Uni test

Một Unit là một thành phần PM nhỏ nhất mà ta có thể kiểm tra được. Theo định nghĩa này, các hàm (Function), thủ tục (Procedure), lớp (Class), hoặc các phương thức (Method) đều có thể được xem là Unit.

Vì ta không khó khăn gì trong việc tổ chức, kiểm tra, ghi nhận và phân tích kết quả kiểm tra. Nếu phát hiện lỗi, việc xác định nguyên nhân và khắc phục cũng tương đối dễ dàng vì chỉ khoanh vùng trong một đơn thể Unit đang kiểm tra. Một nguyên lý đúc kết từ thực tiễn: Unit được chọn để kiểm tra thường có kích thước nhỏ và chức năng hoạt động đơn giản, chúng thời gian tốn cho Unit Test sẽ được đền bù bằng việc tiết kiệm rất nhiều thời gian và chi phí cho việc kiểm tra và sửa lỗi ở các mức kiểm tra sau đó.

Unit Test thường do lập trình viên thực hiện. Công đoạn này cần được thực hiện càng sớm càng tốt trong giai đoạn viết code và xuyên suốt chu kỳ PTPM. Thông thường, Unit Test đòi hỏi kiểm tra viên có kiến thức về thiết kế và code của chương trình. Mục đích của Unit Test là bảo đảm thông tin được xử lý và xuất (khỏi Unit) là chính xác, trong mối tương quan với dữ liệu nhập và chức năng của Unit. Điều này thường đòi hỏi tất cả các nhánh bên trong Unit đều phải được kiểm tra để phát hiện nhánh phát sinh lỗi. Một nhánh thường là một chuỗi các lệnh được thực thi trong một Unit, ví dụ: chuỗi các lệnh sau điều kiện If và nằm giữa then ... else là một nhánh. Thực tế việc chọn lựa các nhánh để đơn giản hóa việc kiểm tra và quét hết Unit đòi hỏi phải có kỹ thuật, đôi khi phải dùng thuật toán để chọn lựa.

Unit Test cũng đòi hỏi phải chuẩn bị trước các tình huống (test case) hoặc kịch bản (script), trong đó chỉ định rõ dữ liệu vào, các bước thực hiện và dữ liệu mong chờ sẽ xuất ra. Các test case và script này nên được giữ lại để tái sử dụng

Acceptant test

* Đây là một kiểm thử liên quan đến nhu cầu của người sử dụng, yêu cầu và quy trình kinh doanh được tiến hành để xác định có hay không một hệ thống đáp ứng các tiêu chí chấp nhận và kiểm tra hệ thống đáp ứng yêu cầu của khách hàng.
* Kiểm thử chấp nhận kiểm thử các chức năng để kiểm tra hành vi của hệ thống bằng cách sử dụng dữ liệu thực tế. Nó cũng được gọi là thử nghiệm người dùng doanh nghiệp.
* Kiểm thử chấp nhận được thực hiện bởi người dùng cuối để kiểm tra hệ thống được xây dựng để phù hợp với yêu cầu kinh doanh của tổ chức.
* Trong kiểm thử này, tất cả các giao diện đã được kết hợp và hệ thống đã hoàn thành và đã được kiểm tra. Người dùng cuối cũng thực hiện các kiểm thử để kiểm tra khả năng sử dụng của hệ thống.
* Nhiều kỹ thuật kiểm thử chức năng sử dụng cho loại này là phân tích giá trị biên giới, phân vùng tương đương, bảng quyết định. Đây là loại kiểm tra tập trung chủ yếu vào các kiểm thử hợp lệ của hệ thống.

Automation test

Là quá trình kiểm tra hệ thống với dữ liệu đầu vào và đầu ra đã được xác định 1 cách tự động.

Ưu điểm

* Tính hiệu quả trong công việc:thay thế con người lặp đi lặp lại đúng quy tắccác bước kiểm thử, không cần xin phép đột suất khi cần, không làm những việc ảnh hưởng đến quá trình test => tránh được hao phí về mặt thời gian.
* Độ tin cậy: một khi đã Execute thành công và nhận về kết quả thì chúng ta có thể yên tâm nó chạy đúng với những gì đã yêu cầu, độ ổn định cao, tránh được rủi ro có thể phát sinh. Ngoài ra, kiểm thử tự động còn giúp tái hiện bug một cách hiệu quả, những bug thỉnh thoảng mới xảy ra hay xảy ra một cách ngẫu nhiên sẽ được repdure một cách dễ dàng nhờ việc lưu lại các bước một cách tự động.
* Cải thiện chất lượng: Kiểm thử tự động làm giảm rủi ro về mặt chất lượng sản phẩm, việc kiểm thử được thực hiện một cách nhanh chóng, tránh trường hợp chủ quan cho những trường hợp test hồi quy, retest, test toàn bộ hệ thống.
* Tốc độ xử lý cực nhanh: chỉ cần mất 30s nếu sử dụng kiểm thử tự động, công việc của bạn là "Run" và ngồi chờ kết quả.
* Chi phí thấp
* Hỗ trợ export ra nhiều ngôn ngữ phổ biến như Ruby, Java, Python…

## Nhược điểm

* Ban đầu thì chi phí cho kiểm thử tự động sẽ cao hơn kiểm thử thủ công
* Để kiểm thử tự động thực hiện được thì vẫn cần con người phải bỏ thời gian, công sức và tiền bạc,...
* Mất chi phí cho các công cụ tự động hóa như bản quyền, bảo trì, tìm hiểu, training.
* Khó mở rộng hơn nhiều so với kiểm thử thủ công
* Yêu cầu những người có trình độ chuyên môn cao mới thực hiện được
* Số lượng công việc phải làm để mở rộng cho kiểm thử tự động sẽ nhiều và khó hơn so với kiểm thử thủ công :rất nhiều công việc như debug, thay dữ liệu đầu vào, cập nhật lại code mới.