Software Testing

5CS024

Objective

- Brief overview of Testing
- Validation and Verification
- Testing Techniques
- Test Design Technique
- Test Case Content
- Bug Report

Motivation?

[Scenario]: Imagine, you want to transfer your money to your friend using mobile banking. And suppose the software has not been adequately tested, and a defect causes the incorrect transaction processing!!

Financial Discrepancies

Customer Dissatisfaction

Legal Consequences

Software Testing

Software Testing

- Process to evaluate the functionality of a software application with an **intent** to find whether the developed software **met** the **specified requirement or not** and helps to identify the **defects** to ensure that the product is **defect free** in order to produce a quality product.
- “Testing is the process of executing the program with the intention of finding errors.” - Myers, The Art of Software Testing
- “Testing can show the presence of bugs but never their absence.” - Dijkstra

Testing Principles

- All tests should be traceable to customer req.
- Tests should be planned long before testing begins.
- The Pareto principle applies to sw testing.(80/20). 80 % of all errors uncovered during testing will likely be traceable to 20% of all program modules.
- Testing should begin “in small” and progress towards “in the large”.
- Exhaustive testing is not possible
- Testing should be conducted by a third party.

Some of the testing questions that you might want to ask before going further

Testing Questions?

1. What do we test?
2. Who does the testing?
3. Why do we test?
4. How do we test?

What do we test?

- Applications / Websites
- All the possible paths in the code
- All possible combinations of the User Interface
- Load Testing

Who Does Testing?

- Testing can depend on the process of the software development life cycle. Generally testing are done by those people who has the responsibility to deliver the quality product.

1. Software Testers
2. Project Developers
3. Project Lead
4. Team Manager
5. End Users

Simply Anyone!! WHO TAKE CARE OF PRODUCT

Why Software Testing is important?

- In today’s world we rely on the machines for almost everything and all the behaviours of machines are controlled by the softwares.

- If we do not test the software well then it can lead to the system error

- Bugs in software or application can likely head to loss of monetary value, jobs or even loss of human

In the world of Software Development, Out of many questions, there are two major questions that we ask!!

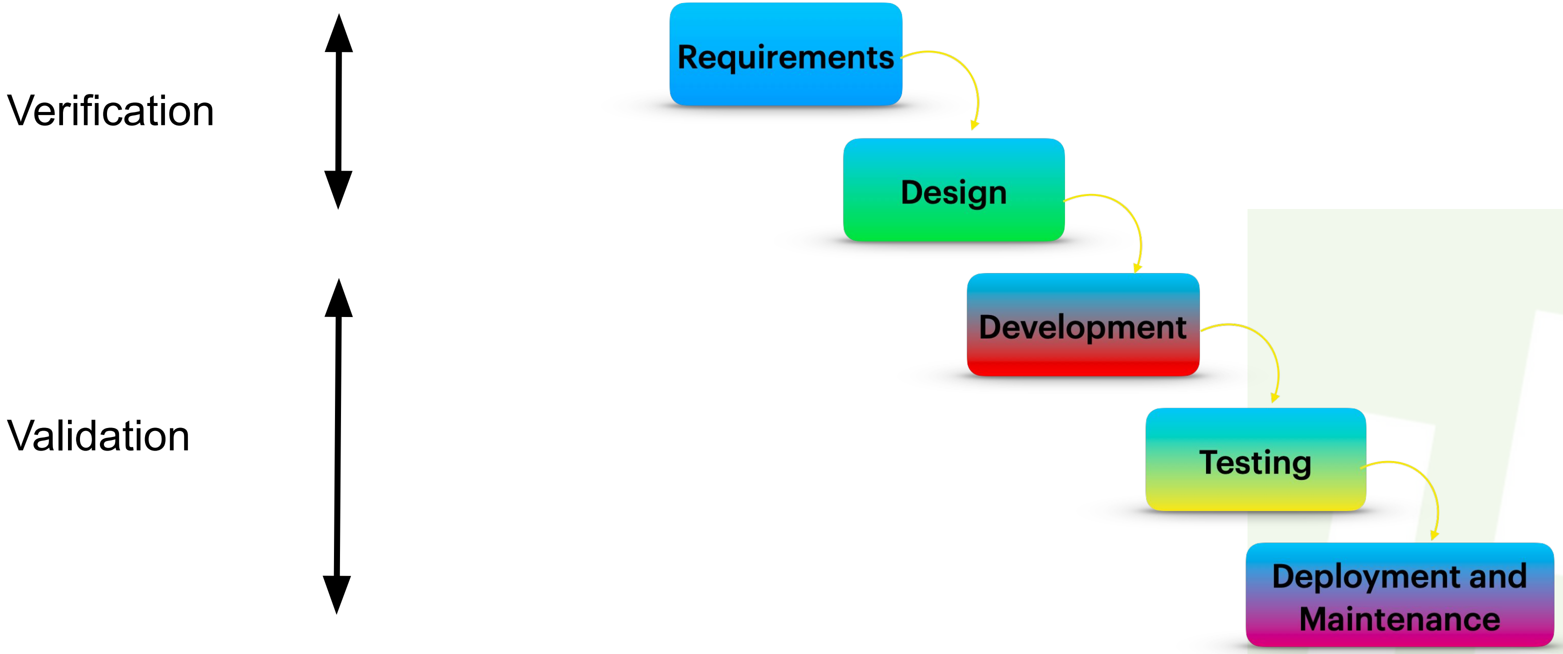
1. **Are we building the product right?**

2. **Are we building the right product?**

We like to verify things before we start and validate later on

BUT WHY DO WE DO IT?

SDLC



Verification
Validation
Requirements
Design
Development
Testing
Deployment and Maintenance

Why do we want to test?

- **Verification (Static Testing)**
- **Validation (Dynamic Testing)**

Note: It is of the software development model just like any other models that we discussed earlier

Verification

- **Proof of compliance with the stated requirements.**

  " Did we proceed correctly when building the system?“
- The software should conform to its specification.
- **Does it meet the requirements**

  Read the user requirements and design tests around the brief
- **Logical Errors**

  Logical errors are caused by the programmer writing code that compiles and runs but causes the wrong actions.

An example might be where the programmer has subtracted rather than added VAT to a total.

Human factors testing

- A system may meet all functional requirements, yet still fail to meet the user's expectations.
- Areas to test human factors include:
  - Correctness - Do buttons do what their labels indicate
  - Completeness Interface provides all required functionality
  - Efficiency
    - Program is efficient & allows user to work efficiently
    - Easy navigation
  - Aesthetics - Pleasing to use – nice colours etc
  - Conformance to business flow
    - Software logic operates same as the business logic.

Verification Technique

- **Review**
  - Whether the document is correct or not and the information provided in the document is complete or not.
  - Each individual in a team can review their document.
  - Example
    - Requirement Review, Design Review,Test Case review
- **Walkthrough**
  - It is formal review and we can discuss/raise the issues at the peer level.
  - Done with team
  - Does Not have minutes of meet.
- **Inspection**
  - Formal review
  - Schedule meeting with a team
  - Member involves: Author,writer,moderator

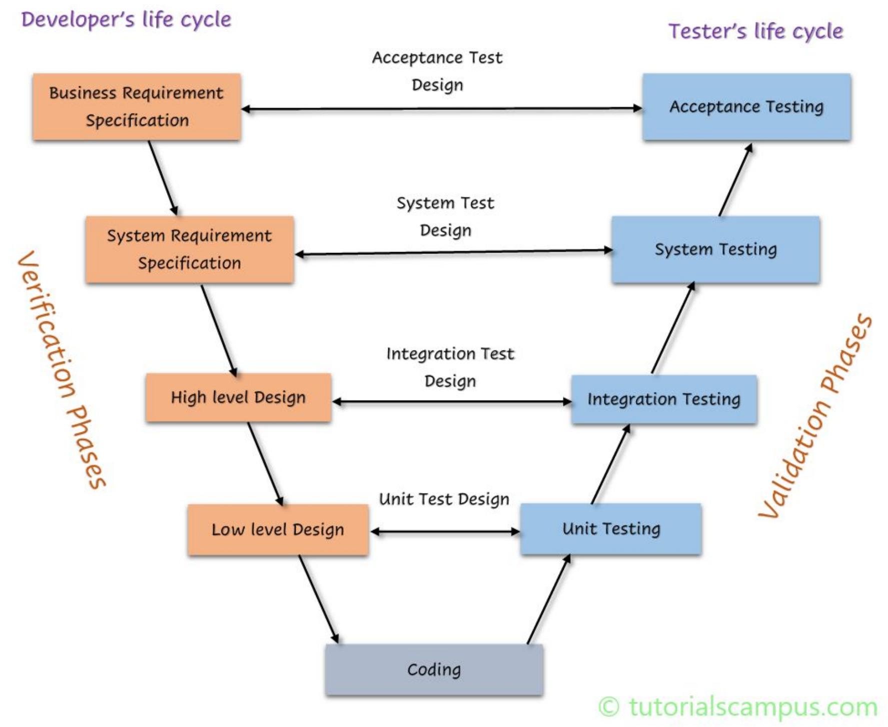
Validation

- Proof of fitness for expected use
  - " Did we build the right system ?“
  - The software should do what the user really requires
- More focus on software
- It comes after verification
- It can find bugs that the verification process can not catch

Validation Technique

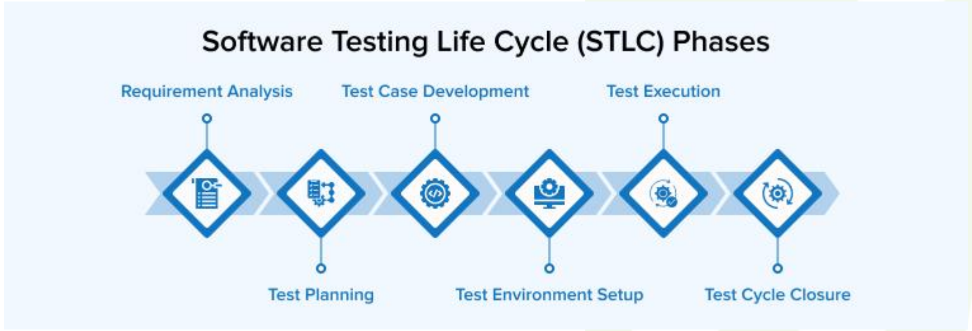
- **Unit Testing**
  - Involves the testing of each unit or an individual component of the software application
  - single testable part of a software system and tested during the development phase of the application software.
- **Integration testing**
  - Units or individual components of the software are tested in a group.
  - The focus of the integration testing level is to expose defects at the time of interaction between integrated components or units.
- **System testing**
  - Testing of a fully integrated software system.
- **Acceptance testing**
  - Involves testing the system before it is accepted for operational use (Somerville 1995)
  - Pilot, Benchmark

Verification and Validation Model



Software Testing Life Cycle

1. Requirement Analysis
2. Test Planning
3. Test Case Development
4. Environment Setup
5. Testing Execution
6. Test Cycle Closure



How Testing is done?

Types of Software Testing

1. **Manual Testing:** Testing of software which is done manually, without the use of automated tool or the applications available in the market
2. **Automation Testing:** Testing of a software where tester writes scripts by own and uses the suitable software to test the softwares.

Recently the world is moving towards automation testing since manual testing is tedious, time consuming, boring and sometime there could be human errors.

Software Testing Methods

1. **White Box Testing**
2. **Black Box Testing**
3. **Grey Box Testing**

White Box Testing

- Testing where the testing of internal structure, logic design and implementation of different modules is done.
- WB - See the code and look at the testing all paths through the code
  - If statements
  - Loops

Black Box Testing

- Testing where the software tests the design and implementation, and UI/UX of the product being tested which is not already known to the tester
- Cannot see the code, all you can test is the interface
- Also, known as the behavioural testing

Grey Box Testing

- Testing where it combines the concepts of both BB and WB testing. In this testing, internal implementation details is partly known to the tester.
- It commonly focuses on context-specific errors related to web systems.

Test Design Technique

- Software testing technique help you design better cases. Since exhaustive testing is not possible; Testing help reduce the number of test cases to be executed while increasing test coverage. They identify test conditions that are otherwise difficult to recognize.
  - Boundary Value Analysis(BVA)
  - Equivalence Partitioning(ECP)
  - Decision Table based testing
  - State Transition
  - Error guessing

Boundary Value Analysis(BVA)

- Boundary value analysis is based on testing at the boundaries between partitions
- Includes maximum, minimum, inside or outside boundaries.
- Dividing boundary
- Range of data check

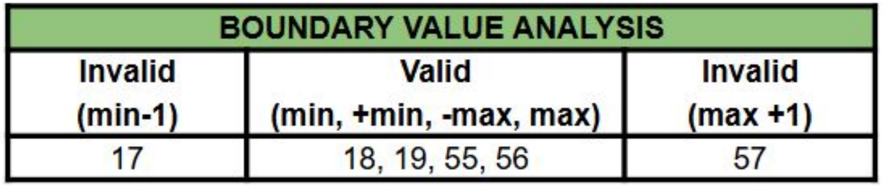
Example

- Minimum boundary value is 18
- Maximum boundary value is 56

AGE

*Accepts value 18 to 56

- Valid Inputs: 18,19,55,56
- Invalid Inputs:17 and 57



BOUNDARY VALUE ANALYSIS		
Invalid (min-1)	Valid (min, +min, -max, max)	Invalid (max +1)
17	18, 19, 55, 56	57

- Test case 1: Enter the value 17(18-1) = Invalid
- Test case 2: Enter the value 18 = Valid
- Test case 3: Enter the value 19(18+1) =Valid
- Test case 4: Enter the value 55(56-1)= Valid
- Test case 5: Enter the value 56 = Valid
- Test case 6: Enter the value 57(56+1) = Invalid

Equivalence partitioning

- Inputs to the software or system are divided into groups that are expected exhibit similar behavior, so they are likely to be proposed in the same way. Hence selecting one input from each group to design the test cases.
- It helps to reduce the total number of test cases from infinite to finite.
- The selected test cases from these groups ensure coverage of all possible scenarios.

Example

- Valid input: 10 digits
- Invalid input: 9 digits, 11 digits

- Valid Class: Enter 10 digit mobile number = 9876543210

MOBILE NUMBER Enter Mobile No. * Must be 10 digits

EQUIVALENCE PARTITIONING		
Invalid	**Valid**	**Invalid**
987654321	9876543210	98765432109

Use case, Test Scenario and Test Case

- **Use Case:**

  - Describes the requirement.

  -Contains Three items: Actor, Action/Flow, Outcome
- **Test Scenario** is "What to be tested" and Test Case is "How to be tested"
- **Test case** consists of set of input values, execution precondition, expected results and executed post condition, developed to cover certain test condition. While Test scenario is nothing but test procedure.

Example

Test Scenario : Checking the functionality of login button

- TC1: Click the button without entering username and password.
- TC2: Click the button only entering Username.
- TC3: Click the button while entering wrong username and wrong password.

Test Case Contents

- Test Case ID
- Test Case Title
- Description
- Pre-condition
- Priority(P0, P1, P2, P3)- order
- Requirement ID
- Steps/Actions
- Expected Result
- Actual Result
- Test Data

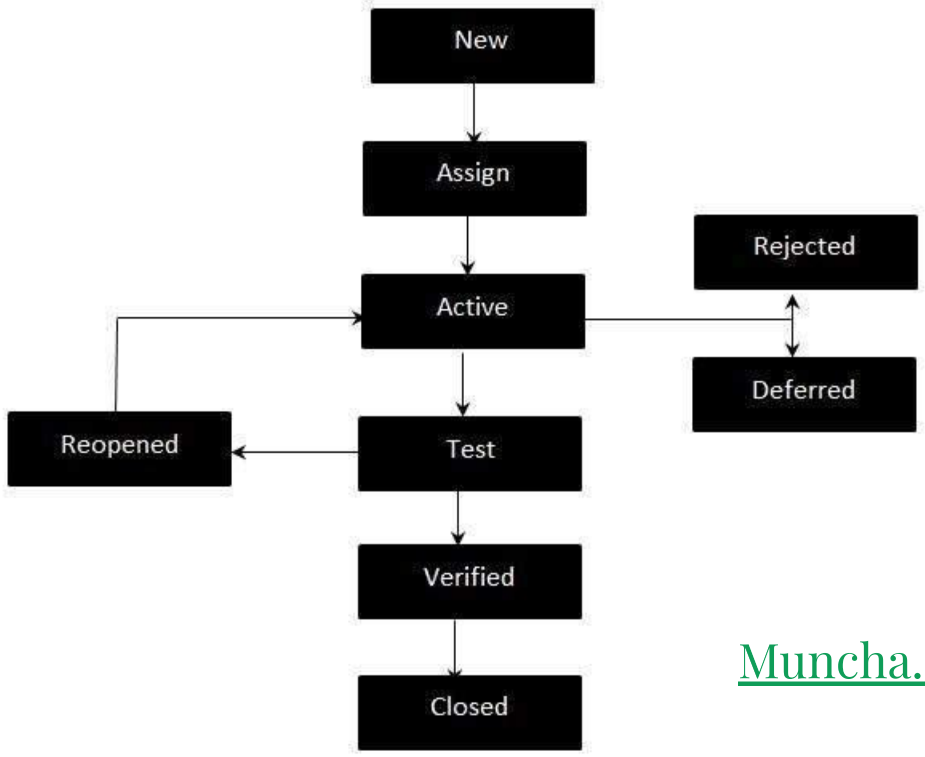
Test Case Contents

Sample : Login Test Case

Bug

- The Bug is the informal name of defects, which means that software or application is not working as per the requirement.
- In software testing, a software bug can also be issue, error, fault, or failure. The bug occurred when developers made any mistake or error while developing the product.
- Like any other lifecycle, bug/defect also have a life cycle and management process

Bug Life-Cycle and a Sample



Muncha.com Bug Report sample

Have Some Research on!!

Before Tutorial, Have some research on!!

- STLC
- Find some Buggy Websites
- Automation Testing
- Selenium

UNIVERSITY OF WOLVERHAMPTON