Week 4: Interactive Assignment, Part IV

Software Design, Implementation, and Testing

Eric Vara

The University of Arizona Global Campus

CST 301: Software Technology & Design

Professor John Howerton

January 26, 2023

**Software Design, Implementation, and Testing**

**Diagram

Description automatically generatedDiagram

Description automatically generated**

**Software Reuse**

Software reuse refers to the use of existing software components, modules, or functions in the development of new software systems. There are three main levels of software reuse. Component-level reuse, which refers to the reuse of pre-existing, low-level components such as libraries, frameworks, or tools. There’s Application-level reuse that refers to the reuse of entire pre-existing applications or modules, as a building block for new applications. Domain-level reuse which refers to the reuse of a common, abstract solution to a problem that has been derived from a particular business or technical domain.

The benefits of software reuse include increased productivity which is Reusing existing code can significantly reduce the time and effort required to develop new software. Improved quality that are reused components generally well tested, debugged, and reliable, which can lead to improved overall quality of the new software. Lastly, there are cost savings which consist of reusing existing software can help reduce development costs, as there is no need to create new code from scratch.

Example: Consider a software development company that specializes in developing e-commerce websites. The company can reuse existing code for common functionality such as shopping cart management, payment processing, and customer management, across multiple e-commerce projects, thus reducing the development time and effort for each new project.

**The Fundamental Configuration Management Activities**

Configuration management (CM) is the process of identifying, organizing, and controlling changes to the software development process and the associated artifacts. The fundamental activities of CM include configuration identification, configuration control, configuration status accounting, configuration auditing, configuration baseline, and configuration change management. These activities are critical for ensuring the integrity and traceability of the software development process, and for ensuring that changes to the product are well controlled and properly documented.

**Host-Target Development Platform**

A host-target development platform refers to a software development environment in which the software is developed on a host computer and then compiled and executed on a target system. The target system can be an embedded system, a single-board computer, or any other system that runs the software.

The host-target development platform provides the following essential capabilities: cross-compilation, debugging, testing, source code management, integration, and deployment. These capabilities allow the software development process to be streamlined and more efficient, by enabling the software to be developed on a more powerful host computer and tested and debugged on the target system.

**Open-Source Development**

Open-source development is a model of software development in which the source code of a software product is made publicly available and can be freely used, modified, and distributed. It has become a widely adopted model for software development, and has led to the creation of many popular and widely-used software products, such as the Linux operating system, the Apache web server, and the Python programming language. Leveraging open-source development can provide many benefits, such as lower costs, access to a large pool of talent, and the ability to leverage existing solutions.

**Software Inspection**

Software inspection is a quality assurance technique that involves reviewing the source code of a software program, usually by a team of peers, to identify defects and improve its quality. This technique helps to complement software testing by identifying problems early in the development process before they become more difficult and costly to fix. Software testing focuses on verifying that a software program meets its specified requirements and works as expected, whereas software inspection focuses on the design and code quality of the software. By combining both techniques, a development team can reduce the number of defects in the final product, save time and resources, and produce higher-quality software.

**Development and User Testing**

Development testing is a software testing process that occurs during the software development lifecycle and is performed by developers, not just by testers. It is aimed at finding and fixing defects as early as possible, reducing the cost of fixing them, and ensuring that the final product meets the requirements and is of high quality. User testing is a type of software testing that focuses on evaluating a software program from the perspective of its intended users. It is aimed at ensuring that the software meets the needs and expectations of its target audience, and that it is user-friendly and easy to use.

**Work Cited**

Y.Y. Ahmaro, I., M. Abualkishik, A., &amp; Mohd Yusoff, M. Z. (2014, June 28). Taxonomy, definition, approaches, benefits, reusability levels, factors and adaption of software reusability: A review of the research literature. Science Alert. Retrieved January 26, 2023, from <https://scialert.net/fulltext/?doi=jas.2014.2396.2421#:~:text=(2010)%20have%20agreed%20that%20the,development%20and%20cut%20operational%20costs>.