# **Defects & Testing Techniques:**

# **Black-Box, White-Box, and Gray-Box Testing**

## **📌 Table of Contents**

1. Defects in Software Testing
2. Black-Box Testing
3. White-Box Testing
4. Gray-Box Testing
5. Comparison Summary
6. Best Practices
7. Real-World Examples
8. References

## **🐞 Defects in Software Testing**

### **What is a Defect?**

A defect (bug) is a deviation from expected behavior in software, caused by errors in code, design, or requirements.

### **Defect Life Cycle**

1. New → Reported by tester.
2. Assigned → Sent to developer.
3. Open → Developer analyzes.
4. Fixed → Code corrected.
5. Retest → Tester verifies fix.
6. Closed/Reopened → If defect persists.

### **Severity vs Priority**

| **Severity** | **Priority** |
| --- | --- |
| Impact on system (Critical/Major/Minor). | Urgency to fix (High/Medium/Low). |
| Example: App crash (High Severity). | Example: Typo in footer (Low Priority). |

## **📦 Black-Box Testing**

### **Definition**

Testing without knowledge of internal code, focusing on inputs and outputs.

### **Techniques**

1. Equivalence Partitioning
   * Divides input into valid/invalid groups.
   * Example: Age field (Valid: 18-60, Invalid: <18, >60).
2. Boundary Value Analysis (BVA)
   * Tests edge values.
   * Example: Password length (Min: 8, Max: 16 → Test 7,8,16,17).
3. Decision Table Testing
   * Tests combinations of inputs/conditions.
   * Example: Login (Valid/Invalid email + password).
4. State Transition Testing
   * Tests system behavior between states.
   * Example: ATM (Card inserted → PIN entered → Cash dispensed).
5. Use Case Testing
   * Validates end-user scenarios.
   * Example: E-commerce checkout flow.

### **Pros & Cons**

| **Pros** | **Cons** |
| --- | --- |
| ✅ No coding knowledge needed. | ❌ Misses internal logic errors. |
| ✅ User-centric (validates UX). | ❌ Limited coverage (unseen paths). |

### **Tools**

* Selenium (Web)
* Postman (API)
* QTP/UFT (Enterprise)

## **📝 White-Box Testing**

### **Definition**

Testing with code access, validating internal structures, logic, and paths.

### **Techniques**

1. Statement Coverage
   * Executes every code statement.
   * Goal: 100% line coverage.
2. Branch Coverage
   * Tests all decision paths (if-else, loops).
   * Example: if (age >= 18) → Test age=17 and age=18.
3. Path Coverage
   * Tests all possible execution paths.
   * Example: Nested loops/complex conditions.
4. Mutation Testing
   * Introduces small code changes to check test robustness.

### **Pros & Cons**

| **Pros** | **Cons** |
| --- | --- |
| ✅ Finds hidden code defects. | ❌ Requires coding expertise. |
| ✅ Optimizes performance. | ❌ Time-consuming for large codebases. |

### **Tools**

* JUnit (Java)
* PyTest (Python)
* SonarQube (Static Analysis)

## **🎭 Gray-Box Testing**

### **Definition**

Hybrid approach with partial code knowledge, combining Black-Box and White-Box techniques.

### **Techniques**

1. Database Testing
   * Validates SQL queries without full DB access.
   * Example: Check if user data saves correctly.
2. API Testing
   * Tests endpoints with limited backend knowledge.
   * Example: Verify GET /users returns 200 status.
3. Reverse Engineering
   * Infers internal logic from outputs.

### **Pros & Cons**

| **Pros** | **Cons** |
| --- | --- |
| ✅ Balanced efficiency. | ❌ Limited depth vs White-Box. |
| ✅ Good for integration testing. | ❌ May miss edge cases. |

### **Tools**

* Postman (API)
* OWASP ZAP (Security)

## **🆚 Comparison Summary**

| **Aspect** | **Black-Box** | **White-Box** | **Gray-Box** |
| --- | --- | --- | --- |
| Code Access | No | Yes | Partial |
| Tester Role | End-user perspective | Developer perspective | Mixed |
| Coverage | Functional | Structural | Integration |
| Best For | UAT, System Testing | Unit, Security Testing | API, DB Testing |

## **🏆 Best Practices**

✔ Combine all three for full coverage.  
✔ Black-Box: Focus on user journeys.  
✔ White-Box: Aim for 90%+ code coverage.  
✔ Gray-Box: Use for third-party integrations.

## **🌍 Real-World Examples**

### **Case 1: Boeing 737 Max (Black-Box Failure)**

* Defect: MCAS system triggered falsely.
* Root Cause: Insufficient scenario testing (missed sensor failures).

### **Case 2: Equifax Breach (White-Box Gap)**

* Defect: Unpatched Apache Struts vulnerability.
* Root Cause: Lack of static code analysis.

### **Case 3: Twitter API (Gray-Box Success)**

* Test: Verified rate limits without full backend access.

## **📚 References**

* [ISTQB Techniques](https://www.istqb.org/)
* [Black-Box vs White-Box (Guru99)](https://www.guru99.com/)
* [OWASP Testing Guide](https://owasp.org/www-project-web-security-testing-guide/)

🔹 Conclusion:

* Black-Box = "What it does"
* White-Box = "How it works"
* Gray-Box = "Partial insight"
* Use all three for bulletproof software!



