**Unit testing**

**Lý thuyết về unit testing:**

Unit testing là một quá trình phát triển phần mềm, trong đó các phần nhỏ nhất có thể kiểm tra được của ứng dụng, được gọi là unit, được xem xét kỹ lưỡng và độc lập để có thể hoạt động tích hợp.

Phương pháp kiểm tra này được thực hiện trong quá trình phát triển bởi lập trình viên và đôi khi là các QA

**Cách hoạt động của unit testing:**

Một unit test thường bao gồm ba giai đoạn: lập kế hoạch, đề ra các trường hợp, kịch bản và các đơn vị kiểm thử. Trong bước đầu tiên, một bài unit test được chuẩn bị và xem xét. Bước tiếp theo là các trường hợp thử nghiệm và kịch bản sẽ được thực hiện, sau đó mã code sẽ được thử nghiệm.

Phát triển phần mềm theo các test case đòi hỏi các lập trình viên trước tiên phải lập ra các bộ test unit thất bại. Sau đó, họ viết mã code và cấu trúc lại ứng dụng cho đến khi các bộ test được thỏa. TDD (Test-driven development) thường dẫn kiến các kết quả rõ ràng và có thể dự đoán được

**Các loại unit testing**

Các unit test có thể thực hiện thủ công hoặc tự động. Người sử dụng phương pháp thủ công có thể có một tài liệu được lập chi tiết từng bước trong quy trình kiểm thử; tuy nhiên, phương pháp kiểm thử tự động là phương pháp phổ biến hơn đối với unit test. Các phương pháp tiếp cận thường sử dụng các bộ khung kiểm thử (testing framework) để phát triển các test case. Các framework này được thiết lập để gắn cờ và báo cáo mọi trường hợp test không thành công, đồng thời cung cấp bản tóm tắt các trường hợp thử nghiệm.

**Các ưu điểm và nhược điểm của unit test (kiểm thử đơn vị):**

Các ưu điểm bao gồm:

- Phát hiện lỗi sớm, hạn chế lỗi có thể xảy ra.

- Phát hiện lỗi càng sớm, chi phí khắc phục càng thấp.

- Quá trình sửa lỗi (debugging) được thực hiện dễ dàng hơn

- Lập trình viên có thể nhanh chóng thay đổi cấu trúc code.

- Tái sử lại cho các dự án khác.

Các nhược điểm bao gồm:

- Các bộ test sẽ không bao trọn hết các lỗi.

- Các bộ unit test chỉ kiểm tra tập hợp dữ liệu và chức năng của một đơn vị – sẽ không bắt được lỗi khi tích hợp các đơn vị lại với nhau.

- Phải viết nhiều code hơn để kiểm tra một đoạn (đơn vị) code nhất định, tiêu tốn thời gian.

- Đòi hỏi người kiểm thử có kiến thức sâu về các công cụ kiểm thử tự động.