Landen Thurston

March 22, 2025

CS – 320

Module two: Journal

When testing code there are usually two types of main testing that can help you progress your code to make it as cohesive and clean as possible. The two types of testing that are used the most by developers are static and dynamic testing. Both are very different but bring a unique and helpful way to test your own code.

Starting with static testing, this type of testing is not based on actually running your code but more on looking and the quality and syntax of the code. When you put your code through a static test you are taking time to look through your code to find any issues that may cause problems later. This type of testing is usually performed earlier in the cycle to catch any big problems early. When you are going through static testing you are not necessarily checking to see if the code will execute but looking to see if the code is built correctly. This type of testing can be done manually or there are automated tools that can help you.

On the other side of testing your code there is dynamic testing. This type of testing is more of a software approach where you will execute your code to see if you are getting the results that you are expecting. It’s not about just the structure anymore but more about runtime, performance, and functionality. This type of testing will identify defects in your code that may cause a problem when ran. Some types of dynamic testing are Unit testing, system testing, and performance testing.

The main differences between dynamic testing and static testing is dynamic testing relies on running the code and compiling to check its functionality while static testing is more early process verification that can be done without running the code. It is important to use both of these because they both help your code throughout the development process. Static testing is crucial in the early process and helps clean up any big errors in your code and keeps your code clean early in the process. Dynamic is used more in the latter half of the process ensuring your code is doing what it was built to do. The complement each other as they help your code get to where it needs to be.