6 Critical Steps

before you roll out a

Security Uplift Program

for Software Developers

SECURE CODE WARRIOR

As more businesses move to agile methodologies, providing more online services to customers and exposing data and APIs, the need for "Shifting Left" and building security into the start of the (continuous) development cycle grows every day.

CIOs and CISOs need to think carefully about when, where and how to deploy a security uplift program to maximize business impact and not overload the core function of the developers. In this e-book, we share six steps companies should take before deploying a program, based on our learnings from more than fifty enterprise deployments. These steps will help you get the strategy, data and processes in place to drive success.

1. Define Objectives of your Security Uplift Program

While you want to make your developers the first line of defense in your security program – by helping them to code securely, that doesn't mean they need to be the most in-depth security experts in your business. You don't need to continuously push new security vulnerabilities and data breach examples to them. While security is important, their primary objective is often building a great product or service and following the pace of the business. If you overload someone with too much security, you run the risk of disengaging them.

Sample objectives of your program could be:

- Get the attention and focus of developers and their management on application security
- Ensure everyone who writes code has the basic security hygiene, awareness and secure coding skills, regardless of role, seniority, location or type (employee/contractor/supplier)
- Provide every team of developers with a go-to security person who they can turn to for advice and support
- Reduce the number of application security weaknesses identified throughout the development life-cycle and the time to fix application security bugs
- Demonstrate the skills of the developers to adhere to compliance standards such as PCI-DSS

2. Define Owners and Security Champions

Identify an owner of the program. Think about who is going to drive this program, measure progress and coordinate between the different stakeholders.

Next, it is critical to find the right security champions in each scrum team for ongoing program success, especially in larger organizations.

In our experience, not every highly skilled security expert or developer has highly developed people and/or communication skills. The best security champions, those who will give your program positive ongoing impact, are those people who are passionate about security and have strong people, influencing and communication skills. Choose wisely and take personality and communication skills into account.

3. Define Team Structures, Reporting and Access Control

A typical organization has a yearly churn rate of 30% amongst their developers. These professionals are often spread across different locations, work for different business lines, write in different coding languages and have different roles (front-end, mobile, back-end, full-stack, devops, etc). Identify these parameters upfront so you effectively create team structures, assign tags to users for reporting purposes and understand their requirements based upon role. Often, federation (ex. Security Assertion Markup Language (SAML)) can be effectively used to reduce the administrative burden on training platforms.

NAME	ROLE	CODING LANGUAGE	BUSINESS UNIT	LOCATION	MANAGER
John Doe	Front-End	JavaScript	Retail Banking	United Kingdom	Jane Smith
Jack Pie	Full-stack	JavaScript and C#	Trading	India	Tim Jones

Any skills-based training provided should focus on secure coding techniques in a particular coding framework (C# MVC, Java Spring, Python Django) because every coding framework has its own weaknesses and its own recommended best practice solutions to remediate those weaknesses.

4. Identify Ideal Metrics to Track

Don't measure how many hours of training your developers take or how many videos they watch; measure the number of weaknesses that get picked up in the development life-cycle through code analysis, bug bounties or classic vulnerability testing before you start the program in each team.

If your training program is working, the number of vulnerabilities you identify will go down. The second thing to measure is the time it takes to fix a vulnerability. If it takes a developer a month to fix it, this clearly shows they don't understand how to do it, but if they can fix it in an hour, you know they mastered the skill. These are the two metrics that clever companies are employing to measure the impact of their developer secure coding training.

Some other metrics that you could use to measure success are:

- **Adoption:** How many teams and/or developers have participated in the program?
- **Engagement:** How many teams and/or developers are actively engaged in the program by participation and providing feedback?
- **Skills:** How many developers are able to pass the baseline assessment?
- **Impact:** Are the number of vulnerabilities reported by static code analysis tools trending downward? Or has the number of vulnerabilities reported by the security assurance functions (pentesting or bug bounty programs) decreased?

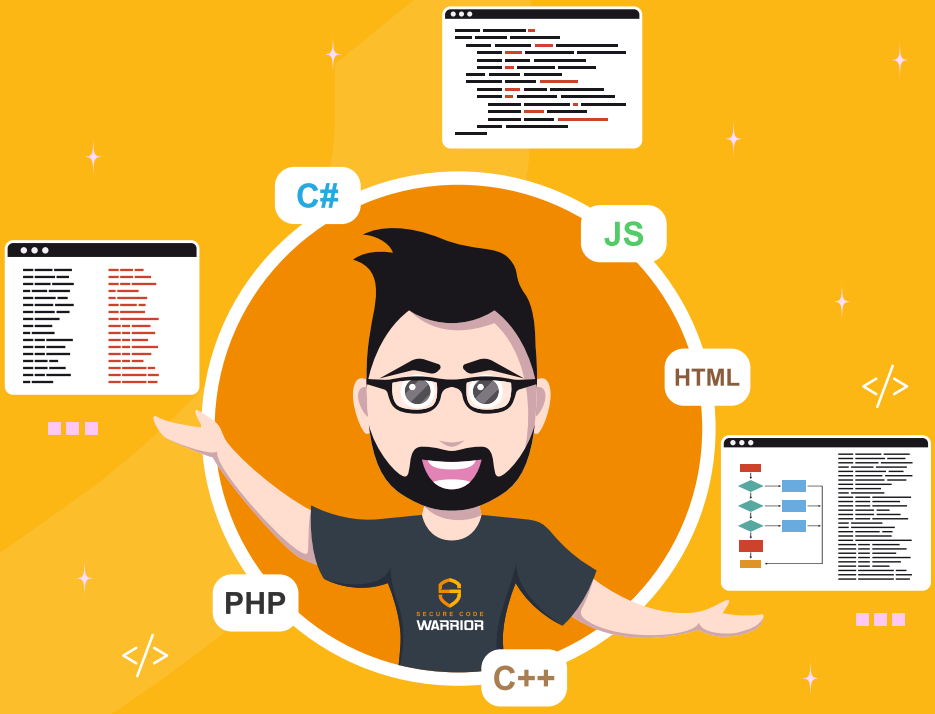
5. Put Some Fun into the Kickoff If You Want Your Program to Have Impact

Making a special kick-off day event, or giving your program a movie/ geeky/gaming theme, can make a big difference to the level of excitement and participation from the start.

One of our large customers created a whole fantasy program around a popular TV series, with t-shirts, badges and stickers, making the whole training program an experience no developer would want to miss. Another themed their event around Star Wars, held conveniently on “May the 4th.”

A bit of fun goes a long way to helping employees get on board. It makes short bursts of important developer secure code skills training feel like a break from work, rather than another task that needs to be done.

Milestone #1 achieved, we've got their attention and they know about the program.



6. Have Cool Rewards That Recognize Skills

In general, every computer geek – including developers – likes to be recognized for their intelligence and skills. When you reward them with specific status stickers, geeky gadgets, special badges or custom printed t-shirts that recognize their skills, they will wear or display them with pride. If someone achieves something significant in the training program, it is important to think of ways to give them recognition and exposure.

We saw a great initiative with one customer who regularly took photos of their Secure Code Warrior champions and posted them alongside a photo and message from the company's CISO in the employee newsletter.

Secure Code Warrior is the smartest and easiest way to improve your application security program.

Find out how we can help you:

- Achieve faster and more secure product development
- Defend yourself with security-conscious developers
- Build a positive security culture

Get in touch with us today via info@securecodewarrior.com or visit securecodewarrior.com for more information.

securecodewarrior.com

insights.securecodewarrior.com

info@securecodewarrior

@seccodewarrior

/company/secure-code-warrior

/securecodewarrior