CS5002NP

Software Engineering

Agendas

- Software Testing
- Types of Software Testing

Software Testing

- **Definition:**
  - Process of exercising a program with the specific intent of finding errors prior to delivering it to the end user

- Involves execution of software components using manual or automated tools to evaluate one or more properties of interest

Software Testing

- **Purpose:**
  - To identify errors, gaps or missing requirements in contrast to actual requirements

IS CORRECTING BUGS PART OF SOFTWARE TESTING?

Why Software Testing is Important?

- Any bugs or errors in the software can be identified early and can be solved before delivery of the software
- Ensures reliability, security and high performance, which further results in time saving, cost effectiveness and customer satisfaction

Why Software Testing is Important?

- Nissan cars recalled over 1 million cars from the market due to software failure in the airbag sensory detectors. There has been reported two accident due to this software failure.

- Starbucks was forced to close about 60 percent of stores in the U.S and Canada due to software failure in its POS system. At one point, the store served coffee for free as they were unable to process the transaction.

Why Software Testing is Important?

- In April 2015, Bloomberg terminal in London crashed due to software glitch affected more than 300,000 traders on financial markets. It forced the government to postpone a 3bn pound debt sale.
- In April of 1999, a software bug caused the failure of a $1.2 billion military satellite launch, the costliest accident in history
- In May of 1996, a software bug caused the bank accounts of 823 customers of a major U.S. bank to be credited with 920 million US dollars.

Software Testing - Benefits

- Cost-effective:
  - Testing any IT project on time helps you to save your money for the long term. In case if the bugs caught in the earlier stage of software testing, it costs less to fix.

- Security:
  - It is the most vulnerable and sensitive benefit of software testing. People are looking for trusted products. It helps in removing risks and problems earlier.

Software Testing - Benefits

- Product quality:
  - It is an essential requirement of any software product. Testing ensures a quality product is delivered to customers.

- Customer Satisfaction:
  - The main aim of any product is to give satisfaction to their customers. UI/UX Testing ensures the best user experience.

Software Testing

- Comprises of 2 components
  - Validation
  - Verification

Validation

- Validation ensures the product under development is as per the user requirements.
- Validation answers the question – "Are we developing the product which attempts all that user needs from this software?".
- Validation emphasizes on user requirements.

Verification

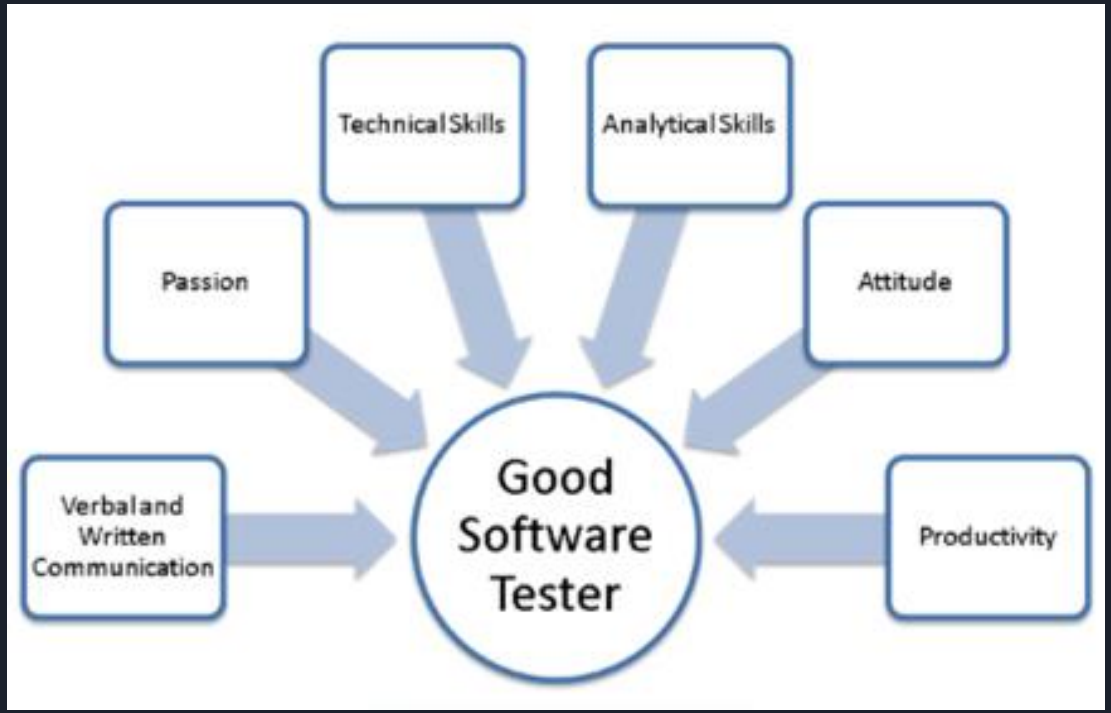
- Verification ensures the product being developed is according to design specifications.
- Verification answers the question– "Are we developing this product by firmly following all design specifications?"
- Verifications concentrates on the design and system specifications.

Targets of a Test

- Errors - These are actual coding mistakes made by developers. In addition, there is a difference in output of software and desired output, is considered as an error.
- Fault - When error exists fault occurs. A fault, also known as a bug, is a result of an error which can cause system to fail.
- Failure - failure is said to be the inability of the system to perform the desired task. Failure occurs when fault exists in the system.

Who Tests the Software?

- Developer
  - Understands the system but will test “gently” and is driven by “delivery”
- Independent Testers
  - Must learn about the system, but will attempt to break it and is driven by “quality”


Technical Skills
Analytical Skills
Passion
Attitude
Verbal and Written Communication
Good Software Tester
Productivity

Manual vs Automated Testing

- Manual Testing:
  - The software tester prepares test cases for different sections and levels of the code, executes the tests and reports the result to the manager
  - Manual testing is time and resource consuming. The tester needs to confirm whether or not right test cases are used. Major portion of testing involves manual testing.

Manual vs Automated Testing

- Automated Testing:
  - This testing is a testing procedure done with aid of automated testing tools. The limitations with manual testing can be overcome using automated test tools
  - There are software and hardware tools which helps tester in conducting load testing, stress testing, regression testing.

Testing Approaches

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

Functional Testing

- Known as Black-box Testing or Behaviour Testing

- Functionality of the software application are tested without having knowledge of internal code structure, implementation details and internal paths

Functional Testing

- Focused on input and output of software applications
- Based on software requirements and specifications
- Black-box testing can be done on any software you want to test, for example, an operating system like Windows, a website like Google, a database like Oracle or even your own custom application

Steps for Black Box Testing

- Initially, the requirements and specifications of the system are examined
- Tester chooses valid inputs (positive test scenario) to check whether system processes them correctly. Also, some invalid inputs (negative test scenario) are chosen to verify that the system is able to detect them
- Tester determines expected outputs for all those inputs

Steps for Black Box Testing

- Software tester constructs test cases with the selected inputs.
- The test cases are executed.
- Software tester compares the actual outputs with the expected outputs.
- Defects if any are reported, fixed and re-tested.

Black Box Testing - Types

- Functional testing – related to the functional requirements of a system; it is done by software testers.
- Non-functional testing – not related to testing of specific functionality, but non-functional requirements such as performance, scalability, usability.
- Regression testing – done after code fixes, upgrades or any other system maintenance to check the new code has not affected the existing code.

Black Box Testing - Techniques

- Syntax Driven Testing
- Equivalence Partitioning
- Boundary Value Analysis
- Cause Effect Graphing
- Requirement Based Testing
- Compatibility Testing

Equivalence Partitioning

- Many type of inputs work similarly so instead of giving all of them separately we can group them together and test only one input of each group.

- The idea is to partition the input domain of the system into a number of equivalence classes such that each member of class works in a similar way, i.e., if a test case in one class results in some error, other members of class would also result into same error.

Equivalence Partitioning

- Involves two steps:
  - Identification of equivalence class
  - Generating test cases

Identification of equivalence class

- Partition any input domain into minimum two sets:
  - valid values
  - invalid values.

- For example, if the valid range is 0 to 100 then select one valid input like 49 and one invalid like 104

Generating test cases

- To each valid and invalid class of input assign unique identification number
- Write test case covering all valid and invalid test case considering that no two invalid inputs mask each other

Boundary Value Analysis

- Boundaries are very good places for errors to occur. Hence if test cases are designed for boundary values of input domain then the efficiency of testing improves and probability of finding errors also increase.

- For example – If valid range is 10 to 100 then test for 10, 100 also apart from valid and invalid inputs.

Cause Effect Graphing

- This technique establishes relationship between logical input called causes with corresponding actions called effect.

- The causes and effects are represented using Boolean graphs. The following steps are followed:
  - Identify inputs (causes) and outputs (effect)
  - Develop cause effect graph
  - Transform the graph into decision table
  - Convert decision table rules to test cases

Requirement based Testing

- It includes validating the requirements given in SRS of software system

Compatibility Testing

- The test case result not only depend on product but also infrastructure for delivering functionality.
- When the infrastructure parameters are changed it is still expected to work properly
- Some parameters that generally affect compatibility of software are:
  - Processor (Pentium 3,Pentium 4) and number of processors
  - Architecture and characteristic of machine (32 bit or 64 bit)
  - Back-end components such as database servers
  - Operating System (Windows, Linux, etc)

THANK YOU!