CS-320 Software Test, Automation QA

Huan Ai

9/14/2025

What is static testing?

Static testing is checking software for defects without actually running the code. It's like proofreading a document before you print it. You examine the work products, such as requirements documents, design plans, code, or test cases, by looking at them. The main types are reviews where people examine documents, and static analysis where tools analyze code for patterns that could lead to errors (Brian, 2015).

What is dynamic testing?

Dynamic testing is the more common type of testing where you actually run the software with test data. You put inputs into the program, execute it, and check the outputs against what you expected to happen. It's used to find bugs in the code's behavior and to see if the software meets its requirements.

What are the differences between static and dynamic testing?

The main difference is all about execution. Static testing is done without executing the program, while dynamic testing requires execution. In addition, static testing involves proofreading documents, design plans, test cases, etc.. Dynamic testing is more about testing the software itself.

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

It's important to use both because they find different types of problems at different times, and finding a bug early saves a lot of time and money. Static testing helps to find bugs early. Catching a mistake in a requirements document is much cheaper to fix than catching that same mistake after it has been coded and tested. It prevents the bug from ever being built into the software in the first place. Dynamic testing finds bugs in the running code. It's the only way to see how the software actually behaves and to find bugs that only show up when the code is executed.

**Reference**

Brian Hambling (2015). Software Testing : An ISTQB-BCS Certified Tester Foundation Guide - 4th Edition. https://ebookcentral.proquest.com/lib/snhu-ebooks/detail.action?docID=5837074