**UML Design Modeling**

**Manuel Madrid**

**GEN499 Capstone for Computer Software Technology**

**Professor Joseph Rangitsch**

**05/12/2025**

Software testing is a crucial and integral aspect of the software development process. It encompasses functionality, meeting requirements, and defect/bug-free, to make a shippable product. Software quality is improved through software testing, as well as a reduction in development cost, and an ease in maintainability. This paper will highlight some aspects involved in software testing and the attributes which make a well-produced product.

Component testing is a software testing technique that verifies the functionality, usability, and behavior of individual components of an application in isolation to ensure they meet specified requirements. In component testing each component is tested independently before integration with other models. This method is used to help identify defects early in the development cycle, making them easier and cheaper to fix. Components tested in isolation allows developers to concentrate on specific parts of the application and are done using debugging tools or test frameworks. The idea of focusing on individual components ensures their correctness and reliability, making for a strong foundation for integration. Some types of component testing techniques include Unit, Integration, Functional, Performance, and Error Handling Testing.

Integration testing is a systematic technique used to evaluate a software applications interface and interaction between integrated units or modules. Integration testing acts as a bridge in the software development lifecycle between unit testing and system testing. With implementation, developers can detect and rectify integration issues early on, reducing the risks of complex problems at later stages, significantly. A way this testing is often approached is module by module, following a sequence that ensures comprehensive coverage of all integration scenarios. This remains a crucial phase in the software development lifecycle where modules are combined and tested. Various types of integration testing techniques include Big-Bang, Bottom-up, Top-Down, Mixed, and Incremental Integration Testing.

System testing is a level of software testing that evaluates the complete and integrated software system to verify that it meets specified requirements. Software requirements, architecture, design, and codes are verified within system testing, meeting the business requirements and standards. Verifying these software artifacts are error-free and built to specifications are also imperative. System testing is a critical phase in the software development lifecycle that ensures the complete application functions as intended before release. It involved validating both functional and non-functional requirements through a structured process which included [planning, design, execution, and closure. System testing techniques include functional, non-functional, interface, stress, and recovery testing.

Acceptance testing is the process of evaluating a software application to ensure it meets the specified business requirements and user needs before it is released. This involves validating functionality, performance, and usability to confirm that the software is ready for deployment and will satisfy end-users. Acceptance testing is the final stage in the software development lifecycle. It focuses on verifying that the software meets both business requirements and user needs before time of release. If acceptance testing is skipped software may not align with the initial requirements. This just shows the importance of the acceptance testing phase in the software development lifecycle. Some acceptance testing techniques include user acceptance, business acceptance, contract acceptance, regulations acceptance, operational acceptance, alpha and beta testing.

These are four major components involved in software testing, proving it to be a crucial part of the software development process. Functionality, maintainability, and defect/bug-free, are all needed to make a well balanced, finished product. Any process which improves software quality, as well as a reduction in development cost, will be an integral part of the success of a product.

**Different UML Diagrams**

Figure 1

A diagram of a course

Description automatically generated

Figure 2

A graph with a diagram

Description automatically generated with medium confidence

**References**

Devi, K. (2024, December 17). *What is Component Testing? (with Examples) | BrowserStack*. BrowserStack. <https://www.browserstack.com/guide/what-is-component-testing>

Elankumaran, Y., & Elankumaran, Y. (2024, February 15). *What is Integration Testing? A Comprehensive Guide*. ACCELQ. <https://www.accelq.com/blog/integration-testing/>

Das, S. (2024, October 7). *What is System Testing? (Examples, Use Cases, Types) | BrowserStack*. BrowserStack. <https://www.browserstack.com/guide/what-is-system-testing>

Das, S. (2024a, September 24). *What is Acceptance Testing? (Importance, Types & Best Practices) | BrowserStack*. BrowserStack. <https://www.browserstack.com/guide/acceptance-testing>