**UML Design Modeling**

Cindy Cruz Gonzalez

The University of Arizona Global Campus

CST499 Capstone for Computer Software Technology

Instructor Rangitsch

May 4, 2025

**Levels of Software Testing: From Components, Integration, System, Acceptance**

Software testing is a crucial phase in the software development lifecycle, aimed at identifying defects and ensuring the final product meets user expectations and requirements. Various levels of testing—component, integration, system, and acceptance—serve specific purposes and progressively build confidence in the software’s final quality and functionality.

**Component Testing**

Also known as unit testing, component testing is the most granular level of testing. It focuses on individual components or modules of a software system, verifying that each performs as intended in isolation. Typically conducted by developers, this level involves testing methods, functions, or classes to detect bugs early in the development process. Unit tests are often automated and cover scenarios such as boundary values, exception handling, and expected outputs for specific inputs. This form of testing ensures that the building blocks of the software are reliable before they are combined into larger systems (Myers, Sandler, & Badgett, 2011).

**Integration Testing**

Once individual components are validated, integration testing examines the interactions between them. It focuses on ensuring that data flows correctly and that combined components collaborate as expected. Integration testing can follow various strategies, such as top-down, bottom-up, or big bang approaches. In my example, a course enrollment module in an academic system might need to integrate seamlessly with user authentication and course catalog modules. Any mismatch in data types, API mismatches, or improper error handling is most likely to be caught during this phase (Beizer, 1995). Integration testing mitigates risks associated with the integration of third-party components, databases, and internal APIs.

**System Testing**

System testing is a higher-level testing process that validates the complete and integrated software system. It is conducted in an environment that closely simulates real-world usage, allowing testers to evaluate both functional and non-functional requirements. Functional system tests might verify use cases such as user registration or course enrollment. Non-functional tests examine performance, security, and usability. This level is usually managed by a dedicated testing team rather than developers. System testing ensures the software behaves correctly from the end-user’s perspective and is crucial before progressing to acceptance testing (Kaner, Falk, & Nguyen, 1999).

**Acceptance Testing**

Acceptance testing is the final level of testing before the software is delivered to the customer. Its primary objective is to determine whether the system meets business requirements and is ready for deployment. This level may involve user acceptance testing (UAT), where real users interact with the software to validate that it meets their needs and expectations. Acceptance criteria are typically derived from the original software requirements specification (SRS). Successful acceptance testing leads to customer approval and sign-off, while failure may result in rework or delay in deployment.

Each of these levels plays a vital role in ensuring software reliability and user satisfaction. When executed effectively, they reduce the cost of fixing bugs and increase the quality of the final product.

**UML Diagrams**

A diagram of a use case diagram

AI-generated content may be incorrect.

A diagram of a class

AI-generated content may be incorrect.

A diagram of a course

AI-generated content may be incorrect.

**References**

Beizer, B. (1995). *Black-box testing: Techniques for functional testing of software and systems*. John Wiley & Sons.

Kaner, C., Falk, J., & Nguyen, H. Q. (1999). *Testing computer software* (2nd ed.). Wiley.

Myers, G. J., Sandler, C., & Badgett, T. (2011). *The art of software testing* (3rd ed.). John Wiley & Sons.