

Software Testing

Why Test?

We test software to ensure it achieves what it is supposed to. The primary approach to testing is to feed in a specific input and assert a certain output is achieved.

The testability of a piece of software is a gauge of how well a system displays any faults when under test.

Types of testing

Verification - Testing that the system meets the specification that has been defined.

Validation - Testing that the system fulfills the client's needs

If the testing shows that verification passes but validation does not then it highlights that the solution/design did not accurately match the client's problem.

There are two ways to approach testing a piece of software, blackbox or white box. With black box testing, the code itself and how it operates is hidden, whereas with white box testing the operation of the code is known. White box testing can allow testing that targets the code better. White box testing can also lead to more fragile tests though, if the implementation changes the tests can also break.

Automatic testing is testing that does not need manual human intervention.

Code smells

Code can 'smell' even if a problem has not occurred. A code smell is a problem with the code but it may indicate a deeper problem.

Approaches to testing

Functional testing is a type of blackbox testing that tests the functional requirements.

Usability Testing tests that the nonfunctional requirements are achieved. Usability is a measure of how easy it is for a user to achieve what they want with the system.

Compatibility Testing, tests that the software will run in the various environments (hardware, operating system) it is supposed to.

Regression testing tests that change to one part of the software do not break anything else in the system.

Performance testing tests that the system achieve its performance targets i.e., responsiveness to user interactions.

Big Bang Testing test the entire system as a whole which in turns tests all the components.

A vulnerability test sees what vulnerabilities exists whereas a penetration tests see's what's the damage of exploiting such vulnerabilities.

Testing strategy's

Specification testing. Specification testing is a type of backbox testing that ensure software achieves what it is specified to do. Three of the techniques for this type of testing are Equivalence Class, Partitioning testing and Equivalence Class Partitioning.

Cyclomatic complexity

Cyclomatic complexity is a metric that indicates how complex a program is. The measurement was developed by Thomas J. McCabe. Higher CC values indicates that a program that is more complex. Cyclomatic complexity can be calculated by working out the maximum number of 'paths' through the code. Every control flow statement in a program increases the number of paths through the code. The minimum cyclomatic complexity is 1, where there is only one path and no control statements, the maximum would be infinite.

The cyclomatic complexity measurement aims to provide an objective measurement for software complexity. The cyclomatic complexity can be compared with the number of test cases to assess the test coverage. Thomas J. McCabe originally thought that the CC measurement could be used to help identify when a 'module' of code should be split. Thomas J. McCabe recommended that if a module had CC greater than 10 then the module should be divided into smaller modules.

The effectiveness of cyclomatic complexity as a measurement has been questioned with academics wondering if it is any more effective a measurement than counting the number of lines of code.