**Dynamic and Static Testing**

Jackson Peterson

CS-320-H7020

Professor Mike Prasad

**Static Testing:**

Static Testing involves reviewing code, design, and documentation without running the program. It’s about inspecting software artifacts to find problems early in development. Techniques include code reviews, walkthroughs, inspections, and desk checks. The aim is to improve software quality and reliability by catching issues before the code is executed.

**Dynamic Testing:**

Dynamic Testing, in contrast to Static Testing, entails running the software to observe how it behaves and assess its performance. This method evaluates the software’s functionality, responsiveness, and its capacity to meet specified requirements. Dynamic Testing encompasses practices such as unit testing, integration testing, system testing, and acceptance testing, with the goal of pinpointing defects that arise during software execution.

**Differences:**

Static Testing occurs early in software development during the design and coding phases, involving a review of software artifacts such as source code and design documents, primarily aimed at identifying issues related to coding standards and design flaws. Dynamic Testing takes place after the software is fully developed, evaluating its behavior through actual execution and focusing on spotting functional defects, runtime errors, and performance issues. Static Testing relies on manual inspection and code analysis tools, while dynamic testing employs automated testing tools and test cases, making it an essential combination to ensure software quality and reliability throughout the development process.

**Importance:**

The importance of using both static and dynamic testing is clear. Static testing helps spot issues early, reducing the cost and effort of fixing defects, while dynamic testing finds defects that only show up during execution. Together, they ensure thorough testing, with static testing improving code quality and dynamic testing checking if the software works correctly. This combo also lowers development risks, as static testing deals with design and code problems, while dynamic testing looks for runtime issues that could affect users. This leads to better software quality and compliance with industry standards and regulations, which often require both static and dynamic testing for reliable and safe software.