a more rewarding work experience.

Not incidentally, 92% of developers say they want more training. But it needs to be the right kind of training. Traditional by-the-book (or by-the-slideshow) compliance training can elicit eye rolls and a grudging acceptance of meeting a requirement, but an agile platform such as SCW's has been shown to engage developers. Training can be fitted to what they are working on and the programming languages they are working with. And delivering training in easily consumed, targeted microbursts that address current real-world issues developers are facing increases the value of training and the level of engagement.

> **Secure code training can be a keystone in a cultural shift toward being a security-first organization, increasing a financial service institution's cybersecurity, performance and, ultimately, profitability.**

About Secure Code Warrior

Secure code learning for today's developers

Secure Code Warrior gives your developers the skills to write secure code. Our learning platform is the most effective secure coding solution because it uses agile learning methods for developers to learn, apply, and retain software security principles. Over 600 enterprises trust Secure Code Warrior to implement agile learning security programs, deliver secure software rapidly, and create a culture of developer-driven security.

Request a demo

Try Secure Code Warrior for free

Find us on social: