Discussion-6 Best practices for designing and executing effective test cases

Discussion Topic:

Please choose one of the following questions to discuss in your initial post:

- What are some of the key challenges of software testing?
- What are some of the best practices for designing and executing effective test cases?
- How can software testing be used to improve the quality of software products?

My Post:

Hello Class,

For this discussion, I chose the “What are some of the best practices for designing and executing effective test cases?” topic.

First, to effectively design and execute test cases, it is important to understand the architecturally significant requirements (ASRs) of the project and then, while focusing on these ASRs, it requires creating artifacts for evaluation, defining a design rubric, and having a plan to help reviewers generate insights, through various activities, so they can form an opinion about the goodness of the architecture (Keeling, 1017).

Focus on Architecturally Significant Requirements (ASRs) implies that test cases should be designed to evaluate the project architecture against these requirements. Creating artifacts implies creating whiteboard sketches, full architecture descriptions, code, and slide-based presentations. Defining a design rubric implies defining criteria with an associated rating scale for rating the fitness of the architecture and how well it aligns with the project’s goals. Having a plan to help reviewers implies having activities such as questionnaires, risk elicitation, and code analysis to help them evaluate the goodness of the architecture. These activities also involve asking good questions, hosting evaluation workshops, and using various testing types.

Note that different testing types serve different purposes. For example, White box testing evaluates the internal structure of the code (CSU Global, 2025). Black box testing evaluates software functionality based on documentation without knowing its internal structure/code/modules. Unit testing can combine white box testing and black box testing. It is typically done by developers, and it tests individual modules in isolation. Integration testing evaluates how modules interact when connected, using approaches like "big bang" or incremental construction. Finally, system testing tests the entire system as a whole.

Other best practices also include balancing cost and value, looking for different kinds of issues, such as unknown problems, and continuous evaluation of the project.

-Alex

References:

CSU Global (2025). *Module 6: Project Testing* [Interactive Lecture]. Canvas. https://csuglobal.instructure.com/courses/110425/pages/module-6-overview?module_item_id=5733358

Keeling, M. (2017). Chapter 12: Give the Architecture a Report Card. *Design it! From programmer to software architect*. Pragmatic Bookshelf. ISBN-13: 978-1-680-50209-1