Software testing fundamentals

Outline

- Software testing fundamentals
  - Goal of testing
  - Testability definition
  - Testability characteristics
  - Attributes of good test
- Internal views of testing
- External views of testing
- Difference between white and black box testing

Software testing fundamentals

- Goal of testing : The goal of testing is to find errors, and a good test is one that has a high probability of finding an error.
- Testability Definition : "*Software testability is simply how easily [a computer program] can be tested.*"
- Testability characteristics :
  - *Operability. "The better it works, the more efficiently it can be tested."*
  - *Observability. "What you see is what you test."*
  - *Controllability. "The better we can control the software, the more the testing can* be automated and optimized."
  - *Decomposability. "By controlling the scope of testing, we can more quickly* isolate problems and perform smarter retesting."
  - *Simplicity. "The less there is to test, the more quickly we can test it."*
  - *Stability. "The fewer the changes, the fewer the disruptions to testing."*
  - *Understandability. "The more information we have, the smarter we will test."*
- Attributes of a "good" test:
  - *A good test has a high probability of finding an error.*
  - *A good test is not redundant.*
  - *A good test should be "best of breed".*
  - *A good test should be neither too simple nor too complex.*

Internal and External views of testing

- *Internal views of testing :* Knowing the specified function that a product has been designed to perform, tests can be conducted that demonstrate each function is fully operational while at the same time searching for errors in each function.
- *External views of testing :* Knowing the internal workings of a product, tests can be conducted to ensure that internal operations are performed according to specifications and all internal components have been adequately exercised.

Internal and External views of testing

- Black-box testing are conducted at the software interface.
- A black-box test examines some fundamental aspect of a system with little regard for the internal logical structure of the software.

- White-box testing of software is predicated on close examination of procedural detail.
- Logical paths through the software and collaborations between components are tested for specific sets of conditions and/or loops.
- Define all logical paths, develop test cases and evaluate results.
- For even small programs, the number of possible logical paths can be very large.