**Andrew Torrez**

**CS-405-10428**

**October 23, 2025**

**Portfolio Reflection Journal**

Looking back over this course, I’ve realized how much my perspective on software development has evolved. Early on, I saw security as something that happened at the end of a project — like a final check before deployment. Now I understand that secure coding isn’t a phase; it’s a mindset that has to exist from the very first line of code.

**Adoption of a Secure Coding Standard**

Through developing the Green Pace Security Policy, I learned how important it is to establish consistent standards such as SEI CERT C++ rules and to use them throughout the development process. By doing that, security becomes part of the foundation rather than an afterthought. I also saw how coding standards reduce human error, improve team collaboration, and make it easier to identify vulnerabilities before they reach production. Adopting secure coding practices from the start not only protects the software but also saves time and cost in the long run by avoiding major rework.

**Evaluation and Assessment of Risk and Cost Benefit of Mitigation**

The risk matrix exercise showed me that not all vulnerabilities carry the same weight. By ranking risks based on likelihood, severity, and remediation cost, I learned how to prioritize effectively and make informed decisions about which threats require immediate attention. This balance between security and practicality mirrors what real development teams face — limited time and budget. Understanding cost-benefit analysis helped me appreciate that mitigation isn’t about fixing everything at once, but about managing risk intelligently.

**Zero Trust**

The concept of Zero Trust completely reshaped how I view system and network security. The idea that “no one is safe” taught me to question every assumption about access and authentication. From a developer’s perspective, it reinforces the need to design applications that verify identity and integrity at every step — whether through MFA, API tokens, or role-based permissions. From a user’s standpoint, it’s a reminder that convenience should never outweigh protection. Zero Trust also helped me think critically about how automation tools, like CI/CD scans and static analyzers, can continuously enforce that same principle within the codebase.

**Implementation and Recommendations of Security Policies**

Creating and presenting the Green Pace policy tied everything together. It gave me hands-on experience with defining encryption policies, AAA frameworks (authentication, authorization, and accounting), and secure DevSecOps automation. It also helped me think like a security lead rather than just a developer — considering organizational policies, compliance, and long-term strategy. Going forward, I’ll carry these lessons into any project I work on by advocating for continuous testing, documentation, and team-wide awareness of security responsibilities.

Overall, this course taught me that security is not just about technology — it’s about culture, discipline, and foresight. A strong security mindset protects not only systems and data but also trust — the foundation every modern organization depends on.