**1. Unitest**

Kiểm thử đơn vị (unit testing) là một phương pháp kiểm thử phần mềm mà mỗi đơn vị mã nguồn, tập hợp một hoặc nhiều các môđun chương trình máy tính cùng với dữ liệu kiểm soát liên quan, thủ tục sử dụng, và các quy trình vận hành, được kiểm tra để xác định chúng có phù hợp để sử dụng hay không?

Thực hiện kiểm tra đơn vị được thiết kế để đơn giản, thường là các bài kiểm tra được viết dưới dạng chức năng sẽ xác định xem một giá trị trả về bằng với giá trị mà bạn mong đợi khi bạn đã viết các chức năng hoặc giá trị mà bạn sẽ mong đợi khi bạn cuối cùng viết nó – điều này được gọi là Thử nghiệm Drive Development khi bạn viết bài kiểm tra đầu tiên.

**Một số công cụ kiểm thử:**

* **Quality Center:** Công cụ quản lý thử nghiệm.
* **QTP**: Dụng cụ thi công.
* **LoadRunner:** Công cụ đánh giá thực thi.
* **Case:** Yêu cầu công cụ quản lý.
* **SourceAnywhere:** Công cụ quản lý cấu hình.
* **InView:** Công cụ review.
* **PMD:** Công cụ phân tíc tổng hợp.
* **Altova:** Công cụ mô phỏng.
* **Clone & Test:** Công cụ chuẩn bị dữ liệu thử nghiệm.
* **Junit:** Dụng cụ kiểm tra khai thác / Kiểm tra đơn vị công cụ.
* **CodeCover:** Công cụ đo độ bao phủ.

**Điểm mấu chốt trong Unit testing và lợi ích:**

* Unit testing được thực hiện trước khi Integration testing bởi các nhà phát triển phần mềm, sử dụng các kỹ thuật kiểm thử hộp trắng (white box testing).
* Tìm kiếm các vấn đề/lỗi ở giai đoạn đầu là rất hữu ích và nó làm giảm tổng chi phí dự án. Là Đơn vị kiểm tra được thực hiện trước khi hội nhập của mã, các vấn đề phát hiện ở giai đoạn này có thể được giải quyết rất dễ dàng và ảnh hưởng của họ cũng bị hạn chế.
* Unit Testing kiểm tra cả đầu vào và đầu ra , ví dụ: đầu ra chính xác trong trường hợp đầu vào hợp lệ, nhưng cũng có những thất bại xảy ra với đầu vào không hợp lệ.
* Một lợi thế quan trọng là các trường hợp đơn vị kiểm tra đơn giản hóa và thực hiện thử nghiệm mã dễ dàng hơn. Vì vậy, nó trở nên dễ dàng hơn để giải quyết các vấn đề ở giai đoạn sau quá bởi vì chỉ có sự thay đổi mới nhất trong mã này là để được kiểm tra.
* Kiểm thử đơn vị (UT) tiết kiệm thời gian và chi phí, và nó là tái sử dụng và dễ dàng để duy trì. Junit (Java framework), PHPUnit (PHP framework), NUnit (.Net framework) etc. là những tool phổ biến được dùng để UT (unit testing) cho các loại ngỗn ngữ khác nhau.

1. **Automation Test**

Trong lĩnh vực kiểm thử phần mềm, thì kiểm thử tự động hay còn gọi là Automation testing đóng một vai trò quan trọng góp phần nâng cao năng suất kiểm thử, giảm thiểu lỗi cũng như sự nhàm chán với việc kiểm thử bằng tay trong một thời gian dài hoặc lặp đi lặp lại.

Kiểm thử tự động là một quá trình xử lý tự động các bước thực hiện một test case. Kiểm thử tự động được thực hiện bởi phần mềm kiểm thử tự động - hay còn gọi là Automation Testing Tool. Một số phần mềm kiểm thử tự động nổi tiếng hiện nay như:

* Quick Test Profressional - (HP)
* Selenium
* Test Architect - (LogiGear)
* Ranorex
* Visual Studio CodedUI Testing
* TestComplete (SmartBear)
* SOAPUI - Web Services Testing (SmartBear)

1. **Stability Test**

* Stability testing(kiểm tra độ ổn định) là khả năng duy trì hoạt động của sản phẩm xuyên suốt và vượt thời hạn sử dụng của nó, mà không hỏng học xảy ra lỗi. Đây là 1 kỹ thuật non-functional, với mục đích đòi hỏi khả năng chịu tải phần mềm tới mức tối đa, kiểm tra hiệu quả của 1 sản phẩm được phát triển vượt qua mức hoạt động bình thường, hay tới 1 điểm dừng
* Đối với 1 ứng dụng đang được thử nghiệm mà số lượng lớn user được giới thiệu và các ứng dụng phải chạy hàng tháng mà không cần khởi động lại, có thể sẽ xảy ra 1 vài vấn đề: hệ thống chạy chậm, vấn đề về function, treo,…
* Stability test giúp user hiểu được hệ thống làm việc trong thực tế, cho phép bạn kiểm tra:
  + Ước lượng độ tin cậy trong sự ổn định của hệ thống đang thử nghiệm
  + Đảm bảo hệ thống xử lý được chương trình lớn
  + Theo dõi hiệu quả và sự ổn định của hệ thống
* Stability đóng 1 vai trò quan trọng trong phát triển sản phẩm, được sử dụng phổ biến đối với kiểm tra website mua sắm trực tuyến, CPU test