Static testing is a form of software testing. This form of testing does not execute any code; it reviews the code or the requirements. There are several benefits to this, to name a few: Early detection of defects can help reduce cost and effect in the long run. It can also reduce rework by catching defects early. It allows for more efficient development. It also improves the team by encouraging collaboration with reviews and more effective communication.

The dynamic test is done in a coding phase. It executes the code and tests its behavior. It tests functionality, performance, and behavior. There are two main types of testing: white-box testing and black-box testing. Black-box testing only focuses on input and output—white-box testing test’s internal structure. Dynamic testing is unit, integration, and system testing. Benefits: it detects runtime issues, improves software quality, and checks security. All these together increase user satisfaction. It also enhances the performance and quality of software.

When it comes to similarities, they work together towards the same goal. They both detect defects in the software. Both help with cost reduction because the team can catch issues sooner rather than later. Overall, both help test for different errors and make higher-quality software.

There are quite a few differences between static and dynamic testing. First, they test differently. Static testing does not test code, while dynamic testing executes the code. The test's timing varies, one before the code and the other during or after. One of the most is what they are testing once is the design, and the other is the software behavior. Lastly, the tools used are different, like static using static analysis tools like Aikido. Dynamic uses tools like JUnit or manual tests. Overall, they are vastly different but essential.

References:

Morgan, P., Samaroo, A., Thompson, G., & Williams, P. (2019). *Software testing : An istqb-bcs certified tester foundation guide - 4th edition*. BCS Learning & Development Limited.

Korando, B. (2018, August 20). Dynamic testing in JUnit 5; A practical guide - Billy Korando - medium. *Medium*. https://medium.com/@BillyKorando/dynamic-testing-in-junit-5-a-practical-guide-a57e3ceaa240

Zelleke, L., & Zelleke, L. (2024, May 10). *The best static code analysis Tools*. Comparitech. https://www.comparitech.com/net-admin/best-static-code-analysis-tools/