**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC ĐÔNG Á**

**KHOA CÔNG NGHỆ THÔNG TIN**

****

**BÀI BÁO CÁO**

**Môn: Kiểm thử phần mềm**

**Đề tài 5: Research và viết báo cáo, demo về kiểm thử phần mềm và selenium (Software Testing, Manual Testing, Automation Testing, White box/ Black box testing, selenium)**

**GVHD: ThS. Tạ Quốc Ý**

**SVTH: Dương Trường Thọ**

**MỤC LỤC**

Nội dung chính báo cáo Kiểm thử phầm mềm

[1. Software Testing: 3](#_Toc101533896)

[1.1. Khái niệm: 3](#_Toc101533897)

[1.2. Mục tiêu của kiểm thử: 3](#_Toc101533898)

[1.3. Tầm quan trọng của kiểm thử: 3](#_Toc101533899)

[1.4. Các nguyên tắc trong kiểm thử: 4](#_Toc101533900)

[1.5. Phân loại kiểm thử: 5](#_Toc101533901)

[2. White box/ Black box Testing: 6](#_Toc101533902)

[2.1. Black box Testing: 6](#_Toc101533903)

[2.2. White box Testing: 7](#_Toc101533904)

[3. Manual Testing: 7](#_Toc101533905)

[3.1. Khái niệm: 7](#_Toc101533906)

[3.2. Các loại Manual testing: 8](#_Toc101533907)

[3.3. Cách thực hiện Manual testing: 8](#_Toc101533908)

[4. Automation Testing: 8](#_Toc101533909)

[4.1. Khái niệm: 8](#_Toc101533910)

[4.2. Ưu nhược điểm: 8](#_Toc101533911)

[4.3. Quy trình kiểm thử tự động: 9](#_Toc101533912)

[5. Selenium: 9](#_Toc101533913)

[5.1. Khái niệm: 9](#_Toc101533914)

[5.2. Một số khái niệm liên quan: 10](#_Toc101533915)

## Software Testing:

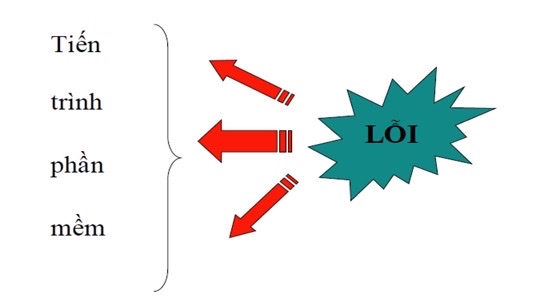
### Khái niệm:

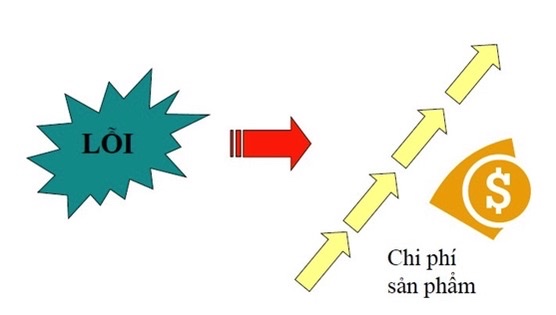
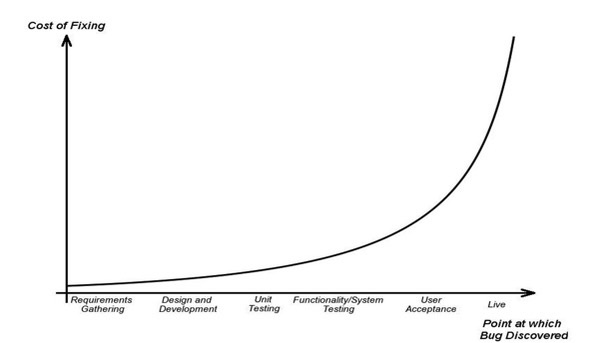
* Theo Glenford Myers:
  + Kiểm thử là quá trình vận hành chương trình để tìm ra lỗi
* Theo IEEE:
  + (1) Là quá trình vận hành hệ thống hoặc thành phần dưới những điều kiện xác định, quan sát hoặc ghi nhận kết quả và đưa ra đánh giá về hệ thống hoặc thành phần đó.
  + (2) Là quá trình phân tích phần mềm để tìm ra sự khác biệt giữa điều kiện thực tế và điều kiện yêu cầu và dựa vào điểm khác biệt đó để đánh giá tính năng phần mềm

### Mục tiêu của kiểm thử:

* Tìm ra được càng nhiều lỗi càng tốt trong điều kiện về thời gian đã định và nguồn lực sẵn có.
* Chứng minh rằng sản phẩm phần mềm phù hợp với các đặc tả của nó.
* Xác thực chất lượng kiểm thử phần mềm đã dùng chi phí và nỗ lực tối thiểu.
* Thiết kế tài liệu kiểm thử một cách có hệ thống và thực hiện nó sao cho có hiệu quả, tiết kiệm được thời gian công sức.

### Tầm quan trọng của kiểm thử:





* Xác định lỗi (Bugs/ Defects).
* Thông tin cho các bên liên quan và uy tín của công ty.
* Cải thiện chất lượng sản phẩm.
* Đưa ra được ứng dụng có kỹ thuật tốt nhất.
* Tạo ra ứng dụng tốt nhất, an toàn để đi trước bất kì nhà phát triển cạnh tranh nào.
* Tránh xa mọi mối nguy hiểm cho ứng dụng.
* Chứng minh tính khả dụng và khả năng vận hành.

### Các nguyên tắc trong kiểm thử:

Trong kiểm thử có 7 nguyên tắc cơ bản:

* Kiểm thử chỉ ra sự hiện diện của lỗi trong phần mềm.
* Kiểm thử tất cả các trường hợp là điểu không thể.
* Nên thực hiện kiểm thử càng sớm càng tốt.
* Sự phân cụm của các lỗi.
* Nghịch lý thuốc trừ sâu.
* Kiểm thử theo các ngữ cảnh độc lập.
* Sự sai lầm về việc không có lỗi.

### Phân loại kiểm thử:

Phân loại kiểm thử dựa trên các yếu tố:

* Mục đích kiểm thử
* Chiến lược kiểm thử
* Phương pháp kiểm thử
* Kỹ thuật kiểm thử

#### Dựa vào mục đích kiểm thử:

* Kiểm thử đơn vị, module
* Kiểm thử cấu hình
* Kiểm thử sơ lược (smoke testing)
* Kiểm thử chức năng
* Kiểm thử tích hợp
* Kiểm thử hồi quy
* Kiểm thử hệ thống
* Kiểm thử tải dữ liệu (load testing)
* Kiểm thử tải trọng (stress testing)
* Kiểm thử hiệu suất (performance testing)
* Kiểm thử chấp nhận (UAT)
* Kiểm thử bảo mật (security testing)

#### Dựa vào chiến lược kiểm thử:

* Kiểm thử thủ công:
* Thực hiện kiểm thử mọi thứ bằng tay, từ viết test case đến thực hiện test.
* Kiểm thử tự động:
* Thực hiện một cách tự động các bước trong kịch bản kiểm thử bằng cách dùng một công cụ trợ giúp.
* Kiểm thử tự động nhằm tiết kiệm thời gian kiểm thử.

#### Dựa vào phương pháp tiến hành kiểm thử:

* Kiểm thử tĩnh:
* Một hình thức của kiểm thử mà phần mềm không được sử dụng thực sự.
* Thường không kiểm thử chi tiết mà chủ yếu kiểm tra tính đúng đắn của code, thuật toán hoặc tài liệu
* Các hoạt động: Đi xuyên suốt (walk through), thanh tra (inspection)
* Kiểm thử động:
* Một hình thức kiểm thử phần mềm chạy mã lập trình thực tế trong các tình huống, diễn ra khi bản thân chương trình đó đang được sử dụng
* Kiểm thử động có thể bắt đầu trước khi chương trình đã hoàn tất.

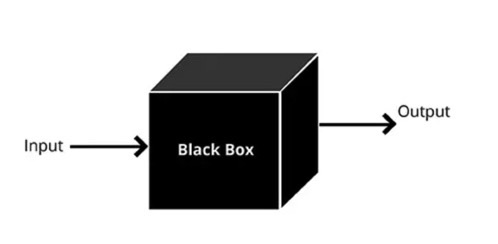
#### Dựa vào kỹ thuật kiểm thử:

* **Kiểm thử hộp trắng**
* Kiểm thử theo góc nhìn thực hiện
* Cần có kiến thức về chi tiết thiết kế và thực hiện bên trong
* Kiểm thử dựa vào phủ các lệnh, các nhánh, phủ các điều kiện con
* **Kiểm thử hộp đen**
* Kiểm thử theo góc nhìn sử dụng
* Kiểm thử dựa trên các yêu cầu và đặc tả sử dụng thành phần phần mềm
* Không đòi hỏi kiến thức về chi tiết thiết kế và thực hiện ở bên trong chương trình

## White box/ Black box Testing:

### Black box Testing:

#### Khái niệm:

Kiểm tra hộp đen (Black box testing) là một phương pháp kiểm thử phần mềm mà việc kiểm tra các chức năng của một ứng dụng không cần quan tâm vào cấu trúc nội bộ hoặc hoạt động của nó.

#### Đối tượng được kiểm thử:

Là thành phần phần mền (TPPM) có thể là 1 hàm chức năng, 1 modul chức năng, 1 phân hệ chức năng...

#### Phương pháp thử nghiệm:

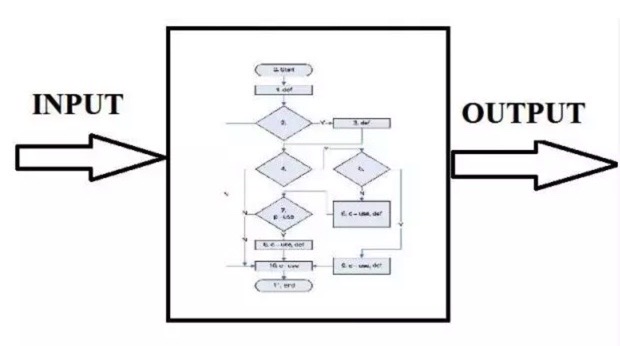
Dựa vào chức năng Kiểm thử hộp đen (Black box test) có thể được áp dụng hầu như đến mọi cấp độ của kiểm thử phần mềm:

* Kiểm thử đơn vị (Unit test)
* Kiểm thử tích hợp (Intergration test)
* Kiểm thử hệ thống (System test)
* Kiểm thử chấp nhận (Acceptance test).

Tuy nhiên, Black box test được sử dụng thích hợp nhất trong kiểm thử hệ thống (System test) và Kiểm thử chấp nhận (Acceptance test).

### White box Testing:

#### Khái niệm:

Kiểm thử hộp trắng (While box test) là phương pháp thử nghiệm phần mềm, trong đó các thiết kế, cấu trúc giải thuật bên trong, và việc thực hiện các công việc đều được biết đến.

#### Đối tượng kiểm thử:

Là 1 thành phần của phần mềm (1 chức năng, 1 module chức năng, 1 phân hệ chức năng....).

#### Phương pháp kiểm thử:

Dựa vào thuật giải Kiểm thử hộp trắng dựa vào thuật giải cụ thể, vào cấu trúc dữ liệu bên trong của đơn vị phần mềm cần kiểm thử để xác định đơn vị phần mềm đó có thực hiện đúng không.

* Với những TPPM quá lớn sẽ tốn rất nhiều thời gian và công sức để kiểm thử nếu như dùng kiểm thử tích hợp (Integration test) hay kiểm thử chức năng (Functional test)).
* Kỹ thuật white box test thích hợp dùng để kiểm thử đơn vị (Unit test).

## Manual Testing:

### Khái niệm:

Manual testing là việc thử nghiệm một phầm mềm hoàn toàn được làm bằng tay bởi người tester. Nó được thực hiện nhằm phát hiện lỗi trong phầm mềm đang được phát triển. Trong manual testing, tester sẽ thực hiện các trường hợp kiểm thử và tạo báo cáo kiểm thử hoàn toàn thủ công mà không có bất kỳ sự trợ giúp của công cụ tự động nào.

### Các loại Manual testing:

Dưới đây là sơ đồ mô tả các loại Manual Testing . Trong thực tế, bất kỳ loại kiểm thử phần mềm nào cũng có thể được thực hiện bằng tay cũng như sử dụng một công cụ tự động hóa.

* Black Box Testing
* White Box Testing
* Unit Testing
* System Testing
* Integration Testing
* Acceptance Testing

### Cách thực hiện Manual testing:

* Đọc và hiểu tài liệu / hướng dẫn dự án phần mềm. Ngoài ra, nghiên cứu Ứng dụng đang thử nghiệm (AUT) nếu có.
* Phác thảo testcases bao gồm tất cả các yêu cầu được đề cập trong tài liệu.
* Xem xét và vạch ra các trường hợp thử nghiệm với Trưởng nhóm, Khách hàng (nếu có)
* Thực hiện các trường hợp thử nghiệm trên AUT
* Báo cáo lỗi.
* Khi các lỗi đã được sửa, một lần nữa thực hiện các trường hợp thử nghiệm thất bại để xác minh chúng không còn sảy ra nữa.

## Automation Testing:

### Khái niệm:

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)

### Ưu nhược điểm:

#### Ưu điểm:

* Độ tin cậy cao
* Khả năng lặp
* Khả năng tái sử dụng
* Tốc độ cao
* Chi phí thấp

#### Nhược điểm:

* Khó mở rộng, khó bảo trì
* Khả năng bao phủ thấp
* Vấn đề về công cụ và nhân lực

### Quy trình kiểm thử tự động:

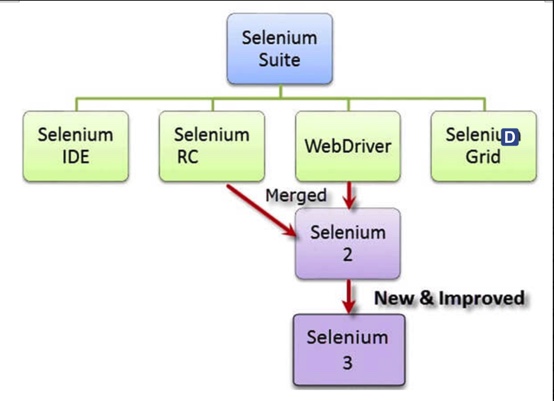
## Selenium:

### Khái niệm:

Selenium là bộ kiểm thử tự động miễn phí (mã nguồn mở) dành cho các ứng dụng web trên các trình duyệt và nền tảng khác nhau. Nó khá là giống với HP Quick Test Pro (QTP bây giờ là UFT) chỉ khác là Selenium thì tập trung vào việc tự động hoá các ứng dụng dựa trên nền tảng web. Kiểm thử được thực hiện bằng cách sử dụng công cụ Selenium thường được gọi là Kiểm thử Selenium. Selenium không chỉ là 1 công cụ độc lập mà là 1 bộ công cụ của phần mềm, mỗi bộ đều đáp ứng được nhu cầu kiểm thử khác nhau của 1 tổ chức.

Gồm 4 thành phần:

* Selenium Integrated Development Environment (IDE)
* Selenium Remote Control (RC)
* WebDriver
* Selenium Grid



### Một số khái niệm liên quan:

* Selenium IDE, một tiện ích bổ sung của Firefox mà bạn chỉ có thể sử dụng trong việc tạo các trường hợp thử nghiệm tương đối đơn giản và bộ phần mềm thử nghiệm.
* Selenium Remote Control, hay còn gọi là Selenium 1, là công cụ Selenium đầu tiên cho phép người dùng sử dụng các ngôn ngữ lập trình để tạo các kiểm thử phức tạp.
* WebDriver, bước đột phá mới hơn cho phép các tập lệnh kiểm tra của bạn liên lạc trực tiếp với trình duyệt, do đó kiểm soát nó từ cấp hệ điều hành.
* Selenium Grid cũng là một công cụ được sử dụng với Selenium RC để thực hiện các kiểm tra song song trên các trình duyệt và hệ điều hành khác nhau.
* Selenium RC và WebDriver được sáp nhập để tạo thành Selenium 2.
* Selenium thuận lợi hơn QTP về mặt chi phí và tính linh hoạt. Nó cũng cho phép bạn chạy kiểm thử song song, không giống như trong QTP, nơi bạn chỉ được phép chạy thử nghiệm theo tuần tự.