Checklist to Evaluate a Secure Coding Training Program

Companies roll out secure coding training to their developers for a variety of reasons – to meet compliance requirements, respond to a security breach[1], or as part of a larger movement to Shift Left.

No matter why, every company wants to ensure their training program is successful. But how do you measure success? The two most common metrics are:

1. **Completion rate (easy to measure)**
2. **How much the ability to code more securely has improved (harder to measure)**

Five factors contribute to these success metrics.

Training Philosophy

Any effective secure coding training program must offer developers two opportunities – the chance to "break" applications (simulating an attacker's actions) and then "fix" what they broke. Developers need exposure to both to build the skills needed to write secure code[2].

EVALUATION QUESTIONS

Who does the training program target? To improve developers' secure coding skillset, look for a training program that puts developers first. The best programs do that and go beyond to offer secure coding training to the entire SDLC. When everyone involved in the development process understands application security, safer applications are delivered.	○
Does the program offer developers the chance to break and fix code? A hands-on, sandbox environment is the most effective way for developers to reinforce the coding knowledge they digest in a lesson.	○

[1] According to Forrester, web applications are the most common attack vector

[2] "Evaluation of the Offensive Approach in Information Security Education" - Mink, Greifeneder

We help enterprises reduce vulnerabilities with application security education for developers and all individuals involved in creating software. Development teams are empowered through practical, skill-oriented secure coding training that easily satisfies compliance needs and goes beyond to build a security-first development culture.

Learning Science Principles

Learning science principles are the underpinning of every successful training program. They are based on proven research about how humans most effectively learn. Without this solid foundation, even the best content in the world will not achieve optimal results.

EVALUATION QUESTIONS

Are lessons bite-sized and spaced out at manageable intervals? Too much information at once overloads learners and impedes their ability to effectively learn. The best programs deliver content in small, consumable bits in a multi-year format customizable to an organization's needs.	○
Is feedback immediate, specific, and focused on improvement? Just-in-time feedback gives learners the opportunity to fix mistakes and reinforce concepts.	○
Do students get a chance to practice what they learn? Learning is most successful when students can immediately apply knowledge in safe, real-world scenarios. Developers should have the opportunity to break and fix code in a hands-on way – not just look at code examples then provide an answer -- to best reinforce the concepts learned.	○
Is the learning environment familiar and fun? Engaged students learn more and can apply that knowledge faster. Giving developers a safe coding environment in which to learn and offering opportunities to compete with other developers, like with challenges or tournaments, satisfies both objectives.	○

Delivery Format

How lessons are constructed and delivered plays a key role in a program's success. The most successful programs deliver information in a variety of ways to meet diverse learning styles while maintaining the integrity of the learning process.

EVALUATION QUESTIONS

Are there a variety of modalities to convey and reinforce lesson information? The most effective way to help developers retain knowledge and build skills is to offer lessons that include expert-led videos (with transcripts available for those who prefer to read) and provide real-world examples and hands-on coding challenges to practice what they do best – writing actual code.	○
Are lessons contextualized? The most effective lessons deliver information in a way the learner is familiar with, and this is especially true for developers. Engagement is higher when developers digest the information in scenarios they are accustomed to – a coding environment.	○
Will developers take the content seriously? Gamification elements can improve motivation and learning outcomes, but that strategy can also go too far. When gamification is included at the expense of contextual learning, the program will not be as effective.	○
Are assessments constructed in a way that demonstrates subject matter mastery? The most successful programs offer solid content and comprehensive assessments at the conclusion of each lesson. Even if a learner breezes through the content, they should still be challenged to prove in-depth knowledge to pass an assessment and move to the next lesson.	○

Curriculum And Content

At the core of every successful secure coding training program is content that resonates with learners. This means it covers vulnerability fundamentals (like the OWASP Top 10), is relevant to your actual code, and offers a library that evolves as new risks and vulnerabilities are detected.

EVALUATION QUESTIONS

Does the content cover the most recent OWASP Top 10? At a bare minimum, the program should cover the known vulnerabilities and risks identified by OWASP.	○
Is the content keeping pace with new threats? Developers should be knowledgeable and skilled at addressing new threats that go beyond OWASP Top 10, like the OWASP API Top 10 and OWASP Mobile Top 10. Make sure the program you choose goes beyond bare minimums.	○
Does the content reflect actual vulnerabilities of today? There should be a mechanism that generates up-to-date content based on actual vulnerabilities as they are discovered and incorporates this content into real-world training scenarios.	○
Is the content relevant to your developers? Look for a program that integrates with your AppSec testing tools and adapts the content offered based on actual vulnerabilities found in your code. When developers are fixing actual threats in your applications, the learning is more meaningful and developers stay engaged.	○
Is there enough content to support a multi-year training program for developers? Studies show there is learning loss within 28 days of completing any training lesson. It is essential to offer secure coding training opportunities throughout the year, allowing learners to regularly reinforce their knowledge gains. Ongoing training turns learners into security champions.	○

Customer Success And Support

During and after any new program roll-out, having resources to answer questions and help troubleshoot issues is essential to smooth adoption by developers. Be sure the secure coding training program you choose offers this type of help.

EVALUATION QUESTIONS

How easy will it be to roll out the program? Deploying a new application across any organization is a challenge, so finding one that is as streamlined and uncomplicated as possible is the key to success. Do not fall for bells and whistles you may never need. Focus on finding the leanest solution for your organization.	○
What type of customer success and support services are offered? Be wary of programs that do not include some level of customer support – or offer it only for customers of a certain size -- for both the initial roll-out and the maintenance phase.	○
How much does customer success and/or support cost? If a program says they offer customer support, be sure to find out if access to that support will cost extra. Nobody wants unexpected financial surprises after the sale.	○