**Brief Summary of 5 Software Testing Styles**

**1. Unit Testing**

**Definition:** Unit Testing focuses on testing individual units or components of a software application. A "unit" refers to the smallest part of an application that can be tested independently, such as functions, methods, or classes.

**Purpose:** To verify that each unit of the software performs as designed.

**Tools:** JUnit, NUnit, PyTest

**Example:** Testing a login function that checks username and password correctness.

**Diagram:**

+----------------+ +----------------+ +----------------+

| Function 1 | | Function 2 | | Function 3 |

| (Login) | | (Register) | | (Logout) |

+----------------+ +----------------+ +----------------+

Each function/unit is tested separately

**2. Integration Testing**

**Definition:** Integration Testing examines how different modules or units of an application work together.

**Purpose:** To detect issues in the interaction between integrated units.

**Tools:** Selenium, Postman, JUnit (for API integration)

**Example:** Testing if the login module properly connects with the dashboard after successful authentication.

**Diagram:**

[Login Module] ----> [Dashboard Module] ----> [User Profile Module]

Integration Testing ensures smooth data flow between modules

**3. System Testing**

**Definition:** System Testing involves testing the complete and fully integrated software product to evaluate its compliance with the specified requirements.

**Purpose:** To validate the end-to-end business flow of the application.

**Tools:** Selenium, TestComplete, QTP

**Example:** Testing an e-commerce website where a user searches for a product, places an order, and receives an order confirmation.

**Diagram:**

[Search] --> [Product Detail] --> [Add to Cart] --> [Checkout] --> [Payment] --> [Confirmation]

Complete System Flow Testing

**4. Acceptance Testing**

**Definition:** Acceptance Testing is performed to determine whether a system meets the business requirements and is acceptable for delivery.

**Purpose:** To validate the software against user needs and business processes.

**Tools:** Cucumber, FitNesse

**Example:** Testing whether a mobile app provides a seamless booking experience for users as per business expectations.

**Diagram:**

[Requirement] --> [Developed Feature] --> [User Acceptance Test] --> [Approval/Release]

**5. Regression Testing**

**Definition:** Regression Testing involves re-running functional and non-functional tests to ensure that previously developed and tested software still performs after a change.

**Purpose:** To catch bugs introduced unintentionally due to new changes, upgrades, or enhancements.

**Tools:** Selenium, QTP, TestComplete

**Example:** After updating the payment gateway, testing if other existing features like order placement, cart update, and notifications still work.

**Diagram:**

[Original Feature] --> [Enhancement/Change] --> [Regression Testing] --> [Stable Release]