**Brief Summary of Any 5 Software Testing Styles**

## **Table of Contents**

1. Black Box Testing
2. White Box Testing
3. Accessibility Testing
4. Acceptance Testing
5. Stress Testing

## **1. Black Box Testing**

**Definition:** Black Box Testing is a software testing technique where testers verify the software’s functionality without knowing its internal code, structure, or design.  
 The tester focuses only on inputs and expected outputs.

**Key Features:**

* Focuses on *functional* and *user-facing* requirements.
* Testers are independent of developers.
* Best suited for *system testing* and *acceptance testing* phases.

**How It’s Done:**

* Review specifications and requirements.
* Create test cases based on expected behavior.
* Provide different inputs and observe outputs.
* Compare actual results with expected results.

**Example:** Testing a login page:  
 Input: username/password → Expected output: Success or error message without needing backend code knowledge.

## **2. White Box Testing**

**Definition:** White Box Testing (also called Clear Box or Glass Box Testing) involves testing the internal workings of the software.  
 Testers have full visibility into code logic, algorithms, and structure.

**Key Features:**

* Focuses on *code paths, loops, decisions, and conditions*.
* Done by developers or testers familiar with coding.
* Used in *unit testing* and *integration testing*.

**How It’s Done:**

* Analyze the code and logic deeply.
* Identify critical paths, loops, and branches.
* Write test cases to cover all code flows.
* Measure code coverage (statement, branch, condition).

**Example:** Testing a function that checks even/odd numbers by ensuring both conditions are executed properly.

**Advantages:**

* Thorough testing of internals; catches hidden issues early.
* Helps optimize and clean up the code.

**Disadvantages:**

* Requires programming knowledge.
* Not scalable for very large applications.

## **3. Accessibility Testing**

**Definition:** Accessibility Testing ensures that a website, app, or software is usable by all people, including those with disabilities such as visual, hearing, motor, or cognitive impairments.

**Key Features:**

* Complies with *WCAG (Web Content Accessibility Guidelines)*.
* Involves using *assistive technologies* like screen readers, voice commands.
* Manual checks include *keyboard navigation*, *contrast ratios*, *alt text verification*.

**How It’s Done:**

* Manual Testing: Use the keyboard only, screen readers, magnifiers.
* Automated Testing: Tools like *Axe*, *WAVE*, *Lighthouse*.
* Validate color contrasts, image descriptions, and navigation flow.

**Example:** Testing a login form where each field and button must be correctly announced by a screen reader.

**Advantages:**

* Improves usability for all users.
* Helps meet legal compliance and accessibility laws.
* Enhances brand reputation.

**Disadvantages:**

* Some testing must be manual; automation cannot detect everything.
* Adds extra time and effort.

## **4. Acceptance Testing**

**Definition:** Acceptance Testing is the final phase before product release where stakeholders verify if the software meets their business requirements and is ready for real-world use.

**Key Features:**

* Focuses on *real-world usage* and *user workflows*.
* Executed by *end users, business analysts,* or *clients*.
* Ensures alignment between business needs and product delivery.

**How It’s Done:**

* Understand business requirements.
* Design acceptance test cases based on user scenarios.
* Perform real-world usage tests (e.g., placing orders, generating reports).
* Log issues and get client sign-off on successful completion.

**Example:** An e-commerce app where clients check search, cart, payment, and order confirmation flows before approval.

**Advantages:**

* Increases customer satisfaction.
* Catches gaps before product reaches end-users.
* Reduces costly post-release fixes.

**Disadvantages:**

* Depends heavily on availability of users/clients.
* Late-stage failures can cause significant rework.

## **5. Stress Testing**

**Definition:** Stress Testing is a form of performance testing where the system is subjected to extreme conditions (high load, low memory, heavy traffic) to test stability and recovery under pressure.

**Key Features:**

* Focuses on system robustness under abnormal conditions.
* Identifies breaking points and crash recovery behavior.
* Helps in preparing for peak usage scenarios.

**How It’s Done:**

* Determine baseline system performance.
* Increase the load beyond normal expectations (users, data volume).
* Monitor how the system reacts — slowdown, crashes, recovery.

**Example:** Simulating 10,000 students logging into an online exam portal at once, instead of the normal 500.

**What is Observed:**

* CPU, memory, network usage.
* Database connections and server stability.
* Response time and error handling.

**Advantages:**

* Prevents real-world failures during heavy usage.
* Improves system resilience and scalability.

**Disadvantages:**

* Needs specialized tools like *JMeter*, *LoadRunner*.
* Can be expensive and resource-heavy.