

# Unit 5

## Black Box Testing

Dr.K.Madheswari

ASP/CSE

# Black box testing

- Using the black box approach, a tester considers the software-under test to be an opaque box. There **is no knowledge of its inner structure** (i.e., how it works). The tester only has knowledge of what it does.
- The size of the software-under-test using this approach can vary from a simple module, member function, or object cluster to a subsystem or a complete software system.

## Source for black box

- The description of behavior or functionality for the software-under-test may come from **a formal specification, an Input/Process/ Output Diagram (IPO)**, or a well-defined set of pre and post conditions.
- Another source for information is a **requirements specification document** that usually describes the functionality of the software-under-test and its inputs and expected outputs.

# Contd...

- The tester provides the specified inputs to the software-under-test, runs the test and then determines if the outputs produced are equivalent to those in the specification. Because the black box approach only considers software behavior and functionality, it is often called **functional, or specification-based testing**.
- This approach is especially useful for **revealing requirements and specification defects**.

# Tools used for Black Box testing

- Black box testing tools are **mainly record and playback tools**.
- These tools are used for **regression testing** that to check whether new build has created any bug in previous working application functionality.
- These record and playback tools records test cases in the form of some scripts like TSL, VB script, Java script, Perl.

# Contd..

- **Advantages of Black Box Testing**
  - Tester can be non-technical.
  - Used to verify contradictions in actual system and the specifications.
  - Test cases can be designed as soon as the functional specifications are complete
- **Disadvantages of Black Box Testing**
  - The test inputs needs to be from large sample space.
  - It is difficult to identify all possible inputs in limited testing time. So writing test cases is slow and difficult
  - Chances of having unidentified paths during this testing

# Methods of black box testing

- **Equivalence Partitioning**
- **Boundary Value Analysis**
- **Error Guessing**
- **Cause and effect graphing**
- [Decision table](#) testing
- State transition testing

# Equivalence Class Partitioning

## Definition

- Equivalence partitioning is a **Test Case Design Technique** to **divide** the input data of software into **different equivalence** data classes. Test cases are designed for equivalence data class. The equivalence partitions are frequently derived from the **requirements specification** for input data that influence the processing of the test object.

Equivalence partitioning is a testing technique where input values set into classes for testing.

- **Valid Input Class = Keeps all valid inputs.**
- **Invalid Input Class = Keeps all Invalid inputs.**

## Example of Equivalence Class Partitioning

- A text field permits only numeric characters
- Length must be 6-10 characters long

Partition according to the requirement should be like this:

Contd..

**0 1 2 3 4 5 | 6 7 8 9 10 | 11 12 13 14**

**Invalid | Valid | Invalid**



# How is this partitioning performed while testing:

## List of Conditions

1. If an input condition specifies a range, one valid and one two invalid classes are defined.

## Example

The value of the variable xyz is lies in the range 1-10;

-2,-1,0	1,2,3....10	11,12,13,14
Invalid class	valid	invalid

## Contd..

2. If an input condition requires a specific value, one valid and two invalid equivalence classes are defined.

### Example

**house of one to four owners**

-2,-1,0	1,2,3,4	5,6,7,8
Invalid class	valid	invalid

## Contd...

3. If an input condition specifies a member of a set, one valid and one invalid equivalence class is defined.

### **example**

if the specification for a paint module states that the colors RED, BLUE, GREEN and YELLOW are allowed as inputs, then select one valid equivalence class that includes the set RED,BLUE, GREEN and YELLOW, and one invalid equivalence class for all other inputs

RED, GREEN, YELLOW	OTHER COLORS
Valid class	Invalid Class

## Contd...

4. If an input condition is Boolean, one valid and one invalid class is defined.

### Example

**X is greater than 1**

1,2,3,4	0,-1,-2,-3
Valid class	Invalid Class

# Best Example:

## Grocery Store Example

### SRS

Consider a software module that is intended to accept the name of a grocery item and a list of the different sizes the item comes in, specified in ounces.

The specifications state that the item name is to be alphabetic characters 2 to 15 characters in length. Each size may be a value in the range of 1 to 48, whole numbers only. The sizes are to be entered in ascending order (smaller sizes first). A maximum of five sizes may be entered for each item. The item name is to be entered first, followed by a comma, then followed by a list of sizes. A comma will be used to separate each size. Spaces (blanks) are to be ignored anywhere in the input.

# Derived Equivalence Classes

- 1.Item name is alphabetic (valid)
- 2.Item name is not alphabetic (invalid)
- 3.Item name is less than 2 characters in length (invalid)
- 4.Item name is 2 to 15 characters in length (valid)
- 5.Item name is greater than 15 characters in length (invalid)
- 6.Size value is less than 1 (invalid)
- 7.Size value is in the range 1 to 48 (valid)
- 8.Size value is greater than 48 (invalid)
- 9.Size value is a whole number (valid)
- 10.Size value is a decimal (invalid)
- 11.Size value is numeric (valid)
- 12.Size value includes non-numeric characters (invalid)
- 13.Size values entered in ascending order (valid)
- 14.Size values entered in non-ascending order (invalid)

# Derived Equivalence Classes contd...

- 15.No size values entered (invalid)
- 16.One to five size values entered (valid)
- 17.More than five sizes entered (invalid)
- 18.Item name is first (valid)
- 19.Item name is not first (invalid)
- 20.A single comma separates each entry in list (valid)
- 21.A comma does not separate two or more entries in the list (invalid)
- 22.The entry contains no blanks (???)
- 23.The entry contains blanks (????)

## Test case (using EC)

#	Test Data	Expected Outcome	Classes Covered
1	xy,1	T	1,4,7,9,11,13,16,18,20,22
2	AbcDefghijklmno,1,2,3,4,48	T	1,4,7,9,11,13,16,18,20,23
3	a2x,1	F	2
4	A,1	F	3
5	abcdefghijklmnp	F	5
6	Xy,0	F	6
7	XY,49	F	8
8	Xy,2.5	F	10
9	xy,2,1,3,4,5	F	14
10	Xy	F	15
11	XY,1,2,3,4,5,6	F	17
12	1,Xy,2,3,4,5	F	19
13	XY2,3,4,5,6	F	21
14	AB,2#7	F	12



# More examples

- Suppose you have very important tool at office, accepts valid User Name and Password field to work on that tool, and accepts minimum 8 characters and maximum 12 characters. Valid range 8-12, Invalid range 7 or less than 7 and Invalid range 13 or more than 13.

## Contd...

- Test Cases 1: Consider password length less than 8.
- Test Cases 2: Consider password of length exactly 8.
- Test Cases 3: Consider password of length between 9 and 11.
- Test Cases 4: Consider password of length exactly 12.
- Test Cases 5: Consider password of length more than 12