**Brandon Hobbs**

**CS-320**

**June 6, 2022**

**Module 2 Journal Entry**

Static and dynamic testing are both valuable tools in the testing of software. Sometimes running the code is not needed to debug code but executing the code does make it easier to locate the bug to begin with. To this end, static testing is done without executing the code and dynamic testing execute the code.

Not only do static testing differ in if the code is executed, they have different goals. Geeks for Geeks (2020) shows this handy table as a comparison:

| **Static Testing** | **Dynamic Testing** |
| --- | --- |
| It is performed in the early stage of the software development. | It is performed at the later stage of the software development. |
| In static testing whole code is not executed. | In dynamic testing whole code is executed. |
| Static testing prevents the defects. | Dynamic testing finds and fixes the defects. |
| Static testing is performed before code deployment. | Dynamic testing is performed after code deployment. |
| Static testing is less costly. | Dynamic testing is highly costly. |
| Static Testing involves checklist for testing process. | Dynamic Testing involves test cases for testing process. |
| It includes walkthroughs, code review, inspection etc. | It involves functional and nonfunctional testing. |
| It generally takes shorter time. | It usually takes longer time as it involves running several test cases. |
| It can discover variety of bugs. | It expose the bugs that are explorable through execution hence discover only limited type of bugs. |
| Static Testing may complete 100% statement coverage in comparably less time. | While dynamic testing only achieves less than 50% statement coverage. |

Static testing is good for debugging (or preventing) the issues determined by dynamic testing. For example, during dynamic testing it might be determined that a certain business rule is being violated in some instances but not the underlying reason for the failure. Static testing, during a code inspection process, might have highlighted that there was a conditional statement that was not written to specification, e.g., daysPerWeek > 48 vs. daysPerWeek >= 48.

In this case, static testing could have prevented the bug from ever making it downstream.

Static testing could also highlight poor coding practices, overusing global variables, or code not meeting institutional coding practices, e.g., camel case vs. Pascal case vs snake case, etc.

What static testing cannot accomplish is testing of non-functional requirements. For example, if there was a non-functional requirement about performance (login time must be less than 10 seconds) static testing cannot verify this. Dynamic testing could.

Both methods have their place within the SDLC and should be employed as early and often as possible.

GeeksforGeeks. (2020, 27 Feb.). *Difference between static and dynamic testing*. https://www.geeksforgeeks.org/difference-between-static-and-dynamic-testing/