

**BỘ GIÁO DỤC VÀ ĐÀO TẠO  
TRƯỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG**

-----



**ISO 9001:2015**

**ĐỒ ÁN TỐT NGHIỆP**

**NGÀNH: CÔNG NGHỆ THÔNG TIN**

**Sinh viên : Bùi Trần Lĩnh**

**Giảng viên hướng dẫn: ThS. Nguyễn Trịnh Đông**

**HẢI PHÒNG - 2018**

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**  
**TRƯỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG**

-----

**KIỂM THỬ PHẦN MỀM TRÊN THIẾT BỊ DI ĐỘNG VÀ ỨNG  
DỤNG PHẦN MỀM APPIUM STUDIO CHO ỨNG DỤNG  
TRÊN IOS**

**ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY**  
**NGÀNH: CÔNG NGHỆ THÔNG TIN**

**Sinh viên : Bùi Trần Linh**

**Giảng viên hướng dẫn : ThS. Nguyễn Trịnh Đông**

**HẢI PHÒNG - 2018**

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**  
**TRƯỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG**

-----

**NHIỆM VỤ ĐỀ TÀI TỐT NGHIỆP**

**Sinh viên: Bùi Trần Linh**

**Mã SV: 1412101135**

**Lớp: CT1801**

**Ngành: Công nghệ thông tin**

**Tên đề tài: Kiểm thử phần mềm trên thiết bị di động và ứng dụng phần mềm Appium Studio cho ứng dụng trên IOS**

## **LỜI CẢM ƠN**

*Được sự phân công của Khoa Công nghệ thông tin Trường Đại Học Dân lập Hải Phòng, và dưới sự hướng dẫn của Thầy giáo hướng dẫn ThS. Nguyễn Trịnh Đông, em đã hoàn thành đề tài “Kiểm thử phần mềm trên thiết bị di động và ứng dụng phần mềm Appium Studio cho ứng dụng trên IOS”.*

*Để hoàn thành khóa luận này, em xin chân thành cảm ơn tới các thầy cô giáo đã tận tình hướng dẫn, giảng dạy trong suốt quá trình học tập, nghiên cứu và rèn luyện ở Trường Đại Học Dân lập Hải Phòng. Đặc biệt xin gửi lời cảm ơn chân thành tới Thầy giáo hướng dẫn ThS. Nguyễn Trịnh Đông đã tận tình, chu đáo hướng dẫn em thực hiện khoá luận này.*

*Mặc dù đã có nhiều cố gắng để thực hiện đề tài một cách hoàn chỉnh nhất. Song do thời gian có hạn, trình độ hiểu biết và nhận thức còn chưa cao cho nên trong đồ án không thể tránh khỏi những thiếu sót, em rất mong nhận được sự đóng góp ý kiến của các thầy cô và bạn bè để em có thể hoàn thiện đồ án này tốt hơn.*

*Em xin chân thành cảm ơn!*

*Hải Phòng, ngày 31 tháng 3 năm 2018*

*Sinh viên thực hiện*

*Bùi Trần Linh*

# MỤC LỤC

LỜI CẢM ƠN.....	1
MỤC LỤC .....	5
DANH MỤC HÌNH VẼ VÀ BẢNG BIỂU .....	7
DANH MỤC TỪ VIẾT TẮT VÀ THUẬT NGỮ .....	8
MỞ ĐẦU .....	10
CHƯƠNG 1: CÁC KIẾN THỨC CƠ BẢN.....	13
1. Phần mềm .....	13
2. Kiểm thử phần mềm và một số khái niệm liên quan .....	13
2.1. Kiểm thử phần mềm .....	13
2.2. Một số khái niệm liên quan .....	14
3. Quy trình kiểm thử phần mềm .....	16
4. Các cấp độ kiểm thử .....	17
4.1. Kiểm thử mức đơn vị .....	18
4.2. Kiểm thử tích hợp .....	19
4.3. Kiểm thử hồi quy .....	19
4.4. Kiểm thử chấp nhận sản phẩm .....	20
4.5. Kiểm thử mức hệ thống .....	20
5. Các kỹ thuật kiểm thử phần mềm .....	20
5.1. Nguyên tắc cơ bản kiểm thử phần mềm .....	21
5.2. Kỹ thuật kiểm thử hộp trắng (White-Box Testing) .....	23
5.3. Kỹ thuật kiểm thử hộp đen (Black-Box Testing).....	25
6. Kỹ thuật thiết kế Ca kiểm thử .....	26
6.1. Cấu trúc của Ca kiểm thử .....	27
6.2. Phân vùng tương đương .....	28
6.3. Phân tích giá trị biên .....	31
6.4. Đoán lỗi.....	33
7. Tạo Bug report .....	34
7.1. Bug và Bug report.....	34
7.2. Cấu trúc một Bug report .....	34
7.3. Severity và Priority.....	36
CHƯƠNG 2: KIỂM THỬ TRÊN THIẾT BỊ DI ĐỘNG .....	38
1. Kiểm thử trên thiết bị di động .....	38
1.1. Các khái niệm cơ bản về ứng dụng di động .....	38

1.2. Phương pháp kiểm thử trên thiết bị di động .....	41
1.3. Các loại kiểm thử di động .....	45
1.4. Các đặc điểm của kiểm thử di động .....	46
2. Kiểm thử tự động .....	48
2.1. Khái niệm kiểm thử tự động .....	48
2.2. Mục tiêu của kiểm thử tự động .....	48
2.3. Nguyên tắc kiểm thử tự động .....	50
2.4. Quy trình kiểm thử tự động .....	52
2.5. Ưu điểm của kiểm thử tự động .....	53
2.6. Một số công cụ kiểm thử tự động .....	54
2.7. So sánh kiểm thử tự động và kiểm thử thủ công .....	55
<b>CHƯƠNG 3: THỰC NGHIỆM SỬ DỤNG APPIUM STUDIO CHO KIỂM THỬ TỰ ĐỘNG TRÊN IOS .....</b>	<b>57</b>
1. Giới thiệu phần mềm Appium Studio .....	57
1.1. Công cụ Appium .....	57
1.2. Phần mềm Appium Studio .....	58
2. Appium Studio tích hợp trong Eclipse .....	59
3. Thực nghiệm với Appium Studio tích hợp trong Eclipse .....	60
3.1. Cài đặt Appium Studio .....	60
3.2. Kết nối với thiết bị trên Cloud .....	61
3.3. Xây dựng bộ ca kiểm thử cho một ứng dụng cần kiểm thử .....	63
3.4. Tạo dự án kiểm thử .....	65
3.5. Báo cáo .....	74
<b>KẾT LUẬN .....</b>	<b>78</b>
<b>TÀI LIỆU THAM KHẢO .....</b>	<b>79</b>

## **DANH MỤC HÌNH VẼ VÀ BẢNG BIỂU**

Hình 1-1: Ví dụ về 1 Kịch bản kiểm thử .....	16
Hình 1-2: Giai đoạn kiểm thử trong xử lý phần mềm.....	16
Hình 1-3: Luồng thông tin kiểm thử .....	22
Hình 1-4: Minh họa Kiểm thử hộp đen.....	26
Hình 1-5: Minh họa của một ca kiểm thử .....	28
Hình 1-6: Minh họa một Form đăng nhập .....	29
Hình 1-7: Minh họa một Bug report .....	35
Hình 2-1: Quy trình Kiểm thử tự động trong mối quan hệ với Kiểm thử phần mềm.....	53
Bảng 2-2: So sánh kiểm thử tự động và kiểm thử thủ công.....	56
Hình 3-1: Kết quả tìm kiếm Appium Studio.....	60
Hình 3-2: Lấy URL để cài đặt Appium Studio .....	60
Hình 3-3: Dán URL vào cửa sổ Install để tiến hành cài đặt .....	61
Hình 3-4: Giao diện trang Cloud của SeeTest .....	61
Hình 3-5: Copy lại Access Key .....	62
Hình 3-6: Kiểm tra kết nối đến máy chủ Cloud.....	62
Hình 3-7: Các thiết bị Cloud được hiển thị trong Eclipse .....	62
Hình 3-8: Màn hình thiết bị được hiển thị sau khi kết nối.....	63
Hình 3-9: Giao diện chương trình máy tính cần kiểm thử.....	63
Hình 3-10: Bộ ca kiểm thử cho ứng dụng máy tính .....	65
Hình 3-11: Đoạn code IOSTest được sinh tự động trong Project.....	66
Hình 3-12: Kết quả tìm kiếm “TestNG” .....	66
Hình 3-13: Kết quả sau khi cài đặt ứng dụng Basic Calculator.....	67
Hình 3-14: Code cài đặt ứng dụng được thêm vào phần setUp.....	68
Hình 3-15: Thêm câu lệnh để chương trình không tự động thoát khi thực hiện kiểm thử ..	68
Hình 3-16: Chọn biểu tượng Dump UI ở cửa sổ Devices.....	69
Hình 3-17: Màn hình được lưu với tên “mainscreen.dump” .....	69
Hình 3-18: Lưu lại đối tượng nút AC của màn hình máy tính.....	70
Hình 3-19: Đoạn mã sinh số thập phân ngẫu nhiên từ -999 đến 999 .....	71
Hình 3-20: Đoạn mã sinh số nguyên ngẫu nhiên từ -999 đến 999 .....	71
Hình 3-21: Đoạn mã sinh dữ liệu kiểm thử tự động .....	73
Hình 3-22: Khởi chạy kiểm thử tự động.....	73
Hình 3-23: Quá trình chạy kiểm thử trên web .....	74
Hình 3-24: Kết quả sinh ca kiểm thử tự động.....	74
Hình 3-25: Toàn bộ báo cáo được sinh tự động trong phần Reports.....	75
Hình 3-26: Chi tiết quá trình thực hiện kiểm thử tự động .....	75
Hình 3-27: Ca kiểm thử đầu tiên không đưa ra kết quả chính xác .....	75
Hình 3-28: Tổng hợp kết quả kiểm thử.....	76
Hình 3-29: Bug report lỗi của nút (+/-).....	77

**DANH MỤC TỪ VIẾT TẮT VÀ THUẬT NGỮ**

STT	KÝ HIỆU	CỤM TỪ ĐẦY ĐỦ	Ý NGHĨA
1	3G	Third-generation technology	Công nghệ truyền thông thế hệ thứ ba, cho phép truyền cả dữ liệu thoại và dữ liệu ngoài thoại
2	API	Application Programming Interface	Giao diện lập trình ứng dụng - là 1 giao tiếp phần mềm được dùng bởi các ứng dụng khác nhau
3	BSD	Berkeley Software Distribution	Tên của một hệ điều hành dẫn xuất từ UNIX được phát hành vào thập niên 1970 từ trường Đại học California tại Berkeley
4	CPU	Central Processing Unit	Bộ xử lý trung tâm
5	Framework	Framework	Framework là một thư viện các lớp đã được xây dựng hoàn chỉnh, bộ khung để phát triển các Phần mềm ứng dụng
6	GPRS	General Packet Radio Service	Dịch vụ vô tuyến gói tổng hợp - là một dịch vụ dữ liệu di động dạng gói dành cho những người dùng Hệ thống thông tin di động toàn cầu
7	GPS	Global Positioning System	Hệ thống định vị toàn cầu
8	GSM	Global System for Mobile Communications	Một công nghệ dùng cho mạng thông tin di động
9	HTTP	HyperText Transfer Protocol	Giao thức truyền tải siêu văn bản
10	ID	Identification number	Mã số



11	IDE	Integrated Development Environment	Phần mềm bao gồm những gói phần mềm khác giúp phát triển ứng dụng phần mềm (Môi trường phát triển tích hợp)
12	IEEE	Institute of Electrical and Electronics Engineers	Viện kỹ nghệ Điện và Điện tử
13	IT	Information Technology	Công nghệ thông tin
14	Linux	Linux	Tên gọi của một hệ điều hành máy tính và cũng là tên hạt nhân của hệ điều hành.
15	QA	Quality Assurance	Người chịu trách nhiệm đảm bảo chất lượng sản phẩm
16	SDK	Software Development Kit	Thuật ngữ được Microsoft, Sun Microsystems và một số công ty khác sử dụng – một bộ công cụ phát triển phần mềm
17	SMS	Short Message Services	Giao thức viễn thông cho phép gửi các thông điệp dạng text ngắn
18	SQA	Software Quality Assurance	Tập hợp các hoạt động đảm bảo chất lượng trong quá trình gia công phần mềm
19	UI	User Interface	Giao diện người dùng
20	URL	Uniform Resource Locator	Định vị tài nguyên thống nhất, được dùng để tham chiếu tới tài nguyên trên Internet
21	V&V	Verification and Validation	Xác minh và thẩm định
22	WAP	Wireless Application Protocol	Giao thức Ứng dụng không dây - là một tiêu chuẩn công nghệ cho các hệ thống truy nhập Internet từ các thiết bị di động

## MỞ ĐẦU

### **Lý do chọn đề tài:**

Với sự phát triển như vũ bão của công nghệ thông tin nói chung và công nghệ phần mềm nói riêng, việc phát triển phần mềm ngày càng được hỗ trợ bởi nhiều công cụ tiên tiến, giúp cho việc xây dựng phần mềm đỡ mệt nhọc và hiệu quả hơn. Tuy nhiên, vì độ phức tạp của phần mềm và những giới hạn về thời gian và chi phí, cho dù các hoạt động đảm bảo chất lượng phần mềm nói chung và kiểm thử nói riêng ngày càng chặt chẽ và khoa học, vẫn không đảm bảo được rằng các sản phẩm phần mềm đang được ứng dụng không có lỗi. Lỗi vẫn luôn tiềm ẩn trong mọi sản phẩm phần mềm và cũng có thể gây những thiệt hại khôn lường.

Kiểm thử phần mềm là một quá trình liên tục, xuyên suốt mọi giai đoạn phát triển phần mềm để đảm bảo rằng phần mềm thoả mãn các yêu cầu thiết kế và các yêu cầu đó đáp ứng các nhu cầu của người dùng. Các kỹ thuật kiểm thử phần mềm đã và đang được nghiên cứu, và việc kiểm thử phần mềm đã trở thành quy trình bắt buộc trong các dự án phát triển phần mềm trên thế giới. Kiểm thử phần mềm là một hoạt động rất tốn kém, mất thời gian, và khó phát hiện được hết lỗi. Vì vậy, việc kiểm thử phần mềm đòi hỏi phải có chiến lược phù hợp, một kế hoạch hợp lý và việc thực hiện được quản lý chặt chẽ.

Và với việc những chiếc điện thoại thông minh đang ngày càng được sử dụng nhiều hơn nhằm đáp ứng nhu cầu giải trí đa dạng của người dùng. Từ một chiếc điện thoại thông thường chỉ được cài đặt sẵn vài ba ứng dụng của nhà sản xuất thì nay với các thiết bị chạy các hệ điều hành nhúng (Android, iOS, v.v.) ta có thể dễ dàng đáp ứng được các nhu cầu của người dùng bằng cách cài thêm các phần mềm bên thứ ba mà không gây ra trở ngại nào. Từ đây lại đặt ra một vấn đề hiển nhiên là kiểm thử các phần mềm chạy trên di động này để xem chúng có đáp ứng được các yêu cầu đề ra ban đầu hay không trước khi phát hành sản phẩm tới tay người tiêu dùng.

Đó là lý em chọn đề tài “Kiểm thử phần mềm trên thiết bị di động và ứng dụng phần mềm Appium Studio cho ứng dụng trên IOS” làm đồ án tốt nghiệp.

Mục đích của đồ án:

Đề tài tìm hiểu cơ sở lý thuyết về kiểm thử nói chung và kiểm thử trên di động nói riêng cũng như cách triển khai công cụ kiểm thử phần mềm tự động để giảm nhân lực kiểm thử và đảm bảo chất lượng phần mềm hơn với công việc kiểm thử bằng tay. Mục tiêu chính của đề tài là nghiên cứu về kiểm thử trên thiết bị di động.

### **Đối tượng và phạm vi nghiên cứu:**

Đồ án nghiên cứu lý thuyết kiểm thử phần mềm. Bên cạnh đó, nghiên cứu các vấn đề về kiểm thử phần mềm trên thiết bị di động và ứng dụng phần mềm Appium Studio cho kiểm thử tự động trên IOS.

### **Phương pháp nghiên cứu:**

Nghiên cứu tổng quan về kiểm thử phần mềm và các kỹ thuật kiểm thử từ đó áp dụng vào kiểm thử phần mềm trên thiết bị di động, tìm hiểu công cụ kiểm thử phần mềm Appium Studio trên IOS.

Với mục tiêu đặt ra như vậy, những nội dung và kết quả nghiên cứu chính của đồ án được trình bày trong ba chương như sau:

Chương 1: Các kiến thức cơ bản

Chương 2: Kiểm thử trên thiết bị di động

Chương 3: Thực nghiệm sử dụng phần mềm Appium Studio cho kiểm thử tự động trên IOS

Phần kết luận đưa ra những đánh giá về những kết quả đạt được và những khó khăn gặp phải trong quá trình nghiên cứu thực hiện đồ án.

Trong quá trình thực hiện đồ án, do thời gian cũng như trình độ của em còn có những hạn chế nhất định nên không thể tránh khỏi những sai sót. Rất mong nhận được sự góp ý của các thầy, cô giáo và các bạn để đồ án hoàn thiện hơn. Em xin chân thành cảm ơn sự hướng dẫn, và giúp đỡ tận tình của thầy giáo **ThS. Nguyễn Trịnh Đông**, các thầy cô trong khoa Công nghệ thông tin Trường Đại học Dân lập Hải Phòng đã giúp đỡ em trong quá trình học tập cũng như trong quá trình làm đồ án.

TaiLieu.vn

## **CHƯƠNG 1: CÁC KIẾN THỨC CƠ BẢN**

Kiểm thử nhằm đánh giá chất lượng hoặc tính chấp nhận được của sản phẩm. Ngoài ra, kiểm thử còn giúp phát hiện lỗi hoặc bất cứ vấn đề gì về sản phẩm. Chúng ta cần kiểm thử vì biết rằng con người luôn có thể mắc sai lầm. Điều này đặc biệt đúng trong lĩnh vực phát triển phần mềm và các hệ thống điều khiển bởi phần mềm. Chương này sẽ giới thiệu các khái niệm trong lĩnh vực kiểm thử phần mềm.

### **1. Phần mềm**

Phần mềm thường được mô tả bởi ba thành phần cấu thành [1]:

- *Tập các lệnh* (chương trình máy tính) trên máy tính khi thực hiện sẽ tạo ra các dịch vụ và đem lại những kết quả mong muốn cho người dùng.
- *Các cấu trúc dữ liệu* (lưu giữ trên các bộ nhớ) làm cho chương trình thao tác hiệu quả với các thông tin thích hợp và nội dung thông tin được số hóa.
- *Các tài liệu* để mô tả thao tác, cách sử dụng và bảo trì phần mềm (hướng dẫn sử dụng, tài liệu kỹ thuật, tài liệu phân tích, thiết kế, kiểm thử, v.v.).

### **2. Kiểm thử phần mềm và một số khái niệm liên quan**

#### **2.1. Kiểm thử phần mềm**

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 [2]. Kiểm thử có thể cung cấp cho doanh nghiệp một quan điểm, một cách nhìn độc lập về phần mềm để từ đó cho phép đánh giá và thấu hiểu được những rủi ro trong quá trình triển khai phần mềm.

Trong kỹ thuật kiểm thử không chỉ giới hạn ở việc thực hiện một chương trình hoặc ứng dụng với mục đích đi tìm các lỗi phần mềm (bao gồm các lỗi và

các thiếu sót) mà còn là một quá trình phê chuẩn và xác minh một chương trình máy tính / ứng dụng / sản phẩm nhằm:

Đáp ứng được mọi yêu cầu hướng dẫn khi thiết kế và phát triển phần mềm.

Thực hiện công việc đúng như kỳ vọng.

Có thể triển khai được với những đặc tính tương tự.

Và đáp ứng được mọi nhu cầu của các bên liên quan.

Tùy thuộc vào từng phương pháp, việc kiểm thử có thể được thực hiện bất cứ lúc nào trong quá trình phát triển phần mềm. Theo truyền thống thì các nỗ lực kiểm thử được tiến hành sau khi các yêu cầu được xác định và việc lập trình được hoàn tất nhưng trong Agile (là một tập hợp các phương pháp phát triển phần mềm linh hoạt dựa trên việc lặp đi lặp lại và gia tăng giá trị) thì việc kiểm thử được tiến hành liên tục trong suốt quá trình xây dựng phần mềm. Như vậy, mỗi một phương pháp kiểm thử bị chi phối theo một quy trình phát triển phần mềm nhất định.

## **2.2. Một số khái niệm liên quan**

*Chất lượng phần mềm* (Software quality): là mức độ mà một hệ thống, thành phần hay quy trình đáp ứng các yêu cầu của đặc tả phần mềm, các nhu cầu mong đợi của khách hàng hoặc người sử dụng [3].

*Đảm bảo chất lượng phần mềm* (Software quality assurance): là một quy trình có kế hoạch và hệ thống của tất cả các hành động cần thiết để cung cấp các thông tin đầy đủ để đảm bảo các sản phẩm có phù hợp với các yêu cầu về kỹ thuật hay không. Mục đích cuối cùng là để đánh giá quy trình sản xuất sản phẩm phần mềm [3].

*Xác nhận* (Validation): là quá trình đánh giá một hệ thống hay cấu phần trong hay cuối của quá trình phát triển để xác định xem nó đáp ứng yêu cầu quy định [3].

*Xác minh, kiểm chứng (Verification)*: là quá trình đánh giá một hệ thống hay thành phần để xác định xem các sản phẩm của một giai đoạn phát triển nhất định đáp ứng các điều kiện áp đặt tại lúc bắt đầu của giai đoạn đó [3]. Xác minh thường là hoạt động có tính kỹ thuật cao hơn, sử dụng những tri thức về các yêu cầu, đặc tả phần mềm. Xác nhận thường phụ thuộc vào tri thức về lĩnh vực tương ứng. Cụ thể là, tri thức về ứng dụng của phần mềm được viết. Ví dụ, xác nhận của phần mềm về máy bay yêu cầu tri thức từ kỹ sư hàng không và phi công.

*Lỗi (Error)*: Lỗi là những vấn đề mà con người mắc phải trong quá trình phát triển các sản phẩm phần mềm [4].

*Sai (Fault)*: Sai là kết quả của lỗi, hay nói khác đi, lỗi sẽ dẫn đến sai [4].

*Thất bại (Failure)*: Thất bại xuất hiện khi một lỗi được thực thi [4].

*Sự cố (Incident)*: Khi thất bại xuất hiện, nó có thể hiển thị hoặc không, tức là rõ ràng hoặc không rõ ràng đối với người dùng hoặc người kiểm thử. Sự cố là triệu chứng liên kết với một thất bại và thể hiện cho người dùng hoặc người kiểm thử về sự xuất hiện của thất bại này [4].

*Ca kiểm thử (Test case)*: Ca kiểm thử gồm một tập các dữ liệu đầu vào và một xâu các giá trị đầu ra mong đợi đối với phần mềm, mục đích là dựa vào đó để kiểm tra xem phần mềm có thỏa các yêu cầu đặt ra hay không.

*Kịch bản kiểm thử (Test script)*: Một kịch bản kiểm thử là một nhóm mã lệnh dạng đặc tả kịch bản dùng để tự động hóa một quy trình hay một ca kiểm tra, giúp cho việc kiểm tra nhanh hơn, hoặc cho những trường hợp mà kiểm tra bằng tay sẽ rất khó khăn hoặc không khả thi.

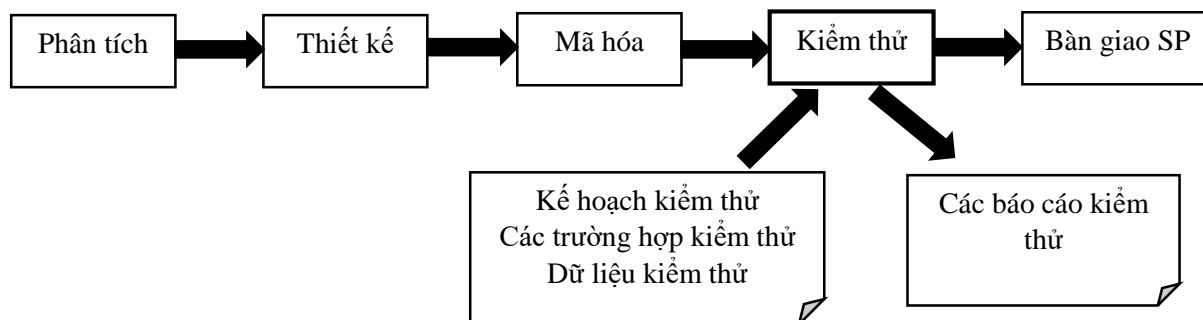


```
nRowCount=DataTable.GlobalSheet.GetRowCount
systemutil.CloseProcessByName "iexplore.exe"
For i = 1 to nRowCount
    Systemutil.Run "iexplore", "http://abc.com"
    Browser("title:=.*").Page("title:=.*").Sync
    DataTable.GlobalSheet.SetCurrentRow(i)
    sEmailid=DataTable("UserName",Global)
    sPassword=DataTable("Password",dtGlobalSheet)
    Browser("title:=.*").Page("title:=.*").WebEdit("name=Email").Set sEmailid
    Browser("title:=.*").Page("title:=.*").WebEdit("name=Passwd").Set sPassword
    Browser("title:=.*").Page("title:=.*").WebButton("name=Sign in").Click
    Browser("title:=.*").Page("title:=.*").Sync
    wait(10)
    If left(browser("title:=.*").Page("title:=.*").getROProperty("title"),13) ="welcome home"Then
        msgbox("signed in successfully")
    Else
        msgbox("Login Unsuccessful")
    End If
    systemutil.CloseProcessByName "iexplore.exe"
Next
```

Hình 1-1: Ví dụ về 1 Kịch bản kiểm thử

### 3. Quy trình kiểm thử phần mềm

Mục đích của kiểm thử là thiết kế một chuỗi các trường hợp kiểm thử mà có khả năng phát hiện lỗi cao. Để cho việc kiểm thử đạt được kết quả tốt cần có sự chuẩn bị về kế hoạch kiểm thử, thiết kế các trường hợp kiểm thử và các dữ liệu kiểm thử cho các trường hợp. Đây chính là đầu vào cho giai đoạn kiểm thử. Và sản phẩm công việc của giai đoạn kiểm thử chính là “báo cáo kiểm thử” mà tài liệu hóa tất cả các trường hợp kiểm thử đã chạy, dữ liệu đầu vào, đầu ra mong đợi, đầu ra thực tế và mục đích của kiểm thử.



Hình 1-2: Giai đoạn kiểm thử trong xử lý phần mềm

Quy trình kiểm thử bao gồm một số giai đoạn:

- Lập kế hoạch kiểm thử: Bước đầu tiên là lập kế hoạch cho tất cả các hoạt động sẽ được thực hiện và các phương pháp được sử dụng. Các chuẩn IEEE bao gồm các thông tin về tác giả chuẩn bị kế hoạch, danh sách liệt kê của kế hoạch kiểm thử. Vấn đề quan trọng nhất đối với kế hoạch kiểm thử:



- Mục đích: Quy định về phạm vi, phương pháp, tài nguyên và lịch biểu của các hoạt động kiểm thử.
  - Các tài liệu tham khảo.
  - Các định nghĩa.
  - Khái quát về xác minh và thẩm định (V&V): tổ chức, tài nguyên, trách nhiệm, các công cụ, kỹ thuật và các phương pháp luận.
  - Vòng đời của V&V: các nhiệm vụ, các dữ liệu vào và các kết quả ra trên một giai đoạn vòng đời.
  - Báo cáo xác minh và thẩm định(V&V) phần mềm: mô tả nội dung, định dạng và thời gian cho tất cả các báo cáo V&V.
  - Các thủ tục quản lý V&V bao gồm các chính sách, thủ tục, các chuẩn, thực nghiệm và các quy ước.
- Giai đoạn bố trí nhân viên kiểm thử: Việc kiểm thử thường phải tiến hành một cách độc lập và các nhóm độc lập có trách nhiệm tiến hành các hoạt động kiểm thử, gọi là các nhóm kiểm thử.
  - Thiết kế các trường hợp kiểm thử: Các trường hợp kiểm thử là các đặc tả đầu vào cho kiểm thử và đầu ra mong đợi của hệ thống cùng với các câu lệnh được kiểm thử.
    - Các kỹ thuật kiểm thử hộp đen để kiểm thử dựa trên chức năng.
    - Các kỹ thuật kiểm thử hộp trắng để kiểm thử dựa vào cấu trúc bên trong.
  - Xử lý đo lường kiểm thử bằng cách thu thập dữ liệu.
  - Đánh giá sản phẩm phần mềm để xác nhận sản phẩm có thể sẵn sàng phát hành được chưa?

#### **4. Các cấp độ kiểm thử**

Các mức kiểm thử phần mềm thông thường:

- *Unit Test* – Kiểm thử mức đơn vị
- *Integration Test* – Kiểm thử tích hợp
- *System Test* - Kiểm thử mức hệ thống

- *Acceptance Test* - Kiểm thử chấp nhận sản phẩm
- *Regression Test* - Kiểm thử hồi quy

#### **4.1. Kiểm thử mức đơn vị**

Một đơn vị kiểm thử là một thành phần phần mềm nhỏ nhất mà ta có thể kiểm thử được. Theo định nghĩa này, các hàm (Function), thủ tục (Procedure), lớp (Class), hoặc các phương thức (Method) đều có thể được xem là đơn vị kiểm thử.

Vì đơn vị kiểm thử được chọn để kiểm thử thường có kích thước nhỏ và chức năng hoạt động đơn giản, chúng ta không khó khăn gì trong việc tổ chức, kiểm thử, ghi nhận và phân tích kết quả kiểm thử. Nếu phát hiện lỗi, việc xác định nguyên nhân và khắc phục cũng tương đối dễ dàng vì chỉ khoanh vùng trong một đơn vị đang kiểm thử. Một nguyên lý đúc kết từ thực tiễn: thời gian tốn cho Kiểm thử đơn vị sẽ được đền bù bằng việc tiết kiệm rất nhiều thời gian và chi phí cho việc kiểm thử và sửa lỗi ở các mức kiểm thử sau đó.

Kiểm thử đơn vị thường do lập trình viên thực hiện. Công đoạn này cần được thực hiện càng sớm càng tốt trong giai đoạn viết code và xuyên suốt chu kỳ phát triển phần mềm. Thông thường, Kiểm thử đơn vị đòi hỏi kiểm thử viên có kiến thức về thiết kế và mã nguồn của chương trình. Mục đích của Kiểm thử đơn vị là bảo đảm thông tin được xử lý và xuất ra là chính xác, trong mối tương quan với dữ liệu nhập và chức năng của đơn vị kiểm thử. Điều này thường đòi hỏi tất cả các nhánh bên trong đơn vị kiểm thử đều phải được kiểm tra để phát hiện nhánh phát sinh lỗi. Một nhánh thường là một chuỗi các lệnh được thực thi trong một đơn vị kiểm thử, ví dụ: chuỗi các lệnh sau điều kiện If và nằm giữa then ... else là một nhánh. Thực tế việc chọn lựa các nhánh để đơn giản hóa việc kiểm thử và quét hết các đơn vị kiểm thử đòi hỏi phải có kỹ thuật, đôi khi phải dùng thuật toán để chọn lựa.

Cũng như các mức kiểm thử khác, Kiểm thử đơn vị cũng đòi hỏi phải chuẩn bị trước các ca kiểm thử hoặc kịch bản kiểm thử, trong đó chỉ định rõ dữ

liệu vào, các bước thực hiện và dữ liệu mong chờ sẽ xuất ra. Các ca kiểm thử và kịch bản này nên được giữ lại để tái sử dụng.

Kiểm thử đơn vị thường sử dụng các Unit Test Framework, đó là các khung chương trình được viết sẵn để hỗ trợ cho việc test các mô đun, các đơn vị phần mềm.

#### **4.2. Kiểm thử tích hợp**

Kiểm thử tích hợp kết hợp các thành phần của một ứng dụng và kiểm thử như một ứng dụng đã hoàn thành. Trong khi Kiểm thử đơn vị kiểm thử các thành phần và đơn vị phần mềm riêng lẻ thì kiểm thử tích hợp kết hợp chúng lại với nhau và kiểm thử sự giao tiếp giữa chúng. Kiểm thử tích hợp có 2 mục tiêu chính:

- Phát hiện lỗi giao tiếp xảy ra giữa các đơn vị kiểm thử.
- Tích hợp các đơn vị kiểm thử đơn lẻ thành các hệ thống nhỏ (subsystem) và cuối cùng là nguyên hệ thống hoàn chỉnh (system) chuẩn bị cho kiểm thử ở mức hệ thống.

#### **4.3. Kiểm thử hồi quy**

Kiểm thử hồi quy không phải là một mức kiểm thử, như các mức khác đã nói ở trên. Nó đơn thuần kiểm tra lại phần mềm sau khi có một sự thay đổi xảy ra, để bảo đảm phiên bản phần mềm mới thực hiện tốt các chức năng như phiên bản cũ và sự thay đổi không gây ra lỗi mới trên những chức năng vốn đã làm việc tốt. Kiểm thử hồi quy có thể thực hiện tại mọi mức kiểm thử. Ví dụ: một phần mềm đang phát triển khi kiểm tra cho thấy nó chạy tốt các chức năng A, B và C. Khi có thay đổi code của chức năng C, nếu chỉ kiểm tra chức năng C thì chưa đủ, cần phải kiểm tra lại tất cả các chức năng khác liên quan đến chức năng C, trong ví dụ này là A và B. Lý do là khi C thay đổi, nó có thể sẽ làm A và B không còn làm việc đúng nữa.

#### **4.4. Kiểm thử chấp nhận sản phẩm**

Thông thường, sau giai đoạn kiểm thử hệ thống là kiểm thử chấp nhận, được khách hàng thực hiện (hoặc ủy quyền cho một nhóm thứ ba thực hiện). Mục đích của kiểm thử chấp nhận là để chứng minh phần mềm thỏa mãn tất cả yêu cầu của khách hàng và khách hàng chấp nhận sản phẩm (và trả tiền thanh toán hợp đồng). Kiểm thử chấp nhận có ý nghĩa hết sức quan trọng, mặc dù trong hầu hết mọi trường hợp, các phép kiểm thử của kiểm thử hệ thống và kiểm thử chấp nhận gần như tương tự, nhưng bản chất và cách thức thực hiện lại rất khác biệt.

#### **4.5. Kiểm thử mức hệ thống**

Mục đích Kiểm thử mức hệ thống là kiểm tra thiết kế và toàn bộ hệ thống (sau khi tích hợp) có thỏa mãn yêu cầu đặt ra hay không. Điểm khác nhau then chốt giữa kiểm thử tích hợp và kiểm thử hệ thống là kiểm thử hệ thống chú trọng các hành vi và lỗi trên toàn hệ thống, còn kiểm thử tích hợp chú trọng sự giao tiếp giữa các đơn vị hoặc đối tượng khi chúng làm việc cùng nhau. Thông thường ta phải thực hiện kiểm thử đơn vị và kiểm thử tích hợp để bảo đảm mọi đơn vị phần mềm và sự tương tác giữa chúng hoạt động chính xác trước khi thực hiện kiểm thử hệ thống. Kiểm thử hệ thống kiểm tra cả các hành vi chức năng của phần mềm lẫn các yêu cầu về chất lượng như độ tin cậy, tính tiện lợi khi sử dụng, hiệu năng và bảo mật. Mức kiểm thử này đặc biệt thích hợp cho việc phát hiện lỗi giao tiếp với phần mềm hoặc phần cứng bên ngoài, chẳng hạn các lỗi “bế tắc” (deadlock) hoặc chiếm dụng bộ nhớ. Sau giai đoạn kiểm thử hệ thống, phần mềm thường đã sẵn sàng cho khách hàng hoặc người dùng cuối cùng kiểm thử để chấp nhận hoặc dùng thử (Alpha/Beta Test).

### **5. Các kỹ thuật kiểm thử phần mềm**

Có thể chia các kỹ thuật kiểm thử phần mềm thành hai loại: các kỹ thuật kiểm thử hộp đen (black-box testing) và kỹ thuật kiểm thử hộp trắng (white-box testing). Các kiểm thử hộp đen tìm các lỗi như thiếu các chức năng, khả