**Test Levels**

### **1. What? (Là gì?)**

* **Unit Test**: Kiểm thử từng đơn vị nhỏ nhất của mã nguồn (thường là hàm, phương thức, class) một cách riêng lẻ.
* **Integration Test**: Kiểm thử sự kết hợp giữa các module hoặc thành phần để đảm bảo chúng hoạt động với nhau như mong đợi.
* **System Test**: Kiểm thử toàn bộ hệ thống như một thể thống nhất dựa trên yêu cầu phần mềm.
* **User Acceptance Test (UAT)**: Kiểm thử do người dùng thực hiện để xác định phần mềm có đáp ứng được yêu cầu nghiệp vụ không.

### **2. Why? (Tại sao?)**

* **Unit Test**: Đảm bảo từng thành phần nhỏ hoạt động đúng logic trước khi tích hợp.
* **Integration Test**: Phát hiện lỗi trong việc giao tiếp giữa các module.
* **System Test**: Xác nhận toàn bộ hệ thống đáp ứng được các yêu cầu chức năng và phi chức năng.
* **UAT**: Kiểm chứng hệ thống có phù hợp với nhu cầu thực tế của người dùng trước khi đưa vào sử dụng chính thức.

### **3. When? (Khi nào?)**

* **Unit Test**: Thực hiện sớm nhất trong quy trình phát triển, ngay sau khi viết xong từng hàm/module.
* **Integration Test**: Sau khi các đơn vị đã được kiểm thử riêng lẻ và sẵn sàng tích hợp.
* **System Test**: Khi toàn bộ hệ thống đã được tích hợp và sẵn sàng để kiểm thử toàn diện.
* **UAT**: Giai đoạn cuối, trước khi phần mềm được triển khai chính thức cho người dùng.

### **4. Where? (Ở đâu?)**

* **Unit Test**: Development Environment. Kiểm thử đơn vị diễn ra trong môi trường phát triển, thường là máy của lập trình viên, tích hợp trực tiếp trong IDE hoặc công cụ CI/CD nội bộ.
* **Integration Test**: Test Environment. Kiểm thử tích hợp được thực hiện trong môi trường kiểm thử – nơi các module đã được triển khai để kiểm tra việc phối hợp giữa chúng. Thường có cấu hình giống production ở mức độ giới hạn.
* **System Test**: Staging Environment. Kiểm thử hệ thống toàn diện được thực hiện trong môi trường staging – bản sao gần như đầy đủ của production để đảm bảo hệ thống chạy ổn định trước khi phát hành.
* **UAT**: Staging Environment. UAT thường được thực hiện trong môi trường staging hoặc một môi trường UAT riêng biệt, mô phỏng chính xác môi trường sản xuất để người dùng kiểm thử nghiệp vụ thực tế.

### **5. Who? (Ai thực hiện?)**

* **Unit Test**: Lập trình viên.
* **Integration Test**: Lập trình viên hoặc kiểm thử viên.
* **System Test**: Kiểm thử viên chuyên nghiệp (QA).
* **UAT**: Người dùng cuối, khách hàng hoặc đại diện nghiệp vụ (business analyst).

### **6. How? (Làm thế nào?)**

* **Unit Test**:
  + Viết mã kiểm thử cho từng đơn vị (unit) nhỏ nhất như hàm, phương thức hoặc class.
  + Các test case nên **độc lập**, **tự động**, và **kiểm tra đầu vào – đầu ra** của từng unit.
  + Thường sử dụng kỹ thuật **Mocking** để giả lập các thành phần bên ngoài.
* **Integration Test**:
  + Viết test để kiểm tra sự tương tác **giữa các module, lớp, API, hoặc thành phần**.
  + Có thể sử dụng **Mock/Stubs** hoặc kiểm tra trên hệ thống đã tích hợp thật.
  + Dùng kỹ thuật **Top-down**, **Bottom-up**, hoặc **Big Bang Integration Testing**.
* **System Test**:
  + Kiểm thử **toàn bộ hệ thống** như một “hộp đen” (black-box testing).
  + Test dựa trên **tài liệu yêu cầu (SRS)** và **test case chi tiết**.
  + Kiểm thử cả chức năng và phi chức năng: bảo mật, hiệu suất, khả năng chịu tải...
* **UAT**:
  + Thực hiện bởi **người dùng cuối, khách hàng, hoặc đại diện nghiệp vụ**.
  + Dựa trên **tình huống sử dụng thực tế (use case)** và quy trình nghiệp vụ.
  + Kết quả UAT quyết định **phần mềm có thể được triển khai chính thức hay không**.