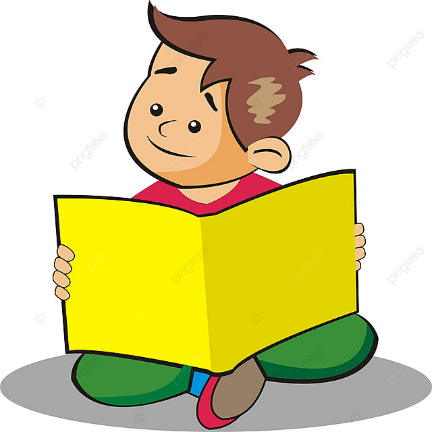
****

**Báo cáo Cuối kì**

**Môn: Kiểm thử phần mềm**

****

**Đề tài 05: Research và viết báo cáo**

**GVHD: TẠ QUỐC Ý**

**Sinh Viên:** Lê Chí Phong

**Id:** 47047

**Lớp:** ST19A1A

# Software Testing

## What is Software Testing ?

Software Testing là một phương pháp để kiểm tra xem sản phẩm phần mềm thực tế có phù hợp với các yêu cầu mong đợi hay không và để đảm bảo rằng sản phẩm phần mềm đó không có khiếm khuyết. Nó liên quan đến việc thực thi các thành phần phần mềm / hệ thống bằng cách sử dụng các công cụ thủ công hoặc tự động để đánh giá một hoặc nhiều thuộc tính quan tâm. Mục đích của kiểm thử phần mềm là xác định các lỗi, khoảng trống hoặc các yêu cầu còn thiếu đối lập với các yêu cầu thực tế.

## Kiến thức cơ bản về Software Testing ?

* Định nghĩa Kiểm thử phần mềm: Kiểm thử phần mềm được định nghĩa là một hoạt động để kiểm tra xem kết quả thực tế có khớp với kết quả mong đợi hay không và để đảm bảo rằng hệ thống phần mềm không có khiếm khuyết.
* Kiểm tra rất quan trọng vì lỗi phần mềm có thể tốn kém hoặc thậm chí nguy hiểm.
* Những lý do quan trọng để sử dụng kiểm thử phần mềm là: hiệu quả về chi phí, tính bảo mật, chất lượng sản phẩm và sự hài lòng của khách hàng.
* Thông thường Thử nghiệm được phân loại thành ba loại thử nghiệm chức năng, thử nghiệm phi chức năng hoặc thử nghiệm hiệu suất và bảo trì.
* Các chiến lược quan trọng trong kỹ thuật phần mềm là: kiểm thử đơn vị, kiểm thử tích hợp, kiểm tra xác nhận và kiểm thử hệ thống.

## How important is software testing?

Software Testing rất quan trọng vì nếu có bất kỳ lỗi hoặc lỗi nào trong phần mềm, nó có thể được xác định sớm và có thể được giải quyết trước khi giao sản phẩm phần mềm. Sản phẩm phần mềm được kiểm tra thích hợp đảm bảo độ tin cậy, bảo mật và hiệu suất cao, giúp tiết kiệm thời gian, hiệu quả chi phí và sự hài lòng của khách hàng.

## What are the benefits of Software Testing?

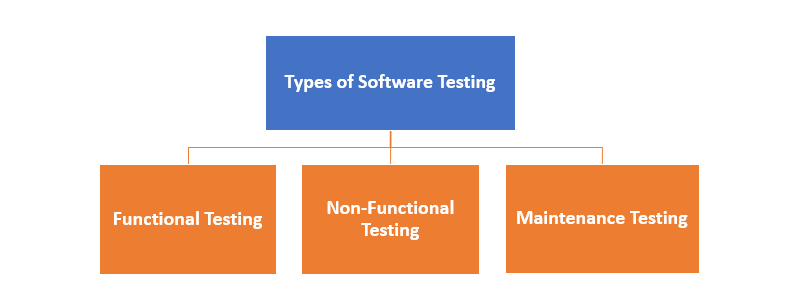
**Cost-Effective( Chi Phí ):** Là một trong những lợi thế quan trọng của kiểm thử phần mềm. Kiểm tra bất kỳ dự án CNTT nào đúng thời hạn sẽ giúp bạn tiết kiệm tiền của mình trong dài hạn. Trong trường hợp nếu các lỗi được phát hiện trong giai đoạn kiểm thử phần mềm trước đó, thì chi phí sửa chữa sẽ thấp hơn

**Security( Bảo Mật ):** Là lợi ích nhạy cảm và dễ bị tấn công nhất của kiểm thử phần mềm. Mọi người đang tìm kiếm các sản phẩm đáng tin cậy. Nó giúp loại bỏ các rủi ro và vấn đề sớm hơn.

**Product quality( Chất Lượng ):** Là yêu cầu thiết yếu của bất kỳ sản phẩm phần mềm nào. Kiểm tra đảm bảo một sản phẩm chất lượng được cung cấp cho khách hàng.

**Customer Satisfaction(Sự Hài Lòng Của Khách Hàng):**Mục đích chính của bất kỳ sản phẩm nào là mang lại sự hài lòng cho khách hàng của họ. Kiểm tra UI / UX đảm bảo trải nghiệm người dùng tốt nhất.

## Types of Software Testing ?



|  |  |
| --- | --- |
| **Testing Category** | **Types of Testing** (150 types More) |
| Functional Testing | Unit TestingIntegration TestingSmokeUAT ( User Acceptance Testing)LocalizationGlobalizationInteroperabilitySo on |
| Non-Functional Testing | PerformanceEnduranceLoadVolumeScalabilityUsabilitySo on |
| Maintenance | RegressionMaintenance |

## Program Testing

### **Program Testing** Trong software testing là một phương pháp thực thi một chương trình phần mềm thực tế với mục đích kiểm tra hành vi của chương trình và tìm ra lỗi. Chương trình phần mềm được thực thi với dữ liệu trường hợp thử nghiệm để phân tích hành vi của chương trình hoặc phản ứng với dữ liệu thử nghiệm. Một thử nghiệm chương trình tốt là một thử nghiệm có khả năng cao tìm thấy lỗi.

# Manual Testing

## What is Manual Testing ?

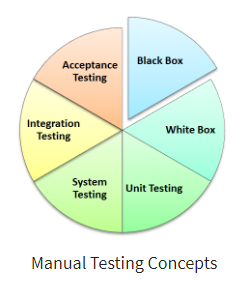
**Manual Testing** là một trong những công việc theo dạng kiểm thử phần mềm, hoặc là một chương trình được thực hiện bằng tay bởi các tester mà không thông qua bất kỳ công cụ hỗ trợ nào.Nó hoạt động dựa vào mục đích phát hiện các lỗi bug từ nhỏ cho đến lớn trong phần mềm.

Từ đó, đưa ra những định hướng giải quyết để có thể đảm bảo cho phần mềm hoạt động ổn định nhất khi giao cho khách hàng

## Mục tiêu của Manual Testing ?

Khái niệm chính của manual testing là nó hoạt động đúng và phù hợp với yêu cầu đặc tả. Test Suites hoặc testcase, được thiết kế trong giai đoạn thử nghiệm và phải có phạm vi kiểm tra bao phủ 100%. Nó cũng đảm bảo rằng các lỗi được báo cáo được sửa chữa bởi các nhà phát triển và kiểm tra lại đã được thực hiện bởi những Tester. Về cơ bản, mục đích là kiểm tra chất lượng của hệ thống và cung cấpsản phẩm không có lỗi cho khách hàng

## Types of 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

## How to perform 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.

## **Manual Testing có những ưu và nhược điểm gì?**

|  |  |
| --- | --- |
| Ưu Điểm | Nhược Điểm |
| * Hầu hết, các tester sẽ dễ dàng hơn trong việc test giao diện giúp mang lại những phản hồi nhanh chóng và trực quan nhất về giao diện ứng dụng của phần mềm cần test. * Nếu như chương trình có thay đổi nhỏ, thì Manual Testing sẽ không phải mất quá nhiều thời gian cho việc kiểm tra đối với những trường hợp kiểm thử. * Các tester sẽ có nhiều hơn cơ hội cho việc khám phá kiếm thử. Từ đó, hiểu thêm về những lỗi phát sinh khó phát hiện để có thể nâng cao được tay nghề cho tester. | * MT sẽ tiêu tốn nhiều thời gian cũng như công sức của tester hơn trong việc phát hiện ra các lỗi bug. * Kết quả tìm thấy thường ít được tin cậy hơn bởi có thể do những sai sót của con người khi thực hiện. * Đối với những task công việc lớn có dạng như: performance testing và stress testing; thì công việc đòi hỏi phải có những tools thực sự chuyên nghiệp để có thể hỗ trợ kỹ thuật manual testing nên việc thực hiện sẽ rất khó. * Việc sử dụng Manual Testing sẽ giúp bạn phát hiện lỗi thường ít hơn so với kỹ thuật tự động test. * Chi phí dành cho dự án có thể phải tăng lên do những đòi hỏi trong việc phải thuê nguồn nhân lực MT để tiến hành kiểm tra. * Quá trình sử dụng tools tự động để test sẽ cho bạn kết quả nhanh hơn cũng như chính xác hơn so với MT. |

## **Một Manual Testing cần làm những công việc gì?**

### Với những MT khi mới bắt đầu vào nghề thì cần trau dồi được mọi kỹ năng cũng như kiến thức cơ bản cho việc nắm bắt để thực hiện tốt công việc của mình. Dưới đây là một số vấn đề bạn cần chuẩn bị như sau:

#### Hiểu rõ những kỹ thuật test manual cơ bản,  cần xây dựng tư duy phân tích để tìm được ra lỗi tốt cũng như nắm vững mọi quy định liên quan đến kỹ thuật test.

#### Phải nâng cao trình độ đọc hiểu tiếng anh để quá trình tìm hiểu các tài liệu hướng dẫn của nước ngoài được dễ dàng hơn. Đây cũng là một trong những yếu tố bạn cần lưu ý để có thể ghi điểm với nhà tuyển dụng.

# Automation Testing

## Khái Niệm

Trong lĩnh vực kiểm thử phần mềm, thì kiểm thử tự động hay còn gọi là Automation testing đóng một vai trò quan trọng góp phần nâng cao năng suất kiểm thử, giảm thiểu lỗi cũng như sự nhàm chán với việc kiểm thử bằng tay trong một thời gian dài hoặc lặp đi lặp lại.

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 St**u**dio CodedUI Testing
* TestComplete (SmartBear)
* SOAPUI - Web Services Testing (SmartBear)

## Ưu, nhược điểm của Automation Testing so với Manual Testing

### **Ưu điểm:**

Độ tin cậy cao: công cụ kiểm thử tự động có sự ổn định cao hơn so với con người, đặc biệt trong trường hợp nhiều test cases, nên độ tin cậy cao hơn so với kiểm thử thủ công.

Khả năng lặp: công cụ kiểm thử tự động ra đời là để giúp cho các tester không phải lặp đi lặp lại các thao tác (ví dụ: nhập dữ liệu, click, check kết quả…) một cách nhàm chán với độ tin cậy và ổn định cao.

Khả năng tái sử dụng: với một bộ kiểm thử tự động, người ta có thể sử dụng cho nhiều phiên bản ứng dụng khác nhau, đây được gọi là tính tái sử dụng.

Tốc độ cao: do thực thi bởi máy nên tốc độ của kiểm thử tự động nhanh hơn nhiều so với tốc độ của con người. Nếu cần 5 phú để thực thi một test case một cách thủ công thì có thể người ta chỉ cần khoảng 30s để thực thi một cách tự động.

Chi phí thấp: nếu áp dụng kiểm thử tự động đúng cách, người ta có thể tiết kiệm được nhiều chi phí, thời gian và nhân lực, do kiểm thử tự động nhanh hơn nhiều so với kiểm thử thủ công, đồng thời nhân lực cần để thực thi và bảo trì scripts không nhiều.

### **Nhược điểm:**

Khó mở rộng, khó bảo trì: trong cùng một dự án, để mở rộng phạm vi cho kiểm thử tự động khó hơn nhiều so với kiểm thử thủ công vì cập nhật hay chỉnh sửa yêu cầu nhiều công việc như debug, thay đổi dữ liệu đầu vào và cập nhật code mới.

Khả năng bao phủ thấp: do khó mở rộng và đòi hỏi nhiều kỹ năng lập trình nên độ bao phủ của kiểm thử tự động thấp xét trên góc nhìn toàn dự án.

Vấn đề công cụ và nhân lực: hiện nay cũng có nhiều công cụ hỗ trợ kiểm thử tự động khá tốt nhưng chúng vẫn còn nhiều hạn chế. Ngoài ra nhân lực đạt yêu cầu (có thể sử dụng thành thạo các công cụ này) cũng không nhiều.

## Tầm quan trọng của automation testing trong thời kỳ công nghiệp 4.0

Cách mạng công nghiệp lần thứ tư đánh dấu kỷ nguyên vạn vật kết nối Internet. Nó xảy ra dựa trên sự hội tụ của nhiều công nghệ trong đó có công nghệ cốt lõi có công nghệ thông tin với sự phát triển không ngừng của công nghệ Internet từ thời kỳ kết nối nội dung như email đến mạng xã hội, Internet vạn vật, Internet kết nối thiết bị máy móc kết nối quá trình vận hành của các nhà máy. Ngoài công nghệ cốt lõi còn có sự hội tụ của công nghệ in 3D, công nghệ vật liệu tiên tiến, công nghệ lưu trữ…

Hiện nay, nền công nghiệp 4.0 đang phát triển mạnh mẽ, kéo theo sự phát triển của rất nhiều ngành nghề, đặc biệt là ngành công nghệ thông tin. Do đó, yêu cầu về nhân lực trong mảng này cũng đòi hỏi cac ứng viên cần phải có kỹ năng tốt hơn nữa về lập trình, technical, các kỹ năng về automation....

Nếu search trên các trang mạng tuyển dụng lớn như ITviec, Vietnamworks, LinkIn, Indeed, Dice, Monster, CareerBuilder … thì bạn sẽ thấy có vô vàn kết quả với các từ khóa tìm kiếm như:

* Test Automation Engineer
* Automation Developer
* Automation Testing/Automation Tester
* QA Automation Engineer
* Software Development Engineer in Test (SDET)

# White box/ Black box testing

## Khái niệm

**White Box Testing:**

White Box Testing là kỹ thuật kiểm thử phần mềm trong đó cấu trúc bên trong, thiết kế và mã hóa của phần mềm được kiểm tra để xác minh luồng đầu vào – đầu ra và cải thiện thiết kế, khả năng sử dụng và bảo mật.

Trong White Box Testing là, mã có thể nhìn thấy đối với người thử nghiệm nên nó còn được gọi là Thử nghiệm hộp rõ ràng, Thử nghiệm hộp mở, Thử nghiệm hộp trong suốt, Thử nghiệm dựa trên mã và Thử nghiệm hộp thủy tinh.

Là một trong hai phần của phương pháp Box Testing đối với kiểm thử phần mềm. Đối tác của nó, thử nghiệm Hộp đen, liên quan đến việc thử nghiệm từ góc độ bên ngoài hoặc kiểu người dùng cuối. Mặt khác, White Box Testing dựa trên hoạt động bên trong của một ứng dụng và xoay quanh kiểm tra nội bộ.

**Black Box Testing:**

Kiểm thử hộp đen: là một phương pháp kiểm thử phần mềm được thực hiện mà không biết được cấu tạo bên trong của phần mềm, là cách mà các tester kiểm tra xem hệ thống như một chiếc hộp đen, không có cách nào nhìn thấy bên trong của cái hộp.

* Nó còn được gọi là kiểm thử hướng dữ liệu hay là kiểm thử hướng in/out.
* Người kiểm thử nên xây dựng các nhóm giá trị đầu vào mà sẽ thực thi đầy đủ tất cả các yêu cầu chức năng của chương trình.
* Cách tiếp cận của các tester đối với hệ thống là không dùng bất kỳ một kiến thức về cấu trúc lập trình bên trong hệ thống, xem hệ thống là một cấu trúc hoàn chỉnh, không thể can thiệp vào bên trong.

Black Box Testing chủ yếu là được thực hiện trong Function test và System test.

Phương pháp này được đặt tên như vậy bởi vì các chương trình phần mềm, trong con mắt của các tester, giống như một hộp đen; bên trong mà người ta không thể nhìn thấy. Phương pháp này cố gắng tìm ra các lỗi trong các loại sau:

* Chức năng không chính xác hoặc thiếu.
* Lỗi giao diện.
* Lỗi trong cấu trúc dữ liệu hoặc truy cập cơ sở dữ liệu bên ngoài.
* Hành vi hoặc hiệu suất lỗi.
* Khởi tạo và chấm dứt các lỗi.

Mọi kỹ thuật nào cũng có ưu điểm và nhược điểm của nó. Các hệ thống thường phải được sử dụng nhiều phương pháp kiểm thử khác nhau để đảm bảo được chất lượng của hệ thống khi đến tay người dùng.

## Ưu Và Nhược Điểm của white box và black box testing

**Ưu / Nhược điểm White Box**

* Ưu điểm
* Test có thể bắt đầu ở giai đoạn sớm hơn, không cần phải chờ đợi cho GUI để có thể test
* Test kỹ càng hơn, có thể bao phủ hầu hết các đường dẫn
* Thích hợp trong việc tìm kiếm lỗi và các vấn đề trong mã lệnh
* 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.
* Nhược điểm
* Vì các bài kiểm tra rất phức tạp, đòi hỏi phải có các nguồn lực có tay nghề cao, với kiến thức sâu rộng về lập trình và thực hiện.
* Maintenance test script có thể là một gánh nặng nếu thể hiện thay đổi quá thường xuyên.
* Vì phương pháp thử nghiệm này liên quan chặt chẽ với ứng dụng đang được test, nên các công cụ để phục vụ cho mọi loại triển khai / nền tảng có thể không sẵn có.

**Ưu / Nhược điểm Black Box**

* **Ưu điểm**
* Các tester được thực hiện từ quan điểm của người dùng và sẽ giúp đỡ trong việc sáng tỏ sự chênh lệch về thông số kỹ thuật.
* Các tester theo phương pháp black box không có “mối ràng buộc” nào với code, và nhận thức của một tester rất đơn giản: một source code có nhiều lỗi. Sử dụng nguyên tắc, "Hỏi và bạn sẽ nhận" các tester black box tìm được nhiều bug ở nơi mà các DEV không tìm thấy.
* Tester có thể không phải IT chuyên nghiệp, không cần phải biết ngôn ngữ lập trình hoặc làm thế nào các phần mềm đã được thực hiện.
* Các tester có thể được thực hiện bởi một cơ quan độc lập từ các developer, cho phép một cái nhìn khách quan và tránh sự phát triển thiên vị.
* Hệ thống thật sự với toàn bộ yêu cầu của nó được kiểm thử chính xác.
* Thiết kế kịch bản kiểm thử khá nhanh, ngay khi mà các yêu cầu chức năng được xác định.
* **Nhược Điểm**
* Dữ liệu đầu vào yêu cầu một khối lượng mẫu (sample) khá lớn
* Nhiều dự án không có thông số rõ ràng thì việc thiết kế test case rất khó và do đó khó viết kịch bản kiểm thử do cần xác định tất cả các yếu tố đầu vào, và thiếu cả thời gian cho việc tập hợp này.
* Khả năng để bản thân kỹ sư lạc lối trong khi kiểm thử là khá cao.
* Chỉ có một số nhỏ các đầu vào có thể được kiểm tra và nhiều đường dẫn chương trình sẽ được để lại chưa được kiểm tra.
* Kiểm thử black box được xem như "là bước đi trong mê cung tối đen mà không mang đèn pin” bởi vì tester không biết phần mềm đang test đã được xây dựng như thế nào. Có nhiều trường hợp khi một tester viết rất nhiều trường hợp test để kiểm tra một số thứ có thể chỉ được test bằng một trường hợp test và/hoặc một vài phần cuối cùng không được test hết.

## SỰ KHÁC NHAU GIỮA BLACK BOX TEST VÀ WHITE BOX TEST

| **Tiêu chuẩn** | **Black box test** | **White box test** |
| --- | --- | --- |
| 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 |

## 