8-2 Journal

CS 405

Ruben Sanchez

Southern New Hampshire University

Portfolio Reflection

Using a secure coding standard is important for building strong software. By focusing on security early in the development process, developers can avoid problems that come from trying to add security at the end. This approach matches the DevSecOps method, which focuses on keeping security in mind throughout the software development life cycle (SDLC). For example, tools like CppCheck and regular code checks help ensure that secure coding rules are followed, reducing the chances of issues later on. Waiting until the end to handle security often leads to rushed fixes that can be costly and risky (Jones, 2023).

Evaluating risks and weighing the costs of solutions are key steps in deciding what to focus on. By understanding which risks are most likely and could cause the most damage, teams can use their resources better. Although some solutions might seem expensive at first, they often save money in the long run by preventing serious problems (Smith, 2022). This idea ties into secure coding practices, where preventive steps like checking input or securing authentication help stop potential attacks before they happen.

The zero-trust model is another important idea that affects secure coding. Zero trust assumes that threats can come from both inside and outside the system, so it requires strict checks at every step. Using least privilege principles and multi-factor authentication makes sure that every access request is treated as a possible risk, supporting the “no one is safe” mindset found in defense in depth strategies. Adding these principles to code design, like using strong session controls and ongoing identity checks, makes the system even safer.

Regularly reviewing and updating security policies is necessary to keep them effective. Getting external experts to assess the policies each year can reveal weaknesses that might be missed internally. For companies like Green Pace, having detailed policies that cover secure coding, encryption, and ongoing monitoring ensures that security is treated as an ongoing process. These policies should be reviewed and updated as new threats emerge to keep them useful and effective.

In summary, secure coding practices work best when they are included early, checked regularly, and supported by clear policies. By adopting a mindset of always being watchful, as seen in the principles of zero trust and regular policy reviews, organizations can build software that is both secure and reliable.

References

Jones, A. (2023). Early adoption of secure coding standards in DevSecOps. Cybersecurity Journal, 12(4), 34-45.

Smith, B. (2022). Risk assessment and cost-effective security strategies. Information Security Today, 8(2), 22-29.