CS-320-T4208 Software Test Automation & QA

Module Two Journal – Dynamic & Static Testing

Computer Science Department

Southern New Hampshire University

Eric Farkas

3/12/2023

Testing is an important aspect performed throughout the software development lifecycle. Although it is all called “testing,” different methods and techniques are employed throughout the development. There are many names associated with these different methods. Some are synonymous with others and some methods have a specific focus. Examples of these are function & non-functional, white box and black box, verification and validation, but the focus of this assignment is static and dynamic. Understanding each method’s advantages and how they are used together under a holistic testing strategy is essential for a thorough test strategy.

Static testing is a testing method that reviews the code and documentation. It is important to note that this review is done without the execution of the code. This means it can be done early in the process and parallels the development. It is common for static testing to reference design documents, user requirements, test cases and other resources related to the development function. Static testing reduces development costs by identifying code errors, quality and functional issues early in development, reducing the required rework. Although many methods exist to perform static testing, most involve review and static analysis phases. The review phase focuses on ambiguity, quality and implementation issues within the design documentation. Static analysis is the first pass review of the code and focuses on function and structural defects. Automating some of the static analysis is possible by leveraging built-in tools or plugins for your integrated development environment. Static testing is often called verification testing or non-execution testing.

Dynamic testing is a testing method that reviews the code’s behavior. This testing comes later in development because the code must be executable to review its behavior. Dynamic testing can take longer than static testing, often including functional and non-functional requirements. The longer test cycles can often be offset by using automated test tools typically deployed in a continuous integration environment. Dynamic testing is often categorized by white box testing and black box testing. White box testing is performed by a development team member familiar with the design and structure. The focus of white box testing is usually performance. Black box testing is performed by a tester with no knowledge of the design and structure. This testing focuses on functionality and does not require working code knowledge. This is most common in usability and acceptance testing but may also be leveraged for integration, security and system testing. Dynamic testing is crucial because it detects errors too complicated to be discovered in the static review. It provides the test team with a complete end-to-end view of the development and functionality, which is ideal for acceptance and security sign-off. This is why dynamic testing is generally synonymous with validation testing.

Although static and dynamic testing are complementary methods and essential to complete a software development lifecycle, some distinguishing differences are worth pointing out. The first is where they occur in the SDLC. Static testing can be performed early, even before any code is produced, whereas dynamic must be late as it requires executable code. The next key difference is the purpose of testing. Static testing is about verification, whereas dynamic testing is about validation. Although these terms sound similar, verification is about meeting needs and ensuring that what you plan to build will do so. Validation is about meeting specifications and building a product accordingly. This is why static testing is often called a preventative process, while dynamic testing is often called a curative one. The last key difference is the tools used in the execution. Static testing is visual and uses spreadsheets and task lists for execution. Dynamic testing tends to be more automated and requires test cases, scripts, applications and injection tools.

Both methods are important to the software development lifecycle. Static testing removes ambiguity, ensures quality and reduces development time. It challenges requirements to ensure they are properly defined and result in proper output before any code is developed. This reduces rework and lowers costs. It also ensures that best practices and special requirements are identified and planned. This happens early in the development process and often before any code has been written. Dynamic testing starts near the end of development because it requires executable code. It is important because it validates that the application accomplishes the task it set out to without errors. This testing may extend to additional systems if integration is required. It is a holistic view of the system, its integrations, security, user interface and other aspects. If static testing were synonymous with design quality, dynamic testing would be synonymous with product quality. You need a great design and execution to build a quality system. Testing will ensure that both phases are executed properly and without errors.