****

**BÁO CÁO KẾT THÚC HỌC PHẦN**

**Đề tài : XÂY DỰNG SELENIUM THEO MÔ HÌNH**

**PAGE OBJECT MODEL**

**Giảng viên hướng dẫn:** Tạ Quốc Ý

**Sinh viên tham gia:** Mạnh Thế Tài

**Lớp:** ST19A1A

*Đà Nẵng, ngày 10 tháng 05 năm 2022.*

**MỤC LỤC**

[**1. Kiểm thử phần mềm là gì? 3**](#_Toc103812342)

[**2. Phân loại kiểm thử phần mềm 3**](#_Toc103812343)

[**a) Kiểm thử full chức năng 3**](#_Toc103812344)

[**b) Kiểm thử phi chức năng 4**](#_Toc103812345)

[**c) Kiểm thử cấu trúc 5**](#_Toc103812346)

[**3. Sơ lược về Page Object – POM 6**](#_Toc103812347)

[**4. Ưu điểm của mô hình Page Object (POM - đối tượng trang) 6**](#_Toc103812348)

# Kiểm thử phần mềm là gì?

Kiểm thử phần mềm là một cuộc kiểm tra được tiến hành để cung cấp cho các bên liên quan thông tin về chất lượng của sản phẩm hoặc dịch vụ được kiểm thử.

# Phân loại kiểm thử phần mềm

Kiểm thử phần mềm không phải là một việc đơn lẻ. Nó có nhiều hình thức khác nhau và được phân loại theo một số tiêu chí. Về cơ bản, kiểm thử phần mềm được chia làm 4 loại:

## Kiểm thử full chức năng

Kiểm thử chức năng là xác minh hệ thống hoạt động theo đúng theo các yêu cầu nghiệp vụ. Hình thức kiểm thử này có thể được thực hiện từ hai khía cạnh: dựa trên yêu cầu (requirements-based) và dựa trên quy trình nghiệp vụ (business – process – based).

Trong kiểm thử dựa trên yêu cầu, các yêu cầu được ưu tiên tùy thuộc vào tiêu chí rủi ro. Điều này sẽ đảm bảo những phần quan trọng nhất sẽ được test đầy đủ. Mặt khác, kiểm thử dựa trên quy trình nghiệp vụ sẽ sử dụng những kiến thức tương ứng. Quy trình nghiệp vụ mô tả các việc liên quan đến nghiệp vụ hằng ngày của hệ thống.

Kiểm thử chức năng bao gồm 5 bước:

* Xác định các chức năng mà phần mềm sẽ thực hiện.
* Tạo các dữ liệu đầu vào dựa trên các tài liệu đặc tả kỹ thuật của các chức năng.
* Xác định các kết quả đầu ra dựa trên các tài liệu đặc tả kỹ thuật của các chức năng.
* Thực hiện các trường hợp kiêm thử.
* So sánh kết quả thực tế và kết quả mong muốn.

Trong đó, kiểm thử chức năng còn được chia nhỏ ra thành các loại:

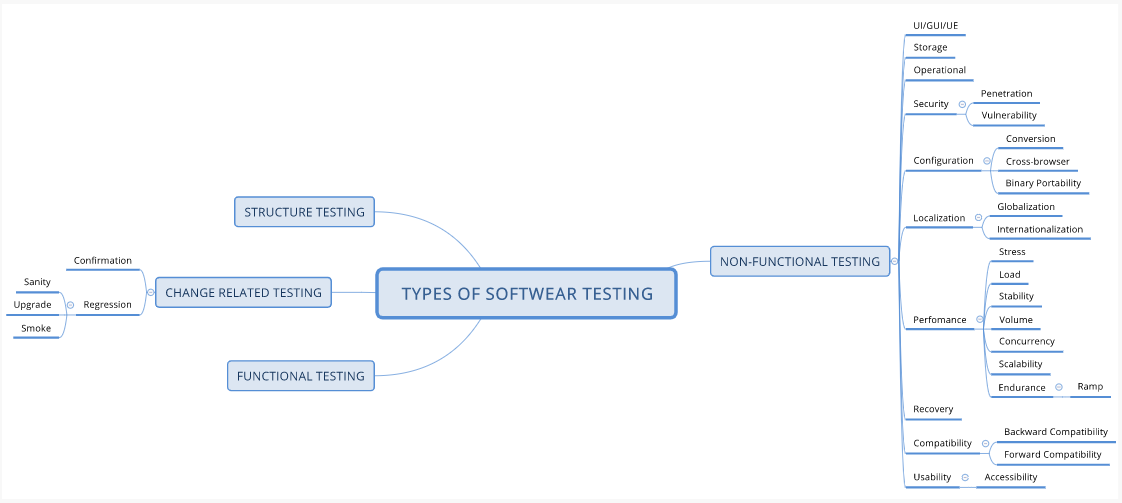
* Kiểm thử đơn vị (Unit testing)
* Smoke Testing
* Sanity Testing
* Kiểm thử giao diện (Interface testing)
* Kiểm thử tích hợp (Integration testing)
* Kiểm thử hệ thống (System testing)
* Kiểm thử hồi quy (Regression testing)
* Kiểm thử chấp nhận (Acceptance testing)

Ưu điểm của kiểm thử chức năng:

* Hình thức kiểm thử này mô phỏng việc sử dụng hệ thống thực tế
* Được thực hiện trong các điều kiện gần với điều kiện của khách hàng
* Không có giả định nào về cấu trúc hệ thống được đưa ra trong khi kiểm thử chức năng
* Rất dễ dàng để thực hiện test thủ công

Ngược lại, kiểm thử chức năng có những giới hạn sau:

* Khả năng cao xảy ra tình trạng test dư thừa
* Các lỗi logic trong phần mềm có thể bị bỏ sót trong khi kiểm thử chức năng



## Kiểm thử phi chức năng

Kiểm thử phi chức năng là kiểm tra các đặc tính chất lượng của hệ thống. Ví dụ, kiểm tra xem bao nhiêu người có thể đăng nhập đồng thời vào một phần mềm. Kiểm tra phi chức năng cũng quan trọng không kém như kiểm tra chức năng và ảnh hưởng đến sự hài lòng của khách hàng.

Tương tự, kiểm thử phi chức năng cũng được chia thành các loại:

* Kiểm thử độ ổn định (Stability testing): đánh giá phần mềm có thể liên tục hoạt động tốt trong hoặc ngay trên khoảng thời gian có thể chấp nhận hay không
* Kiểm thử khả năng chịu tải (Load testing): đánh giá hoạt động của hệ thống khi khối lượng công việc ngày càng tăng
* Kiểm thử áp lực (Stress testing): ước tính hoạt động của hệ thống ở trong hoặc vượt quá giới hạn khối lượng công việc dự kiến
* Kiểm thử tính khả dụng (Usability testing): sản phẩm được test về tính thân thiện với người dùng
* Kiểm thử bảo trì (Maintainability testing): kiểm tra mức độ đánh giá, thay đổi và test sản phẩm
* Kiểm thử độ tin cậy (Reliability testing): sử dụng công cụ để tìm, ngăn chặn và loại bỏ lỗi trước khi hệ thống được triển khai
* Kiểm thử tính tương thích (Portability testing): xác định mức độ dễ dàng hoặc khó khăn mà phần mềm có thể di chuyển từ môi trường này sang môi trường khác

## Kiểm thử cấu trúc

Kiểm thử cấu trúc thường được gọi là “hộp trắng” hoặc “hộp thủy tinh” bởi vì phương pháp này quan tâm đến việc tìm kiếm những gì đang xảy ra bên trong, kiểm tra dựa trên phân tích cấu trúc bên trong của thành phần hoặc hệ thống. Nó thường được sử dụng như một cách đo lường của kiểm thử, thông qua độ bao phủ của một tập hợp các yếu tố cấu trúc. Kiểm thử cấu trúc chủ yếu được áp dụng ở kiểm thử thành phần, kiểm thử tích hợp.

Các mục tiêu chính của kiểm thử cấu trúc bao gồm:

* Nhận ra những điểm bất cập
* Test chức năng bổ sung
* Xác định những phần bị thiếu trong bộ kiểm thử

Ưu điểm của kiểm thử cấu trúc:

* Loại bỏ code chết
* Có khả năng tìm ra lỗi ở giai đoạn đầu
* Đảm bảo kiểm tra phần mềm kỹ lưỡng hơn
* Tiết kiệm thời gian

Bên cạnh đó, nhược điểm của kiểm thử cấu trúc:

* Kiểm tra kết cấu khá tốn kém
* Yêu cầu kiến thức về code
* Đòi hỏi kiến thức vững chắc về công cụ được sử dụng để test

# Sơ lược về Page Object – POM

* + OOP (Object-Oriented Programming) nhìn tất cả mọi vật là Object – đối tượng. Đối tượng được định nghĩa trong các class có các thuộc tính và các hành động khác nhau.
  + POM (Page Object Model) là 1 design pattern (mẫu thiết kế) giúp mô hình hóa các pages, hoặc các phần (component: header, footer, menu…) trong page của trang web thành mỗi đối tượng riêng biệt. Mỗi component / page sẽ gói gọn tất cả các hành động và các thuộc tính của component / page đó.

# Ưu điểm của mô hình Page Object (POM - đối tượng trang)

* + **Giúp bảo trì dễ dàng**: POM hữu ích khi có sự thay đổi trong phần tử giao diện người dùng hoặc có sự thay đổi trong một hành động. Ví dụ: một menu thả xuống được thay đổi thành một nút radio. Trong trường hợp này, POM giúp xác định trang hoặc màn hình cần sửa đổi. Vì mọi màn hình sẽ có các tệp java khác nhau, nên việc xác định này là cần thiết để thực hiện các thay đổi bắt buộc đối với các tệp đó. Điều này làm cho các trường hợp kiểm thử dễ bảo trì và giảm lỗi.
  + **Giúp sử dụng lại mã (dùng lại code đã viết)**: Như đã thảo luận, tất cả các màn hình đều độc lập. Bằng cách sử dụng POM, người ta có thể sử dụng mã thử nghiệm cho một màn hình và sử dụng lại nó trong một trường hợp thử nghiệm khác. Không cần phải viết lại mã, do đó tiết kiệm thời gian và công sức.
  + **Dễ đọc code**: Khi tất cả các màn hình có các tệp java độc lập, người ta có thể dễ dàng xác định các hành động sẽ được thực hiện trên một màn hình cụ thể bằng cách điều hướng qua tệp java đó thôi. Nếu một thay đổi ảnh hưởng đến một phần mã code nhất định thì nó có thể được thực hiện một cách hiệu quả mà không ảnh hưởng đến các tệp (class/package) khác.
  + **Tạo kho lưu trữ**: Có thể một kho lưu trữ duy nhất cho các xử lý chung hoặc hoạt động chung cho các trang thay vì có các xử lý này nằm rải rác trong các test case riêng lẻ. VD: getTitlePage(), verifyHeaderPage(),...

