Test Level

1. Unit test

* Được thực trong quá trình ứng dụng và được thực hiện bởi dev , mục đích là tìm ra lỗi ngay khi vừa code xong trước khi tích hợp và chuyển sang giai đoạn khác.
* Unit test thực hiện để kiểm tra xem các đơn vị riêng lẻ của chương trình có đúng hay không
* 1 unit (components) là phần nhỏ nhất có thể kiểm tra của một ứng dụng , ví dụ như là một chức năng , một thủ tục hoặc chương trình nhỏ
* Unit test đúng và đầy đủ sẽ tiết kiệm thời gian , chi phí cho dự án
* Unit test cần để làm gì ?
* Đảm bảo code thực hiện theo đúng thiết kế chi tiết ban đầu
* Đảm bảo chất lượng đơn vị phần mềm , những lỗi đc tìm thấy sớm sẽ được log ra và sửa lỗi tránh để sang giai đoạn sau
* Giảm chi phí và nhân công cho phần đảm bảo chất lượng
* Giảm nhân công để sửa lỗi

1. Integration test

* Test sự kết nối(giao tiếp) giữa các components với nhau.
* Không tập chung vào các chức năng của đơn vị riêng lẻ mà tập trung vào kết nối giữa các đơn vị đó
* Lỗi được tìm trong giai đoạn integration test là lỗi được tìm khi hai components giao tiếp với nhau không đúng .
* Lý do:
* Đảm bảo các functions , các modules , các components được tích hợp với nhau và hoạt động đúng theo thiết kế tổng quan.
* Đảm bảo sự kết nối giữa các components , các chức năng hoạt động đúng .
* Mức độ thích hợp của integration test:
* Tích hợp ở mức Components : Tích hợp giữa các components với nhau
* Tích hợp ở system : có nhiều hệ thống tích hợp lại với nhau tạo thành một hệ thống lớn hơn.
* Các loại tích hợp :

+ Big-bang integration : tích hợp tất cả các components , systems cùng 1 lúc . Việc tích hợp này không có thứ tự ưu tiên đều tích hợp ngay cùng một thời điểm.

+ Incremental integration : tích hợp tăng dần- tích hợp từng components lại với nhau

Big-bang integration:

|  |  |
| --- | --- |
| Advantage | Disadvantage |
| * tất cả mọi thứ được tích hợp cùng một lúc trước khi kiểm thử, không mất thời gian làm giả lập các module | * Tốn nhiều thời gian, khó để theo dõi nguyên nhân lỗi * Việc tích hợp chỉ hiệu quả khi mà các chức năng hoặc module không có lỗi. |

Incremental integration:

|  |  |
| --- | --- |
| Advantages | Disadvantages |
| * Lỗi được tìm thấy sớm ngay khi bắt đầu tích hợp, dễ tìm nguyên nhân lỗi | * Tốn thời gian để tạo các bộ giả lập |

* Trong khi thực hiện test, có các loại tích hợp như sau:
* Top-down : Tích hợp từ trên xuống , từ menu đến các chức năng (sd sơ khai)
* Bottom-up : Tích hợp từ dưới lên , từ chức năng chi tiết lên menu
* Functional incremental: mỗi một chức năng được tích hợp , kiểm thử từng cái một.
* Stub : là chương trình hoặc chức năng được gọi bởi module mà mình dự định test(Nếu như module A gọi module B khi cần giả lập module B thì người ta gọi module B là stub- top-down )
* Driver : là chương trình đề gọi module mình định test( ngược lại vd trên)

1. System test

* Là tập trung vào hành vi của tất cả hệ thống sản phẩm xem hệ thống của sản phẩm đã thực hiện chức năng và thỏa mãn các đặc tính của nó được định nghĩa trong phạm vi của bài toán hoặc sản phẩm hay không
* Là kiểm tra toàn bộ hệ thống xem hệ thống có thỏa mãn yêu cầu đặt ra hay không , và để tìm được nhiều lỗi nhất có thể . System test thường được thực hiện độc lập với đội phát triển
* Để thực hiện system test cần có các yếu tố :
* Risk - phân tích rủi ro
* Requirement specification : dựa vào đặc tả yêu cầu
* Business process : quy trình nghiệp vụ
* Use cases
* Interaction with OS or system resources: sự tương tác giữa các hệ điều hành và nguồn lực hệ thống.
* Tại sao cần thực hiện system test ?
* Đảm bảo của nghiệp vụ hệ thống hoạt động đúng , đảm bảo đặc tả yêu cầu tồn tại trong hệ thống
* Bao gồm các loại :
* Functional testing : test chức năng
* Performance testing:test hiệu năng
* Security testing:test bảo mậts
* Installation testing:test về cài đặt.

1. Acceptance test

* Được thực hiện bởi khách hàng.
* Môi trường acceptance test nên thực hiện gần giống với môi trường production càng tốt.
* Việc AT tập chung vào kiểm tra xem các loại test của hệ thống có phù hợp , dễ sử dụng hay không
* Khi đưa vào sử dụng thì AT cần có :
* Không còn một lỗi nghiêm trọng nào trong hệ thống
* Quy trình nghiệp vụ hệ thống hoạt động đúng
* Nên có một buổi họp và thống nhất kết quả nghiệm thu của các bên liên quan ( đội dự án và đội khách hàng)
* Tại sao cần có AT :
* Chấp nhận sản phẩm dựa trên các tiêu chí nghiệm thu
* Đảm bảo các chức năng cần và các chức năng khách hàng mong muốn , hoạt động đúng trên sản phẩm.
* Các loại AT :
* Test nghiệm thu vận hành ( Operational AT ) : Backup/restore(phục hồi), disaster recovery(khôi phục sau sự cố ), maintenance task (nhiệm vụ bảo trì), security weakness ( điểm yếu bảo mật)
* Test nghiệm thu tính tuân thủ của hệ thống ( Compliance AT) : hệ thống được thực hiện test phải tuân thủ nghiệm thu trong hợp đồng (VD theo các điều luật của chính phủ, các điều kiện nghiệm thu dự án)
* Thị trường AT có 2 giai đoạn: Kiểm thử alpha : do người dùng trong cùng một tổ chức phát triển thực hiện. Kiểm thử beta : do người dùng trong môi trường thật thực hiện.

=> Development testing : Unit test ( kiểm thử độc lập ) , integration test (kiểm thử tích hợp ) , system test( test khi tất cả hệ thống đã sẵn sàng ) : tìm càng nhiều lỗi để khắc phục sớm nhất

=> Acceptance test : được thực hiện bởi khách hàng để nghiệm thu sản phẩm : Đảm bảo chắc chắn hệ thống làm việc chắc chắn đúng như mong đợi của khách hàng, hệ thống thỏa mãn các yêu cầu của người dùng

=> Với mỗi một giai đoạn kiểm thử , sẽ có một chức năng khác nhau , một mục đích khác nhau .

=> Maintenance testing : Kiểm tra xem không còn một lỗi nào mới được sinh ra sau khi thay đổi yêu cầu, thay đổi hệ thống

=> During operational testing : mục đích chính có thể được truy cập vào các đặc điểm của hệ thống như độ tin cậy hoặc tính khả dụng.