**CS 405 Journal**

Security should never be an afterthought. Adopting a secure coding standard from the start makes a huge difference in preventing vulnerabilities before they happen. If security is pushed to the end of development, it becomes way harder and more expensive to fix. Best practices like input validation, memory safety, and least privilege access should be part of the coding process from day one. Following industry standards like OWASP and NIST helps make sure applications are built with security in mind.

Risk assessment is just as important. Every security measure has trade-offs, so it’s critical to weigh the risks and costs of different mitigation strategies. Some security features might seem expensive upfront, but when you compare them to the potential damage of a security breach, they’re usually worth it. For example, implementing multi-factor authentication adds an extra step for users, but it significantly reduces the risk of compromised accounts. Zero trust takes security even further by assuming no one is automatically trusted. Every user, device, and connection needs to be verified before gaining access. While it can feel like an inconvenience at times, it’s one of the best ways to prevent unauthorized access and stop lateral movement inside a network.

To enforce security policies, developers need to follow secure coding standards, perform regular security reviews, and integrate automated tools to catch vulnerabilities early. Getting developers on board who might be resistant to zero trust means showing them how these practices actually make their lives easier in the long run. At the end of the day, security is an ongoing process, not a onetime fix, and continuously improving security strategies is the only way to stay ahead of threats.