Testing & Quality Assurance

Lecture 10

Mr. Alfred Edwin Worthington

Learning Outcomes

End of this lecture you will be able to learn ,

LO1: **Understand how the software testing and quality assurance relate with system analysis and design.**

LO2: **Apply the knowledge of different testing methodologies in to the problems.**

Testing & Quality Assurance

- Systems are not 100% accurate
- 40% of the total project time is spent on testing.
- For life critical software, testing can cost more than all other activities.
- Developer should discard preconceived notion of developed software.
- Must be done from a different perspective as that of a developer

Testing

- Testing is a process to explore a system to find defects.

Quality Assurance

- Quality assurance is about the activities designed to make sure the project is conforming to the expectations of the stakeholders.

What is an error?

- If there is a difference between expected output of a software and what was achieved there is an error.
- The main causes of errors are:
  - Not obtaining the right requirements.
  - Not getting the requirements right.
  - Not translating the requirements in a clear and understandable manner for the developers.

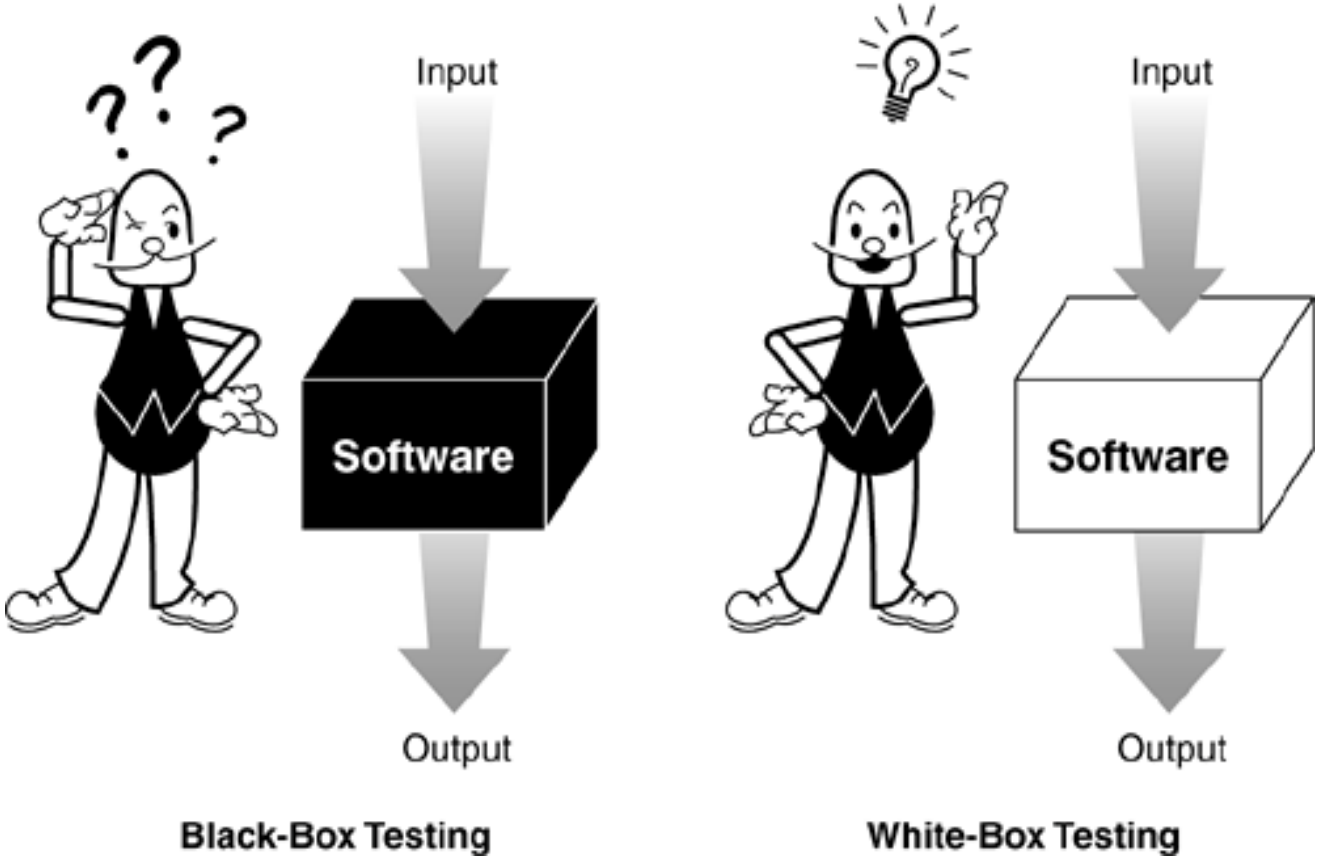
Software Testing Methods

- Black Box Testing

  A technique of testing without having any knowledge of the interior workings of the application.

- White Box Testing

  The detailed investigation of internal logic and structure of the code.


Input
Software
Output
Black-Box Testing
Input
Software
Output
White-Box Testing

Black Box Testing

- Also known as
    - Behavioral
    - Closed box testing
    - Data driven testing
    - Functional testing.
- Software testing method in which the internal structure/ design/ implementation of the application being tested is **not known** to the tester.

Black Box Testing

- This method attempts to find errors in the following categories:
  - Incorrect or missing functions
  - Interface errors
  - Errors in data structures or external database access
  - Behavior or performance errors
  - Initialization and termination errors

Black box testing Techniques

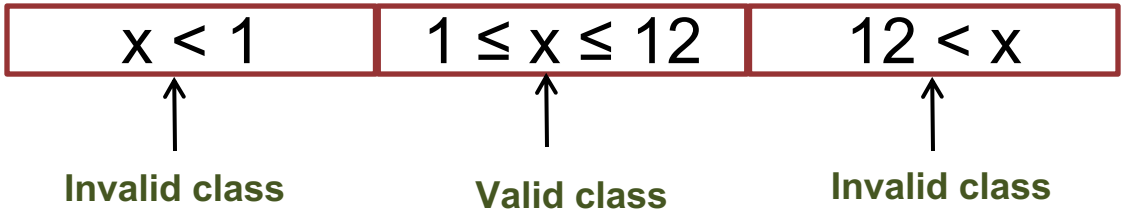
- Equivalence Class Partitioning
- Boundary value Analysis

Equivalence Class Partitioning

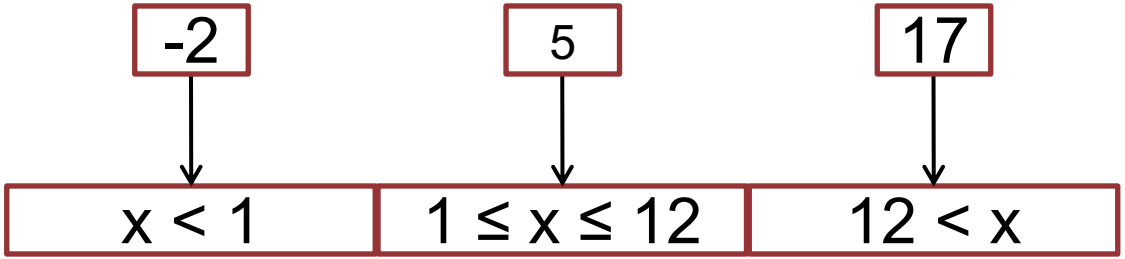
- A software testing technique that divides the input data of a software unit into partitions.
- The equivalence partitions are usually derived from the requirements specification for input attributes.
- Test cases are designed to cover each partition at least once.
- Usually, the input data gets partitioned. However, depending on the software unit to be tested, output data can also be partitioned.

Example 01:

- The function which takes a parameter “month”.
- The valid range for the month is 1 to 12, representing January to December. This valid rang is called a **partition**.
- In this example there are two further partitions of invalid ranges.



- Test cases are chosen so that each partition would be tested.



Example 02:

- In an Examination a candidate has to score minimum of 24 marks in order to clear the exam. The maximum that he can score is 40 marks.

Identify the Valid Equivalence values

a) 22,23,26
b) 21,39,40
c) 29,30,31
d) 0,15,22

Sample Answer

- The classes will be as follows:

  Class I: values < 24 => invalid class

  Class II: 24 to 40 => valid class

  Class III: values > 40 => invalid class

- We have to identify Valid Equivalence values. Valid Equivalence values will be there in Valid Equivalence class. All the values should be in Class II.

- a) 22,23,26

  b) 21,39,40

  c) 29,30,31

  d) 0,15,22

Answer is 'C'

Guidelines for Equivalence Class Partitioning

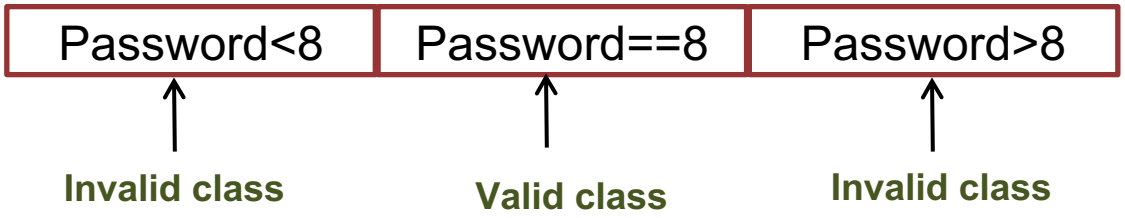
1. If an **input condition specifies a range**, one valid and two invalid equivalence classes are defined.
2. If an **input condition requires a specific value**, then one valid and two invalid equivalence classes are defined.
3. If an **input condition specifies a member of a set**, then one valid and one invalid equivalence class are defined.
4. If an **input condition is Boolean**, then one valid and one invalid equivalence class are defined.

Question

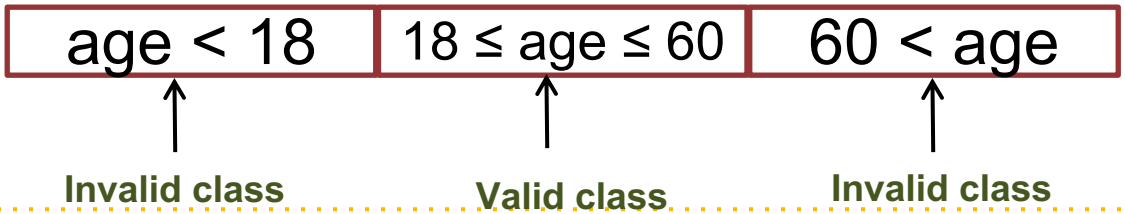
Perform Equivalence Class Partitioning for the following scenario.

Login Interface

- Number of Password Character should be 8.



- Age should be between 18-60



Boundary Value Analysis

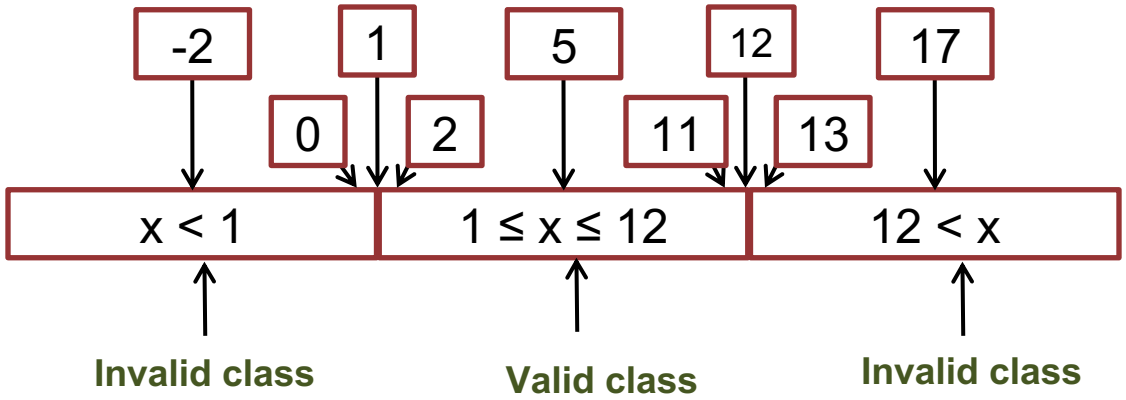
- Focuses on **boundaries and their values**.
- An extension of equivalence class partitioning where boundaries also get focused at the same time.
- In the test case includes
    - Boundary value.
    - Just below the boundary.
    - Just above the boundary.
- Example: The values in between 1-100

    Lower Boundary= 1 (0,1,2)

    Upper Boundary = 100(99,100,101)

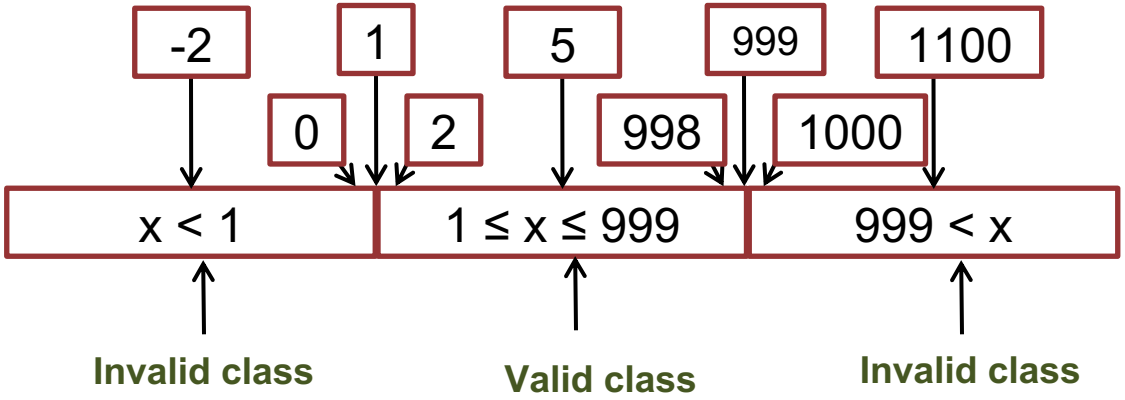
Equivalence partitioning with boundary value analysis

- Test cases are supplemented with boundary values.



Try it yourself

- Consider the values in a range of 1-999
  - Identify the all the classes using Equivalence Class Partitioning
  - Identify all the test case values using Boundary Value Analysis



White box testing

- A testing technique, that examines the internal structure of the component or system.
- It focuses primarily on strengthening security, the flow of inputs and outputs through the application, and improving design and usability.
- White box testing is also known as Clear Box testing, Open Box testing, Structural testing, Transparent Box testing, Code-Based testing, and Glass Box testing.

White box testing Contd….

- White Box Testing done by the testers.
- Learning parts of the code & finding out the weakness in the software program under test.
- One of the best method to find out the errors in the software application in early stage of software development life cycle.

White Box Testing Techniques

- Statement Coverage
  - Aimed at exercising all programming statements with minimal tests.
- Branch Coverage
  - Running a series of tests to ensure that all branches are tested at least once.
- Path Coverage
  - Testing all possible paths which means that each statement and branch is covered.

Black Box Testing Vs. White Box Testing

Black Box Testing	White Box Testing
The internal workings of an application are not required to be known.	Tester has full knowledge of the internal workings of the application
Performed by end users and by testers and developers	Normally done by testers and developers
This is the least time consuming and least exhaustive	The most exhaustive and time-consuming type of testing
Not suited for algorithm testing	Suited for algorithm testing

Debugging vs. Testing

- Testing is the process to find errors.
- Debugging is the process of fixing those errors. Debugging should be incorporated in testing strategies.

Verification and Validation in Software Testing

Verification :

- Evaluates whether the built product is meeting the requirements and design specifications.

Are we building the product right?

Validation:

- Evaluates whether built product is satisfying specified business requirements and fulfilling its intended use.

Are we building the right product?

Functional testing

Software is usually tested at several levels of complexity.

1.Unit Testing

Concentrates on each component/function of the software in isolation.

2. Integration Testing

Focuses on the design and construction of the software architecture.

(test a growing set of integrated modules each time a new one is added)

3. System Testing

The software and other system elements are tested as a whole.

Regression Testing

- Whenever a change in a software application is made it is quite possible that other areas within the application have been affected by this change.
- To verify that a fixed bug hasn't resulted in another functionality or business rule violation is Regression testing .
- Regression testing ensures that a bug fix did not result in another fault in the application.

Acceptance Testing

- The aim is to determine whether the system meets its specification.
- Types of Acceptance Testing
  - User acceptance Testing
  - Business acceptance Testing
  - Alpha Testing
  - Beta Testing

Acceptance Testing Types

User Acceptance Testing

- Clients/end users involve in testing the product to validate the product against their requirements.

- Performed at client location or at developer's site.

Acceptance Testing Types

Alpha Testing

- Takes place at the developer's site by the internal teams, before release to external customers.
- Performed without the involvement of the development teams.

Beta Testing

- Takes place at the end users' site by the end users to validate the usability, functionality, compatibility, and reliability.
- Beta testing adds value to the software development life cycle as it allows the customer an opportunity to provide real inputs.

Non-Functional Testing : Recovery Testing

- Performed in order to determine how quickly the system can recover after it has gone through system crash or hardware failure.

- Recovery testing is the forced failure of the software to verify whether the recovery is successful.

Non-Functional Testing : Stress Testing

- Stress testing is performed as part of performance testing.

- During stress testing, the system is monitored after subjecting the system to overload to ensure that the system can sustain the stress.

Non-Functional Testing : Security Testing

- Security testing is to determine if an information system protects data and maintains functionality as intended.
- It also aims at verifying 6 basic principles as listed below:
    - Confidentiality
    - Integrity
    - Authentication
    - Authorization
    - Availability
    - Non-repudiation

Any questions?