Software Testing

Testing Concepts

Why should we do Testing?

- What is “Success” for a Tester?
  - Proving that Software Works – NO
  - Improving the quality of Software – NO
  - Finding max. defects #
- Tester needs a different mind-set. Mindset of destroying a software with intent of finding max. defects ##

How much should we Test

- Till all bugs are found?
  - No. Why?
  - There is nothing like a bug free software.
  - Example: Puzzle – A person starts travel from City A to City B. First day he travels half the distance. Second day, half of the remaining. Third day, half of the remaining distance, and continues like this. When will he reach City B?
    - Never
  - Similarly, Testing with intent of finding all defects will take Infinite time. As, there is nothing Perfect in this world. #
- How much to Test, depends on Criticality of the application,
  - Applications with less criticality – Around 20-30% of the SDLC time
  - Products – around 50-60% of the total SDLC time
  - Mission Critical applications – 100% to 1000% or more. Examples,
    - NASA's space mission
    - Railway signaling system
    - Medical Electronics – Pacemaker ##

Introduction

- **Testing is the process of executing a program with the intent of finding errors.**
- **Testing is a process used to help identify the correctness, completeness and quality of a developed computer software.**
- **Testing helps is Verifying and Validating if the Software is working as it is intended to be working.**

Need for Testing

- Contribute to the delivery of higher quality software product.
- Undetected errors are costly to detect at a later stage.
- Satisfied users and to lower maintenance cost.

How Testing is conducted

- By examining the users' requirements.
- By reviewing the artifacts like design documents.
- By examining the design objectives.
- By examining the functionality.
- By examining the internal structures and design.
- By executing code.

Principles

- **Economics of Testing**
  - It is both the driving force and the limiting factor.

- **Driving - Earlier the errors are discovered and removed in the lifecycle, lower the cost of their removal.**

- **Limiting - Testing must end when the economic returns cease to make it worth while i.e. the costs of testing process significantly outweigh the returns.**

Exhaustive Testing

- **Exhaustive Testing**
  - Testing every possible input over every possible output
  - Can use every possible input condition as a test case.
  - Is Exhaustive Testing feasible?
    - E.g. Online railway reservation system.
    - Too complex to create test cases to represent all valid and invalid cases of source and destination cities.

Exhaustive Testing (Contd.)

- Exhaustive testing is hence nearly impossible.
- Implications are:: One cannot test a program completely to guarantee that it is error free.

Economics

- Objective is to therefore find maximum errors with a finite number of test cases.

Limitations of Software Testing

- **Even if we could generate the input, run the tests, and evaluate the output, we would not detect all faults:**

- **Correctness is not checked:**

  The programmer may have misinterpreted the specs, the specs may have misinterpreted the requirements.

-

Psychology of Testing

- Test Engineers pursue defects not people.
- Don’t assume that no error(s) will be found.
- Test for Valid and Expected as well as Invalid and Unexpected.
- The probability of the existence of more errors in a section of a program is proportional to the number of errors already found in that section.
- Testing is extremely creative and intellectually challenging.

Test Case

- **“A set of test inputs, execution conditions, and expected results developed for a particular objective, such as to exercise a particular program path or to verify compliance with a specific requirement.”**

  (…IEEE)

- **In other words, a planned sequence of actions (with the objective of finding errors)**

A good Test Case

- Has a high probability of detecting error(s)
- Test cases help us discover information
- Maximize bug count
- Help managers make ship / no-ship decisions.
- Minimize technical support costs.

Other Terminologies

- **Test Suite – A set of individual test cases/scenarios that are executed as a package, in a particular sequence and to test a particular aspect.**

  E.g. Test Suite for a GUI or Test Suite for functionality

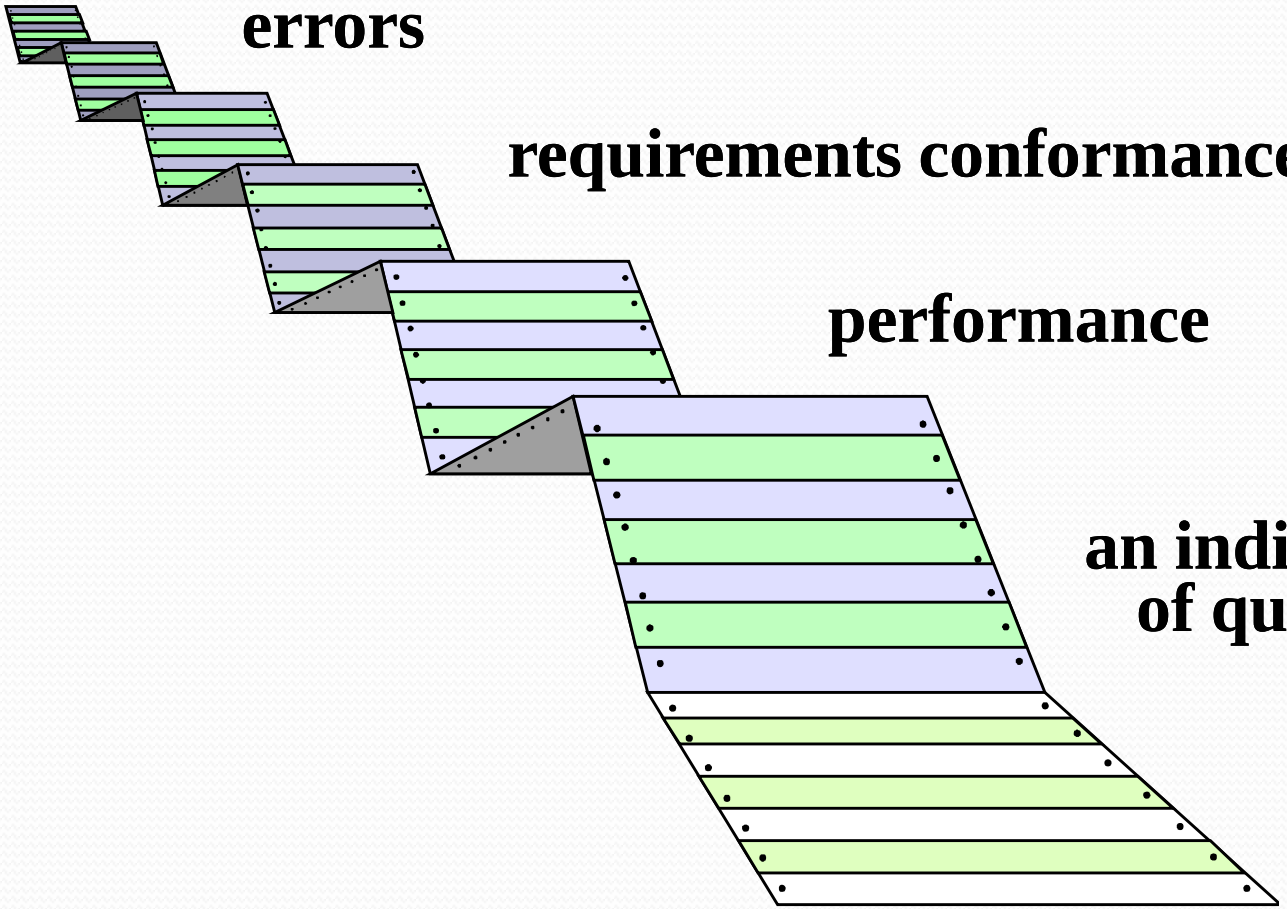
- **Test Cycle – A test cycle consists of a series of test suites which comprises a complete execution set from the initial setup to the test environment through reporting and clean up.**

  E.g. Integration test cycle / regression test cycle

Objectives of Testing

To find greatest possible number of errors with manageable amount of efforts applied over a realistic time span with a finite number of test cases.

What does Software Testing reveal



errors

requirements conformance or the lack of it

performance

an indication of quality