



Chương 5: **Thực thi Kiểm thử trong** **BĐCLPM**

Khoa Công nghệ thông tin
Trường Đại học Nguyễn Tất Thành

- 5.1/. Kiểm thử đơn vị
- 5.2/. Kiểm thử tích hợp
- 5.3/. Kiểm thử hệ thống
- 5.4/. Kiểm thử hồi qui
- 5.5/. Báo cáo lỗi và tái phục hồi lỗi
- 5.6/. Kiểm toán phần mềm

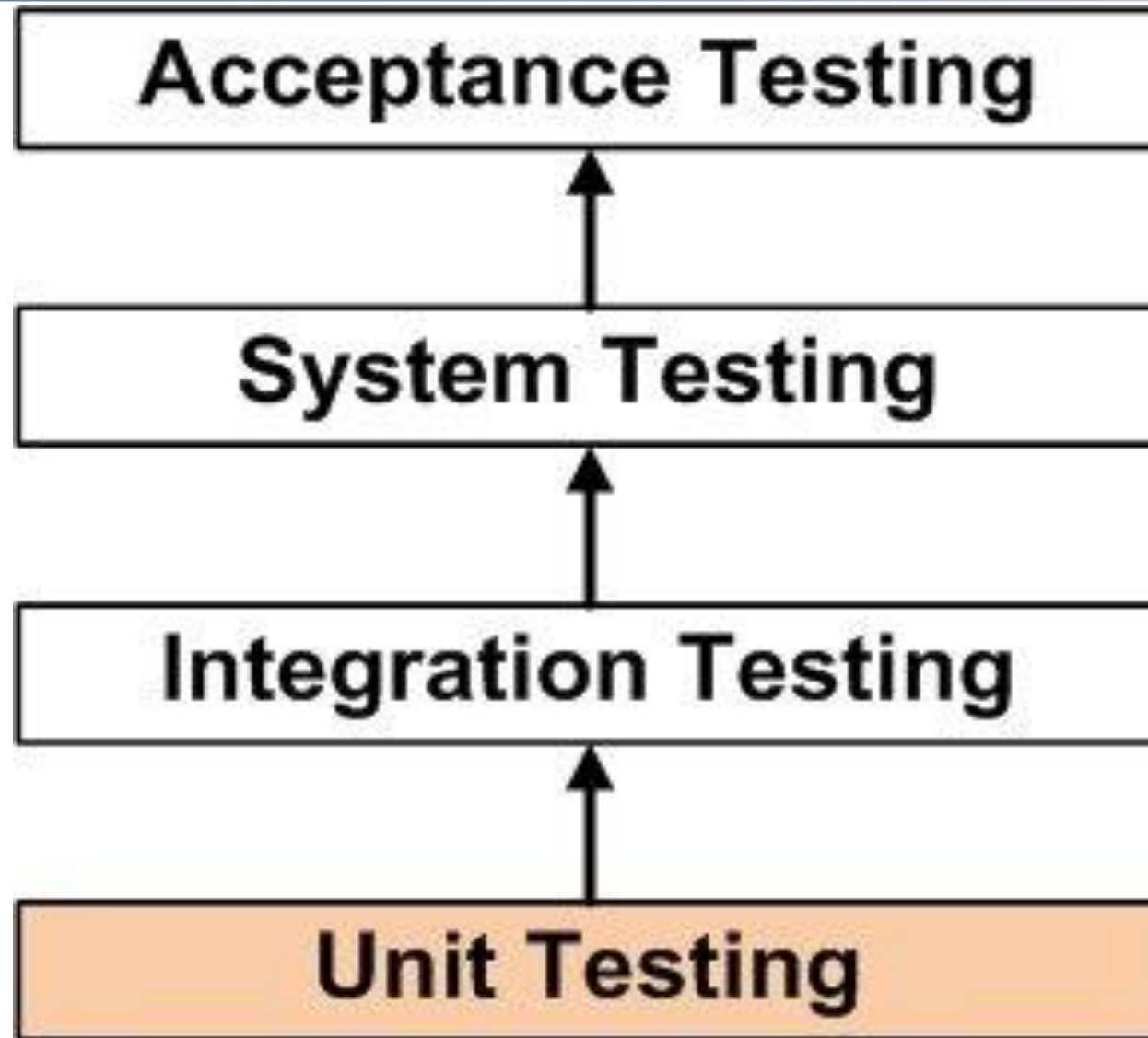
1. Kiểm thử đơn vị

Kiểm thử đơn vị (Unit Testing) là mức kiểm thử cơ bản, trong đó các đơn vị / thành phần riêng lẻ của một phần mềm được kiểm tra. Mục đích là để xác nhận rằng mỗi đơn vị của phần mềm thực hiện như thiết kế. Một đơn vị là thành phần nhỏ nhất có thể kiểm tra, mỗi đơn vị phải có đầu vào và đầu ra để có thể kiểm tra được (loại kiểm thử White Box).

- Trong lập trình thủ tục: đơn vị là một chương trình riêng lẻ như hàm, Module
- Trong lập trình hướng đối tượng: đơn vị là một phương thức, có thể thuộc về một lớp cơ sở, lớp cha, lớp trừu tượng hoặc lớp dẫn xuất / lớp con.

1. Kiểm thử đơn vị

Sinh viên xem hình và xác định vị trí, vai trò của Kiểm thử đơn vị.



1. Kiểm thử đơn vị

Chi phí sửa chữa một khiếm khuyết được phát hiện trong quá trình kiểm thử ở mức đơn vị là ít hơn so với các lỗi được phát hiện ở các cấp cao hơn.

Các công cụ kiểm thử đơn vị:

- MS Test trong Visual Studio
- NUtnit
- ...

2. Kiểm thử tích hợp

Kiểm thử tích hợp (Integration Testing) là một cấp độ kiểm thử phần mềm trong đó các đơn vị riêng lẻ được kết hợp và kiểm tra theo một nhóm. Mục đích của kiểm thử tích hợp này là để lộ các lỗi trong tương tác giữa các đơn vị tích hợp.

Kiểm thử tích hợp xảy ra sau kiểm thử đơn vị và trước kiểm thử hệ thống.

Trình điều khiển kiểm thử và các tình huống kiểm thử được sử dụng để hỗ trợ Kiểm thử tích hợp.

2. Kiểm thử tích hợp

Ví dụ về Kiểm thử tích hợp: Giả sử bạn làm việc cho công ty phát triển PM đã được yêu cầu phát triển trang Web mua bán trực tuyến cho một công ty bán dụng cụ làm vườn. Sau khi thu thập yêu cầu, phân tích và thiết kế hoàn tất, nhà phát triển đã phát triển từng Module dưới đây:

- Đăng ký và xác thực người dùng / Đăng nhập
- Danh mục sản phẩm
- Giỏ hàng
- Thanh toán
- Tích hợp cổng thanh toán
- Theo dõi vận chuyển và gói hàng

2. Kiểm thử tích hợp

- Nhà phát triển phải kiểm tra từng chức năng là một phần của kiểm thử đơn vị và tìm thấy một số khiếm khuyết. Sau đó sửa chữa những khuyết điểm này. Tại thời điểm này, xem như các Module đã hoàn thành.
- Kiểm thử tích hợp sẽ được thực hiện để xác nhận rằng tất cả các Module hoạt động cùng nhau, ví dụ như xem danh mục sản phẩm và chọn đưa vào giỏ hàng có phối hợp hoạt động tốt không, .. có các lỗi đăng nhập, thêm xóa sửa giỏ hàng, thanh toán có hoạt động đúng như thiết kế không?
- Theo cách này, Kiểm thử tích hợp giúp chúng ta xác định, khắc phục các sự cố và đảm bảo rằng toàn bộ ứng dụng phối hợp hoạt động như mong đợi.

2. Kiểm thử tích hợp

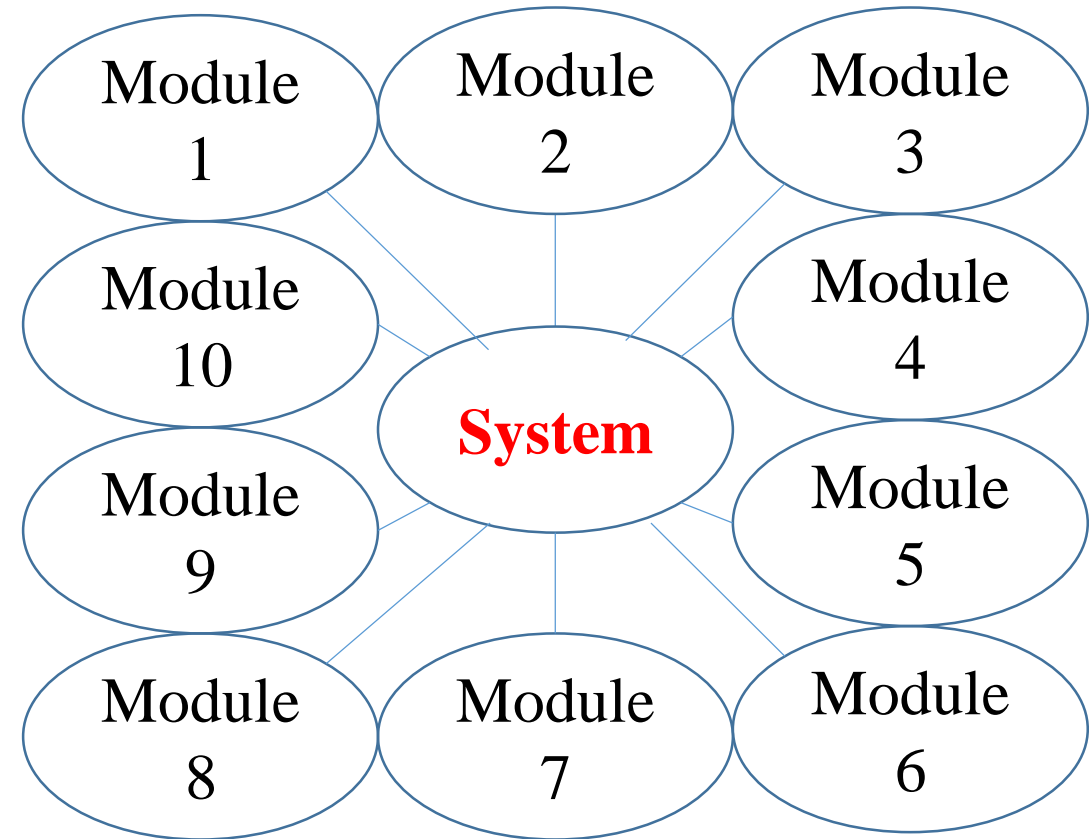
Cách tiếp cận, phương pháp, chiến lược của kiểm thử tích hợp:

- Có nhiều cách tiếp cận khác nhau để kiểm thử tích hợp. Các phương pháp phổ biến và được sử dụng thường xuyên nhất là Kiểm thử Big Bang, Kiểm thử Top_Down, Kiểm thử Bottom_Up.
- Sự lựa chọn của phương pháp phụ thuộc vào các yếu tố khác nhau như chi phí, độ phức tạp, mức độ quan trọng của ứng dụng, v.v. Ngoài ra, có nhiều loại kiểm thử tích hợp ít được biết đến như tích hợp dịch vụ phân tán, tích hợp Sandwich, tích hợp đường trực, tích hợp tần số cao, tích hợp lớp, v.v.

2. Kiểm thử tích hợp

Kiểm thử tích hợp Big Bang:

- Tất cả các thành phần hoặc Module được tích hợp lần lượt và đồng thời, sau đó mọi thứ được kiểm tra tổng thể. Theo hình ảnh trên, tất cả các Module từ Module 1, đến Module N, được tích hợp dần dần sau đó kiểm thử được thực hiện đồng thời cho tất cả.

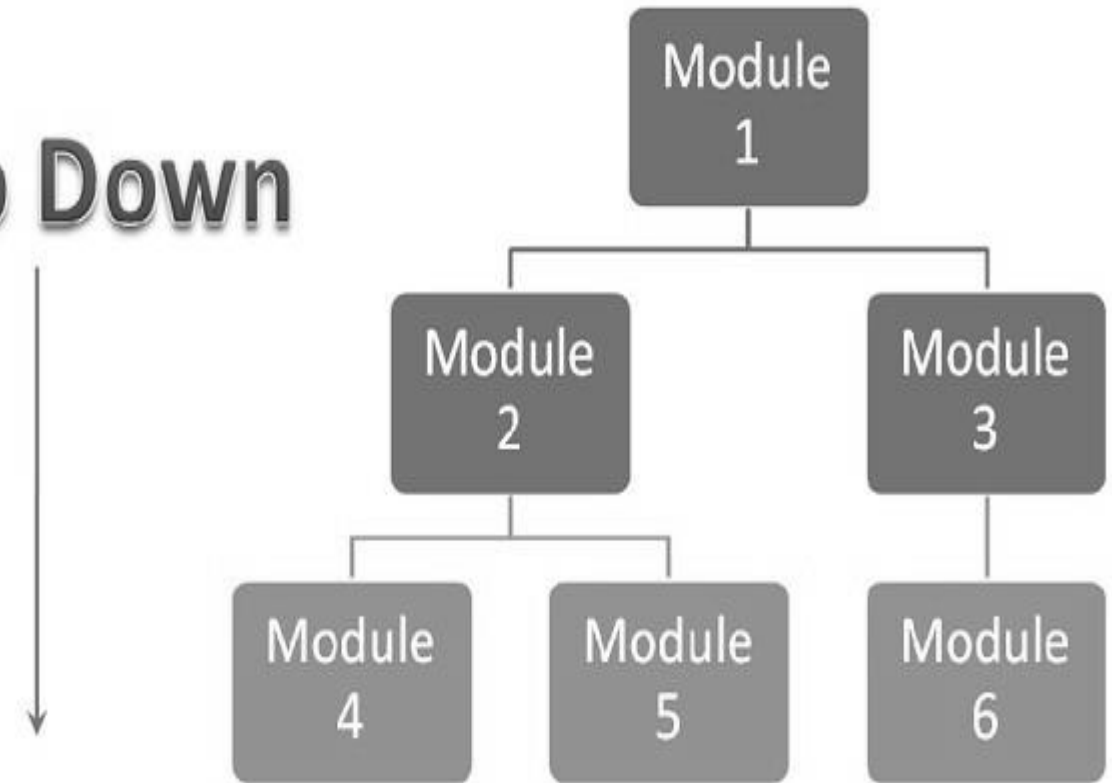


2. Kiểm thử tích hợp

Kiểm thử tích hợp Top_Down:

- Việc kiểm tra diễn ra từ trên xuống dưới, theo dòng điều khiển hoặc cấu trúc kiến trúc (ví dụ: bắt đầu từ menu chính). Nó được sử dụng cho Stub testing. Dưới đây là sơ đồ của cách tiếp cận từ trên xuống:

Top Down

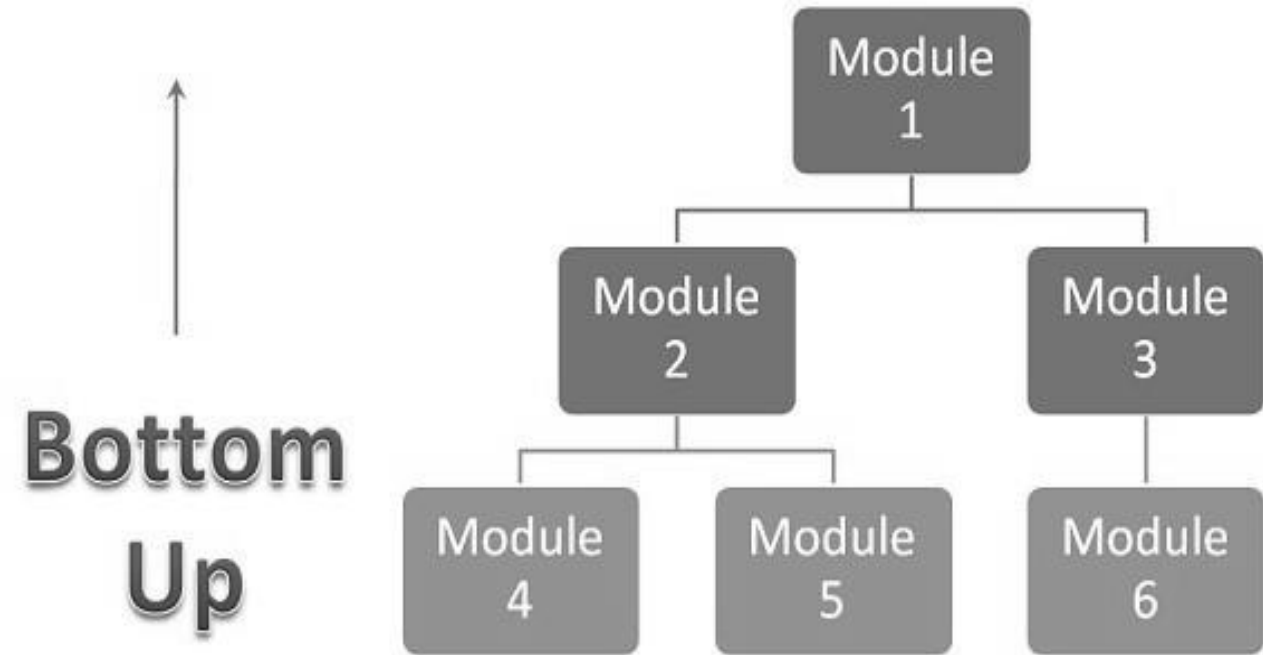


2. Kiểm thử tích hợp

Kiểm thử tích hợp

Bottom_Up:

- Mỗi Module ở mức thấp được kiểm thử với các Module cao hơn cho đến khi tất cả các Module đều được kiểm tra. Nó được sử dụng cho Driver Testing. Thể hiện bằng biểu đồ dưới đây:



2. Kiểm thử tích hợp

Kiểm thử tích hợp gia tăng:

- Trong phương pháp này, kiểm thử được thực hiện bằng cách kết hợp hai hoặc nhiều Module có liên quan một cách hợp lý cho từng chức năng. Sau đó, tiếp tục kết hợp cho các chức năng khác đến khi tất cả các Module được tham gia và kiểm thử thành công.

3. Kiểm thử hệ thống

- Kiểm thử hệ thống (System Testing) là kiểm thử một sản phẩm PM hoàn chỉnh và tích hợp đầy đủ. Thông thường, phần mềm chỉ là một thành phần của hệ thống trên máy tính. Phần mềm được tương tác với các hệ thống phần mềm hay phần cứng khác. Kiểm thử hệ thống thực hiện một loạt các kiểm thử khác nhau với mục đích kiểm tra phần mềm có tương thích với toàn bộ hệ thống trên máy tính hay không?
- Kiểm thử hệ thống thuộc loại kiểm thử hộp đen (Black Box Testing), dựa trên đầu vào và đầu ra của chương trình để kiểm thử, bao gồm kiểm tra các chức năng và phi chức năng của sản phẩm PM.

3. Kiểm thử hệ thống

Sau đây là danh sách các loại kiểm thử hệ thống mà các công ty phát triển phần mềm thường sử dụng:

- **Kiểm thử khả năng sử dụng - Usability Testing:** Kiểm thử khả năng sử dụng chủ yếu tập trung vào việc người dùng dễ dàng sử dụng ứng dụng, linh hoạt trong việc kiểm soát xử lý và khả năng của hệ thống để đáp ứng các mục tiêu.
- **Kiểm thử tải - Load Testing:** Kiểm thử tải là cần thiết để biết rằng một phần mềm sẽ thực hiện theo tải thực tế.

3. Kiểm thử hệ thống

- **Kiểm thử phục hồi - Recovery Testing:** Kiểm thử phục hồi được thực hiện để chứng minh một giải pháp phần mềm là đáng tin cậy và có thể phục hồi thành công khi các sự cố xảy ra.
- **Kiểm thử di chuyển - Migration Testing:** Kiểm thử di chuyển được thực hiện để đảm bảo rằng phần mềm có thể được chuyển từ cơ sở hạ tầng từ hệ thống cũ sang cơ sở hạ tầng hệ thống mới mà không gặp sự cố nào.

3. Kiểm thử hệ thống

- **Kiểm thử chức năng - Functional Testing:** Còn được gọi là kiểm thử tính đầy đủ của chức năng. Tester có thể lập danh sách các chức năng bổ sung mà sản phẩm có thể phải cải thiện trong quá trình kiểm thử chức năng.
- **Kiểm thử phần cứng / phần mềm - Hardware/Software Testing:** IBM gọi kiểm thử phần cứng / phần mềm là Kiểm thử CTNH / SW, là khi tester tập trung sự chú ý của mình vào các tương tác giữa phần cứng và phần mềm trong quá trình kiểm thử hệ thống.

4. Kiểm thử hồi quy

Kiểm thử hồi quy - Regression Testing: Kiểm thử hồi quy bao gồm kiểm thử được thực hiện để đảm bảo không có sự thay đổi nào phát sinh ra lỗi mới trong quá trình triển phần mềm. Nó cũng đảm bảo không có lỗi cũ xuất hiện từ việc bổ sung các module mới theo thời gian.



4. Kiểm thử hồi quy

Thông thường có 3 phương pháp để kiểm thử hồi quy:

- i. Kiểm thử lại tất cả: là kiểm tra lại tính toàn vẹn của phần mềm từ trên xuống dưới.
- ii. Kiểm tra hồi quy lựa chọn: Phương pháp này cho phép nhóm chọn một lựa chọn đại diện để kiểm thử (xấp xỉ kiểm thử đầy đủ). Điều này cần ít thời gian hoặc chi phí hơn nhiều so với bộ kiểm thử tất cả.
- iii. Kiểm thử hồi quy đơn vị: nghĩa là chỉ cần Kiểm thử lại với Module / vùng bị thay đổi của ứng dụng.

4. Kiểm thử hồi quy

Thảo luận:

- Kiểm thử hồi quy có phải là kiểm thử lại ?

5. Báo cáo lỗi và tái phục hồi lỗi

Báo cáo lỗi là một khía cạnh quan trọng của kiểm thử phần mềm. Một báo cáo lỗi tốt giúp giao tiếp tốt với đội ngũ phát triển hiệu quả và tránh nhầm lẫn, bên cạnh đó cũng là để phòng ngừa và phát hiện những vấn đề nghiêm trọng xảy ra, liên quan và ảnh hưởng trực tiếp đến chất lượng sản phẩm PM.

Báo cáo lỗi cần thiết phải là báo cáo lỗi hiệu quả và trung thực.

5. Báo cáo lỗi và tái phục hồi lỗi

Báo cáo lỗi hiệu quả:

- Một báo cáo lỗi tốt phải rõ ràng và súc tích mà vẫn không thiếu các điểm chính. Bất kỳ sự thiếu rõ ràng nào sẽ dẫn đến sự hiểu nhầm và làm ảnh hưởng đến quá trình phát triển dự án. Viết báo lỗi là một trong những giai đoạn quan trọng nhất trong việc kiểm thử giúp cho người đọc có thể dễ dàng nhìn ra được vấn đề đang xảy ra.
- Viết báo cáo là một quá trình quan trọng để gửi một lỗi đến nhà phát triển (Dev). Hãy luôn tạo sự thoải mái giữa Tester và Dev.

5. Báo cáo lỗi và tái phục hồi lỗi

Báo cáo lỗi hiệu quả: (tt)

- Nên kiểm tra danh sách các lỗi đã có sẵn nhằm tránh báo cáo lại những bug bị trùng. Lỗi trùng lặp xảy ra sẽ khiến Dev dễ mất lòng tin vào mình, và gây tốn thời gian của dự án
- Thông tin quan trọng mà báo cáo lỗi phải có là H2W: “How_Làm thế nào?” , “Where _ Ở đâu?” và “When _ Xảy ra khi nào?”. Báo cáo lỗi phải trả lời rõ ràng các hỏi : "cách kiểm tra đã được thực hiện như thế nào và nơi xảy ra lỗi là ở đâu?" Người đọc sẽ dễ dàng hiểu và tìm lỗi ở đâu.

5. Báo cáo lỗi và tái phục hồi lỗi

Báo cáo lỗi hiệu quả: (tt)

- Luôn nhớ là mục tiêu của việc viết báo cáo lỗi là để cho người đọc hình dung ra vấn đề đang xảy ra. Hãy tập trung vào một vấn đề cụ thể đang gặp phải, cung cấp đầy đủ thông tin có liên quan mà Dev đang tìm kiếm. Tránh mô tả chung nhiều vấn đề vào 1 bug sẽ gây khó hiểu cho người đọc, khiến Dev mất nhiều thời gian để có thể giải quyết và giải quyết được bug.

5. Báo cáo lỗi và tái phục hồi lỗi

Các nội dung trong báo cáo lỗi: Mỗi công ty, dự án khác nhau sẽ có cách quản lý lỗi khác nhau, các biểu mẫu và cách báo cáo lỗi khác nhau. Nhưng đều phải đảm bảo và lưu ý một số yếu tố dưới đây:

1/. Bug number / ID: Số lỗi / ID giúp cho báo cáo lỗi và đề cập đến một lỗi dễ dàng hơn. Giúp đội phát triển có thể dễ dàng kiểm tra lỗi đã được sửa chữa (fix) hay chưa? Giúp Tester có thể tìm kiếm thực hiện Test và Test lại dễ dàng.

5. Báo cáo lỗi và tái phục hồi lỗi

Các nội dung trong báo cáo lỗi: (tt)

2/. Bug Title: Tiêu đề lỗi sẽ được đọc nhiều nhất trước khi mở lỗi và đọc những phần tiếp theo. Tiêu đề lỗi nên gợi ý đủ về những gì lỗi đang xảy ra. Một tiêu đề rõ ràng làm cho người đọc hiểu rõ hơn, có thể biết được lỗi đã được báo cáo trước đó hay chưa hoặc đã được sửa hay chưa?

3/. Priority: Dựa vào mức độ ưu tiên là sự nghiêm trọng của lỗi, ta có thể đặt ra độ ưu tiên cho nó để cho thấy mức độ ảnh hưởng của lỗi đến hệ thống. Thường thì sẽ dựa vào chỉ số này để ưu tiên các bug nào sẽ phải được/nên được sửa chữa trước.

5. Báo cáo lỗi và tái phục hồi lỗi

Các nội dung trong báo cáo lỗi: (tt)

4/. Description: Mô tả lỗi giúp Dev hiểu được lỗi. Nó mô tả các vấn đề đang gặp phải. Mô tả không rõ ràng sẽ tạo ra sự nhầm lẫn và lãng phí thời gian của Dev. Cần phải thông báo rõ ràng về hiệu quả trong mô tả. Sẽ rất hữu ích nếu sử dụng các câu hoàn chỉnh. Một báo cáo tốt là có thể tách riêng từng vấn đề một cách riêng biệt thay vì gộp chung vấn đề. Không sử dụng cụm từ như ‘Tôi nghĩ’ hoặc ‘Tôi tin’, ‘rất có khả năng’,...

5. Báo cáo lỗi và tái phục hồi lỗi

Các nội dung trong báo cáo lỗi: (tt)

5/. Step: Một báo cáo lỗi tốt nên đề cập rõ đến các bước để **tái phục hồi lỗi**. Các bước nên bao gồm các hành động gây ra lỗi. Không đưa ra các tuyên bố chung, hãy cụ thể trong các bước để làm theo. Vd: Các bước kiểm tra thông tin profile hiển thị:

- Đăng nhập vào hệ thống
- Chuyển đến màn hình Profile
- Kiểm tra thông tin cá nhân hiển thị

5. Báo cáo lỗi và tái phục hồi lỗi

Các nội dung trong báo cáo lỗi: (tt)

6/. Environment: Môi trường Test và cấu hình hệ thống là 1 yếu tố cần thiết cho một báo cáo lỗi rõ ràng. Đó là cách tốt nhất để **tái phục hồi lỗi**. Nếu môi trường không chính xác, PM có thể hoạt động khác và lỗi không thể tái hiện được. Vì vậy tốt nhất nên đề cập đến rõ ràng môi trường phát hiện lỗi. Ví dụ như:

- Thiết bị : Đang sử dụng loại phần cứng nào? Cấu hình cụ thể nào?
- OS : Sử dụng phiên bản nào của hệ điều hành ?
- Browser: trình duyệt được dùng là gì?

5. Báo cáo lỗi và tái phục hồi lỗi

Các nội dung trong báo cáo lỗi: (tt)

7/. Expected and Actual Result: Mô tả lỗi sẽ không đầy đủ nếu không có kết quả mong đợi và kết quả thực tế. Cần nêu được kết quả cụ thể sau khi tiến hành cc bước kiểm thử là gì và mong đợi kết quả như thế nào.

8/. Attachment (đính kèm bằng chứng: Video, Ảnh chụp màn hình lỗi): Chụp ảnh màn hình, quay video cho các trường hợp lỗi / thất bại với mô tả phù hợp để làm rõ hơn lỗi đang báo cáo là gì, giúp người đọc có thể kiểm tra và làm theo dễ dàng.

5. Báo cáo lỗi và tái phục hồi lỗi

Xem 1 báo cáo lỗi.

Bug #165802

Edit Log time Watch Copy

FGAS_PERSONAL DATA : Home: Profiles : Dead page display when user click on "Profile" icon

Added by Sac Duc Thi

Status:	Closed	Start date:	03/23/2018
Priority:	High	Due date:	
Assignee:	Sac Duc Thi	% Done:	0%
Category:	-	Spent time:	-
Target version:	1.0.2 (22/01/2018 - 26/01/2018)		
Story point:		Bug Severity:	High(Major/ Severe)

Description

Quote

Hiện tại, khi người dùng click Profile ở màn hình Home thì hiển thị trang chết

Mong muốn, khi người dùng click Profile ở màn hình Home thì hiển thị chính xác thông tin Profile

Step:

1. Đăng nhập vào TK:

| nguyen.thi.hong@framgia.com.edev/123456

2. Tới màn hình "Home"

3. Click vào icon "Profile"

4. Kiểm tra màn hình hiển thị

Actual result: Hiển thị trang chết khi người dùng click Profile ở màn hình Home

Expected result: Hiển thị chính xác thông tin Profile của người dùng khi click vào icon "Profile"

Information:

+ Env: STG
+ Date: 15/03/2018
+ OS: Win 8.1
+ Browser: Chrome
+ Attachment: <https://sac.tinytake.com/sf/MjQzODU5M183MzczMTE4>

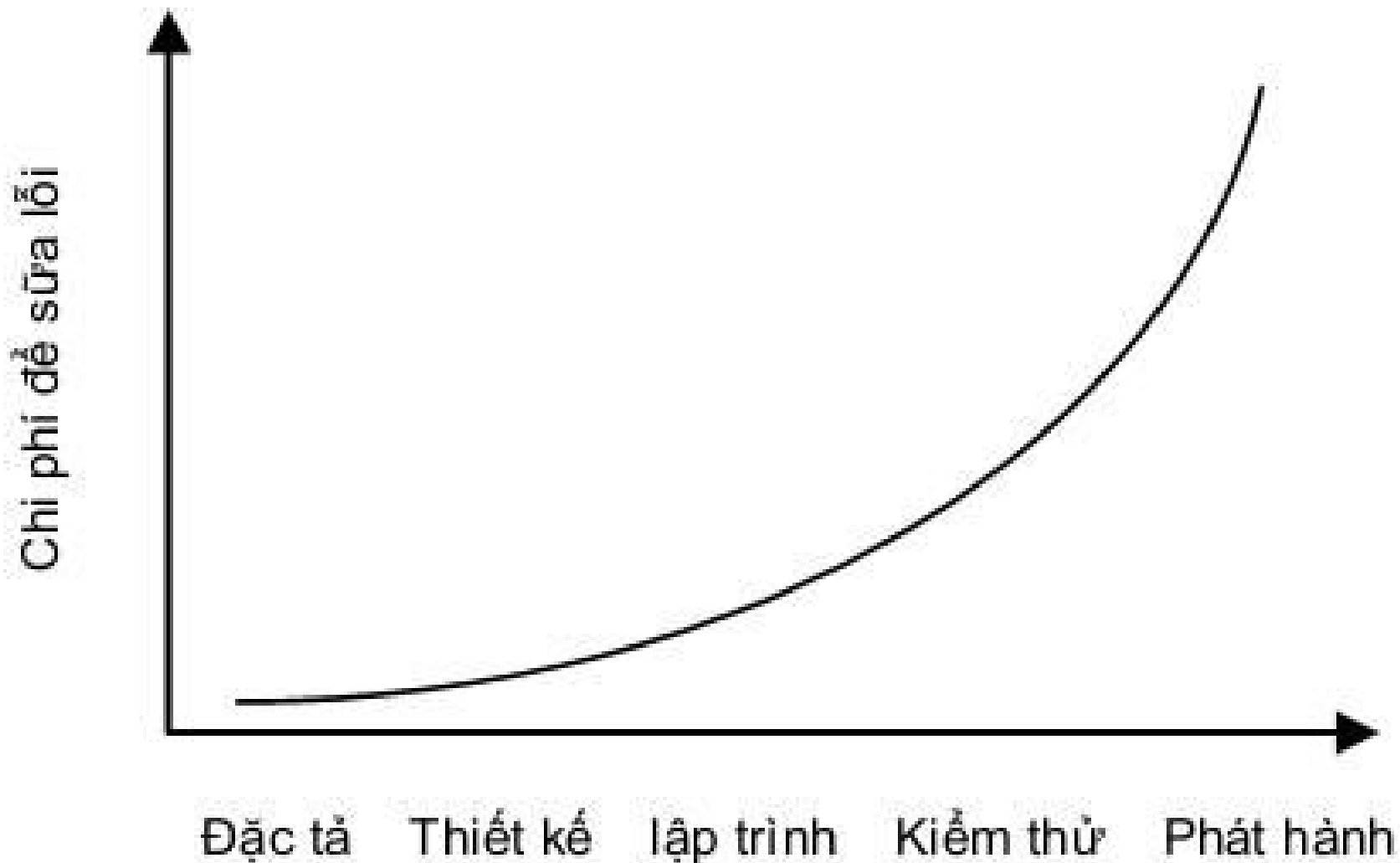
6. Kiểm toán phần mềm

Kiểm toán chính là kiểm tra và xác minh tính trung thực của những báo cáo tài chính đó, để rồi cung cấp những thông tin chính xác nhất về tình hình tài chính của tổ chức / dự án. Nói cách khác: “***Kiểm toán là quá trình thu thập và đánh giá bằng chứng liên quan đến những thông tin tài chính được kiểm tra (cung cấp bởi kế toán) nhằm xác định và báo cáo về mức độ phù hợp giữa thông tin đó với các chuẩn mực đã được thiết lập***”. Kiểm toán hướng đến rất nhiều đối tượng.

Người quản trị dự án phát triển PM thường không có nghiệp vụ về tài chính, kế toán. Do đó, cần có các kiểm toán viên để đánh giá tình hình tài chính giúp họ có những quyết định đúng đắn nhất.

6. Kiểm toán phần mềm

Người quản trị
xác định các y
khi khắc phục
Xem hình:



ừ cần
kém

