Throughout this course, I’ve learned that security must be built into the development process from the start. Adopting a secure coding standard, such as the SEI CERT guidelines, ensures that common vulnerabilities are addressed early, instead of waiting until the end when it’s usually too late or costly to fix.

Evaluating and assessing risk is also essential. Not every vulnerability requires immediate action. By using tools like threat matrices and performing cost-benefit analyses, companies can prioritize fixes that provide the greatest impact. For example, input validation is a low-cost method that effectively reduces risks like injection attacks.

The concept of Zero Trust further reinforces this mindset. Rather than assuming users or systems are trustworthy based on their location within a network, Zero Trust requires continuous verification and least-privilege access. This is especially important in cloud-based environments, where the traditional security methods no longer applies.

Implementing strong, clear security policies is also key to maintaining consistency across development teams. These policies should cover secure authentication, data protection, and regular security reviews. Automated tools can help ensure policies are followed and updated as new threats emerge.

I plan to apply these principles by integrating secure coding practices early, assessing risk, adopting Zero Trust strategies, and recommending clear, enforceable policies.