**🎯 What is Black-Box Testing?**

|  |  |
| --- | --- |
| Aspect | Explanation |
| Definition | Testing based on requirements and specifications **without knowing the internal code**. |
| Scope | Applied at **unit**, **integration**, **system**, and **acceptance** testing levels. |
| Goal | Validate the software’s **functionality** against its expected outputs for given inputs. |

**📦 Key Techniques in Black-Box Testing**

|  |  |
| --- | --- |
| Technique | Purpose |
| Exhaustive Testing | Tests every possible input |
| Equivalence Class Testing | Reduces input test cases by grouping similar inputs |
| Boundary Value Analysis | Focuses on input limits (boundaries) |
| Decision Table Testing | Tests logic and rules in tabular format |
| State-Transition Testing | Tests behavior changes based on state and events |

**🔁 Exhaustive Testing**

|  |  |
| --- | --- |
| Definition | Test with every possible input value |
| Example | Password field with 3 characters: 256×256×256 = 16.7 million+ combinations |
| ⚠️ Limitation | **Impractical** due to huge number of possible inputs |

**🧩 Equivalence Class Testing (EC)**

|  |  |
| --- | --- |
| Definition | Divide input into classes expected to behave similarly |
| Goal | Test one representative from each class |
| Example | Bus Fare System: • Age 0–1 → Free • Age 2–14 → $10 • Age 15–64 → $15 • Age 65+ → $5 |

**Valid/Invalid ECs:**

|  |  |
| --- | --- |
| Input Range | Class |
| 1–50 | Valid EC |
| < 1 or > 50 | Invalid ECs |

✅ **Best for**: Systems with input ranges or categorical inputs.

**🧮 Boundary Value Analysis (BVA)**

|  |  |
| --- | --- |
| Definition | Test values at the edges (boundaries) of input ranges |
| Why? | Most defects occur **near limits** |
| Steps |  |

1. Identify ECs
2. Identify boundaries
3. Test at, above, and below each boundary

| **Example** | Input must be a 5-digit number ≥ 10000:<br>Test cases: 9999 (below), 10000 (at), 10001 (above) |

**📋 Decision Table Testing**

|  |  |
| --- | --- |
| Definition | Represent combinations of conditions and actions in tabular format |
| Purpose | Capture **complex business logic** |
| Structure | Conditions (top), Actions (bottom), Rules (columns) |

**Example:**

|  |  |  |  |
| --- | --- | --- | --- |
| Rule # | Email Valid | Password Valid | Action |
| 1 | T | T | Login Success |
| 2 | T | F | Show Password Error |
| 3 | F | T/F | Show Email Error |

✅ **Good for**: Input combinations, login systems, business rules.

**🔄 State-Transition Testing**

|  |  |
| --- | --- |
| Definition | Tests system behavior based on states and events |
| Components |  |

* **State**: Circle (e.g., ON, OFF)
* **Transition**: Arrow (event that triggers state change)
* **Event**: Action or input (e.g., "switch ON")
* **Action**: Output (e.g., “Light ON”)

**Example:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Case | Start State | Input | Output | End State |
| TC1 | OFF | switch ON | light ON | ON |
| TC2 | ON | switch OFF | light OFF | OFF |

✅ **Great for**: Interactive systems, UIs, clocks, media players

**🔥 Advanced Testing Concepts**

|  |  |
| --- | --- |
| Technique | Definition |
| Stress Testing | Test system **beyond normal limits** to assess stability under extreme conditions |
| Fuzz Testing | Provide **random/invalid inputs** to detect system crashes or vulnerabilities |
| Progressive Testing | Also called **Incremental Testing**; test modules **one by one** using: • Top-down • Bottom-up • Hybrid |
| Re-testing | Run failed tests again after fixing the issue to confirm the fix |

**🧠 Bonus: Summary Table of Techniques**

|  |  |  |
| --- | --- | --- |
| Technique | Main Use | Example Scenario |
| Exhaustive Testing | All input combinations | 3-digit password brute force testing |
| Equivalence Class | Input range grouping | Bus fare by age |
| Boundary Value | Edge testing ofs ranges | Age input: test 14, 15, 16 |
| Decision Table | Rule-based logic | Login system |
| State-Transition | State-driven systems | Light switch, electronic clocks |
| Stress Testing | Extreme loads | Crash a server with overload |
| Fuzz Testing | Random inputs | Send junk values to web form |
| Progressive Testing | Module-by-module | Test parent-child modules incrementally |
| Re-testing | Confirm fixed bugs | Test failed login after patch |