In software development the statement “don’t leave security to the end” is a call of action to follow best practices by focusing on secure coding during every stage of the software development lifecycle. In planning it requires the analyzing the threat landscape and planning to minimize the risks associated with each threat. In designing a focus on secure coding includes modularization and designing in a way that is test driven and supports the use of differing access levels to limit functionality as much as possible. Testing is required prior to release that includes things like JUnit tests, static analysis, full stack testing, and analysis by outside security experts to ensure that the system meets secure coding guidelines prior to release. After release a focus on secure coding includes penetration testing, monitoring, swift detection of attempted attacks as well as the ability to quickly respond when an attack is detected.

If developers leave security to the end, the measures are not as layered or effective as when developers consider security at each stage. It is like hoping to stop an arterial bleed with a small Band-Aid, it is simply not going to cut it. One way to ensure that security is addressed intrinsically and not left to the end is automated testing that is repeated after any major update to the code. This should include positive and negative tests to ensure that actions that should be successful are and that activities that should be unsuccessful are appropriately blocked. This frequent, repeated testing ensures that changes do not introduce new vulnerabilities and if the changes do it can be quickly addressed prior to deployment when the vulnerability can be discovered and exploited.