CS5002NP

Software Engineering

Testing Approaches

- Conducted based on two Approaches
  - Functional testing
  - Implementation testing

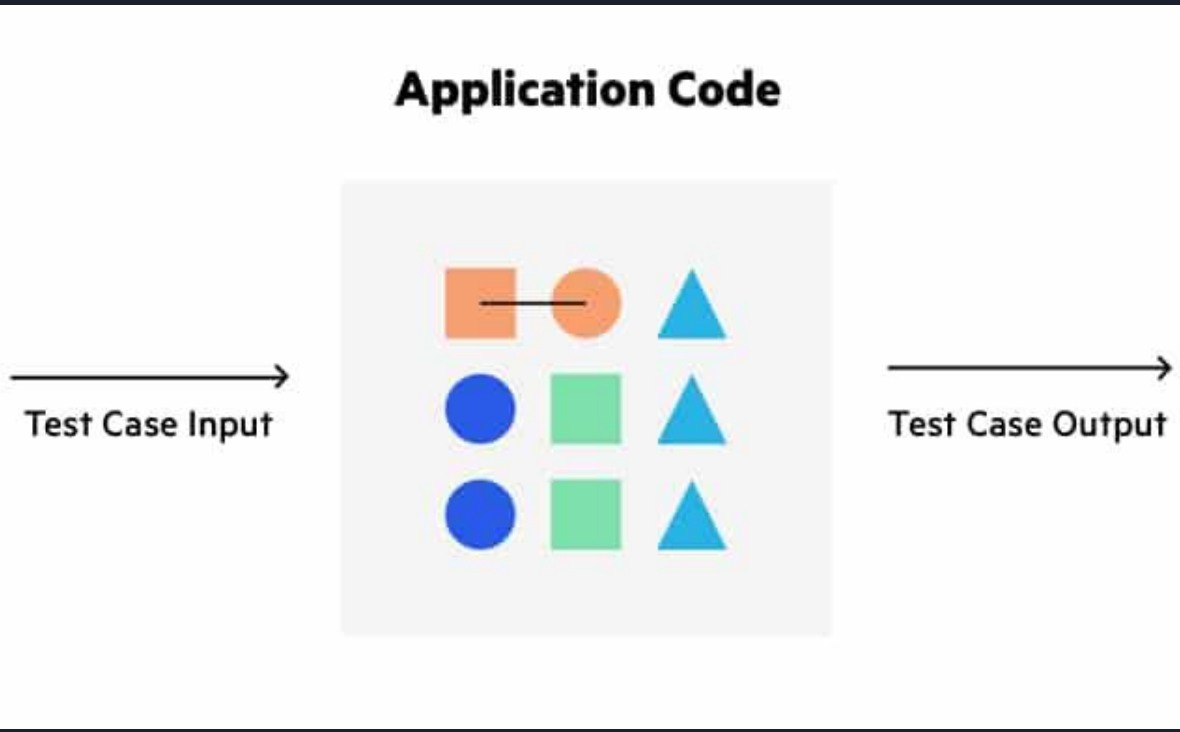
Agendas

- White Box Testing
- Black Box vs White Box Testing

White Box Testing

- Approach that allows testers to inspect and verify the inner workings of a software system - its code, infrastructure and integrations with external systems

- Often referenced in the context of Static Application Security Testing, an approach that check source code or binaries automatically and provides feedback on bugs and possible vulnerabilities


Application Code
Test Case Input
Test Case Output

White Box Testing

- known as glass box testing, structural testing, clear box testing, open box testing and transparent box testing

- The ability to see through the software’s outer shell into its inner workings.

White Box Testing - Pros

- Ability to achieve complete code coverage
- Easy to automate
- Reduces communication overhead between testers and developers
- Allows for continuous improvement of code and development practices

White Box Testing - Cons

- Requires a large effort to automate
- Sensitive to changes in code base, automation requires expensive maintenance
- Cannot test expected functionality that does not exist in the code base
- Cannot test from the user’s perspective

White Box Testing - Steps

- Design all test scenarios, test cases and prioritize them according to high priority number.
- This step involves the study of code at runtime to examine the resource utilization, not accessed areas of the code, time taken by various methods and operations and so on.
- In this step testing of internal subroutines takes place. Internal subroutines such as nonpublic methods, interfaces are able to handle all types of data appropriately or not.

White Box Testing - Steps

- This step focuses on testing of control statements like loops and conditional statements to check the efficiency and accuracy for different data inputs.

- In the last step white box testing includes security testing to check all possible security loopholes by looking at how the code handles security.

White Box Testing - Types

- Unit Testing
  - tests written as part of the application code, which test that each component is working as expected

- Mutation Testing
  - checks the robustness and consistency of the code by defining tests, making small, random changes to the code and seeing if the tests still pass

White Box Testing - Types

- Integration Testing
  - tests specifically designed to check integration points between internal components in a software system, or integrations with external systems

- White box penetration Testing
  - an ethical hacker acts as a knowledge insiders, attempting to attack an application based on intimate knowledge of its code and environment

White Box Testing - Types

- Static code analysis
  - Automatically identifying vulnerabilities or coding errors in static code, using predefined patterns or machine learning analysis

White Box Testing - Focus

- Security gaps and vulnerabilities
  - checking to see if security best practices were applied when coding the application, and if the code is vulnerable to known security threats and exploits
- Broken or poorly structured paths
  - identifying conditional logic that is redundant, broken or inefficient

White Box Testing - Focus

- Expected output
  - executing all possible inputs to a function to see if it always returns the expected output
- Loop testing
  - checking single loops, concatenated loops and nested loops for efficiency, conditional logic and correct handling of local and global variables
- Data Flow testing
  - tracking variables and their values as they pass through the code to find variables that are not correctly initialized, declared but never used, or incorrectly manipulated

Testing Techniques and Code Coverage

- White Box Testing Goal - cover the source code as comprehensively as possible

- Metric that shows how much of an application’s code has unit tests checking its functionality

Testing Techniques and Code Coverage

- Within code coverage, it is possible to verify how much of an application’s logic is actually executed and tested by the unit test suite, using concepts like
  - Statement Coverage
  - Branch Coverage
  - Path Coverage

Statement Coverage

- All executable statements in the code are run and tested at least once
- For example, if there are several conditions in a block of code, each of which is used for a certain range of inputs, the test should execute each and every range of inputs, to ensure all lines of code are actually executed

Statement Coverage

- Helps uncover unused statements, unused branches, missing statements that are reference by part of the code, and dead code left over from previous versions

Branch Coverage

- Maps the code into branches of conditional logic, and ensures that each and every branch is covered by unit tests

- Tester identifies all conditional and unconditional branches and writes code to execute as many branches as possible

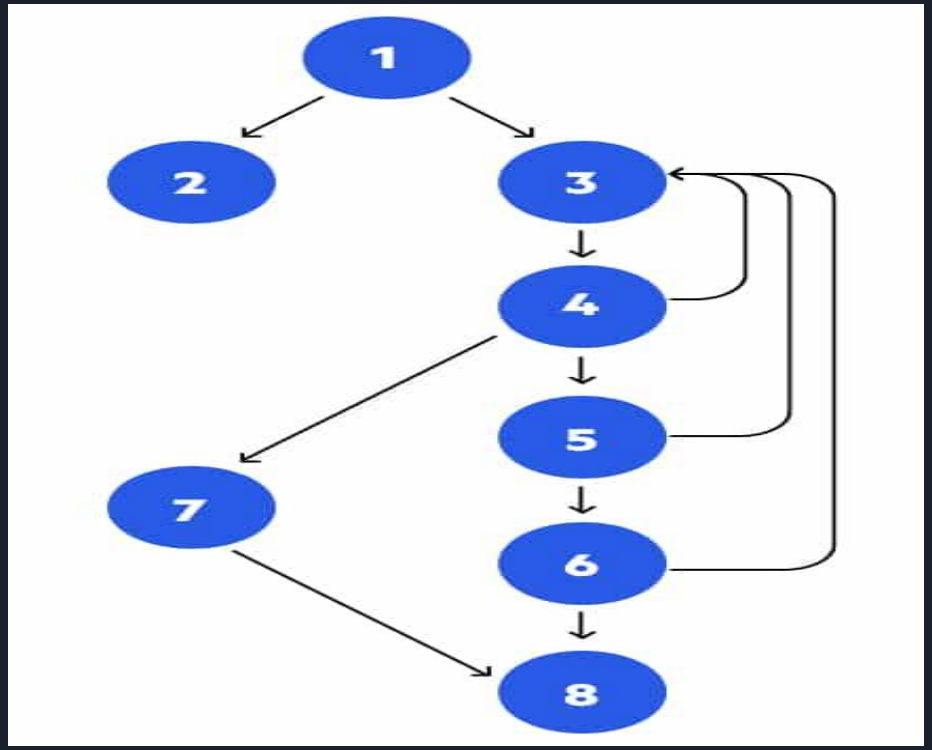
```
if X then..
   if Y then..
      A
      B
   else
      if Z then..
         C
      else..
         D
```

Branch Coverage

- A, C, and D are conditional branches, because they occur only if a condition is satisfied
- B is an unconditional branch, because it always executes after A

Path Coverage

- Concerned with linearly independent paths through the code
- Testers draw a control flow diagram of the code for testing


1
2
3
4
5
7
6
8

Path Coverage

- The tester writes the unit tests to execute as many paths as possible through the program’s control flow
- Identify paths that are broken, redundant or inefficient

Black Box vs White Box Testing

- Black box testing is used to test the software without knowing the internal structure of code or program. While white box testing tests internal structures of the software.

- Black box testing is carried out by testers, while white box testing is generally carried out by developers

Black Box vs White Box Testing

- Black Box testing is applicable on higher levels of testing like System Testing, and Accepting Testing whereas White Box testing is application on lower levels of testing like Unit Testing, and Integration Testing.

- Black Box testing aims at checking on functionality performed under the test while While Box testings aims and checking how the system performs.

Black Box vs White Box Testing

- Programming and implementation knowledge is not required to carry out Black Box testing while its important to have knowledge of both programming and implementation to carry out White Box Testing

- Black Box testing means functional test or external test whereas White Box testing means structural test or interior testing

THANK YOU!