Unit: IV
Lecture: 2

Black Box Testing

Black box testing also called behavioral testing focuses on the functional requirements of the software i.e. black box testing enables the software engineer to derive sets of input conditions that fully exercise all functional requirements for a program.

Black box testing 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 and
- initialization and termination errors.

Unlike white-box testing, which is performed early in the testing process, black box testing tends to be applied during later stages of testing.

Tests are designed to answer the following questions:

- ✓ How is functional validity tested?
- ✓ How is system behavior and performance tested?
- ✓ What classes of input will make good test cases?
- ✓ Is the system particularly sensitive to certain input values?
- ✓ How are the boundaries of a data class isolated?
- ✓ What data rates and data volume can the system tolerate?
- ✓ What effect will specific combinations of data have on system operation?

Black Box Testing Techniques:

- **Equivalence Partitioning:** Equivalence partitioning is a black-box testing method that divides the input domain of a program into classes of data from which test cases can be derived.
  - It is often seen that 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.

An equivalence class represents a set of valid or invalid states for input conditions. Typically, an input condition is either a specific numeric value, a range of values, a set of related values or a Boolean condition.

e.g. To calculate the square root of a number, the equivalence classes will be:
a) Valid Inputs:
- Whole number which is a perfect square- output will be an integer.
- Whole number which is not a perfect square-output will be decimal number
- Positive decimals

b) Invalid Inputs:
- Negative numbers (integer or decimal)
- Characters other than numbers like “a”,”!”, etc.

- **Boundary Value Analysis:** In general, a greater number of errors tend to occur at the boundaries of the input domain rather than in the center.
  - It is for this reason, boundary value analysis (BVA) has been developed as a testing technique.
  - Boundary value analysis is a test case design technique that complements equivalence partitioning. Rather than selecting any element of an equivalence class. BVA leads to the selection of test cases at the edges of the class.

Black-box testing tools:
- Selenium
- Appium
- Applitools
- HP QTP
- Microsoft coded UI