

BÀI GIẢNG KIỂM THỬ PHẦN MỀM



BÀI 4:



ÔN TẬP BÀI TRƯỚC

- ▶ **Có mấy phương pháp kiểm thử?**
- ▶ **Black box testing và White box testing khác nhau thế nào?**
- ▶ **Bạn biết những kỹ thuật Black box testing nào?**

ÔN TẬP BÀI TRƯỚC

- ▶ **Có mấy loại kiểm thử (testing types)**
- ▶ **Intergration testing khác với System test ntn?**
- ▶ **Test Alpha là gì?**
- ▶ **Test Beta là gì?**

QUY TRÌNH KIỂM THỬ (Testing process)

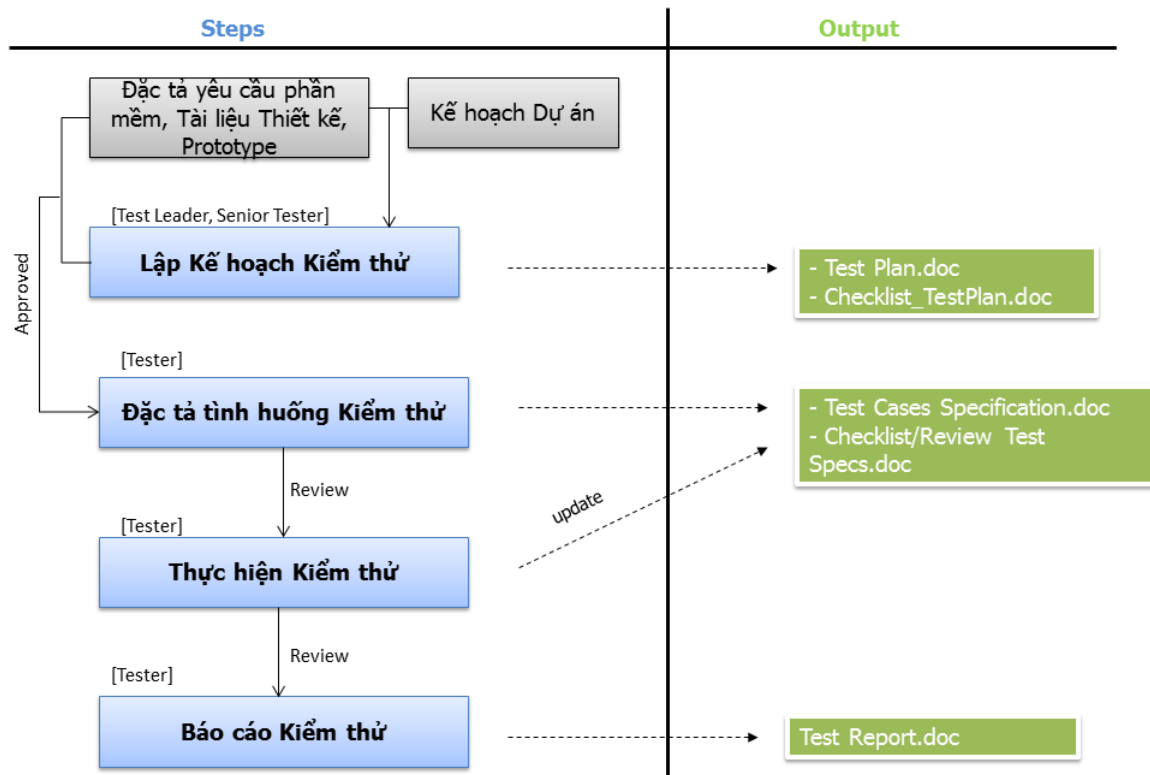
► Software Testing Process



SOFTWARE TESTING là gì?

- ▶ Kiểm thử phần mềm (software testing) là hoạt động nhằm tìm kiếm, phát hiện các lỗi của phần mềm
- ▶ Kiểm thử phần mềm đảm bảo sản phẩm phần mềm đáp ứng chính xác, đầy đủ và đúng theo yêu cầu của khách hàng, yêu cầu của sản phẩm đề đã đặt ra.
- ▶ Kiểm thử phần mềm tạo điều kiện cho bạn tận dụng tối đa tư duy đánh giá và sáng tạo để bạn có thể phát hiện ra những điểm mà người khác chưa nhìn thấy.
- ▶ Software testing cũng có thể xem như là quá trình thẩm định và thẩm tra (validating and verifying) phần mềm/chương trình/ứng dụng/sản phẩm để:
 - Đáp ứng được các yêu cầu công việc và kỹ thuật đã được quy định trong thiết kế và trong lúc phát triển
 - Làm việc như mong đợi

SOFTWARE PROCESS là gì?



TEST PLAN là gì?

- ▶ Một kế hoạch kiểm thử dự án phần mềm (test plan) là một tài liệu mô tả các mục tiêu, phạm vi, phương pháp tiếp cận, và tập trung vào nỗ lực kiểm thử phần mềm. Quá trình chuẩn bị test plan là một cách hữu ích để suy nghĩ tới những nỗ lực cần thiết để xác nhận khả năng chấp nhận một sản phẩm phần mềm.
- ▶ Cấu trúc chung của một test plan:
 - Tên project
 - Danh sách các Module cần test
 - Ngày bắt đầu, ngày kết thúc
 - Danh sách các Test case
 - Nhân sự tham gia
 - Tài nguyên sử dụng (Servers, Workstations, Printers,...)
 - Kế hoạch thực hiện (sử dụng Ms Project lập kế hoạch)...
- ▶ Xem biểu mẫu Test Plan

TEST CASE là gì?

- ▶ Test case mô tả một dữ liệu **đầu vào (input)**, **hành động (action)** và một **kết quả mong đợi (expected response)**, để xác định một chức năng của ứng dụng phần mềm hoạt động đúng hay không.
- ▶ Một test case có thể có các phần đặc thù khác nhau như **mã test case, tên test case, mục tiêu test, các điều kiện test, các yêu cầu data input, các bước thực hiện và các kết quả mong đợi**. Mức chi tiết có thể được định nghĩa khác nhau dựa vào quy mô của dự án hay quy mô của công ty sản xuất phần mềm.
- ▶ Quá trình phát triển test case có thể giúp tìm ra lỗi trong các yêu cầu hoặc thiết kế của ứng dụng, vì nó đòi hỏi phải tư duy hoàn toàn thông qua các hoạt động của ứng dụng. Vì lý do này, việc chuẩn bị test case sớm nhất có thể trong qui trình phát triển phần mềm là rất hữu ích.

TEST CASE là gì?

- ▶ Nó bao gồm **3 bước cơ bản**:
 - Mô tả : đặc tả các điều kiện cần có để tiến hành kiểm tra.
 - Nhập : đặc tả đối tượng hoặc dữ liệu cần thiết, được sử dụng làm đầu vào để thực hiện kiểm tra.
 - Kết quả mong chờ : kết quả trả về từ đối tượng kiểm tra.
- ▶ **Biểu mẫu Test case:**

ID	Test Case Description	Test Case Procedure	Expected Result	Test Result
1	<Brief description of this case: what is tested?> Test username and password = empty	<Describe steps to perform this case> 1. Username and Password textbox = empty 2. Click "Log in" button	<Describe results which meet customer's requirement> Message displays 'Username and Password are required'	Pass
2	<Test case 2>			Fail

TESTING EXECUTION là gì?

- ▶ Thực hiện test trực tiếp trên ứng dụng phần mềm sau khi lập trình viên bàn giao
- ▶ Dựa trên các test cases đã được viết trước đó để thực hiện test trên phần mềm
- ▶ Thực hiện ghi nhận kết quả kiểm thử vào cột kết quả của tài liệu kịch bản kiểm thử. Nếu kết quả kiểm thử là thất bại thì nhóm kiểm thử ghi nhận lỗi lên hệ thống quản lý lỗi.
- ▶ Nhóm kiểm thử, QTDA, nhóm lập trình, nhóm giải pháp tham gia vào quá trình quản lý/xử lý lỗi (bug/ defect)

TEST REPORT là gì?

- ▶ Trong giai đoạn testing, một số document và report sẽ được tạo ra, ví dụ như là Test plan, Testcase,... trong đó Test report là được tạo sau khi hoàn thành giai đoạn Testing. Ngoài ra cũng Daily Report, Weekly Report tùy theo yêu cầu đặc thù của dự án
- ▶ Báo cáo test cũng thể hiện tiến độ kiểm thử, tiến độ sửa lỗi và số lượng lỗi được tìm thấy hay còn tồn của dự án. Báo cáo kiểm thử cũng là công cụ để phục vụ cho đánh giá hay giám sát dự án có kịp tiến độ hay không, có thể bàn giao hay release cho khách hàng hay không và các vấn đề cần giải quyết khi mà số lượng lỗi còn nhiều, gây ra các rủi ro về tiến độ hoàn thành của dự án để có những điều chỉnh kịp thời

TEST REPORT là gì?

- ▶ Test Report thường chứa các nội dung như:
 - Số lượng test cases đã viết/ số lượng testcase đã test
 - Số lượng test cases passed/failed
 - Số lượng defects tìm ra và status, severity của defects
 - Số lượng defects trên từng module
 - Các vấn đề liên quan đến testing, bản build, tiến độ sửa lỗi...

TEST REPORT là gì?

I. ĐÁNH GIÁ CHẤT LƯỢNG SẢN PHẨM

Kết quả	Tuần 4/11	Tuần 1/12	Tuần 2/12	Tuần 3/12	Tuần 4/12
Tổng số Lỗi trong Tuần	45	60	61	57	19
Tổng số Lỗi chưa Fix	30	25	40	29	10
Tổng số Lỗi Re-assign	3	3	5	10	0

Nhận xét - Thiếu nhân sự fix bug, số lượng bug nhiều nên gây chậm tiến độ kiểm thử.

- Module Báo cáo được test lần 1 không kỹ nên phải kiểm tra lại và có thêm nhiều bug ngoài phạm vi định rà soát ban đầu

Thông kê mức độ nghiêm trọng của Lỗi

Chức năng	Severity			Tổng Lỗi
	Crash	Major	Minor	
Module A	0	6	1	7
Module B	0	5	5	10
Module C	1	20	33	54
Tổng	1	31	39	71

Số lượng Lỗi được assign cho Developer

Tên Dev	Lỗi
Dev A	20
Dev B	11
Dev C	23
Tổng	54

Defect/ Bug/ Fault Management System

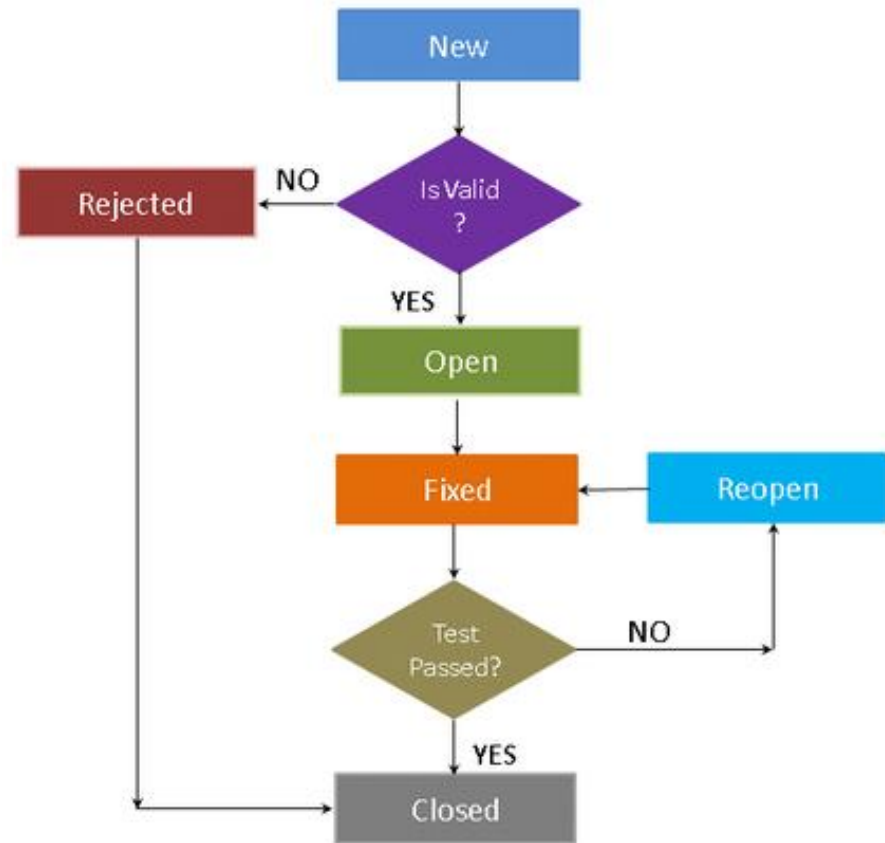
- ▶ Defect, Bug, Fault là gì?
- ▶ Độ ưu tiên, Độ nghiêm trọng trong quản lý Bug?
- ▶ Quy trình xử lý Bug?
- ▶ Tool quản lý Lỗi: như Redmine, Mantis, Jira

Defect/ Bug/ Fault là gì?

- ▶ Testing là công việc không thể thiếu trong quá trình xây dựng sản phẩm phần mềm. Cho dù sản phẩm của bạn là một đoạn chương trình, một chức năng hay toàn bộ ứng dụng thì bạn đều phải thực hiện testing trước khi bàn giao. Đó là lúc kiểm tra lại xem sản phẩm làm có đúng yêu cầu khách hàng không? Có vận hành ổn định trên nhiều tình huống giả định không? Có lỗi phát sinh nào không, nguyên nhân là gì để biết cách khắc phục và hoàn thiện sản phẩm.
- ▶ Lỗi phần mềm được gọi là Defect hoặc Bug hoặc Fault trong tiếng anh.
- ▶ Không phải tất cả các Lỗi gây ra đều xảy ra do code, lỗi có thể đến từ Đặc tả yêu cầu, thiết kế...

Defect/ Bug/ Fault Life Cycle (vòng đời của Lỗi)

- ▶ Tester report 1 bug (trạng thái New) > Xem xét xem có đúng là lỗi không:
 - Nếu không phải là lỗi thì Reject và Closed
 - Nếu là Lỗi thì chuyển sang trạng thái Open > Developer fix lỗi > Tester thực hiện test lại > Nếu lỗi đã Pass thì Close. Nếu Lỗi vẫn còn lỗi thì lại Reopen cho Dev fix lại
- ▶ Tóm lại: tester phát hiện ra Bug, ghi Bug lên hệ thống quản lý Lỗi và assign cho lập trình viên. Lập trình viên fixed lỗi và tester thực hiện test lại. Nếu đã OK thì closed, nếu vẫn lỗi thì chuyển trạng thái bug là reopen để lập trình viên xem xét sửa lỗi.



Redmine tool: Tool quản lý Defect

[Overview](#) [Activity](#) **[Issues](#)** [New issue](#) [Gantt](#) [Calendar](#) [News](#) [Documents](#) [Wiki](#) [Files](#) [Settings](#)

Issues

▼ Filters

☒ Status

closed ▼

Add filter

► Options

✓ Apply

🔄 Clear

💾 Save































✓ #	Độ ưu tiên	Tracker	Status	Priority	Subject	Assignee	Updated	Start date
<input type="checkbox"/> 44316		Bug	Closed	Normal	Ko đổi mã PIN mới trùng với mã PIN mặc định		08/07/2014 09:21 am	08/01/2014
<input type="checkbox"/> 44193		Bug	Closed	Normal	Sai format sms.thông báo tiền điện		08/05/2014 10:16 am	07/29/2014
<input type="checkbox"/> 44188		Bug	Closed	Normal	Lỗi nhập ngày sau khi nhập sai định dạng ngày		08/07/2014 09:22 am	07/29/2014
<input type="checkbox"/> 44186		Bug	Closed	Normal	Sai dữ liệu thuế màn hình thông báo gạch nợ thành công		08/05/2014 10:04 am	07/29/2014
<input type="checkbox"/> 44185		Bug	Closed	Normal	Hiển thị sai thông tin còn nợ trên màn hình hiển thị thông báo gạch nợ thành công		08/07/2014 09:22 am	07/29/2014
<input type="checkbox"/> 44115		Bug	Closed	Normal	Thêm màn hình thông báo gửi SMS thành công hay thất bại		07/28/2014 11:22 am	07/28/2014

Redmine tool: Màn hình viết 1 defect

Tracker *

Subject *

Description

B *I* U ~~S~~ **C** H1 H2 H3                              

Redmine tool: Màn hình Defect đã được Fixed

Issues

▼ Filters



Status

open

Add filter

► Options



Apply



Clear




Save

✓ # ▼	Độ ưu tiên	Tracker	Status	Priority	Subject	Assignee	Updated	Start date	% Done
<input type="checkbox"/> 44691		Bug	Resolved	Immediate	Lỗi hệ thống ko ổn định	Nguyễn Văn Hòa	08/08/2014 05:15 pm	08/07/2014	


Redmine tool: Màn hình Chi tiết Lỗi

Bug #54121

 Update  Watch  Copy

☐ Không có nút back trên các màn hình

« Previous | 9 of 14 | Next »

Added by Vũ  5 days ago. Updated 2 days ago.

Status: Resolved

Priority: High

Assignee:  Đức

Category: Android_R7628

Target version: -

Severity: minor

Plan Start Date:

Plan End Date:

Start date: 01/16/2015

Due date:


% Done:  0%

Spent time: -

Độ ưu tiên:

Nguyên nhân lỗi:

Description

 Quote

1. Đăng nhập vào hệ thống trên phiên bản android
 2. vào các màn hình trang 02, xem chi tiết tài liệu
 3. Kiểm tra các hiển thị nút back trên màn hình
- KQ:
Trên các màn hình đều không có nút back
MM:
Thực hiện bổ sung thêm các nút back vào các chương trình

Cách viết Bug/ Defect

- ▶ Khi viết Bug thì miêu tả lỗi gồm 2 nội dung quan trọng :
 - + **Thực tế:** là miêu tả chức năng đang bị lỗi (cũng có thể capture cả ảnh lỗi)
 - + **Mong muốn:** miêu tả chức năng chạy chính xác/ cách hoạt động đúng

Ví dụ 1:

- ▶ Lỗi như sau: Khi đăng ký email, ko có ký tự @ mà vẫn thành công

Cách viết mô tả lỗi lên tool quản lý lỗi:

- + **Thực tế:** Đăng ký 1 email không đúng định dạng là ko có @ mà vẫn đăng ký thành công
 - + **Mong muốn:** Email phải đúng format mới cho phép đăng ký thành công (có @)
- Định dạng format chuẩn: abc@xy.aa

Cách viết Bug/ Defect

Ví dụ 2:

- ▶ Lỗi như sau: Đăng nhập không thành công khi có khoảng trắng đầu và cuối ở username

Cách viết mô tả lỗi lên tool quản lý lỗi:

- + **Thực tế:** Đăng nhập không thành công khi có khoảng trắng đầu và cuối ở username
- + **Mong muốn:** Đăng nhập thành công khi có khoảng trắng đầu và cuối ở username.
Trim space ở 2 đầu tên đăng nhập

Cách viết Bug/ Defect

- ▶ Cách viết bug bằng tiếng anh như sau:

Username & password are empty but the required message does not display

Added by [REDACTED] 2 days ago. Updated less than a minute ago.

Status:	New	Start date:
Priority:	Normal	Due date:
Assignee:	[REDACTED] Nguyen	% Done:
Category:	-	Spent time:
Target version:	-	

Description

1. Username = empty
2. Password = empty
3. Click on button 'Login'

- Actual Result: The required message does not display
- Expected result: message display ' Username is required', 'Password is required'

Độ Ưu tiên (Priority) & Độ nghiêm trọng (Severity) trong quản lý Bug

- ▶ Trong kiểm thử phần mềm thì hai khái niệm Độ ưu tiên (Priority) và Độ nghiêm trọng (Severity) cũng không quá xa lạ, đặc biệt là trong quản lý bug.
- ▶ Phụ thuộc vào độ ưu tiên mà lập trình viên lần lượt thực hiện fix bug .



Low



Medium



High



Critical

Độ nghiêm trọng: Severity Bug

- ▶ Mức độ nghiêm trọng của một con bug thường chỉ mức độ tác động của con bug đó đến sản phẩm
- ▶ Mỗi dự án hay sản phẩm có tiêu chí đánh giá độ nghiêm trọng khác nhau nhưng thông thường sẽ có 4 mức độ khác nhau từ nghiêm trọng nhất đến ít nghiêm trọng hơn:
 - **Mức độ 1 (Critical):** Rất nghiêm trọng, có thể làm cho PM "chết cứng" và không sử dụng được.
 - **Mức độ 2 (Major):** Nghiêm trọng, buộc phải sửa chữa để có thể sử dụng được như yêu cầu đề ra.
 - **Mức độ 3 (Minor):** Nhẹ, tuy không làm PM ngưng chạy, nhưng làm cho việc sử dụng PM khó khăn hoặc gây bất tiện cho người dùng
 - **Mức độ 4 (Cosmetic):** Không ảnh hưởng đến chức năng hay hiệu năng của PM được quy định trong yêu cầu (như vấn đề thẩm mỹ hoặc thông báo sai chính tả).
- ▶ Thực tế việc xác định độ nghiêm trọng của con bug không hẳn lúc nào cũng mang tính chất trắng-đen và tuyệt đối.

Độ ưu tiên: Priority Bug

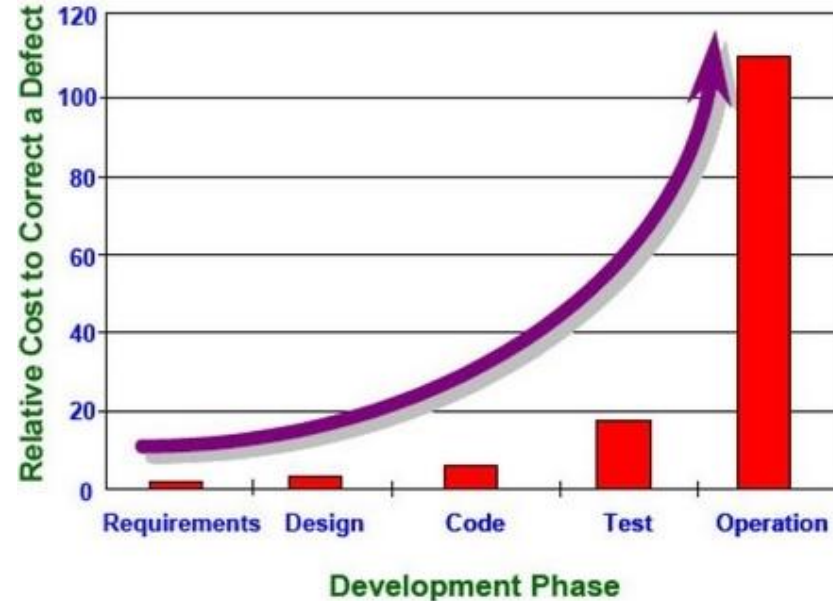
- ▶ Đã là bug thì sẽ phải sửa. Tuy nhiên, có một thực tế là đội phát triển khó có thể sửa hết tất cả bug một lượt cũng như không đáng để sửa hết tất cả các bug. Do đó, đội phát triển sẽ phải cần đến độ ưu tiên của con bug để biết được bug nào cần được sửa trước bug nào sau. Độ ưu tiên của bug cũng thường được chia thành 3-4 cấp độ:
 - **Mức độ 1 (Immediate):** Bug sẽ phải sửa ngay lập tức, nếu không công việc sẽ không thể tiếp tục.
 - **Mức độ 2 (High):** độ ưu tiên cao; công việc sẽ bị ngăn trở rất nhiều nếu như lỗi vẫn chưa được sửa.
 - **Mức độ 2 (Medium):** độ ưu tiên trung bình; công việc sẽ gặp vài khó khăn nếu như lỗi vẫn chưa được sửa.
 - **Mức độ 3 (Low):** độ ưu tiên thấp nhất; công việc không bị ảnh hưởng nhưng lỗi vẫn phải được sửa.

Độ ưu tiên: Priority Bug

- ▶ Xác định độ ưu tiên? Bug nào sửa trước bug nào sửa sau (hoặc không sửa)? Dựa vào độ nghiêm trọng của bug. Bug nào nghiêm trọng nhất, tác động đến người dùng nhiều nhất thì sẽ được ưu tiên sửa trước. Bug nào ít nghiêm trọng hơn sẽ được sửa sau.
- ▶ Xác định độ ưu tiên được khuyến khích đối với kỹ sư kiểm thử nhưng không phải bắt buộc.

Chi phí của Lỗi (cost of defect) là gì?

- ▶ Nếu một lỗi xảy ra và được phát hiện ở bước Requirement, thì sẽ dễ dàng sửa với chi phí rẻ. Tuy nhiên nếu lỗi xảy ra ở bước Requirement nhưng lại được phát hiện ở bước Acceptance test thì sẽ phải tốn nhiều chi phí hơn để sửa. Bởi vì chúng ta sẽ phải sửa lại cho đúng với yêu cầu khách hàng sau đó phải design và code lại theo yêu cầu đã sửa, việc này rất tốn thời gian và nỗ lực, chưa kể đến việc còn có thể làm mất uy tín của công ty.
- ▶ Biểu đồ bên cạnh biểu thị của chi phí gia tăng của việc sửa Lỗi theo từng giai đoạn phát triển



Software Testing: Verification & Validation (V&V)

- **Verification**

“Are we building the product right?”

- **Validation**

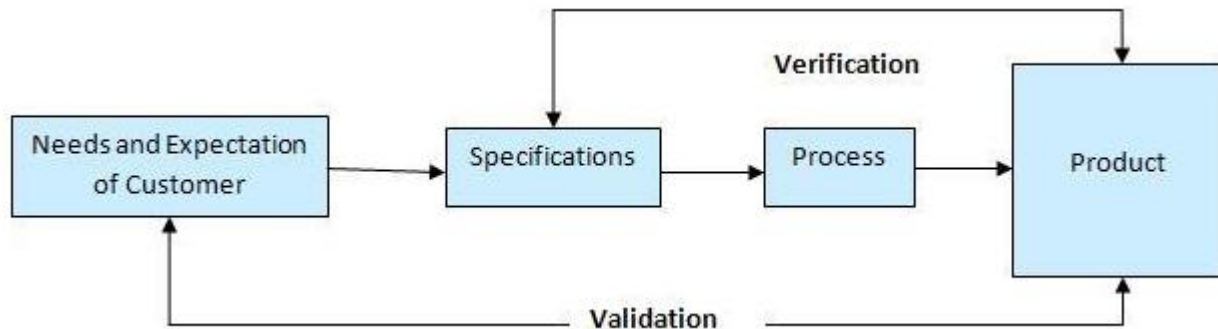
“Are we building the right product?”

Barry Boehm, 1979



Verification and Validation (Xác minh và Thẩm định)

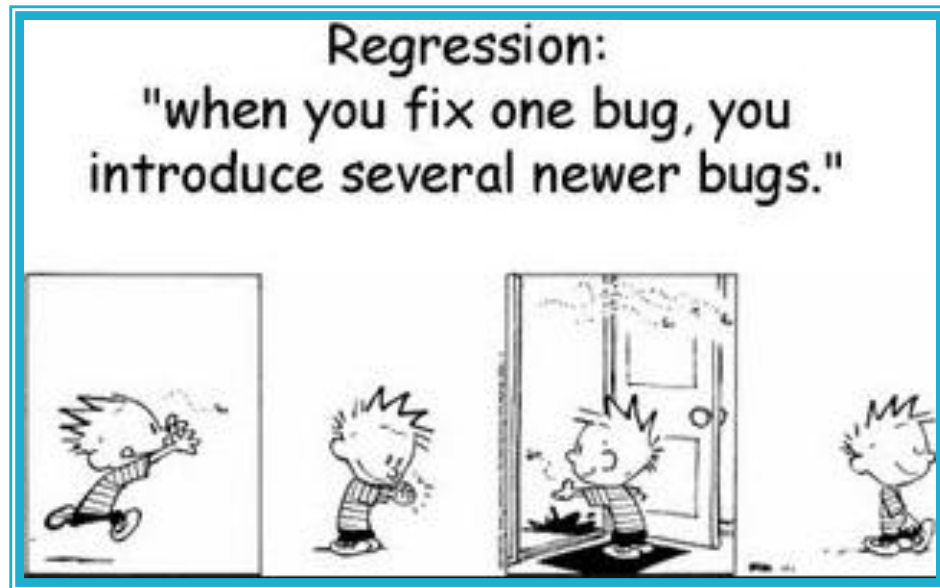
- ▶ Software Testing là:
 - Là một quá trình thực thi một chương trình với mục đích tìm lỗi.
 - Là hoạt động kiểm tra xem phần mềm có chạy chính xác hay không (**Verification**) và có thoả mãn yêu cầu của khách hàng hay không (**Validation**) nhằm hướng tới mục tiêu chất lượng của phần mềm.
- ▶ Quality Testing = Verification + Validation



Verification and Validation (V&V)

Verification (xác minh)	Validation (thẩm định)	Software Testing
<ul style="list-style-type: none">- Đảm bảo phần mềm thực hiện <u>đúng đặc tả yêu cầu, có đúng thiết kế hay không</u>.- Phát hiện lỗi lập trình	<ul style="list-style-type: none">- Đảm bảo phần mềm <u>đáp ứng nhu cầu người dùng</u>.- Phát hiện lỗi phân tích, thiết kế.	<ul style="list-style-type: none">- Là cả hai quá trình : kiểm tra phần mềm có chạy chính xác hay không và có thỏa mãn yêu cầu khách hàng hay không.- Đảm bảo chất lượng phần mềm
<ul style="list-style-type: none">- Verification đảm bảo rằng "you built it right".	<p>Validation đảm bảo rằng "you built the right thing".</p>	

Re-testing # Regression testing (test hồi quy)

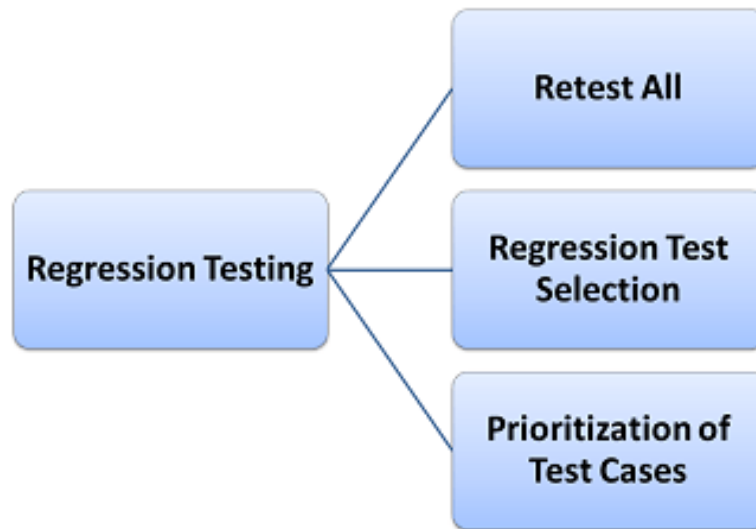


Test Hồi Quy là gì?

- ▶ Khi một chức năng mới được thêm vào phần mềm, chúng ta cần chắc chắn rằng phần chức năng mới được thêm vào không phá hỏng các phần khác của ứng dụng. Hoặc khi lỗi đã được chỉnh sửa, chúng ta cần chắc chắn rằng lỗi chỉnh sửa không phá hỏng các phần khác trong ứng dụng. Để test điều này chúng ta thực hiện kiểu test lặp đi lặp lại gọi là test hồi quy.
- ▶ Test hồi quy được thực hiện đối với những phần nằm trong phạm vi bị ảnh hưởng khi mà có sự sửa đổi một chức năng nào đó hoặc là thêm mới, đảm bảo những sự thay đổi không gây ra lỗi mới trên những chức năng vốn đã làm việc tốt.
- ▶ Regression Test không phải là một mức kiểm tra (giống như unit test, integration testing, system test & acceptance test). Regression test có thể thực hiện tại mọi mức kiểm tra.

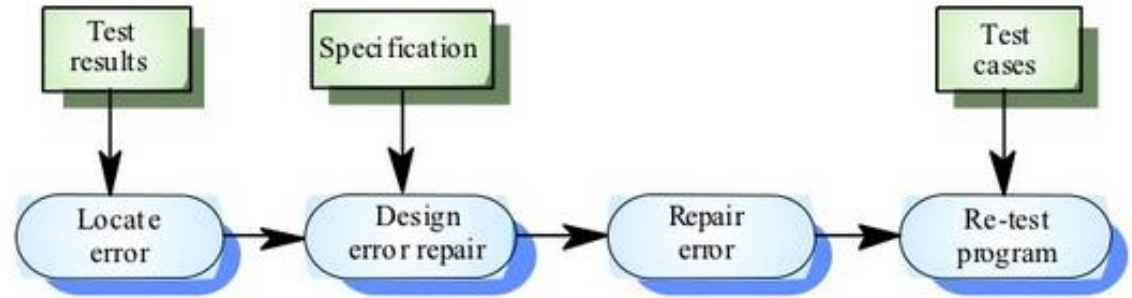
Test Hồi Quy là gì?

- ▶ Mặc dù không là một mức kiểm tra, thực tế lại cho thấy Regression Test là một trong những loại kiểm tra tốn nhiều thời gian và công sức nhất. Tuy thế, việc bỏ qua Regression Test là "không được phép" vì có thể dẫn đến tình trạng phát sinh hoặc tái xuất hiện những lỗi nghiêm trọng, mặc dù ta "tưởng rằng" những lỗi đó hoặc không có hoặc đã được kiểm tra và sửa chữa rồi!



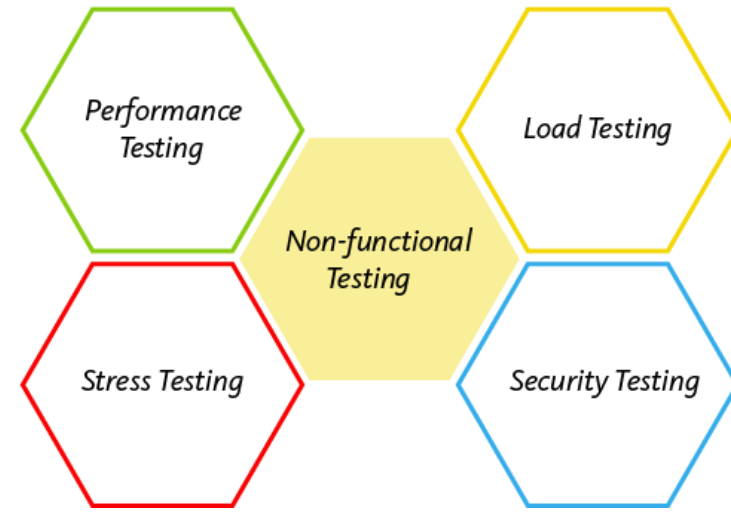
Re-test là gì?

- ▶ Re-test: Đồng nghĩa với confirmation testing (kiểm tra xác nhận)
Là thực hiện test để kiểm tra xem bug mình đã post có được fixed hay chưa (kiểm tra lại xem đã hết bị lỗi mà mình đã gặp chưa)
Nếu đã được sửa xong thì mình báo cáo Close bug
Ngược lại, nếu vẫn còn lỗi thì báo cáo re-open để DEV sửa lại.
- ▶ Là một loại kiểm thử thực hiện để kiểm tra lỗi được fix đã ok chưa



Functional testing # Non-functional testing?

- ▶ **Kiểm thử Chức năng là gì?**
- ▶ **Kiểm thử Phi Chức năng là gì**



Functional testing là gì?

- ▶ Kiểm thử chức năng: tương tự black box testing (kiểm tra đến chức năng của chương trình mà không quan tâm đến cấu trúc bên trong). kiểm thử chức năng là chỉ tập trung kiểm tra chức năng của ứng dụng đó có hoạt động đúng như khách hàng mong đợi không? Khi test sẽ dựa vào tài liệu yêu cầu (requirement) hoặc tài liệu mô tả chi tiết (specification) để test
- ▶ Functional testing cũng nằm trong System testing:

Functional testing là gì?

- ▶ System Test kiểm tra cả các hành vi chức năng của phần mềm lẫn các yêu cầu về chất lượng như độ tin cậy, tính tiện lợi khi sử dụng, hiệu năng và bảo mật. System Test lại gồm nhiều loại kiểm tra khác nhau, phổ biến nhất gồm:
 - Kiểm tra chức năng (Functional Test): bảo đảm các hành vi của hệ thống thỏa mãn đúng yêu cầu thiết kế.
 - Kiểm tra khả năng vận hành (Performance Test)
 - Kiểm tra khả năng chịu tải (Stress Test hay Load Test)
 - Kiểm tra cấu hình (Configuration Test)
 - Kiểm tra khả năng bảo mật (Security Test)
- ▶ Lưu ý: không nhất thiết phải thực hiện tất cả các loại kiểm tra nêu trên. Tùy yêu cầu và đặc trưng của từng hệ thống, tùy khả năng và thời gian cho phép của dự án, khi lập kế hoạch sẽ quyết định áp dụng những loại kiểm tra nào.

Non-functional testing là gì?

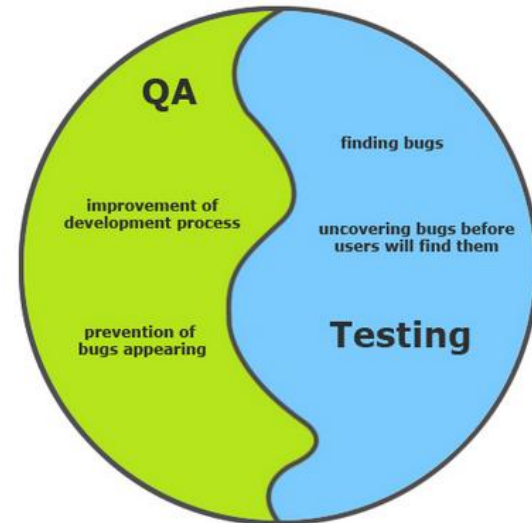
- ▶ Kiểm thử phi chức năng như:
 - Performance testing (kiểm thử hiệu năng)
 - Stress testing: kiểm tra các giới hạn của hệ thống
 - Security testing (kiểm thử bảo mật)
 - Usability testing (kiểm thử tính khả dụng)

Testing Performance là ?

- ▶ Kiểm thử hiệu năng được thực hiện để xác định tốc độ một hệ thống thực hiện hay xử lý một khối lượng công việc cụ thể.
- ▶ Kiểm tra khả năng chịu tải (Stress Test hay Load Test): bảo đảm hệ thống vận hành đúng dưới áp lực cao (ví dụ nhiều người truy xuất cùng lúc). Stress Test tập trung vào các trạng thái tới hạn, các "điểm chết", các tình huống bất thường...
- ▶ Kiểm tra khả năng bảo mật (Security Test): bảo đảm tính toàn vẹn, bảo mật của dữ liệu và của hệ thống.
- ▶ Các thuật ngữ load testing, performance testing, reliability testing, và volume testing thường có thể sử dụng thay thế cho nhau.
- ▶ Để test hiệu năng, thì sử dụng tool test tự động như Load Runner, Apache Jmeter
- ▶ Ví dụ: Kiểm thử load test cho Đăng nhập: cho 100user login cùng lúc, sau đó thử 200user, 500user, 1000user,... và xem kết quả xử lý của website: thời gian đáp ứng bao nhiêu ms, bao nhiêu giao dịch thất bại/ thành công, có lỗi xảy ra trong quá trình thực hiện ko....?

QC (Quality Control) # QA (Quality Assurance)

- ▶ **QA: Giám sát, quản lý và đảm bảo chất lượng**
- ▶ **QC: Kiểm soát chất lượng sản phẩm**

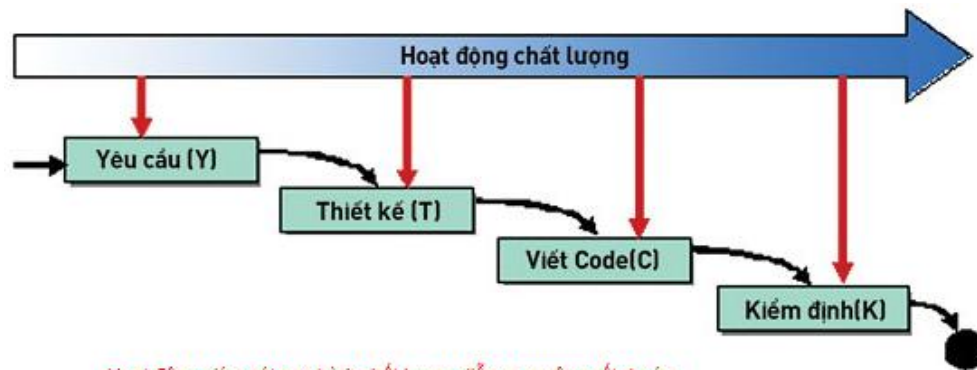


QA = Đảm bảo Chất lượng

- ▶ Cả 2 lĩnh vực QA và QC đều làm bên mảng quản lý chất lượng
- ▶ QA có nhiệm vụ giám sát để bảo đảm các tiêu chuẩn và quy trình sản xuất PM được định nghĩa và tuân thủ nghiêm túc, hướng đến mục tiêu các sản phẩm (SP) trung gian cũng như SP sau cùng của dự án thỏa mãn các tiêu chuẩn và yêu cầu đã định trước đó. Công việc của QA liên quan đến quy trình (process).
- ▶ Thông thường, trong một dự án PM, khách hàng chỉ trả tiền cho lực lượng sản xuất trực tiếp như developer và tester, do đó đầu tư vào lực lượng QA là một đầu tư mang tính nội bộ và nền tảng, giúp công ty cải tiến và kiểm soát các quy trình đảm bảo chất lượng
- ▶ Một khi quy trình sản xuất được tuân thủ nghiêm túc, lỗi xuất hiện ở các khâu sản xuất sẽ được nhận diện và ngăn chặn sớm, SP sau cùng (ở chặng kiểm định) sẽ ít lỗi và khả năng thành công của dự án được bảo đảm hơn, do đó tổng chi phí sẽ thấp hơn.

QA = Đảm bảo Chất lượng

- Quy trình bao gồm các biểu mẫu, các bước và hướng dẫn cụ thể giúp thành viên dự án thực hiện một công việc nào đó một cách nhất quán và có kiểm soát. Có rất nhiều quy trình khác nhau được thiết kế tùy theo nhu cầu của một dự án như Quy trình phát triển yêu cầu SP (Requirement); Quy trình thiết kế SP (Design); Quy trình triển khai và viết code (Coding); Quy trình kiểm định SP (Testing); Quy trình cài đặt và hỗ trợ (Delivery); Quy trình bảo trì SP (Maintenance); Quy trình quản trị dự án (Project management); Quy trình quản lý cấu hình (Configuration management); Quy trình đảm bảo chất lượng (Quality Assurance)



Hoạt động giám sát quy trình chất lượng diễn ra xuyên suốt dự án

QA = Đảm bảo Chất lượng

- ▶ QA là người tham gia phát triển quy trình hoạt động ở cấp công ty và ở cấp dự án, đánh giá các tài liệu. Ngoài ra, họ còn phải giám sát và kiểm tra (*audit*) các hoạt động được thực hiện trong dự án xem chúng có tuân thủ các quy trình (*process*) đã được định ra; xác định các điểm không tương thích với quy trình (*process noncompliance* – gọi tắt là *NC*) và báo cáo cho những người liên quan và các cấp quản lý đồng thời giám sát để bảo đảm chúng được giải quyết đến khi hoàn tất.
- ▶ Nhân sự QA cũng có thể phản hồi về những bất cập của quy trình và đề xuất cải tiến quy trình.
- ▶ QA cũng hay gọi là PQA (Process Quanlity Assurance – Cán bộ quản lý chất lượng quy trình)

QC = Quality Control

- ▶ QC chính là Tester. Nhiều công ty gọi là QC chứ không phải là tester.
- ▶ Quality Control: Điều khiển chất lượng
 - Tập hợp các hoạt động được tạo ra nhằm đánh giá chất lượng sản phẩm, bảo đảm sản phẩm đúng đặc tả yêu cầu
 - Trực tiếp kiểm tra chất lượng của sản phẩm
 - Có nhiệm vụ khảo sát, chạy thử và báo cáo lỗi

Hỏi & Đáp!



TESTER là?

- T** *Take care of quality (Chăm lo cho chất lượng)
- E** *Eager for finding defect (Ham tìm lỗi)
- S** *Standardize software (Chuẩn hóa phần mềm)
- T** *Thought of logic (Tư duy logic)
- E** *Enjoyable job (Nghề thú vị)
- R** *Raise of carefulness (Tăng thêm sự cẩn thận)

Trân trọng cảm ơn!