Philip Trinh

SNHU/ CS-320

04 July 2023

CS 320 Module Two Journal

To put it in simple terms, static testing technique that examine a software system or component without running any code. Normally static testing is done manually but an analysis may be done automatically with tools. Static testing is used to find the cause of flaws, deficiencies, and prevent future problems since the earlier the problem is found the quicker and less expensive it will be to fix it. The test involves evaluating the software's requirements, design, and source code. Static testing reviews include informal, walkthroughs, technical reviews, and inspection listed in order from low to high in level of formality.

On the other hand, dynamic testing involves running the software to verify its operation and behavior. It focuses on executing the program and evaluating its functionalities in accordance with predefined test cases and anticipated results. Some examples of dynamic testing are unit testing, integration testing, system testing, performance testing, and acceptance testing.

The most obvious different between static and dynamic testing is in their name, static testing examines the software without running the code while dynamic requires software to be running. Static testing is normally performed early in the development phase before code execution or often before any code is written so the developers can ensure the foundation of the software such as requirement, specification are solid before moving on to plan and write code. Dynamic testing focuses more on the performance and functionality aspects of the software. Running dynamic testing helps in uncovering functional defects and ensures that the software meets the specified requirements.

It is important to use both static and dynamic testing techniques because they provide a more comprehensive approach with static testing focuses on structural defects while dynamic testing focuses on functional aspects, performance, and behaviors. By combining the two testing techniques, the developers can further reduce overall risk associated with software deficiencies, defects, save time on fixing major bugs later or deliver a failure product to the clients. Additionally, thorough testing and recording can help to boost the confidence of clients in the software performance, help establish a strong background, and to identify or isolate problems that can occur later.

References

*SOFTWARE TESTING : an istqb-bcs certified tester foundation guide.* (2019). Bcs, The Chartered Instit.

‌