Victor Feight

CS320 Software Test and Automation

Dynamic vs Static Testing

* What is static testing?

According to this week’s reading, static testing involves using techniques that test software without actually executing it. The general idea is to find errors or defects in the code before it is executed. The text mentions “static analysis tools”, which are automatic tools that enable code to be analyzed for programmatic weaknesses. Such weaknesses may lead to defects, which can cause failures during system use. Static testing can be considered as the verification stage.

* What is dynamic testing?

Dynamic testing is done while code is executed, and can be broken into categories of black box testing and white box testing. Black box testing involves techniques based on analysis of the specification – “including both functional and non-functional aspects” (Graham). White box testing involves testing the code, using techniques such as equivalence partitioning, boundary value analysis, decision table testing, state transition testing, and use case testing. Dynamic testing can be considered as the validation stage.

* What are the differences between static and dynamic testing?

The main difference is that static testing is done quite early in the lifecycle to prevent defects from occurring, without executing any code. The test basis and source code can be reviewed early on. Dynamic testing checks nonfunctional and functional programmatic features, including unit testing, integration testing, and system testing. In static testing the cost of finding defects are low, whereas the cost of finding defects are high in dynamic testing. Static testing can be performed on source code whereas dynamic testing is performed after compiling the code.

* Why is it important to use both static and dynamic testing?

By utilizing static testing, ambiguities or errors can be cleared from test basis and requirement specifications, which can prevent the bug from being propogated up the testing lifecycle. For each stage in the testing lifecycle, the cost of discovering and fixing a bug is exponentially higher, thus it is beneficial to find it earliest in the test stage as possible. As an example from the book, if such a bug was found during the dynamic testing stage, the team would “incur the cost of creating/testing code, diagnosing source of defect, correcting the problem, and rewriting code to eliminate that defect.”

Works Cited

Graham, D., Black, R., & Veenendaal, R. V. (2020). *Foundations of software testing: ISTQB Certification*. Andover, Hampshire: Cengage Learning, EMEA.