**ĐẠI HỌC ĐÔNG Á**

**KHOA CÔNG NGHỆ THÔNG TIN**



**KIỂM THỬ PHẦN MỀM**

**Giảng viên hướng dẫn:**

**TẠ QUỐC Ý**

**Sinh viên thực hiện:**

**LÊ VĂN QUỐC**

**Lớp : ST19A1A**

**Khoá** **: 2019 – 2023**

**Hệ** **: Chính quy**

**Đà Nẵng, tháng 05 /2022**

MỤC LỤC

[**PHẦN 1. KIỂM THỬ PHÂN MỀM** 3](#_Toc102657197)

[**I.** **Software testing là gì?** 3](#_Toc102657198)

[1. Khái niệm 3](#_Toc102657199)

[**2.** Phân loại testing 3](#_Toc102657200)

[**3.** Các công việc của một tester là gì? 4](#_Toc102657201)

[**II.** **Black box testing và While box testing** 4](#_Toc102657202)

[**1.** Black box testing. 4](#_Toc102657203)

[**2.** While box testing. 5](#_Toc102657204)

[**3.** Ưu điểm, nhược điểm của Black box testing và While box testing. 7](#_Toc102657205)

[**4.** Sự khác nhau của Black box testing và While box testing. 8](#_Toc102657206)

[**III.** **Định nghĩa Selenium là gì?** 9](#_Toc102657207)

[1. Khái niệm. 9](#_Toc102657208)

[2. Phân loại. 9](#_Toc102657209)

[3. Những điều mà Selenium hỗ trợ là gì? 10](#_Toc102657210)

# **PHẦN 1. KIỂM THỬ PHÂN MỀM**

1. **Software testing là gì?**
2. Khái niệm

Software testing là cụm từ dùng để chỉ quá trình kiểm thử phần mềm. Quá trình này được thực hiện nhằm mục đích tìm ra các lỗi  trong phần mềm  trước khi tung ra thị trường. Nếu không có quá trình này, phần mềm có thể hoạt động không như mong đợi hoặc gặp những trục trặc không đáng có.

Software testing còn có thể được xem là quá trình validating (xác thực) và verifying (xác minh) rằng chương trình phần mềm, ứng dụng hay sản phẩm đã đáp ứng các yêu cầu về kỹ thuật, kinh doanh và hoạt động đúng như mong đợi.



1. **Phân loại testing**

Dưới đây là hai loại testing được tester sử dụng trong quá trình kiểm thử phần mềm của mình:

* 1. ***Kiểm thử thủ công (manual testing)***

Đây là quá trình kiểm thử được tester thực hiện thủ công và không thông qua bất kỳ công cụ nào. Từ công đoạn viết test case, test, nhập điều kiện đầu vào cho đến các thao tác khác như click nút và quan sát kết quả thực tế đều được thực hiện bằng tay. Đây là phương pháp testing được phần lớn các công ty phần mềm sử dụng.

* 1. ***Kiểm thử tự động (automation testing)***

Đây là quá trình kiểm thử sử dụng các công cụ, phần mềm hỗ trợ. Điều này giúp các tester không phải lặp đi lặp lại các bước kiểm thử nhàm chán. Các công cụ kiểm thử có thể lấy các dữ liệu từ file bên ngoài và nhập vào ứng dụng, so sánh kết quả với kết quả thực tế sau đó xuất báo cáo kết quả kiểm thử.



1. **Các công việc của một tester là gì?**

Cơ bản, công việc chính của một tester là kiểm tra nhằm phát hiện các lỗi đang tồn tại trong phần mềm, sản phẩm. Công việc của mỗi tester sẽ được quy định cụ thể tùy thuộc vào từng doanh nghiệp và từng dự án khác nhau.

Nếu theo hướng manual testing, bạn sẽ không cần quá nhiều kiến thức về lập trình trong lúc làm. Bạn chỉ cần nắm vững các định nghĩa, các kỹ thuật kiểm thử thủ công và có tư duy tìm lỗi tốt là có thể đảm bảo hiệu quả công việc.

Còn khi chọn automation testing, nhiệm vụ của bạn sẽ có phần phức tạp hơn. Automation tester thường  được ví là developer trong testing. Công việc chính của bạn là viết code để thực hiện việc kiểm thử một cách tự động.

Người làm automation testing sẽ cần phải nắm rõ về các automation tools & frameworks, có thể làm việc với nhiều ngôn ngữ lập trình khác nhau như Java, AutoIT, C++, Python,… tùy theo yêu cầu của từng dự án.

1. **Black box testing và While box testing**
2. Black box testing.

* **Định nghĩa** 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:

1. Kiểm thử đơn vị (Unit test)
2. Kiểm thử tích hợp (Intergration test)
3. Kiểm thử hệ thống (System test)
4. 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)

* 1. ***Đặc điểm***
* Là chiến lược kiểm thử TPPM dựa vào thông tin duy nhất là các đặc tả về yêu cầu chức năng của TPPM tương ứng.
* Người kiểm thử không cần thiết phải có kiến thức về việc mã hoá, cấu trúc bên trong của TPPM, cũng như không yêu cầu phải biết lâp trình phần mềm.
* Việc kiểm thử được tiến hành dựa vào việc kiểm thử TPPM làm được gì, có phù hợp với yêu cầu của người dùng hay không. Các tester nhập số liệu vào phần mềm và chỉ cần xem kết quả của phần mềm và các mục tiêu kiểm tra.
* Mức test này thường yêu cầu các tester phải viết test case đầy đủ trước khi test; khi test, đơn giản chỉ cần thực hiện theo các bước mô tả trong test case thao tác và nhập data vào, sau đó xem kết quả trả về hoặc hành vi của phần mềm, rồi so sánh với kết quả mong đọi được viết trong testcase
  1. ***Tạo test case và Thực hiện test case***
* Khi viết test case: Dựa vào yêu cầu và giao diện bên ngoài của chương trình (Không can thiệp vào bên trong code của chương trình)
* Khi thực hiện test: Thực hiện trên giao diện của chương trình (yêu cầu chương trình phải chạy được mới test được, không can thiệp vào code)

1. While box testing.

* **Định nghĩa** 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 thử nghiệm:** Dựa vào thuật giải Kiểm thử hộp trắng dựa vào thuật giải cụ thể, vào cấu trúc dữ liệu bên trong của ₫ơn vị phần mềm cần kiểm thử ₫ể xác ₫ịnh ₫ơn vị phần mềm ₫ó có thực hiện ₫úng 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)

* 1. ***Đặc điểm***
* Là chiến lược kiểm thử TPPM dựa vào giải thuật, cấu trúc bên trong chức năng của TPPM tương ứng.
* Người kiểm thử phải có kiến thức nhất định về việc mã hoá, cấu trúc bên trong của chức năng, biết lâp trình phần mềm.
* Việc kiểm thử được tiến hành dựa vào việc kiểm xem giải thuật, mã lệnh đã làm có đúng không.
* Mức test này thường yêu cầu các tester phải viết test case đầy đủ các nhánh trong code; khi test, sẽ set điều kiện và data để chạy vào đủ tất cả các nhánh trong giải thuật, đảm bảo thực hiện đầy đủ.
  1. ***Tạo testcase và thực hiện test***
* Khi viết test case: Dựa vào yêu cầu và nội dung Source Code (can thiệp vào bên trong Code của chương trình)
* Khi thực hiện test: Thực thi test trong code (không cần thực thi chương trình, vì thực hiện test white box sẽ sử dụng framework nào đó hỗ trợ (Ví dụ như test kiểu debug)
* Trong kiểm tra này, đòi hỏi người tester phải có kiến thức và kỹ năng nhất định về ngôn ngữ lập trình được dùng, hiểu thuật giải trong thành phần phần mềm, để có thể hiểu được chi tiết về đoạn code cần kiểm thử

1. Ưu điểm, nhược điểm của Black box testing và While box testing.

|  |  |  |
| --- | --- | --- |
| **Nội dung** | **Black box test** | **White box test** |
| 1. Ưu điểm | - Thích hợp trong việc kiểm tra từng phân đoạn lớn các mã lệnh, chức năng lớn  - Người thử nghiệm không cần hiểu biết về mã lệnh được viết trong chương trình - Tách biệt giữa quan điểm của người sử dụng và người phát triển phần mềm | - Thích hợp trong việc tìm kiếm lỗi và các vấn đề trong mã lệnh  - Biết được yêu cầu bên trong của phần mềm, kiểm tra sẽ sát hơn  - Cho phép tìm kiếm các lỗi ẩn bên trong  - Các lập trình viên có thể tự kiểm tra  - Giúp tối ưu việc mã hoá  - Do yêu cầu kiến thức cấu trúc bên trong của phần mềm, nên việc kiểm soát lỗi tối đa nhất |
| 2. Nhược điểm | - Độ bao phủ hạn chế vì chỉ có một phần nhỏ trong số các kịch bản thử nghiệm được thực hiện  - Kiểm tra không hiệu quả do người thử nghiệm không hiểu biết gì về cấu trúc bên trong phần mềm.  - Tester có hạn chế về hiểu biết về ứng dụng | - Không thể tìm thấy tính năng chưa thực hiện hoặc bỏ sót  - Đòi hỏi hiểu sâu về cấu trúc bên trong của phần mềm được thử nghiệm  - Yêu cầu truy xuất mã lệnh bên trong chương trình |

1. Sự khác nhau của Black box testing và While box testing.

|  |  |  |
| --- | --- | --- |
| **Tiêu chuẩn** | **Black box testing** | **While box testing** |
| 1. Định nghĩa | - Kiểm tra hộp đen là phương pháp thử nghiệm phần mềm được sử dụng để kiểm tra các phần mềm mà không quan tâm đến cấu trúc bên trong của chương trình. | - Kiểm tra hộp trắng là phương pháp kiểm thử phần mềm, sử dụng để kiểm tra phần mềm mà yêu cầu phải biết cấu trúc bên trong của chương trình. |
| 2. Trách nhiệm | - Thử nghiệm được thực hiện bên ngoài, không liên quan đến nhà phát triển phần mềm. | - Thông thường, các thử nghiệm được thực hiện bởi nhà phát triển phần mềm. |
| 3. Cấp độ test sử dụng | - Thử nghiệm áp dụng ở cấp độ cao như: kiểm tra hệ thống (System test), kiểm tra chấp nhận (Acceptance test) | - Thử nghiệm được áp dụng ở mức độ thấp hơn như thử nghiệm đơn vị (Unit Test), thử nghiệm hội nhập (Integration test) |
| 4. Biết lập trình | - Không yêu cầu hiểu biết về Lập trình | - Yêu cầu hiểu biết nhất định về lập trình. |
| 5. Biết việc thực hiện chương trình | - Không yêu cầu hiểu về cấu trúc bên trong chức năng, và không cẩn hiểu làm thế nào để có được chức năng đó | - Yêu cầu hiểu cấu trúc bên trong chức năng được thực hiện như nào. |
| 6. Cơ sở tạo Test Cases | - Kiểm tra hộp đen được bắt đầu dựa trên tài liệu yêu cầu kỹ thuật | - Kiểm tra hộp trắng được bắt đầu dựa trên các tài liệu thiết kế chi tiết. |

1. **Định nghĩa Selenium là gì?**
2. Khái niệm.

Selenium là một bộ công cụ chuyên dụng trong kiểm thử tự động open source dành cho các ứng dụng web, cũng như hỗ trợ hoạt động trên các trình duyệt có nền tảng khác nhau như Mac, Linux, Windows,.. Với Selenium thì bạn hoàn toàn có thể viết các test script bằng nhiều ngôn ngữ lập trình khác nhau như: Java, PHP, C#, Ruby hoặc Python.

Selenium được sử dụng để có thể automate cho các thao tác với trình duyệt hoặc dễ hiểu hơn là nó hỗ trợ giả lập lại các tương tác nằm trên trình duyệt tương tự như một người dùng thực thụ. Chính vì thế, bạn có thể lập trình để có thể bật tự động các trình duyệt, để open một link, input cho dữ liệu, upload, download dữ liệu từ web page hoặc thậm chí get info page.

Ngoài ra, bạn còn có thể sử dụng để tùy biến tận dụng tối đa sức mạnh của nó. Ngoài các mục đích sử dụng trong kiểm thử thì bạn hoàn toàn có thể xây dựng một project để automate các công việc mang lại cảm giác nhàm chán vì phải lặp đi lặp lại một thao tác.

1. Phân loại.

Selenium là một trong những khái niệm chung để miêu tả một phần mềm chuyên dụng trong automation. Mà ở đó, mỗi loại trong nó sẽ đáp ứng được các yêu cầu testing khác nhau. Còn về cơ bản thì Selenium bao gồm 4 thành phần chính là:

* Selenium IDE (IDE là từ viết tắt của Integrated Developer Environment): là một plug-in nằm trên trình duyệt Fire-fox, ta có thể sử dụng để record và play lại các thao tác đó dựa theo một quy trình hay một test case nào đó.
* Selenium RC: Selenium Remote Control, Selenium server sẽ khởi chạy và tương tác với các trình duyệt web.
* WebDriver: Selenium WebDriver có nhiệm vụ gửi lệnh khởi chạy rồi thực hiện tương tác trực tiếp với các trình duyệt mà không cần thông qua bất cứ server như Selenium RC.
* Selenium Grid: Selenium Hub được sử dụng để khởi chạy nhiều các test thông qua các máy cũng như các trình duyệt khác nhau tại cùng một thời điểm nhất định.

1. Những điều mà Selenium hỗ trợ là gì?

**Những vấn đề mà Selenium có thể giải quyết được.**

* Thực hiện kiểm thử hồi quy thường xuyên hơn.
* Cần phản hồi nhanh chóng tới những người phát triển.
* Hầu như nó không giới hạn về việc lặp lại của những việc thực thi kiểm thử khác. Selenium hỗ trợ các phương pháp luận về Agile và phát triển cực đoan.
* Selenium cung cấp tài liệu kỷ luật cho các ca kiểm thử.
* Tùy chỉnh cho các nhược điểm của báo cáo.
* Tìm kiếm các nhược điểm đã bị bỏ qua do quá trình kiểm thử thủ công bỏ qua.

Các tính năng nổi bật của Selenium là gì?

**Dưới đây là những tính năng nổi bật mà Selenium giới thiệu.**

* Bạn có thể kết hợp với nhiều ngôn ngữ lập trình như: Java,.Net, Python, Ruby một cách dễ dàng.
* Selenium sẽ giả cho các thao tác người dùng ở trên web page và các web element.
* Script được base dựa theo HTML nên rất dễ học và việc ứng dụng cũng sẽ rất nhanh.
* Bạn có thể tạo ra một bộ test suite có chứa nhiều test case khác.
* Bạn có thể run các test suite thông qua Selenium IDE hoặc Selenium command line
* Selenium API sẽ hỗ trợ rất nhiều cho các loại trình duyệt, do vậy mà sẽ rất dễ dàng khi test ứng dụng web với nhiều loại trình duyệt khác nhau.