

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

-----



**Phan Thị Ngọc**

**NGHIÊN CỨU PHƯƠNG PHÁP KIỂM THỬ TỰ ĐỘNG  
TRONG KIỂM THỬ PHẦN MỀM CỦA TRUNG TÂM  
THÔNG TIN TÍN DỤNG QUỐC GIA VIỆT NAM**

**LUẬN VĂN THẠC SĨ KỸ THUẬT**

*(Theo định hướng ứng dụng)*

HÀ NỘI – NĂM 2021

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

-----



**Phan Thị Ngọc**

**NGHIÊN CỨU PHƯƠNG PHÁP KIỂM THỬ TỰ ĐỘNG  
TRONG KIỂM THỬ PHẦN MỀM CỦA TRUNG TÂM  
THÔNG TIN TÍN DỤNG QUỐC GIA VIỆT NAM**

**Chuyên ngành: Hệ thống thông tin**

**MÃ SỐ: 8.48.01.04**

**LUẬN VĂN THẠC SĨ KỸ THUẬT**

*(Theo định hướng ứng dụng)*

**NGƯỜI HƯỚNG DẪN: TS. PHẠM THỊ HÀ**

**HÀ NỘI - NĂM 2021**

## **LỜI CAM ĐOAN**

Em xin cam đoan luận văn là thành quả công trình nghiên cứu độc lập với sự giúp đỡ của giáo viên hướng dẫn. Kết quả nghiên cứu chưa từng được công bố trong bất kỳ công trình nào khác. Tất cả sự giúp đỡ cho việc thực hiện luận văn đã được cảm ơn và các thông tin trích dẫn đều có nguồn gốc rõ ràng.

**TÁC GIẢ LUẬN VĂN**

**PHAN THỊ NGỌC**

## LỜI CẢM ƠN

Trước tiên, em xin cảm ơn Học viện Công nghệ Bưu chính viễn thông, Khoa Đào tạo sau đại học, các thầy cô giáo của Học viện đã tạo điều kiện và trực tiếp giảng dạy, giúp đỡ em hoàn thành chương trình đào tạo Thạc sĩ kỹ thuật.

Em xin gửi lời cảm ơn sâu sắc tới Tiến sĩ Phan Thị Hà đã dành nhiều thời gian, tâm huyết hướng dẫn tận tình và đóng góp các ý kiến để em hoàn thiện luận văn này.

Em xin chân thành cảm ơn tới Ban lãnh đạo, anh chị em đồng nghiệp tại Trung tâm thông tin tin dụng Quốc gia Việt Nam đã nhiệt tình góp ý trong quá trình em thực hiện khảo sát và viết bài.

Cuối cùng, em muốn bày tỏ sự biết ơn đối với gia đình, người thân, bạn bè đã luôn bên cạnh khích lệ, động viên, giúp đỡ em trong suốt thời gian học tập và hoàn thiện luận văn.

Em xin trân trọng cảm ơn.

Hà Nội, ngày      tháng      năm 2021.

**Người viết**

**Phạm Thị Ngọc**

## MỤC LỤC

<b>LỜI CAM ĐOAN.....</b>	<b><i>i</i></b>
<b>LỜI CẢM ƠN.....</b>	<b><i>ii</i></b>
<b>DANH MỤC CÁC KÝ HIỆU, CÁC CHỮ VIẾT TẮT.....</b>	<b><i>v</i></b>
<b>DANH MỤC CÁC HÌNH.....</b>	<b><i>vii</i></b>
<b>DANH MỤC CÁC BẢNG.....</b>	<b><i>viii</i></b>
<b>I. MỞ ĐẦU.....</b>	<b><i>1</i></b>
1. Tính cấp thiết của đề tài.....	<b><i>1</i></b>
2. Tổng quan nghiên cứu.....	<b><i>2</i></b>
3. Mục tiêu nghiên cứu.....	<b><i>4</i></b>
4. Đối tượng và phạm vi nghiên cứu.....	<b><i>4</i></b>
5. Phương pháp nghiên cứu.....	<b><i>4</i></b>
<b>II. NỘI DUNG.....</b>	<b><i>4</i></b>
<b>CHƯƠNG I: TỔNG QUAN VỀ KIỂM THỬ PHẦN MỀM.....</b>	<b><i>6</i></b>
<b>1.1. Tổng quan về kiểm thử phần mềm.....</b>	<b><i>6</i></b>
1.1.1. Định nghĩa kiểm thử.....	<b><i>6</i></b>
1.1.2. Vai trò của kiểm thử.....	<b><i>6</i></b>
1.1.3. Các mục đích của kiểm thử phần mềm.....	<b><i>8</i></b>
1.1.4. Các giai đoạn kiểm thử phần mềm.....	<b><i>8</i></b>
<b>1.2. Vai trò của kiểm thử trong dự án của Trung tâm Thông tin Tín dụng Quốc gia Việt Nam.....</b>	<b><i>10</i></b>
1.2.1. Khái quát về Trung tâm Thông tin Tín dụng Quốc gia Việt Nam.....	<b><i>10</i></b>
1.2.2. Chức năng và nhiệm vụ của Trung tâm thông tin tín dụng Quốc gia Việt Nam.....	<b><i>11</i></b>
1.2.3. Bộ máy quản lý của Trung tâm thông tin tín dụng Quốc gia Việt Nam.....	<b><i>13</i></b>
<b>1.3. Quy trình kiểm thử, chất lượng phần mềm và việc đảm bảo chất lượng phần mềm đầu ra.....</b>	<b><i>16</i></b>
1.3.1. Định nghĩa quy trình kiểm thử phần mềm.....	<b><i>16</i></b>
1.3.2. Phân tích quy trình kiểm thử phần mềm.....	<b><i>17</i></b>
<b>1.4. Vai trò của kiểm thử đối với Trung tâm Thông tin Tín dụng Quốc gia Việt Nam.....</b>	<b><i>22</i></b>
<b>CHƯƠNG II: PHƯƠNG PHÁP KIỂM THỬ TỰ ĐỘNG VÀ CÁC CÔNG CỤ...25</b>	
<b>2.1. Tổng quan về kiểm thử tự động.....</b>	<b><i>25</i></b>
2.1.1. Khái niệm kiểm thử tự động.....	<b><i>25</i></b>
2.1.2. Sự khác biệt giữa kiểm thử thủ công và kiểm thử tự động.....	<b><i>26</i></b>

2.1.3. So sánh ưu và nhược điểm của Kiểm thử thủ công và Kiểm thử tự động	28
2.1.4. Kiểm thử thủ công và kiểm thử tự động.....	30
2.1.5. Quy trình kiểm thử phần mềm tự động [11] (Automation testing) .....	31
<b>2.2. Tìm hiểu về các mô hình kiểm thử tự động hiện nay.....</b>	<b>32</b>
2.2.1. Mô hình kiểm thử tự động .....	32
2.2.2. Cấp độ kiểm thử tự động. ....	39
<b>2.3. Tổng quan về các công cụ kiểm thử tự động hiện nay.....</b>	<b>40</b>
2.3.1. Công cụ kiểm thử tự động .....	40
2.3.2. Một số công cụ kiểm thử tự động mới hiện nay. ....	41
<b>2.4. Phân tích so sánh các công cụ kiểm thử tự động. ....</b>	<b>45</b>
<b>2.5. Phân tích đánh giá và nêu ra phương pháp kiểm thử đối với phần mềm được kiểm thử.....</b>	<b>45</b>
<b>CHƯƠNG III: ỨNG DỤNG.....</b>	<b>47</b>
<b>3.1. Đặt vấn đề cho bài toán ứng dụng công cụ kiểm thử.....</b>	<b>47</b>
<b>3.2. Đề xuất áp dụng công cụ kiểm thử chức năng cho website. ....</b>	<b>47</b>
3.2.1. Hiện trạng kiểm thử phần mềm tại Trung tâm Thông tin tin dụng Quốc gia Việt Nam. ....	47
3.2.2. Đề xuất quy trình kiểm thử tự động tại Trung tâm Thông tin Tín dụng Quốc gia Việt Nam .....	51
3.2.3. Các bước để tiến hành Kiểm Thử Tự Động .....	52
<b>3.3. Phân tích sử dụng công cụ. ....</b>	<b>52</b>
3.3.1. Công cụ kiểm thử tự động Selenium. ....	53
3.3.2. Một số tính năng nổi bật chung: .....	56
3.3.3. Các câu lệnh sử dụng trong Selenium WebDriver .....	57
<b>3.4. Lý do để lựa chọn Selenium để kiểm thử ứng dụng web cho CIC. ....</b>	<b>60</b>
<b>3.5. Xây dựng Testcase để thực hiện kiểm thử. ....</b>	<b>62</b>
3.5.1. Test case là gì?.....	62
3.5.2. Vai trò của test case .....	62
3.5.3. Cấu trúc cơ bản của test case .....	63
3.5.4. Những bước quan trọng để bạn viết được một test case hiệu quả.....	63
3.5.5. Test case kiểm thử web CIC.....	65
<b>3.6. Xây dựng phần mềm và ứng dụng.....</b>	<b>72</b>
3.6.1. Xây dựng chương trình kiểm thử tự động .....	72
3.6.2. Xuất báo cáo kết quả.....	73
<b>KẾT LUẬN .....</b>	<b>79</b>
<b>DANH MỤC TÀI LIỆU THAM KHẢO.....</b>	<b>80</b>

## DANH MỤC CÁC KÝ HIỆU, CÁC CHỮ VIẾT TẮT

Từ viết tắt	Giải thích	Tên tiếng anh
CIC	Trung tâm Thông tin Tín dụng Quốc gia Việt Nam- Ngân hàng Nhà nước Việt Nam	National credit information center of vietnam
BCTC	Báo cáo tài chính	
NHNN	Ngân hàng nhà nước	
TCTD	Tổ chức tín dụng	
TTTD	Thông tin tín dụng	
XHTD	Xếp hạng tín dụng	
DN	Doanh nghiệp	
HTML	Ngôn ngữ Đánh dấu Siêu văn bản	Hypertext Markup Language
XML	Ngôn ngữ đánh dấu mở rộng	Extensible Markup Language
POM	Mô hình đối tượng trang	Page Object Model
PM	Người quản lý dự án	Project Manager
BA	Đội phân tích kinh doanh	Business Analysis

QA	Người chịu trách nhiệm sản phẩm đầu ra	Quality Assurance.
QTP	Kiểm thử chức năng	Quicktest professional
IDE	Môi trường phát triển tổng hợp	Integrated development enviroment
GUI	Giao diện đồ họa người dùng	Graphic User Interface
SSL	Giao thức bảo mật	Secure Sockets Layer
WLV	Trình xác thực liên kết web	Web link Validator



## DANH MỤC CÁC HÌNH

Hình 1.1 Các giai đoạn kiểm thử .....	8
Hình 1.2 Sơ đồ bộ máy theo cấp của CIC .....	14
Hình 1.3 Quy trình kiểm thử phần mềm .....	16
Hình 2.1 Quy trình kiểm thử tự động .....	31
Hình 2.2 Mô hình kiểm thử tự động .....	33
Hình 2.3 Mô hình kiểm thử hướng mô - đun .....	34
Hình 2.4 Mô hình kiểm thử hướng thư viện .....	35
Hình 2.4 Mô hình kiểm thử hướng dữ liệu .....	36
Hình 2.5 Mô hình kiểm thử hướng từ khóa .....	37
Hình 2.6 Mô hình kiểm thử hướng kết hợp .....	38
Hình 2.7 Cấp độ kiểm thử tự động .....	39
Hình 3.1 Mô hình chuyển đổi kênh giao tiếp của CIC .....	48
Hình 3.2 Các tầng cấu trúc của dự án .....	72

## DANH MỤC CÁC BẢNG

<i>Bảng 2.1 Sự khác biệt giữa kiểm thử thủ công và kiểm thử tự động .....</i>	<i>26</i>
<i>Bảng 2.2 So sánh ưu nhược điểm của kiểm thử thủ công và kiểm thử tự động .....</i>	<i>28</i>
<i>Bảng 2.3 Kiểm tra mô hình kiểm thử hướng từ khóa .....</i>	<i>37</i>
<i>Bảng 2.4 Mô hình kiểm thử hướng kết hợp .....</i>	<i>38</i>
<i>Bảng 2.5 So sánh công cụ kiểm thử tự động katalon và selenium .....</i>	<i>45</i>
<i>Bảng 3.1 Tổng kết lỗi .....</i>	<i>77</i>

## I. MỞ ĐẦU

### 1. Tính cấp thiết của đề tài.

Công nghệ thông tin ngày nay đang phát triển nhanh chóng, kéo theo đó là hệ thống mạng và phần mềm cũng tăng lên về số lượng, quy mô. Nhưng cũng từ đó mà đã nảy sinh ra nhiều vấn đề về lỗi, hỏng hóc không đáng có gây ảnh hưởng lớn đến kinh tế, xã hội, uy tín... Những lỗi này không đơn thuần là khách quan mà do tự bản thân phần mềm lỗi mà do không được kiểm thử chặt chẽ, kỹ lưỡng trước khi đưa cho người dùng, ngoài yếu tố trên thì có thể do có hacker cố tính muốn xâm nhập hệ thống phần mềm để lấy thông tin khách hàng, như tài khoản ngân hàng, họ tên, điện thoại, email... hoặc do virus xâm nhập phá hoại. Từ đây thì ta thấy mặc dù phần mềm phát triển ngày càng cao, càng phức tạp nhưng vấn đề chất lượng vẫn là một vấn đề cần xem xét cẩn thận. Vì vậy, kiểm thử phần mềm là một công cuộc kiểm tra để cung cấp cho khách hàng sản phẩm hoặc dịch vụ được kiểm thử.

Đối với phần mềm cần phải thực hiện kiểm thử kỹ lưỡng nhằm ngăn chặn các lỗi hay hỏng hóc còn tiềm ẩn bên trong phần mềm mà ta chưa kịp nhận ra. Tuy nhiên vì phần mềm càng ngày càng lớn, với hàng nghìn mô - đun, có thể do cả một công ty lớn, hàng nghìn người phát triển vì thế để có thể kiểm thử được một phần mềm với quy mô lớn như vậy nếu chúng ta làm thủ công sẽ tốn rất nhiều công sức và thời gian mà chất lượng phần mềm chưa cao. Vì vậy cần có các hệ thống kiểm thử phần mềm một cách tự động cho phép ta thực hiện được các công việc một cách nhanh chóng và độ an toàn, chính xác cao nhất có thể giảm thời gian, con người và chất lượng phần mềm được nâng cao. Theo nhiều tính toán thì công việc kiểm thử đóng vai trò hết sức quan trọng trong quy trình phát triển phần mềm, nó đóng góp tới 40% tổng toàn bộ chi phí cho việc sản xuất phần mềm.

Trung tâm Thông tin Tín dụng Quốc gia Việt Nam trực thuộc Ngân hàng Nhà nước Việt Nam, thực hiện chức năng, nhiệm vụ đăng ký tín dụng; phân tích xếp hạng tín dụng, thu thập, xử lý, lưu trữ thông tin tín dụng; phòng tránh và hạn chế các rủi ro tín dụng; chấm điểm, xếp hạng tín dụng pháp nhân và thể nhân trên đất nước Việt

Nam, phục vụ yêu cầu quản lý nhà nước của Ngân hàng Nhà nước Việt Nam; cung cấp sản phẩm dịch vụ thông tin tín dụng theo quy định của Nhà nước và pháp luật.

Kho dữ liệu Thông tin tín dụng Quốc gia Việt Nam do Trung tâm Thông tin tín dụng Quốc gia Việt Nam quản lý lưu giữ thông tin của hơn 40 triệu khách hàng vay, với sự tham gia báo cáo thông tin của 100% các tổ chức tín dụng, hoạt động tại Việt Nam, hơn 1100 quỹ tín dụng nhân dân, các công ty bán lẻ cũng như các tổ chức khác trong và ngoài hệ thống Ngân hàng. Với số lượng dữ liệu lớn, số lượng khách hàng lớn, nên đòi hỏi phải có một hệ thống quản lý lớn, phần mềm được tạo ra phải hoạt động chính xác, kịp thời, nên việc tạo ra một phần mềm để quản lý trong Trung tâm Thông tin tín dụng Quốc gia Việt Nam luôn đòi hỏi cao, việc kiểm thử đầu ra của phần mềm phải được thực hiện chặt chẽ, chính xác, kịp thời.

Vì những lý do trên nên em đã lựa chọn đề tài “Nghiên cứu phương pháp kiểm thử tự động trong kiểm thử phần mềm của Trung tâm Thông tin Tín dụng Quốc gia Việt Nam” nhằm nghiên cứu các vấn đề về kiểm thử tự động và áp dụng kiểm thử tự động trong các dự án công nghệ thông tin.

## **2. Tổng quan nghiên cứu.**

Kiểm thử chương trình, phần mềm được thực hiện bởi nhiều người kiểm thử. Năm 1980 kiểm thử viên đã được sử dụng trong các dự án phần mềm, nhưng sau đó mới được coi là một nghề. Liên quan đến giai đoạn và mục đích khác nhau trong kiểm thử phần mềm, ứng dụng thì những vai trò khác nhau đã được thiết lập cho các nhà quản lý, nhóm trưởng kiểm thử, người phân tích kiểm thử, người thiết kế kiểm thử, người kiểm thử, người phát triển tự động hóa và quản trị viên kiểm thử.

Kiểm thử (testing) và việc gỡ lỗi (sửa lỗi debugging) lần đầu tiên được Glenford J. Myers đưa ra vào năm 1979. Năm 1988, Dave Gelperin và William C. Hetzel đã phân loại các giai đoạn và mục tiêu trong kiểm thử phần mềm theo trình tự sau:

Trước 1956: Giai đoạn hướng về việc kiểm soát lỗi.

1957-1978: Giai đoạn hướng về chứng minh lỗi.

1979-1982: Giai đoạn hướng về tính phá hủy của lỗi.

1983–1987: Giai đoạn hướng về đánh giá lỗi.

1988–2000: Giai đoạn hướng về việc phòng ngừa lỗi.

Kiểm thử phần mềm là việc kiểm tra nhằm cung cấp cho các bên liên quan thông tin về sản phẩm phần mềm, chất lượng của sản phẩm, ứng dụng sau khi kiểm thử. Kiểm thử cung cấp cho doanh nghiệp một cách nhìn độc lập về phần mềm để từ đó cho phép đánh giá và hiểu được những rủi ro trong quá trình triển khai phần mềm.

Trong các kỹ thuật kiểm thử ngoài giới hạn việc thực hiện một chương trình, ứng dụng với mục đích tìm lỗi phần mềm mà còn là một quá trình phê duyệt và xác minh chương trình, ứng dụng, sản phẩm để: đáp ứng được yêu cầu của khách hàng khi thiết kế và phát triển phần mềm, đáp ứng được mọi nhu cầu của các bên liên quan.

Tùy từng phương pháp, việc kiểm thử có thể được thực hiện bất cứ lúc nào trong quá trình xây dựng và phát triển phần mềm. Việc kiểm thử được thực hiện sau khi các yêu cầu được xác định và việc lập trình được hoàn tất nhưng trong Agile (là 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 thực hiện liên tục trong quá trình xây dựng và phát triển phần mềm.

Trong kiểm thử phần mềm, kiểm thử tự động (test automation) là việc sử dụng công cụ (tool) để thực hiện các kiểm tra và so kết quả thực tế với kết quả dự đoán. Kiểm thử tự động là có thể tự động hóa một số nhiệm vụ kiểm thử lặp đi lặp lại nhưng cần thiết trong một quá trình thử nghiệm, hay là các kiểm thử bổ sung nhưng sẽ khó thực hiện thủ công. Kiểm thử tự động rất quan trọng kiểm thử liên tục.

Nhiệm vụ của kiểm thử phần mềm, ứng dụng như kiểm thử hồi quy, có thể tốn thời gian và công sức để thực hiện. Ngoài ra, phương pháp kiểm thử thủ công không phải lúc nào cũng có hiệu quả trong việc tìm ra lỗi. Kiểm thử tự động thực hiện các loại kiểm thử một cách nhanh chóng, hiệu quả. Khi kiểm thử tự động được phát triển, chúng có thể chạy nhanh, hiệu quả và liên tục.

Tự động hóa trong kiểm thử làm tăng hiệu quả phần mềm tổng thể và đảm bảo chất lượng phần mềm. Có những công cụ hỗ trợ có thể thực thi hiệu quả các trường hợp thử nghiệm tự động và so sánh các kết quả thực tế và kết quả dự kiến. Theo

phương pháp này, kiểm thử tự động có thể đảm bảo sự thành thạo phần mềm mà không cần can thiệp thực hiện lặp lại một cách thủ công.

Một trong những đặc quyền lớn nhất trong kinh doanh phần mềm mà kiểm thử tự động mang lại là có thể được thực hiện bám sát thời gian độ chính xác tối đa. Những lợi ích mà kiểm thử tự động có thể mang lại: Phản hồi nhanh hơn, kết quả chính xác, giảm chi phí trong kinh doanh, kiểm tra cải thiện hiệu quả, mức độ kiểm tra tổng thể cao, có khả năng sử dụng lại các thử nghiệm tự động, triệt để trong kiểm thử, bảo mật thông tin.

### **3. Mục tiêu nghiên cứu.**

Đề tài luận văn nghiên cứu và tìm hiểu về cơ sở lý thuyết về kiểm thử và triển khai kiểm thử phần mềm tự động để giảm nguồn nhân lực kiểm thử và đảm bảo chất lượng phần mềm so với công việc kiểm thử thủ công.

Mục tiêu chính của đề tài là nghiên cứu các giai đoạn cần áp dụng công cụ kiểm thử tự động vào các phần mềm, đảm bảo phần mềm hạn chế lỗi đến mức tối thiểu.

### **4. Đối tượng và phạm vi nghiên cứu.**

Luận văn bao gồm đối tượng và phạm vi nghiên cứu như sau:

- Tìm hiểu về kiểm thử.
- Mô hình kiểm thử tự động.
- Thực hiện kiểm thử tự động trực tiếp cho phần mềm của Trung tâm Thông tin Tín dụng Quốc gia Việt Nam.

### **5. Phương pháp nghiên cứu.**

Phương pháp lý thuyết: Khảo sát, phân tích các tài liệu khoa học liên quan về các mô hình kiểm thử, kiểm thử tự động.

Phương pháp thực nghiệm: Xây dựng hệ thống kiểm thử tự động, sử dụng công nghệ để thử nghiệm và đánh giá hiệu quả của hệ thống kiểm thử tự động đối với chương trình đã lựa chọn.

## **II. NỘI DUNG**

Từ những mục tiêu nghiên cứu, phần mở đầu đã nói về tính cấp thiết, mục đích nghiên cứu, đối tượng nghiên cứu, phạm vi nghiên cứu, phương pháp nghiên cứu và những đóng góp của luận văn, ngoài những phần trên thì luận văn bao gồm các chương sau.

## CHƯƠNG I: TỔNG QUAN VỀ KIỂM THỬ PHẦN MỀM

Chương này chủ yếu tìm hiểu về kiểm thử phần mềm và thực tế của kiểm thử là như thế nào.

### 1.1. Tổng quan về kiểm thử phần mềm.

#### 1.1.1. Định nghĩa kiểm thử

Kiểm thử phần mềm có rất nhiều định nghĩa khác nhau được đề xuất bởi nhiều các nhân, tổ chức khác nhau

Kiểm thử phần mềm là hoạt động khảo sát thực tế sản phẩm phần mềm trong đúng môi trường dự định sẽ được triển khai nhằm cung cấp cho người có lợi ích liên quan những thông tin về chất lượng của sản phẩm hay dịch vụ phần mềm ấy. Mục đích của kiểm thử phần mềm là tìm ra các lỗi phần mềm nhằm đảm bảo hiệu quả hoạt động tối ưu của phần mềm trong nhiều ngành khác nhau. (Theo Wikipedia).

Kiểm thử phần mềm là quá trình khảo sát một hệ thống hay thành phần dưới những điều kiện xác định, quan sát và ghi lại các kết quả, và đánh giá một khía cạnh nào đó của hệ thống hay thành phần đó. (Theo Bảng chú giải thuật ngữ chuẩn IEEE của Thuật ngữ phần mềm- IEEE Standard Glossary of Software Engineering Terminology).

#### 1.1.2. Vai trò của kiểm thử.

##### 1.1.2.1. Vai trò của kiểm thử phần mềm như sau:

Kiểm thử phần mềm là hoạt động không thể thiếu vì không có bất cứ một sản phẩm phần mềm nào được tạo ra một cách hoàn hảo, không có lỗi. Một số lỗi sẽ không gây ảnh hưởng nhiều đến sản phẩm tuy nhiên có một số lỗi rất lớn cần phải được sửa chữa kịp thời.

Kiểm thử phần mềm có nhiệm vụ chỉ ra các lỗi và những sai sót hình thành trong các giai đoạn phát triển sản phẩm phần mềm.

Kiểm thử cần thiết vì nó đảm bảo độ tin cậy cho khách hàng và sự hài lòng của họ về phần mềm.

Kiểm thử quan trọng trong việc đảm bảo chất lượng của sản phẩm. Sản phẩm chất lượng được giao cho khách hàng giúp họ sử dụng hiệu quả hơn.

Kiểm thử phần mềm là công việc cần thiết để cung cấp các phần mềm chất lượng cho khách hàng hoặc ứng dụng phần mềm đòi hỏi chi phí bảo trì thấp hơn và do đó dẫn đến kết quả chính xác, nhất quán và đáng tin cậy hơn.

Thử nghiệm là cần thiết cho hiệu suất hiệu quả của ứng dụng phần mềm hoặc sản phẩm.

Điều quan trọng của kiểm thử là đảm bảo ứng dụng không dẫn đến bất kỳ lỗi nào vì nó có thể rất tốn kém trong tương lai hoặc trong các giai đoạn sau của quá trình phát triển phần mềm.

Kiểm thử là khâu quan trọng để đảm bảo hoạt động kinh doanh phần mềm.

Đảm bảo uy tín của chủ sở hữu sản phẩm đối với khách hàng.

#### 1.1.2.2. Vai trò và trách nhiệm của một người kiểm thử (Tester):

Trong giai đoạn lập kế hoạch và chuẩn bị thử nghiệm, người kiểm thử nên xem xét và đóng góp vào các kế hoạch thử nghiệm, cũng như phân tích, xem xét và đánh giá các yêu cầu và thông số kỹ thuật thiết kế. Người kiểm thử có thể tham gia hoặc là những người chính xác định các điều kiện thử nghiệm và tạo ra các thiết kế thử nghiệm, các trường hợp kiểm tra, các đặc tả thủ tục kiểm tra và dữ liệu thử nghiệm, và có thể tự động hóa.

Người kiểm thử thường thiết lập các môi trường thử nghiệm hoặc hỗ trợ quản trị hệ thống và nhân viên quản lý mạng.

Khi thực hiện thử nghiệm bắt đầu, số lượng người thử nghiệm thường tăng lên, bắt đầu với công việc cần thiết để thực hiện các bước thử nghiệm trong môi trường thử nghiệm.

Những kiểm thử viên thực hiện và đăng nhập các bài kiểm tra, đánh giá kết quả và các vấn đề tài liệu được tìm thấy.

Kiểm thử viên thường theo dõi thử nghiệm và môi trường thử nghiệm, sử dụng các công cụ cho tác vụ này và thu thập số liệu hiệu suất.



Trong suốt vòng đời kiểm thử, Kiểm thử viên xem xét công việc, hỗ trợ lẫn nhau, bao gồm các thông số kiểm tra, báo cáo lỗi và các kết quả kiểm tra.

### 1.1.3. Các mục đích của kiểm thử phần mềm

Mục tiêu chính của Kiểm thử phần mềm như sau:

Mục đích trực tiếp:

Xác định và tìm kiếm được nhiều lỗi nhất có thể trong phần mềm được kiểm thử.

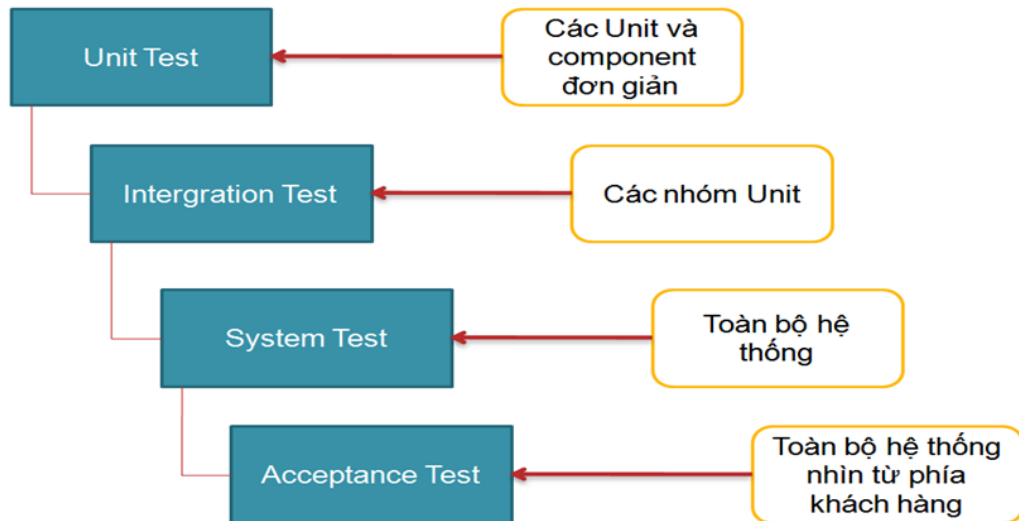
Sau khi sửa lỗi đã được phát hiện cần phải xác định và kiểm tra loại, giúp cho phần mềm được kiểm thử đạt đến mức độ chấp nhận được về chất lượng phần mềm.

Mục đích gián tiếp:

Biên kịch một bản ghi về lỗi phần mềm để sử dụng trong công tác phòng chống lỗi (bằng các hành động khắc phục và ngăn ngừa).

### 1.1.4. Các giai đoạn kiểm thử phần mềm.

Các giai đoạn trong kiểm thử phần mềm bao gồm 4 giai đoạn sau:



**Hình 1.1 Các giai đoạn kiểm thử**

Việc hiểu được các giai đoạn kiểm thử giúp các bạn khi thực hiện công việc liên quan sẽ hiểu rõ để đưa ra được các cách thức kiểm thử, phát hiện lỗi.

#### a. Kiểm thử mức đơn vị (Unit Test)

Kiểm thử đơn vị là kiểm thử trên từng đơn vị nhỏ nhất của phần mềm mà chúng ta có thể thực hiện kiểm thử. Đó có thể là một hàm (function), một thủ tục (stored procedure).

Giai đoạn này nếu phát hiện ra có lỗi, việc xác định nguyên nhân và khắc phục tương đối dễ chỉ khoanh vùng một đơn thể đơn vị đang kiểm tra. Nguyên lý đúc kết từ thực tiễn: thời gian tiêu tốn cho kiểm thử đơn vị tiết kiệm hơn rất nhiều ở các mức kiểm tra sau đó. Bên cạnh đó chi phí cho kiểm thử và sửa lỗi cũng là thấp nhất. Việc thực hiện kiểm thử đơn vị thường là lập trình viên. Các kịch bản kiểm thử (testcase) ở giai đoạn này giữ lại để tái sử dụng cho các dự án khác.

### **b. Kiểm thử tích hợp (Integration Test)**

Kiểm thử đơn vị là kiểm thử riêng lẻ thì kiểm thử tích hợp lại là kết hợp lại với nhau và kiểm tra sự giao tiếp giữa chúng. Kiểm thử tích hợp sẽ kết hợp các thành phần của một ứng dụng và kiểm tra như một ứng dụng đã hoàn thành.

Mục tiêu chính của kiểm thử tích hợp:

- Tìm ra lỗi giao tiếp xảy ra giữa các đơn vị với nhau

Tích hợp các đơn vị (unit) đơn lẻ thành các hệ thống nhỏ (subsystem) và hoàn chỉnh hệ thống (system) để chuẩn bị cho kiểm tra ở mức hệ thống (System Test).

- Tích hợp dần từng đơn vị (unit).

Một đơn vị sau khi được tích hợp vào một nhóm các đơn vị khác đã tích hợp trước đó thì lúc này chỉ cần kiểm tra giao tiếp của đơn vị mới thêm vào với nhóm các đơn vị đã được tích hợp trước đó. Việc này làm cho số lượng trường hợp kiểm thử sẽ giảm đi rất nhiều, sai sót sẽ giảm đi đáng kể.

Người thực hiện kiểm thử (test) tích hợp thường là lập trình viên.

### **c. Kiểm thử hệ thống (System test)**

Mục đích của kiểm thử hệ thống là kiểm tra về thiết kế và toàn bộ hệ thống (sau khi tích hợp) có thỏa mãn yêu cầu đặt ra hay không?

Trước khi kiểm thử hệ thống thì phải thực hiện kiểm thử đơn vị và kiểm thử tích hợp để bảo đảm sự tương tác giữa chúng hoạt động chính xác.

Đặc điểm kiểm thử hệ thống: Tốn rất nhiều công sức và mất nhiều thời gian.

Bởi vì trọng tâm của kiểm thử hệ thống là đánh giá về hoạt động, thao tác, sự tin cậy và những yêu cầu khác liên quan đến chất lượng của toàn hệ thống.

Điểm khác biệt giữa kiểm thử hệ thống và kiểm thử tích hợp.

- Kiểm thử hệ thống (System Test): Chú trọng đến các hành vi và lỗi trên toàn bộ hệ thống.
- Kiểm thử tích hợp (Integration Test): Chú trọng sự giao tiếp giữa các đơn thể (Unit) khi chúng làm việc cùng với nhau.

Người thực hiện kiểm thử hệ thống thường là người kiểm thử. Một hoặc nhóm kiểm thử hoàn toàn độc lập so với nhóm phát triển dự án.

**d. Kiểm thử chấp nhận sản phẩm (Acceptance Test)** hay còn gọi là kiểm thử nghiệm thu.

Kiểm thử nghiệm thu hết sức quan trọng. Giai đoạn này thường được khách hàng thực hiện (hoặc ủy quyền cho nhóm thứ ba thực hiện) để kiểm tra xem sản phẩm có đáp ứng được yêu cầu đặt ra của mình trước đó hay không. Trong mọi trường hợp, các phép kiểm thử trong kiểm thử nghiệm thu và kiểm thử hệ thống gần như giống nhau, nhưng cách thức và bản chất lại khác nhau.

Thực tế cho thấy, Khách hàng phải là người tham gia vào quá trình phát triển phần mềm, nếu khách hàng không quan tâm và không tham gia thì kết quả của kiểm thử nghiệm thu sẽ bị sai lệch, cho dù phần mềm đã trải qua tất cả các bước kiểm tra trước đó.

## **1.2. Vai trò của kiểm thử trong dự án của Trung tâm Thông tin Tín dụng Quốc gia Việt Nam.**

### ***1.2.1. Khái quát về Trung tâm Thông tin Tín dụng Quốc gia Việt Nam***

#### **1.2.1.1. Quá trình hình thành và phát triển.**

Trung tâm Thông tin Tín dụng Quốc gia Việt Nam (CIC) là đơn vị sự nghiệp công lập trực thuộc Ngân hàng Nhà nước Việt Nam. Tiền thân của CIC là một phòng nhỏ trực thuộc Vụ tín dụng, CIC hiện tại đã trở thành một tổ chức tín dụng hàng đầu

khu vực với hệ thống cơ sở dữ liệu đầy đủ, hệ thống công nghệ hiện đại, chuyên nghiệp, hoạt động dựa trên nguyên tắc chia sẻ thông tin.

Các mốc thời gian quan trọng trong lịch sử phát triển của CIC [2]:

- 9/1992: Phòng Thông tin phòng ngừa rủi ro- tiền thân của Trung tâm thông tin tín dụng CIC được thành lập trực thuộc Vụ Tín dụng – NHNN.

- 4/1995: Phòng Thông tin phòng ngừa rủi ro được đổi tên và tách thành Trung tâm thông tin tín dụng trực thuộc Vụ Tín dụng – NHNN.

- 02/1999: CIC trở thành một đơn vị sự nghiệp thuộc NHNN theo Quyết định số 68/1999/QĐ-NHNN ngày 27/2/1999 của Thống đốc NHNN, tổ chức lại CIC thuộc Vụ Tín dụng.

- 9/2007: Kỷ niệm 15 năm hoạt động tín dụng và được Thủ tướng Chính phủ trao tặng Bằng khen.

- 12/2008: CIC được thành lập lại theo Quyết định số 3289/QĐ-NHNN ngày 31/12/2008 của Thống đốc NHNN trở thành một tổ chức sự nghiệp công lập thuộc NHNN.

- 2012: Kỷ niệm 20 năm hoạt động thông tin tín dụng.

- 3/2014: Đổi tên thành Trung tâm thông tin tín dụng Quốc gia Việt Nam theo Quyết định số 342/QĐ – NHNN ngày 26/2/2014 của thống đốc NHNN.

- 2019: Kỷ niệm 20 năm thành lập Trung tâm thông tin tín dụng Quốc gia Việt Nam và đón nhận Huân chương lao động hạng Nhất của Chủ tịch nước

### ***1.2.2. Chức năng và nhiệm vụ của Trung tâm thông tin tín dụng Quốc gia Việt Nam***

#### **a. Chức năng của Trung tâm thông tin tín dụng Quốc gia Việt Nam**

Trung tâm Thông tin Tín dụng Quốc gia Việt Nam (CIC) là đơn vị dự toán độc lập, thực hiện chế độ tự chủ, tự chịu trách nhiệm về tài chính, tự đảm bảo toàn bộ chi phí hoạt động thường xuyên theo quy định của Nhà nước và của pháp luật. Chức

năng của CIC là thực hiện đăng ký tín dụng quốc gia; thu nhận, xử lý, lưu trữ, phân tích, dự báo thông tin tín dụng; phòng ngừa và hạn chế rủi ro tín dụng; chấm điểm, xếp hạng tín dụng pháp nhân và thể nhân trên lãnh thổ Việt Nam phục vụ cho yêu cầu quản lý nhà nước của Ngân hàng Nhà nước, cung ứng sản phẩm dịch vụ thông tin tín dụng theo quy định của Ngân hàng Nhà nước và của pháp luật.

Trung tâm thông tin tín dụng Quốc gia Việt Nam đảm bảo cung cấp thông tin một cách độc lập, khách quan, công bằng, minh bạch, đúng hạn và đúng đối tượng. Cung cấp hệ thống sản phẩm, dịch vụ đa dạng, phong phú cho Ngân hàng Nhà nước, tổ chức tín dụng và thể nhân khác và là đầu mối thông tin tín dụng quốc gia.

### **b. Nhiệm vụ của Trung Tâm Thông Tin Tín Dụng Quốc gia Việt Nam**

Những nhiệm vụ và quyền hạn của CIC:

- Xây dựng, trình Thống đốc ký ban hành các văn bản về hoạt động thông tin tín dụng; tổ chức triển khai thực hiện sau khi được Thống đốc ký ban hành.

- Xây dựng, trình Thống đốc phê duyệt các đề án, dự án, kế hoạch, chương trình về phát triển Trung tâm Thông tin tín; tổ chức triển khai và thực hiện sau khi được Thống đốc phê duyệt.

- Lập và trình Thống đốc phê duyệt danh mục, tiêu chuẩn về thông tin tín dụng; tổ chức thu thập, xử lý, lưu trữ, quản lý kho dữ liệu quốc gia về thông tin tín dụng.

- Tổ chức, thu thập, mua thông tin tín dụng từ các nguồn trong và ngoài nước; cung cấp đầy đủ, kịp thời, các sản phẩm thông tin tín dụng cho Ngân hàng nhà nước, các tổ chức tín dụng và các tổ chức khác theo quy định nhà nước và pháp luật

- Thực hiện phân tích, xếp hạng và chấm điểm tín dụng đối với các tổ chức, cá nhân có quan hệ vay vốn của các tổ chức tín dụng

- Thực hiện dịch vụ thông tin tín dụng, cảnh báo sớm rủi ro tín dụng theo quy định của pháp luật

- Được liên doanh, liên kết với các tổ chức thể nhân hoặc thuê chuyên gia trong và ngoài nước để thực hiện nhiệm vụ được giao theo quy định của Thống đốc và pháp luật

- Tổ chức các hoạt động nghiên cứu, hội thảo, trao đổi kinh nghiệm về lĩnh vực thông tin tín dụng, phối hợp với Trường bồi dưỡng cán bộ ngân hàng và các tổ chức liên quan tổ chức đào tạo, bồi dưỡng về chuyên môn, nghiệp vụ cho đội ngũ cán bộ, viên chức, làm công tác thông tin tín dụng của Trung tâm thông tin tín dụng và của ngành Ngân hàng

- Quản lý biên chế và sử dụng cán bộ, viên chức

- Thực hiện chế độ báo cáo định kỳ, đột xuất theo quy định của Ngân hàng nhà nước và pháp luật

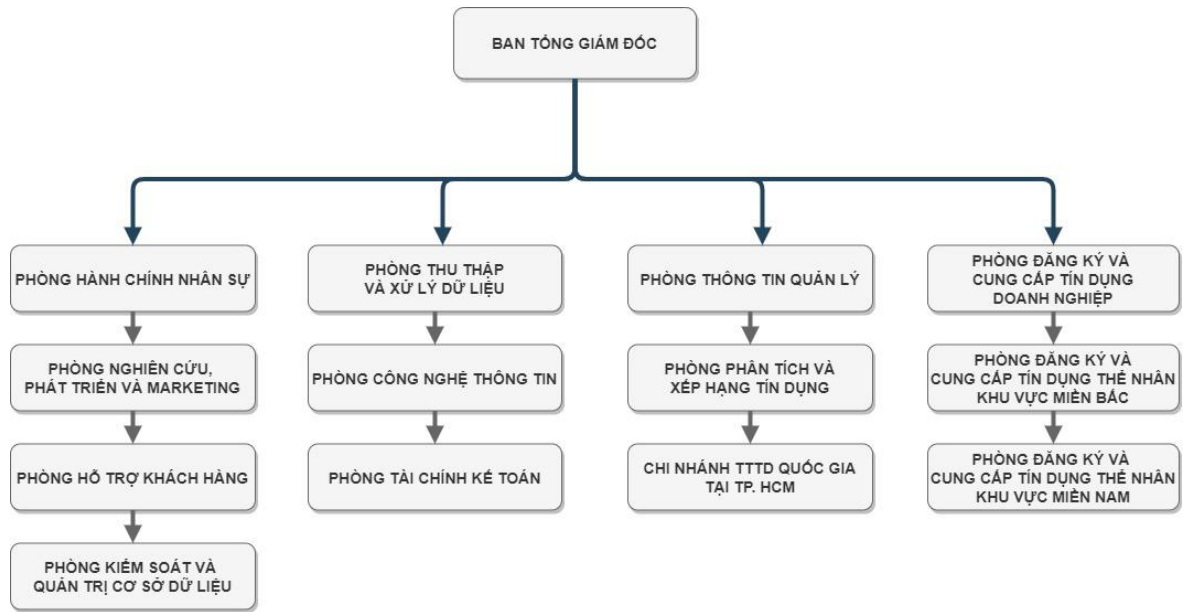
- Phối hợp kết hợp với các đơn vị chức năng thực hiện hợp tác quốc tế, tiếp nhận và quản lý các dự án trợ giúp kỹ thuật của các tổ chức quốc tế

- Quản lý và chi tiêu tài chính, tài sản của CIC theo đúng quy định của pháp luật

- Thực hiện nhiệm vụ khác do cấp trên giao

### ***1.2.3. Bộ máy quản lý của Trung tâm thông tin tín dụng Quốc gia Việt Nam***

#### **a. Sơ đồ bộ máy tổ chức theo cấp của CIC**



**Hình 1.2 Sơ đồ bộ máy theo cấp của CIC**

(Nguồn: Văn phòng CIC)

### **b. Chức năng và nhiệm vụ của các phòng ban trong bộ máy quản lý**

**Ban Tổng giám đốc:** Là ban tổ chức và thực hiện nhiệm vụ của CIC, chịu trách nhiệm trước Thống đốc và pháp luật về toàn bộ hoạt động của CIC; quyết định chương trình, kế hoạch của CIC, ký các văn bản thuộc thẩm quyền theo quy định.

**Nhiệm vụ của các phòng ban trong CIC:**

- Văn phòng: Nghiên cứu, dự thảo và trình giám đốc các văn bản về chế độ, công tác hành chính, đối nội, đối ngoại của CIC; xây dựng kế hoạch công tác của Trung tâm, ...

- Phòng nghiên cứu phát triển và marketing: Nghiên cứu các sản phẩm mới, chỉnh sửa sản phẩm hiện tại cho phù hợp với điều kiện hiện tại, Nghiên cứu và đưa ra các văn bản hướng dẫn các tổ chức tín dụng trong việc thực hiện các văn bản của Ngân hàng Nhà nước ban hành, ...

- Phòng Tài chính – kế toán: Tham mưu cho ban tổng giám đốc xây dựng các chế độ quản lý tài chính: Xây dựng quy chế chi tiêu nội bộ phù hợp với cơ chế, quy

chế tài chính của Nhà nước đối với đơn vị sự nghiệp có thu; Lập Kế hoạch tài chính, mua sắm - sửa chữa tài sản cố định, thiết bị máy móc phục vụ công việc.

- Phòng Thu thập và Xử lý dữ liệu: Theo dõi, đôn đốc các tổ chức tín dụng thực hiện báo cáo đầy đủ, chính xác, đúng thời hạn các nội dung thông tin theo chỉ tiêu, mẫu biểu thông tin quy định tại Quy chế hoạt động thông tin tín dụng.

- Phòng đăng ký và cung cấp tin Doanh nghiệp: Nghiên cứu, xây dựng quy trình cấp tin doanh nghiệp trong nước và ngoài nước. Thực hiện việc phân tích, xếp loại tín dụng của doanh nghiệp, lập báo cáo cung cấp cho NHNN, cho các tổ chức tín dụng và các tổ chức, thể nhân khác khi có yêu cầu. Định kì theo quý hoặc năm tổng hợp kết quả xếp loại tín dụng doanh nghiệp để phân tích, đánh giá theo ngành, vùng kinh tế; trên cơ sở đó CIC đưa ra dự báo và cảnh báo rủi ro tín dụng.

- Phòng cấp đăng ký và cung cấp tin tín dụng thể nhân Miền Nam: Nghiên cứu, xây dựng quy trình cấp tin thể nhân khu vực Miền Nam; Cung cấp sản phẩm báo cáo tín dụng khách hàng thể nhân khu vực Miền Nam cho các TCTD; Cung cấp sản phẩm khách hàng vay cho khách hàng thể nhân khu vực miền nam;

- Phòng cấp đăng ký và cung cấp tin tín dụng thể nhân Miền Bắc: Nghiên cứu, xây dựng quy trình cấp tin thể nhân khu vực Miền Bắc. Cung cấp sản phẩm báo cáo tín dụng khách hàng thể nhân khu vực Miền Bắc cho các TCTD.

- Phòng Thông tin quản lý: Tham mưu cho ban tổng giám đốc trong việc xuất bản, phát hành các ấn phẩm Thông tin Tín dụng phù hợp với hoạt động của CIC và quy định của Ngân hàng Nhà nước. Cung cấp các sản phẩm cảnh báo tín dụng cho TCTD và NHNN. Báo cáo phục vụ cho NHNN và các đơn vị nhà nước

- Phòng phân tích và xếp hạng tín dụng: Cung cấp các báo cáo về mức hạn tín dụng của các doanh nghiệp (dựa trên báo cáo tài chính). Nghiên cứu quy trình mới về xếp hạng tín dụng doanh nghiệp

- Phòng Công nghệ thông tin: Tham mưu cho ban tổng Giám đốc và hoạch định chiến lược phát triển hệ thống công nghệ thông tin trong các hoạt động thông tin



tín dụng phù hợp với sự nghiệp Ngành. Xây dựng hệ thống bảo mật, đảm bảo an toàn an ninh mạng và phần mềm; quản lý vận hành mạng và hệ thống máy chủ đảm bảo hoạt động ổn định, phục vụ nhu cầu khai thác thông tin của các Vụ, Cục NHTW, các TCTD

- Phòng cấp tin Hỗ trợ khách hàng: Nghiên cứu, xây dựng quy trình hỗ trợ, tư vấn cho các TCTD. Giải đáp thắc mắc, đối chiếu số liệu và báo cho các phòng liên quan kiểm tra bản trả lời tin cho TCTD

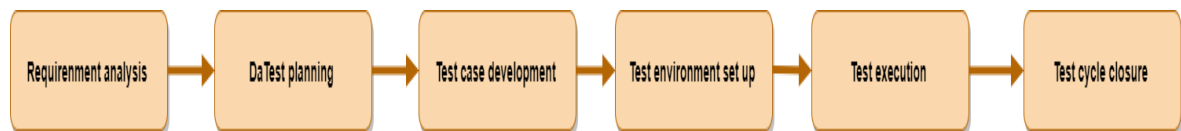
- Chi nhánh Tp. HCM: Đơn đốc các TCTD thực hiện theo đúng quy định về hoạt động thông tin tín dụng khu vực Miền Nam. Thực hiện thu thập và xử lý thông tin với các Quỹ khu vực Miền Nam. Giải đáp thắc mắc, đối chiếu số liệu bản trả lời tin khi TCTD phản hồi

- Phòng Kiểm soát và Quản trị cơ sở dữ liệu: Kiểm tra tính đúng đắn, chính xác của các báo cáo thông tin, dữ liệu trước khi cập nhật dữ liệu vào kho của CIC theo quy định; Điều chỉnh sai sót dữ liệu khi có công văn từ các tổ chức tín dụng.

### 1.3. Quy trình kiểm thử, chất lượng phần mềm và việc đảm bảo chất lượng phần mềm đầu ra.

#### 1.3.1. Định nghĩa quy trình kiểm thử phần mềm

Quy trình kiểm thử phần mềm là xác định các giai đoạn trong kiểm thử phần mềm. Tuy nhiên, không có vòng đời kiểm thử phần mềm (software testing life cycle – STLC) tiêu chuẩn cố định nào trên thế giới, nhưng về cơ bản quy trình kiểm thử bao gồm những giai đoạn sau:



**Hình 1.3 Quy trình kiểm thử phần mềm**

- Phân tích yêu cầu (Requirement analysis)
- Lập kế hoạch kiểm thử (Test planning)
- Thiết kế kịch bản kiểm thử (Test case development)

- Thiết lập môi trường kiểm thử (Test environment set up)
- Thực hiện kiểm thử (Test execution)
- Đóng chu trình kiểm thử (Test cycle closure)

Các giai đoạn kiểm thử phần mềm được thực hiện một cách tuần tự từng bước. Mỗi giai đoạn kiểm thử sẽ có những mục tiêu khác nhau, nhưng mục tiêu cuối cùng vẫn là đảm bảo sản phẩm phần mềm tốt nhất.

### ***1.3.2. Phân tích quy trình kiểm thử phần mềm***

#### **a. Phân tích yêu cầu (Test cycle closure)**

- Đầu vào trong giai đoạn phân tích yêu cầu:

Trong giai đoạn này bao gồm các tài liệu: mô tả yêu cầu, thiết kế hệ thống, yêu cầu của khách hàng và những tiêu chí mà sản phẩm chấp nhận được, bản nguyên mẫu (prototype) của khách hàng yêu cầu (nếu có) ...

- Hoạt động của giai đoạn phân tích yêu cầu:

Hoạt động phân tích yêu cầu là do đội đảm bảo chất lượng (Quality Assurance - QA) sẽ thực hiện: đọc hiểu, nghiên cứu và phân tích cụ thể các yêu cầu trong tài liệu đặc tả của dự án hoặc tài liệu yêu cầu của khách hàng. Từ hoạt động này, Đội QA là đội phải nắm được các yêu cầu mà dự án đưa ra bao gồm việc yêu cầu kiểm thử chức năng và phi chức năng.

Trong quá trình phân tích và nghiên cứu tài liệu, nếu có phát sinh mới hay đề xuất giải quyết yêu cầu, Đội QA sẽ phải đưa ra câu hỏi đối với các bên liên quan như người quản lý dự án (Project Manager - PM), đội phân tích kinh doanh (Business Analysis - BA), trưởng nhóm (team leader) và khách hàng để tất cả những bên liên quan hiểu chính xác hơn về yêu cầu của sản phẩm. Những câu hỏi này sẽ được lưu trữ vào file câu hỏi và trả lời (Question and Answer - Q&A). Các câu hỏi thường được đưa ra dưới dạng có/không (Yes/No) hoặc đưa ra các lựa chọn để tiết kiệm thời gian trả lời. Trường hợp, đối với những khách hàng không có chuyên môn về công nghệ, khi trao đổi với khách hàng thì không thể trao đổi với ngôn ngữ chuyên ngành. Chính QA sẽ là những người hỗ trợ và đưa ra giải pháp phù hợp cho khách hàng lựa chọn.

- Đầu ra của giai đoạn phân tích yêu cầu:

Đầu ra của giai đoạn phân tích yêu cầu bao gồm tài liệu chứa các câu hỏi và câu trả lời liên quan đến chức năng và nghiệp vụ của hệ thống, tài liệu báo cáo có tính khả thi, phân tích những rủi ro có thể xảy ra trong việc kiểm thử phần mềm.

#### **b. Lập kế hoạch kiểm thử (Test planning)**

- Đầu vào của giai đoạn lập kế hoạch kiểm thử:

Giai đoạn lập kế hoạch kiểm thử là các tài liệu đặc tả đã được thông qua các câu hỏi và trả lời được đưa ra trong giai đoạn phân tích yêu cầu, tài liệu báo cáo tính khả thi, phân tích rủi ro của việc kiểm thử phần mềm.

- Hoạt động của giai đoạn lập kế hoạch kiểm thử:

Dựa vào các tài liệu được cung cấp và cập nhật mới nhất trong giai đoạn phân tích yêu cầu, quản lý kiểm thử là người lập kế hoạch kiểm thử cho cả đội QA.

- Lập kế hoạch kiểm để xác định một số yếu tố quan trọng sau:

Xác định phạm vi của dự án: Thời gian thực hiện dự án là bao lâu? Thực hiện những công việc gì trong thời gian xác định? Từ đó đưa ra lịch trình thực hiện cho từng công việc nhỏ cho phù hợp với toàn bộ đội dự án.

- Phương pháp tiếp cận:

Phương pháp tiếp cận thì phải dựa vào nhiều yếu tố, ví dụ: Thời gian phù hợp cho việc kiểm thử phần mềm dài hay ngắn, yêu cầu chất lượng từ phía khách hàng Cao hay thấp Công nghệ, kỹ thuật sử dụng để phát triển ứng dụng này là gì? Mới hay cũ. Lĩnh vực của hệ thống, sản phẩm đang được kiểm thử là gì?...

Từ đó, Quản lý kiểm thử (test manager) có thể đưa ra những phương pháp và kế hoạch phù hợp nhất cho cả quá trình thực hiện dự án, tiêu chí chấp nhận của sản phẩm và kịp tiến độ với các mốc thời gian bàn giao.

- Xác định các nguồn lực:

- Con người: Số lượng người tham gia dự án, phân công công việc hợp lý.
- Thiết bị: số lượng thiết bị, môi trường để thực hiện kiểm thử: số lượng máy chủ (server), phiên bản (version), máy tính, mobile để thực hiện.

- Lập kế hoạch thiết kế công việc kiểm thử.

Thống kê các chức năng cần kiểm thử.

Xác định những điều kiện bắt đầu: xác định những điều kiện tối thiểu để bắt đầu hoạt động kiểm thử cho từng chức năng.

Xác định điều kiện kết thúc: khi có những điều kiện nào thì sẽ kết thúc việc kiểm thử.

Đầu ra của giai đoạn lập kế hoạch kiểm thử:

Đầu ra của giai đoạn lập kế hoạch kiểm bao gồm các tài liệu kế hoạch kiểm thử (test plan), ước tính quy mô kiểm thử (test estimation), lịch kiểm thử (test schedule).

### **c. Thiết kế kịch bản kiểm thử (Test case development)**

- Đầu vào của giai đoạn thiết kế kịch bản kiểm thử:

Giai đoạn thiết kế kịch bản kiểm thử là kế hoạch kiểm thử (test plan), quy mô kiểm thử (test estimation), lịch kiểm thử (test schedule), các tài liệu đặc tả đã được cập nhật.

- Hoạt động của giai đoạn thiết kế kịch bản kiểm thử:

Xem tài liệu: đây là bước đầu tiên mà các kiểm thử cần xem lại tất cả các tài liệu để xác định công việc cần làm, chức năng nào cần kiểm thử, chức năng nào không cần kiểm thử lại. Từ đó, có thể tiết kiệm thời gian mà vẫn đưa ra được một kịch bản kiểm thử đầy đủ và hiệu quả.

Viết kịch bản kiểm thử (test case), người kiểm thử bắt đầu vào việc viết test case. Kịch bản kiểm thử cần bao phủ được tất cả các trường hợp kiểm thử có thể xảy ra và đáp ứng đầy đủ các tiêu chí của sản phẩm. Người kiểm thử cũng cần đánh giá mức độ ưu tiên cho từng kịch bản kiểm thử.

- Chuẩn bị dữ liệu kiểm thử:

Đội kiểm thử cũng cần chuẩn bị trước các dữ liệu kiểm thử cho các trường hợp như kiểm thử dữ liệu (test data), bản kiểm thử chi tiết (test script).

Xem kịch bản kiểm thử (test case): Đội kiểm thử và trưởng nhóm kiểm thử cũng

cần xem lại kịch bản kiểm thử đã được tạo để có thể bổ sung, hỗ trợ lẫn nhau nhằm tránh những sai sót trong thiết kế kịch bản kiểm thử và rủi ro phát sinh về sau.

Đầu ra của giai đoạn thiết kế kịch bản kiểm thử:

Sau khi hoàn thành thiết kế kịch bản kiểm thử, đội kiểm thử sẽ có các tài liệu bao gồm: thiết kế kiểm thử (test design), kịch bản kiểm thử (test case), kiểm thử danh mục (check list), kiểm thử dữ liệu (test data), kịch bản kiểm thử tự động (test automation script).

#### **d. Thiết lập môi trường kiểm thử (Test environment set up)**

Đầu vào của giai đoạn thiết lập môi trường kiểm thử:

Đầu vào của giai đoạn thiết lập môi trường kiểm thử là kế hoạch kiểm thử, dữ liệu kiểm thử.

Hoạt động của giai đoạn thiết lập môi trường kiểm thử:

Việc cài đặt môi trường kiểm thử là rất quan trọng trong phát triển phần mềm. Môi trường kiểm thử dựa trên những yêu cầu của khách hàng, hay đặc thù của sản phẩm: như server, client, network...

Người kiểm thử phần mềm cần chuẩn bị một vài kịch bản test để kiểm tra môi trường cài đặt.

Đầu ra của việc thiết lập môi trường kiểm thử:

Đầu ra thiết lập môi trường kiểm thử là môi trường đã được cài đặt đúng theo đúng yêu cầu, sẵn sàng cho việc kiểm thử.

#### **e. Thực hiện kiểm thử (Test execution)**

- Đầu vào của thực hiện kiểm thử:

Tài liệu đầu vào của giai đoạn thực hiện kiểm thử này là test plan, test design, test case, check list, test data, test automation script.

- Hoạt động của giai đoạn thực hiện kiểm thử:

Thực hiện theo kịch bản kiểm thử như thiết kế.

So sánh với kết quả sau khi báo cáo các lỗi xảy ra lên công cụ quản lý lỗi, theo dõi trạng thái của lỗi đến khi được sửa thành công.

Thực hiện kiểm tra lại để kiểm chứng các bug đã được sửa và kiểm thử hồi quy khi có sự thay đổi liên quan.

Trong quá kiểm thử, người kiểm thử cũng có thể hỗ trợ, đề xuất cho đội dự án những giải pháp hợp lý và phối hợp thực hiện để công việc đạt hiệu quả cao.

- Phân tích và đo tiến độ:

Người kiểm thử cũng cần kiểm soát chặt chẽ tiến độ công việc bằng cách so sánh tiến độ thực tế với kế hoạch, nếu tiến độ chậm thì cần phải điều chỉnh sao cho kịp tiến độ dự án, nếu nhanh cũng cần điều chỉnh vì có thể việc lên kế hoạch chưa sát với thực tế dự án. Từ đó có thể điều chỉnh lại test plan để phù hợp với tiến độ dự án đưa ra.

Báo cáo thường xuyên cho người quản lý dự án và khách hàng về tình hình thực hiện dự án: Cung cấp thông tin trong quá trình kiểm thử những chức năng nào đã làm được, những chức năng nào chưa làm được (lý do), báo cáo tỷ lệ phần trăm công việc hoàn thành và chưa hoàn thành, báo cáo các trường hợp phát sinh, đảm bảo tiến độ công việc.

- Đầu ra của giai đoạn thực hiện kiểm thử:

Đầu ra của giai đoạn này là kết quả kiểm thử, danh sách các lỗi tìm được.

#### **f. Đóng chu trình kiểm thử (Test cycle closure)**

- Đầu vào đóng chu trình kiểm thử:

Đầu vào của giai đoạn này là bao gồm những tài liệu liên quan đã được tổng hợp, ghi chép và hoàn thiện trong suốt quy trình kiểm thử: tài liệu phân tích đặc tả yêu cầu, kế hoạch kiểm thử, kết quả kiểm thử, tài liệu QA...

- Hoạt động của đóng chu trình kiểm thử:

Là giai đoạn cuối cùng trong một quy trình kiểm thử phần mềm.

Ở phần này, đội kiểm tra chất lượng phần mềm thực hiện tổng kết, báo cáo kết quả về việc thực thi kịch bản kiểm thử, bao nhiêu lỗi vào nhiều case không lỗi, bao nhiêu case đã được sửa, mức độ nghiêm trọng của lỗi, lỗi còn ở chức năng nào, người lập trình nào làm nhiều lỗi.

Đánh giá các tiêu chí hoàn thành như phạm vi kiểm tra, chất lượng, thời gian, chi phí, mục tiêu kinh doanh.

- Đầu ra của đóng chu trình kiểm thử:

Đầu ra của giai đoạn này bao gồm các tài liệu: báo cáo kiểm thử, kết quả kiểm thử.

#### **1.4. Vai trò của kiểm thử đối với Trung tâm Thông tin Tín dụng Quốc gia Việt Nam.**

Số hóa trong ngành ngân hàng nói chung và Trung tâm Thông tin Tín dụng Quốc gia Việt Nam nói riêng là xu hướng tất yếu để tồn tại và phát triển trong thời đại mới trước sự tác động của cuộc cách mạng công nghiệp (CMCN) 4.0 và sự lớn mạnh của công nghệ tài chính (Fintech).

Bên cạnh những cơ hội từ ứng dụng công nghệ 4.0 (như giảm chi phí, gia tăng lợi nhuận, đưa lại nhiều tiện ích cho khách hàng), những tác động của cuộc cách mạng công nghiệp 4.0 tới ngành Ngân hàng khiến rủi ro an ninh mạng như lừa đảo, hacker... trong lĩnh vực này ngày càng trở nên lớn hơn và thường trực hơn, do sự kết nối mở, liên tục, đa chiều, phức tạp. Những lo ngại về vấn đề bảo mật trong các giao dịch thanh toán ngày càng tăng. Thiệt hại liên quan đến lĩnh vực này tăng theo cấp số nhân. Nhận diện những rủi ro công nghệ trong hoạt động ngân hàng số để có những giải pháp ngăn ngừa, hạn chế là cần thiết và đang được ngành Ngân hàng và CIC đặc biệt quan tâm.

Với CIC

Trong 9 tháng đầu năm 2020, Trung tâm Thông tin tín dụng Quốc gia Việt Nam (CIC) đã thu thập được thông tin từ 123 đầu mối tổ chức tín dụng (TCTD) và chi nhánh ngân hàng nước ngoài, trên 1.160 quỹ tín dụng nhân dân, các tổ chức tài chính vi mô chính thức và 52 tổ chức tự nguyện tham gia hoạt động thông tin tín dụng (TTTD). Trong Quý III/2020, độ phủ TTTD tiếp tục được cải thiện so với tháng trước, tăng trên 766.000 khách hàng vay (trên 2.241.793 hồ sơ vay mới), nâng tổng số khách hàng trong kho dữ liệu TTTD quốc gia lên trên 44.600.000 khách hàng. CIC cũng tiếp tục duy trì thu thập ổn định các loại thông tin về dư nợ, hợp đồng tín dụng, tài

sản bảo đảm, thẻ tín dụng, tài chính doanh nghiệp, đầu tư trái phiếu doanh nghiệp từ các TCTD, đảm bảo chất lượng thông tin được duy trì, đều đạt từ trên 98% đến 100%, điều này góp phần cải thiện khả năng tiếp cận tín dụng của khách hàng vay.

Tính riêng trong Quý III/2020, CIC đã có trên 41.000 khách hàng cá nhân đăng ký tài khoản trực tuyến, trong đó, có 23.718 khách hàng được phê duyệt. Qua đó, nâng cao chất lượng kho dữ liệu, đảm bảo tính minh bạch của TTTD.

Hỗ trợ các TCTD kết nối trực tuyến với người dân và doanh nghiệp khi giao dịch truyền thống bị hạn chế trong giai đoạn giãn cách xã hội, Cổng thông tin kết nối khách hàng vay (cic.gov.vn - CIC Credit Connect) và app CIC Credit Connect liên tục được miễn phí sử dụng trong Quý III/2020.

Ngoài ra, CIC cũng hướng dẫn TCTD báo cáo thông tin khách hàng vay được điều chỉnh giữ nguyên nhóm nợ theo Thông tư 01/2020/TT-NHNN quy định về việc TCTD, chi nhánh ngân hàng nước ngoài cơ cấu lại thời hạn trả nợ, miễn, giảm lãi, phí, giữ nguyên nhóm nợ nhằm hỗ trợ khách hàng chịu ảnh hưởng do dịch Covid-19. Cùng với đó, CIC đã ký mới 62 hợp đồng với các TCTD, nâng tổng số người sử dụng trên toàn quốc lên trên 53.478 tài khoản khai thác; tiếp nhận và xử lý 10.141 yêu cầu hỗ trợ, trong đó, có 53 trường hợp khiếu nại bằng văn bản từ TCTD, giải đáp 1.280 trường hợp thắc mắc qua email, 770 phiên chat qua webchat, thực hiện 23 lượt tiếp công dân.

CIC cũng đã xây dựng được một cơ sở dữ liệu TTTD quốc gia thống nhất trên nền tảng công nghệ hiện đại, tích hợp được nhiều nguồn dữ liệu khác nhau, bao gồm 100% TCTD, kể cả gần 1.200 quỹ tín dụng nhân dân và 04 tổ chức tài chính vi mô. Dữ liệu lịch sử được phân tổ, lưu trữ tối thiểu 5 năm trên nền tảng công nghệ hiện đại, có thể truy xuất phục vụ cho các hoạt động nghiệp vụ khác nhau. Tổng số chủ thẻ thông tin được cập nhật và lưu trữ trong kho dữ liệu TTTD quốc gia đến hết Quý I/2020 là trên 43.400.000, bao gồm 1.200.000 pháp nhân và trên 42.200.000 thẻ nhân.

Đặc thù của ngành ngân hàng nói chung và của CIC nói riêng liên quan đến đến bảo mật thông tin, hệ thống dữ liệu lớn, cung cấp thông tin nhanh chóng kịp thời, chương trình sử dụng phải được cập nhật để phù hợp với thời đại mới, chương trình



thay đổi thì việc kiểm thử phải hoạt động, để đảm bảo khi thay đổi thì chương trình sẽ không bị gặp sự cố lỗi nào.

### **KẾT LUẬN CHƯƠNG.**

Trong chương 1, luận văn trình bày và giới thiệu tổng quan các nghiên cứu về kiểm thử phần mềm và các đặc trưng của kiểm thử trong các phần mềm khác nhau, đồng thời đưa ra các phương pháp tiếp cận cho quá trình nghiên cứu.

## CHƯƠNG II: PHƯƠNG PHÁP KIỂM THỬ TỰ ĐỘNG VÀ CÁC CÔNG CỤ

Chương này sẽ tìm hiểu về tổng quan về kiểm thử tự động, một số mô hình kiểm thử tự động hiện nay.

### 2.1. Tổng quan về kiểm thử tự động

#### 2.1.1. *Khái niệm kiểm thử tự động*

Tự động hóa kiểm thử làm tăng hiệu quả phần mềm tổng thể và đảm bảo chất lượng phần mềm mạnh mẽ. Có những công cụ cụ thể có thể thực thi hiệu quả các trường hợp thử nghiệm tự động và giúp so sánh các kết quả thực tế và dự kiến. Theo cách này, kiểm tra tự động có thể đảm bảo sự thành thạo phần mềm mà không cần can thiệp thực hiện để lặp lại một cách thủ công.

Một trong những đặc quyền lớn nhất trong vấn đề kinh doanh mà thử nghiệm tự động mang lại là nó có thể được thực hiện bám sát thời gian với nỗ lực tối thiểu và độ chính xác tối đa.

Kiểm thử tự động hiện nay quan trọng như thế nào?

Trong thời đại công nghệ 4.0 đi liền với sự phát triển vượt bậc của ngành công nghệ tạo ra cuộc cách mạng vạn vật kết nối Internet (internet of things). Internet đang phát triển không ngừng. Trong đó công nghệ thông tin giúp ta kết nối các nội dung với nhau dễ dàng hơn. Ngành công nghệ mang lại cho chúng ta từ Email đến các mạng xã hội, Internet vạn vật và tạo ra nhiều sản phẩm công nghệ tiên tiến như in 3D, công nghệ lưu trữ... Đi liền với sự phát triển nhanh chóng đó thì nhu cầu về ứng dụng công nghệ của con người trong đời sống và trong sản xuất ngày một tăng lên. Xã hội đang ngày càng trở nên tự động hóa. Nó mang lại cho con người nhiều lợi ích lớn giúp con người giải quyết nhiều việc hơn, tạo ra năng suất lao động cao hơn, ứng dụng sản xuất mạnh mẽ hơn..... Vì vậy, các yếu tố liên quan đến tự động hóa ngày càng được coi trọng.

### 2.1.2. Sự khác biệt giữa kiểm thử thủ công và kiểm thử tự động

Với một dự án, có ba vấn đề quan trọng cần chú ý đó là chi phí, thời gian và chất lượng. Tất cả các dự án đều có mục tiêu là có một sản phẩm chất lượng cao nhưng chi phí thì thấp và thời gian cần thiết để hoàn thành dự án. Vì vậy nên kiểm thử là một phần không thể thiếu của bất kỳ dự án phần mềm nào. Kiểm thử phần mềm được phân loại gồm Kiểm thử thủ công (maunual testing) và Kiểm thử tự động (automation testing).

**Bảng 2.1 Sự khác biệt giữa kiểm thử thủ công và kiểm thử tự động**

	<b>Kiểm thử thủ công (Kiểm thử thủ công)</b>	<b>Kiểm thử tự động (Automaiton Testing)</b>
<b>Định nghĩa</b>	Kiểm thử thủ công là việc thử nghiệm phần mềm được làm thủ công bởi con người. Người thực hiện các công việc đó gọi là người kiểm thử. Người kiểm thử thực hiện kiểm thử đảm bảo một phần mềm/ứng dụng hoạt động bình thường bằng cách tuân theo các điều kiện được viết trong kịch bản kiểm thử. Người kiểm thử là người đánh giá thiết kế kiểm thử, chức năng và hiệu suất của ứng dụng bằng cách click vào các phần khác nhau của ứng dụng/hệ thống đó.	Kiểm thử tự động là thực hiện kiểm thử phần mềm bằng công cụ với rất ít hoặc không có sự tương tác của con người, hỗ trợ người kiểm thử không phải lặp đi lặp lại các bước nhàm chán. Trong kiểm thử tự động, có các kịch bản kiểm thử được viết sẵn và chạy tự động. Các kịch bản kiểm thử được chạy để so sánh kết quả thực tế với kết quả mong đợi.

Độ tin cậy (Reliability)	Độ tin cậy không cao.	Độ tin cậy rất là cao
Thời gian yêu cầu (Time Requires)	Kiểm thử thủ công Tốn nhiều thời gian cho một bộ kịch bản kiểm thử (test case) trên môi trường khác nhau Chạy trên nhiều nền tảng nhiều giao diện khác nhau	Kiểm thử tự động được thực hiện bởi các công cụ kiểm thử nên thời gian thực hiện nhanh hơn so với kiểm thử thủ công
Sử dụng khi nào? (When to use ?)	Chỉ phục vụ cho một số loại kiểm thử nhất định Đòi hỏi nhận định của con người, những người có kinh nghiệm. Một số trường hợp con người mới có thể làm được như đoán lỗi, kiểm thử giao diện.	Kiểm thử tự động phù hợp với kiểm tra hồi quy, kiểm tra hiệu suất, Phục vụ cho các bạn kiểm thử phù hợp (test suited) lặp đi lặp lại hoặc cần một dữ liệu lớn
Hiệu suất và hàng loạt (Performance & Batch Testing)	Kiểm tra hiệu suất và hàng loạt không thể làm được với kiểm thử thủ công Vì con người không thể đăng nhập một ngàn user hoặc kêu gọi 1 ngàn người dùng (user) click vào trang web để xem phản hồi và báo cáo.	Đối với kiểm tra hiệu suất và hàng loạt thì kiểm thử tự động giúp cho bạn có thể giả làm số lượng người dùng (user) đăng nhập vào hệ thống và đo độ phản hồi của hệ thống.
Chi phí đầu tư (Investment Cost)	Chi phí ban đầu của kiểm thử thủ công thời gian đầu sẽ tiết	Chi phí ban đầu cao, chi phí để thuê một người kiểm thử tự động với chi phí cao.

	kiệm, tiết kiệm tiền thuê nhân viên kiểm thử, Nhưng về sau dự án lớn, thời gian lập trình lâu thì chi phí cho Kiểm thử thủ công cao vì khi vào dự án người kiểm thử phải kiểm tra đi kiểm tra lại nhiều lần. Mỗi lần thay đổi thì phải kiểm tra lại từ đầu.	Công ty cần mua công cụ hỗ trợ (công cụ) nếu như công cụ có phí. Về lâu dài thì kiểm thử tự động sẽ giảm chi phí hơn so với kiểm thử thủ công Vì những test case có thể chạy lại tự động giảm thời gian xử lý hoặc có thể lên lịch cho chương trình tự động chạy.
Con người (Human element)	Tốn rất nhiều người để ngồi kiểm thử, người kiểm thử tốn thời gian xem và kiểm tra đi kiểm tra lại theo kịch bản kiểm thử	Còn Kiểm thử tự động thì chỉ cần một người ngồi chạy và ghi báo cáo lỗi.

### 2.1.3. So sánh ưu và nhược điểm của Kiểm thử thủ công và Kiểm thử tự động

**Bảng 2.2 So sánh ưu nhược điểm của kiểm thử thủ công và kiểm thử tự động**

	<b>Kiểm thử thủ công (Kiểm thử thủ công)</b>	<b>Kiểm thử tự động (Kiểm thử tự động)</b>
Ưu điểm	Dễ dàng cho việc kiểm thử giao diện, người kiểm thử sẽ có phản hồi nhanh và trực quan về giao diện ứng dụng. Mất ít chi phí hoặc có thể không mất phí cho các công cụ tự động và quy trình khi có thay đổi nhỏ kiểm thử thủ	Sử dụng công cụ tự động giúp tìm kiếm được nhiều lỗi hơn kiểm thử tự động nhanh và hiệu quả Quá trình kiểm thử được ghi lại, điều đó giúp chạy lại kịch bản kiểm thử nhiều lần và thực hiện trên nhiều nền tảng khác nhau

	<p>công không bị mất thời gian để thay đổi các trường hợp kiểm thử</p> <p>Nếu như chương trình có thay đổi nhỏ, thì Kiểm thử thủ công sẽ không phải mất quá nhiều thời gian cho việc kiểm tra đối với những trường hợp kiểm thử.</p>	<p>Kiểm thử tự động được thực hiện bằng các công cụ phần mềm, do đó nó hoạt động không mệt mỏi không giống con người kiểm thử.</p> <p>Kiểm thử tự động năng suất và chính xác.</p> <p>Phạm vi kiểm thử rộng vì kiểm tra tự động không quên kiểm tra cả đơn vị nhỏ nhất.</p> <p>Phản hồi nhanh hơn</p> <p>Bảo mật thông tin</p>
Nhược điểm	<p>Kết quả kiểm thử ít tin cậy hơn vì có thể sai sót thay đổi do yếu tố con người, kiểm thử thủ công không có tính tái sử dụng.</p> <p>Không thực hiện được với kiểm thử hiệu năng và kiểm thử khả năng chịu tải.</p> <p>Kiểm thử thủ công sẽ tiêu tốn nhiều thời gian cũng như công sức của người kiểm thử hơn trong việc phát hiện ra các lỗi (bug).</p> <p>Kết quả tìm thấy thường ít được tin cậy</p> <p>Chi phí dành cho dự án có thể phải tăng lên do những đòi hỏi trong việc phải thuê</p>	<p>Rất khó có cái nhìn đúng và trực quan về giao diện người dùng như font chữ, màu sắc, vị trí, kích thước các button nếu như không có yếu tố con người.</p> <p>Chi phí cho các công cụ kiểm thử có thể tốn kém, nếu có một thay đổi nhỏ cũng sẽ mất thời gian để cập nhật kịch bản kiểm thử.</p> <p>Công cụ có thể mất phí.</p> <ul style="list-style-type: none"> <li>- Chi phí cho người kiểm thử cao</li> <li>- Chi phí phát triển và bảo trì test script cao.</li> <li>- Đòi hỏi người kiểm thử phải có kinh nghiệm công nghệ và kỹ năng lập trình.</li> <li>- Đòi hỏi thời gian chuẩn bị dài hơn để thiết kế, cài đặt kỹ càng trước khi cần đưa dự án đi kiểm thử.</li> <li>- Có những dự án không nên chạy kiểm thử tự động, nhưng nhiều người kiểm thử</li> </ul>

nguồn nhân lực để tiến hành kiểm tra.  Quá trình sử dụng công cụ tự động để kiểm thử sẽ cho bạn kết quả nhanh hơn cũng như chính xác hơn.	vấn hiệu nhằm và chạy kiểm thử tự động, dẫn đến mất thời gian, resource, công sức.
---	--

#### **2.1.4. Kiểm thử thủ công và kiểm thử tự động**

Kiểm thử thủ công phù hợp với:

Kiểm thử thăm dò (Exploratory Testing): Là kiểm thử đòi hỏi phải thử nghiệm của kiến thức, kinh nghiệm, phân tích, kỹ năng, sáng tạo và trực giác.

Kiểm thử khả dụng (Usability Testing): Đây là kiểm thử mà bạn cần để đo độ thân thiện, hiệu quả thuận tiện cho người dùng cuối.

Kiểm thử Ad hoc (Ad-hoc Testing): Trong kịch bản, không có phương pháp cụ thể. Là phương pháp hoàn toàn không có kế hoạch kiểm thử cụ thể.

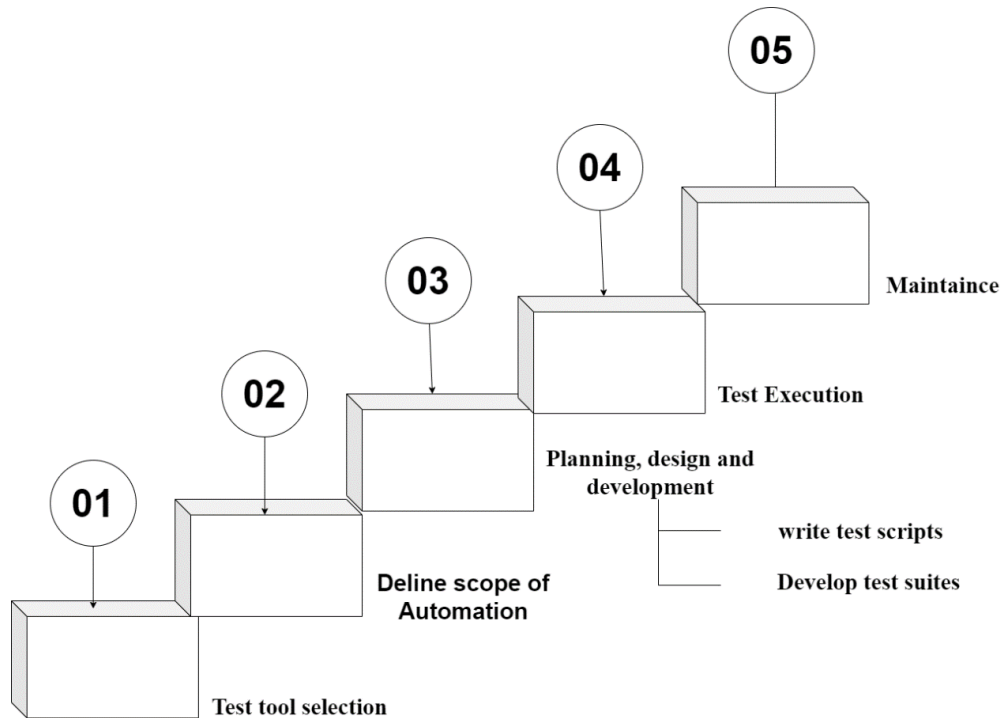
Kiểm thử tự động phù hợp với:

Kiểm thử hồi quy (Regression Testing): Kiểm thử hồi quy rất phù hợp với kiểm thử tự động vì các thay đổi mã thường xuyên và khả năng chạy các hồi quy một cách kịp thời, đúng lúc.

Kiểm thử khả năng chịu tải (Load Testing): Là kiểm thử phù hợp với kiểm thử tự động để hoàn thành các thử nghiệm có hiệu quả khi nó đi kèm để tải thử nghiệm.

Kiểm thử hiệu năng (Performance Testing): Là thử nghiệm đòi hỏi sự mô phỏng của hàng ngàn người dùng đồng thời đòi hỏi tự động hóa.

### 2.1.5. Quy trình kiểm thử phần mềm tự động [11] (Automation testing)



**Hình 2.1 Quy trình kiểm thử tự động**

#### a. Chọn công cụ kiểm thử (Test tool selection)

Chọn công cụ kiểm thử tự động dựa trên một số quy tắc sau đây:

Nếu công cụ hỗ trợ kiểm thử không phải miễn phí thì công ty đã có giấy phép cho công cụ đó chưa? có thể tìm các công cụ mã nguồn mở (open source) nhưng cần lựa chọn, các thành viên trong nhóm biết về công cụ này đã dùng chưa? có cần thêm người vào nhóm không hay cần đào tạo cho người cũ.

#### b. Xác định phạm vi tự động (Define scope of automation)

Phạm vi tự động là khu vực của ứng dụng đang được thử nghiệm sẽ được tự động hóa. Chúng ta có thể xác định được phạm vi của kiểm thử tự động dựa trên các yếu tố sau đây:

- Tính năng quang trọng đối với nghiệp vụ;
- Các tình huống (Scenarios) có lượng dữ liệu lớn;
- Các ứng dụng kết nối với nhau bằng các chức năng;
- Dựa vào tính khả thi về mặt kỹ thuật (có thể thực hiện được về mặt kỹ thuật);



- Mức độ tái sử dụng của các thành phần kinh doanh (business component);
- Dựa vào tính phức tạp của các trường hợp kiểm thử;
- Khả năng sử dụng kiểm thử chéo các trường hợp giống nhau.

### **c. Lập kế hoạch thiết kế và phát triển (Planning, design and development)**

Trong giai đoạn này tạo chiến lược và kế hoạch tự kiểm thử tự động, bao gồm những công việc như sau:

- Chọn công cụ kiểm thử tự động để sử dụng cho dự án.
- Thiết kế mô hình và các tính năng của kế hoạch kiểm thử tự động.
- Các hạng mục trong phạm vi ngoài phạm vi của tự động hóa.
- Tự chuẩn bị lớp thử nghiệm.
- Lập lịch kiểm thử, kế hoạch, thời gian viết kịch bản và thời gian thực hiện.
- Các tài liệu liên quan đến kiểm thử tự động, dự án.

### **d. Thực hiện kiểm thử (Test Execution).**

Các script tự động được thực hiện trong giai đoạn này. Các kịch bản phải sử dụng dữ liệu kiểm thử đầu vào trước khi thiết lập để chạy. Sau khi thực hiện cung cấp báo cáo kiểm thử chi tiết. Được thực hiện bằng phương pháp sử dụng công cụ kiểm thử tự động hoặc thông qua công cụ Quản lý kiểm thử - được gọi chung là công cụ tự động.

### **e. Bảo trì (Maintaince)**

Khi các chức năng mới được thêm vào thì hệ thống đang kiểm thử (system under test) với các chu trình liên tiếp, scripts tự động cần phải được bổ sung, xem xét và duy trì cho mỗi chu kỳ phát hành.

## **2.2. Tìm hiểu về các mô hình kiểm thử tự động hiện nay.**

### **2.2.1. Mô hình kiểm thử tự động**

Mô hình kiểm thử tự động là mô hình bắt đầu bằng việc tập hợp các quy luật, nguyên tắc dùng trong quá trình viết mã kiểm thử. Những quy luật này giúp chúng ta viết mã theo cách “giảm thiểu việc chỉnh sửa mã kiểm thử khi ứng dụng có thay đổi”.

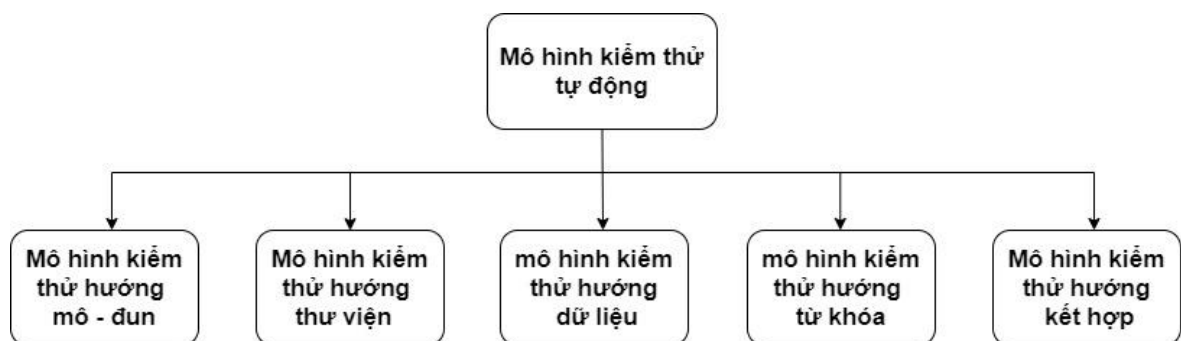
Các mô hình tự động hóa là mô hình được thiết lập để cung cấp môi trường thực thi cho các kịch bản kiểm thử tự động. Mô hình cung cấp cho người dùng nhiều lợi ích khác nhau giúp họ phát triển, thực thi và báo cáo các tập lệnh kiểm thử tự động hóa một cách hiệu quả. Các mô hình kiểm thử tự động cung cấp kiến trúc riêng cho mỗi dự án kiểm thử. Mỗi mô hình kiểm thử lại có những quy tắc riêng, các giao thức, các chỉ dẫn và những thủ tục riêng dành cho các việc khác nhau như tạo kịch bản kiểm thử, tổ chức và thực thi các kịch bản kiểm thử đó.

Lợi thế của mô hình kiểm thử tự động: Khả năng tái sử dụng mã, bảo mật tối đa, có thể khôi phục kịch bản, bảo trì với chi phí thấp, hạn chế việc can thiệp thủ công, dễ báo cáo tổng hợp.

Hiện nay có rất nhiều loại mô hình kiểm thử tự động khác nhau [8], các mô hình này khác nhau dựa trên việc hỗ trợ của chúng đối với các yếu tố chính khác nhau để thực hiện tự động hóa như khả năng tái sử dụng, dễ bảo trì...

Mô hình kiểm thử tự động được sử dụng phổ biến hiện nay:

- Mô hình kiểm thử dựa trên mô – đun - Module Based Testing Framework
- Mô hình kiểm thử kiến trúc thư viện - Library Architecture Testing Framework
- Mô hình kiểm thử theo hướng dữ liệu - Data Driven Testing Framework
- Mô hình kiểm thử theo hướng từ khóa - Keyword Driven Testing Framework
- Mô hình kiểm thử theo hướng hỗn hợp - Hybrid Testing Framework



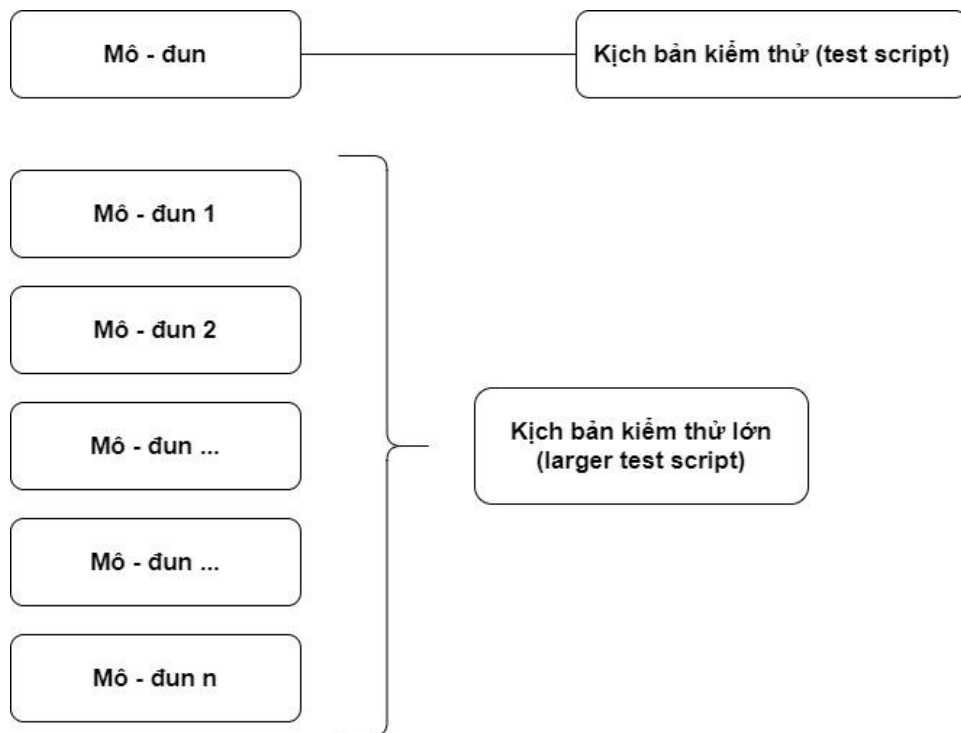
**Hình 2.2 Mô hình kiểm thử tự động**

### **Mô hình kiểm thử hướng Mô - đun**

Mô hình kiểm thử dựa trên các Mô - đun. Dựa vào một trong những khái niệm lập trình hướng đối tượng (OOP-Object-Oriented Programming) phổ biến - trừu tượng.

Mô hình phân chia “ứng dụng đang kiểm thử” thành một Mô - đun logic và riêng biệt. Đối với mỗi mô - đun, chúng tạo lập một lệnh kiểm tra riêng biệt và độc lập. Khi đó, các tập lệnh kiểm tra này kết hợp với nhau sẽ xây dựng một tập lệnh kiểm thử lớn hơn đại diện cho nhiều mô - đun nhỏ.

Với một script của một Mô - đun sẽ tương ứng với các thao tác và dữ liệu tương ứng dành cho nó. Nếu như có sự thay đổi về kiểm thử dữ liệu (database) thì các script cũng phải thay đổi.



**Hình 2.3 Mô hình kiểm thử hướng mô - đun**

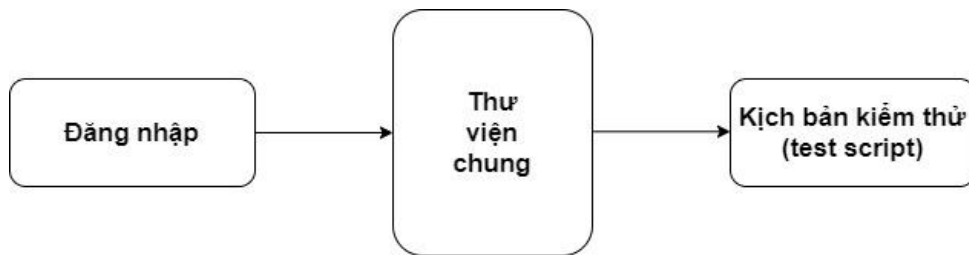
**Ưu điểm:** Mô hình kiểm thử Mô - đun dễ bảo trì, tiết kiệm chi phí. Mô hình được mở rộng. Nếu có thay đổi được triển khai trong một phần của ứng dụng, thì chỉ cần sửa tập lệnh kiểm tra đại diện cho phần đó của ứng dụng để không ảnh hưởng đến tất cả các phần khác.

**Nhược điểm:** Trong triển khai các tập lệnh kiểm thử cho từng mô - đun riêng biệt, chúng ta phải nhúng dữ liệu thử nghiệm vào các tập lệnh kiểm thử. Do đó, bất cứ khi nào cũng phải kiểm tra tập dữ liệu kiểm tra khác, nó yêu cầu thao tác phải thực hiện trong các kịch bản kiểm thử.

### a. Mô hình kiểm thử kiến trúc thư viện

Mô hình được xây dựng có nền tảng dựa trên mô hình kiểm thử hướng mô - đun nhưng nó có một số ưu điểm tốt hơn. Thay vì chia ứng dụng đang kiểm thử thành các tập lệnh nhỏ, thì mô hình kiểm thử này tạo ra một thư viện chung bao gồm các chức năng đang được kiểm thử. Do đó, thư viện này có thể được gọi từ các kịch bản kiểm thử bất kỳ lúc nào có yêu cầu.

Ví dụ: Các bước đăng nhập có thể được kết hợp thành một hàm và được lưu trữ trong một thư viện. Do đó, tất cả các kịch bản kiểm thử yêu cầu để đăng nhập ứng dụng có thể gọi hàm đó thay vì viết lại toàn bộ mã



**Hình 2.4 Mô hình kiểm thử hướng thư viện**

Ưu điểm: Giống như mô hình hướng mô - đun, mô hình này cũng giới thiệu mức độ mô - đun hóa cao, bảo trì dễ dàng và hoạt động hiệu quả, chi phí thấp.

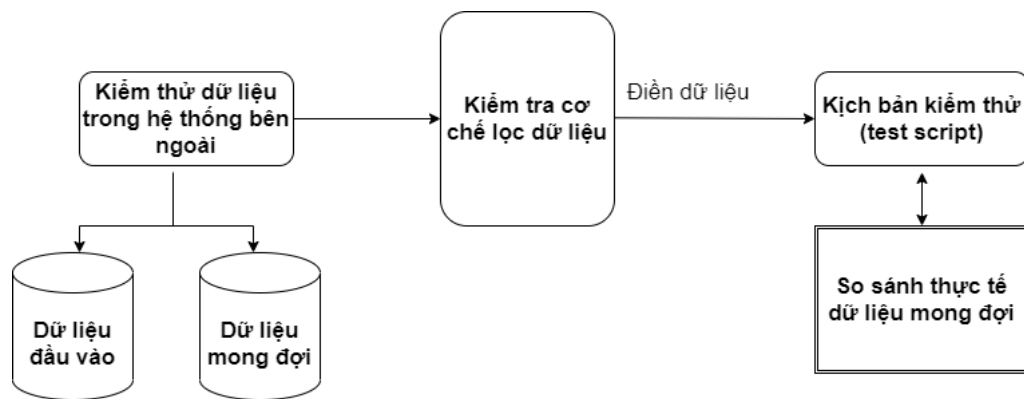
Khi tạo các chức năng có thể sử dụng các tập lệnh kiểm thử khác nhau trên mô hình. Do đó, mức độ tái sử dụng lớn.

Nhược điểm: Cũng giống với mô hình hướng mô - đun, dữ liệu kiểm thử được đưa vào các tập lệnh kiểm thử. Do đó, bất kỳ thay đổi nào trong dữ liệu thử nghiệm cũng sẽ yêu cầu thay đổi các tập lệnh kiểm thử. Với sự ra đời của thư viện thì mô hình thư viện phức tạp hơn so với mô hình mô - đun.

### b. Mô hình kiểm thử tự động hướng dữ liệu

Trong quá trình kiểm thử tự động bất kỳ ứng dụng nào, việc kiểm thử một chức năng lặp đi lặp lại nhiều lần với tập dữ liệu đầu vào khác nhau. Do đó, trong những trường hợp như vậy thì không thể để dữ liệu kiểm thử nhúng vào trong script. Vì vậy, người ta phải nghĩ đến việc lưu trữ dữ liệu kiểm thử vào một cơ sở dữ liệu bên ngoài các tập lệnh script.

Mô hình kiểm thử này giúp người kiểm thử tách biệt dữ liệu logic, tập lệnh kiểm thử và dữ liệu kiểm thử với nhau. Nó cho phép người dùng lưu trữ dữ liệu kiểm thử vào cơ sở dữ liệu bên ngoài. Cơ sở dữ liệu bên ngoài định dạng, tệp xml, tệp excel, tệp doc, tệp csv... . Dữ liệu được lưu trữ theo quy ước trong cặp từ khóa “Key-Value”. Do đó các từ khóa này được sử dụng để truy cập và ghi dữ liệu vào một thư viện chung.



**Hình 2.4 Mô hình kiểm thử hướng dữ liệu**

Ưu điểm:

Tính năng quan trọng nhất của mô hình này là nó làm giảm đáng kể tổng số tập lệnh cần thiết của các kịch bản kiểm thử. Vì vậy, số lượng mã ít hơn được yêu cầu để kiểm thử một tập hợp các tình huống hoàn chỉnh.

Bất kỳ thay đổi nào trong dữ liệu kiểm thử sẽ không làm ảnh hưởng đến tập lệnh kiểm thử, tăng tính linh hoạt và dễ bảo trì

Một kịch bản kiểm thử duy nhất có thể được thực hiện để thay đổi các giá trị dữ liệu kiểm thử

Nhược điểm:

Quá trình này phức tạp và đòi hỏi nhiều nỗ lực để tìm ra các nguồn dữ liệu kiểm thử và cơ chế đọc.

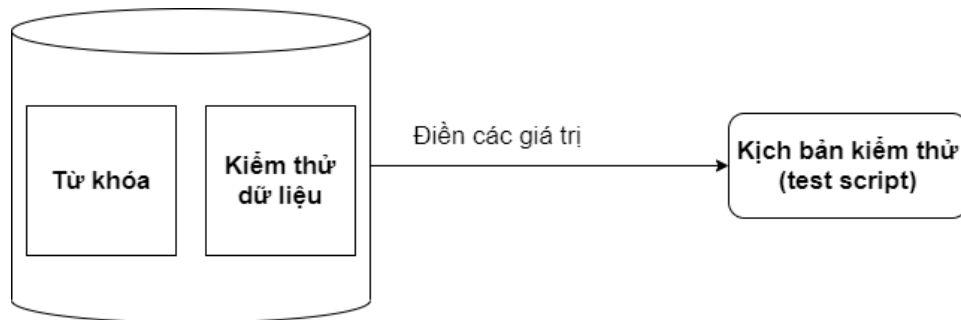
Yêu cầu người kiểm thử thành thạo về lập trình, có thể lập trình và hiểu về lập trình.

### c. Kiểm thử theo hướng từ khóa

Kiểm thử theo hướng từ khóa là một phần của kiểm thử hướng dữ liệu. Kiểm thử theo hướng từ khóa không chỉ tách dữ liệu ra khỏi tập lệnh mà còn giữ một bộ mã nhất định thuộc tập lệnh kiểm thử vào một dữ liệu ngoài.

Bộ mã được gọi là từ khóa, từ khóa hướng dẫn về những hành động cần được thực hiện trên ứng dụng

Các từ khóa và dữ liệu thử nghiệm được lưu trữ trong cấu trúc dạng bảng vì vậy nó cũng được gọi mô hình điều khiển bảng



**Hình 2.5 Mô hình kiểm thử hướng từ khóa**

Ví dụ: bảng kiểm tra của mô hình kiểm thử hướng từ khóa có dạng sau:

**Bảng 2.3 Kiểm tra mô hình kiểm thử hướng từ khóa**

Bước	Miêu tả	Từ khóa	Location(định vị)/Dữ liệu
1	Đăng nhập vào ứng dụng	Login_link	<code>"/span[text()='Đăng nhập']"</code>
2	Click nút đăng nhập	Close_button	<code>"/button[@id='acceptBtn']";</code>
3	Nhập tên đăng nhập	Username_textbox	<code>"/input[@name='username']";</code>
4	Nhập mật khẩu	Password_textbox	<code>"/input[@name='password']";</code>

Trong ví dụ trên, các từ khóa đăng nhập, click chọn xác minh liên kết được xác định trong mã.

Tùy thuộc và bản chất của ứng dụng mà từ khóa có thể được lấy ra. Và tất cả các từ khóa có thể được sử dụng lại nhiều lần trong một trường hợp kiểm thử duy

nhất. Cột định vị chứa giá trị định vị được sử dụng để xác định các phần tử của web trên màn hình hoặc sẽ kiểm thử cần được cung cấp.

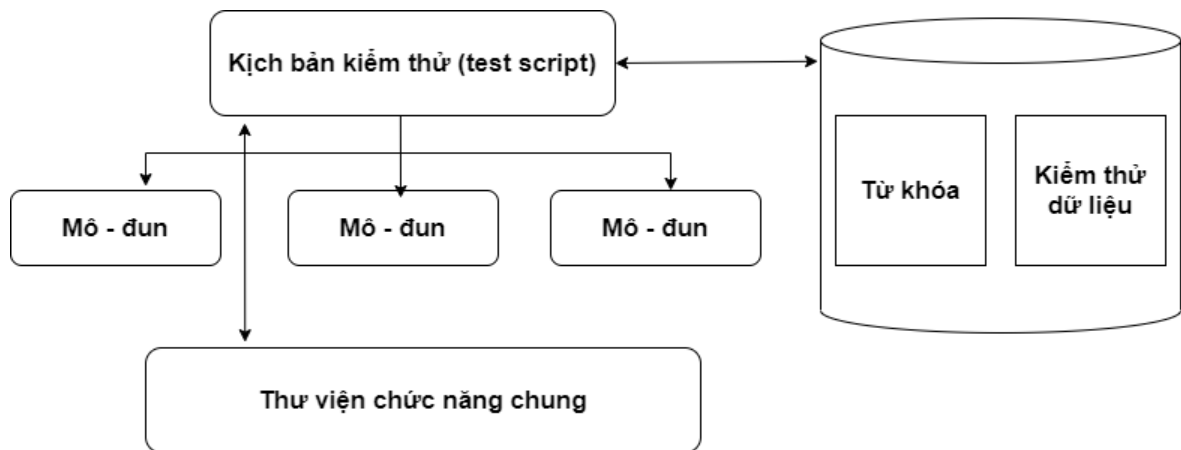
Tất cả các từ khóa được yêu cầu đều được thực hiện thiết kế và đặt trong mã cơ sở dữ liệu của mô hình.

Ưu điểm: ngoài những lợi thế kiểm thử theo hướng dữ liệu mang lại, mô hình kiểm thử hướng từ khoa không yêu cầu người dùng có kiến thức về kịch bản. Một từ khóa duy nhất có thể được sử dụng trên nhiều tập lệnh kiểm thử.

Nhược điểm: Người dùng phải hoàn thành thao tác với cơ chế tạo từ khóa để có nền tảng tận dụng một cách hiệu quả các lợi ích do ứng dụng cung cấp. Mô hình trở nên phức tạp dần lên là khi nó phát triển và một số từ khóa mới được đưa vào kiểm thử.

#### d. Kiểm thử hướng kết hợp

Kiểm thử hướng kết hợp là sự kết hợp giữa hai hay nhiều mô hình trên lại với nhau. Điều quan trọng nhất về việc kết hợp nhiều mô hình là tận dụng những ưu điểm của tất cả các mô hình khác.



**Hình 2.6 Mô hình kiểm thử hướng kết hợp**

Ví dụ:

Một phiếu kiểm thử sẽ chứa từ khóa và dữ liệu

**Bảng 2.4 Mô hình kiểm thử hướng kết hợp**

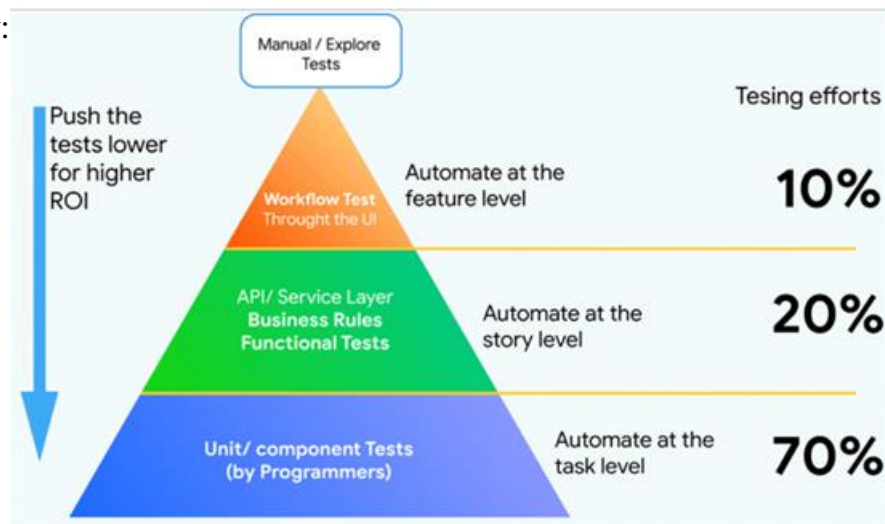
Bước	Mô tả	Từ khóa	Định vị (location)	Dữ liệu
------	-------	---------	--------------------	---------

1	Đăng nhập vào ứng dụng	Login_link	"/span[text()='Đăng nhập']"	
2	Nhập tên đăng nhập	Username_textbox	"/input[@name='username']";	0986668038
3	Nhập mật khẩu	Password_textbox	"/input[@name='password']";	11111
4	Click nút đăng nhập	Login_button	"/button[@class='btn-login']";	

Trong ví dụ trên, cột từ khóa chứa tất cả các từ khóa bắt buộc được sử dụng trong trường hợp kiểm thử và cột dữ liệu chứa tất cả dữ liệu được yêu cầu trong kịch bản kiểm thử.

### 2.2.2. Cấp độ kiểm thử tự động.

Kiểm thử tự động chia thành ba cấp độ [9] được mô phỏng theo hình chóp dưới đây:



Hình 2.7 Cấp độ kiểm thử tự động

#### a. Kiểm tra quy trình hoạt động của phần mềm thông qua giao diện người dùng (Workflow test through the - UI)

Ở cấp độ này chiếm tỉ lệ 10%, có thể kiểm tra không chỉ giao diện người dùng mà còn cả chức năng bằng cách thực hiện hoạt động tạo ra logic nghiệp vụ của ứng dụng.



Loại kiểm tra xuyên suốt này mang lại hiệu quả lớn hơn chỉ kiểm thử lớp chức năng, vì cần phải thực hiện kiểm thử chức năng bằng cách mô phỏng các hoạt động của người dùng cuối thông qua giao diện đồ họa.

**b. Kiểm thử phương thức giao diện kết nối với thư viện và ứng dụng khác**  
(Application Programming Interface – **API**)

Ở cấp độ này chiếm tỉ lệ 20%, không phải tất cả logic nghiệp vụ của ứng dụng đều có thể được kiểm thử bằng cách sử dụng lớp UI. Đây có thể là một tính năng của việc triển khai, ẩn logic nghiệp vụ với người dùng. Quyền truy cập trực tiếp vào lớp chức năng, mang lại cơ hội kiểm tra logic nghiệp vụ của ứng dụng trực tiếp bỏ qua giao diện người dùng, chỉ được cung cấp cho nhóm kiểm thử khi có thỏa thuận với các nhà phát triển.

**c. Kiểm thử đơn vị / Kiểm thử thành phần (Unit / component tests)**

Ở cấp độ này chỉ chiếm tỉ lệ 70%, kiểm tra tự động là thành phần của các kiểm tra đơn vị do các nhà phát triển thiết kế. Người kiểm thử cũng có thể viết các bài kiểm thử như vậy nếu họ có các kỹ năng cần thiết. Các bài kiểm tra trong giai đoạn đầu của dự án, cũng như việc thực hiện liên tục của chúng và bổ sung các kiểm thử mới, kiểm tra "các bản sửa lỗi", có thể giúp bảo vệ dự án khỏi nhiều vấn đề nghiêm trọng.

Các mô hình có thể có để tự động hóa thử nghiệm theo mức độ của nó. Để có một bức tranh toàn cảnh về việc tổ chức quy trình tự động hóa trong một dự án, chúng ta hãy xem xét các khả năng mô hình của nó về mức độ tự động hóa.

## **2.3. Tổng quan về các công cụ kiểm thử tự động hiện nay.**

### **2.3.1. Công cụ kiểm thử tự động**

#### **2.3.1.1. Khái niệm**

Công cụ kiểm thử tự động là các công cụ hỗ trợ người kiểm thử thực hiện công việc kiểm thử phần mềm một cách tự động.

#### **2.3.1.2. Ý nghĩa của các công cụ kiểm thử tự động:**

- Tăng độ tin cậy cho việc kiểm thử.
- Giảm bớt thời gian và công sức thực hiện quá trình kiểm thử.

- Hỗ trợ người kiểm thử phần mềm rèn luyện kỹ năng lập trình.
- Giúp giảm chi phí cho quá trình kiểm thử.
- Giảm sự nhàm chán khi phải thực hiện kiểm thử đi kiểm thử lại cho con người khi phải kiểm thử thủ công.

#### 2.3.1.3. Những thuận lợi và khó khăn khi áp dụng công cụ kiểm thử tự động

##### **Thuận lợi:**

- Kỹ thuật viên không cần can thiệp vào hệ thống.
- Giả lập tình huống không thể thực hiện thủ công bằng tay.
- Giảm chi phí thực hiện kiểm tra số lượng lớn kịch bản kiểm thử hoặc kịch bản kiểm thử lại nhiều lần.

##### **Khó khăn:**

- Tốn chi phí để tạo ra các script để thực hiện kiểm thử tự động.
- Mất chi phí cho việc bảo trì các script.
- Đòi hỏi kỹ sư kiểm thử phần mềm phải có khả năng tạo script kiểm thử tự động.
- Những lỗi mới thì không áp dụng được trong phần mềm.

#### 2.3.1.4. Các yếu tố cần thiết khi chọn một công cụ kiểm thử tự động

- Khả năng tương thích giữa các nền tảng.
- Bảo trì dễ dàng.
- Chất lượng hỗ trợ khách hàng.
- Dễ dàng tạo các kịch bản thử nghiệm.
- Chi phí giấy phép, nếu có.
- Trong trường hợp của một dự án thuê ngoài, bạn cần tính đến mức độ ưa thích của khách hàng.
- Chi phí liên quan đến việc đào tạo nhân viên về một công cụ.
- Yêu cầu phần cứng / Phần mềm của công cụ kiểm thử phần mềm.
- Chính sách hỗ trợ và cập nhật của công cụ.
- Nhận xét của công ty.

#### 2.3.2. Một số công cụ kiểm thử tự động mới hiện nay. [6]

##### a. Công cụ kiểm thử Selenium

Selenium là một trong những công cụ kiểm thử phần mềm phổ biến hiện nay. Selenium được thiết kế để hỗ trợ kiểm thử tự động hóa các chức năng của các ứng dụng web, hỗ trợ trên nhiều nền tảng và trình duyệt.

Đặc trưng:

Selenium là một trong những công cụ kiểm thử tự động tốt nhất cung cấp sự hỗ trợ cho việc thực hiện kiểm thử song song giúp giảm thời gian thực hiện. Selenium là công cụ cần ít tài nguyên hơn khi so sánh với các công cụ kiểm thử thủ công khác. Các trường hợp kiểm thử được chuẩn bị bằng công cụ kiểm thử selenium có thể thực hiện trên tất cả các hệ điều hành.

Selenium hỗ trợ nhiều ngôn ngữ lập trình như Java, Python, C#, Perl, PHP, JavaScript...

#### b. Công cụ kiểm thử TestRail

TestRail là công cụ để quản lý trường hợp kiểm thử trên web, có thể mở rộng, có thể tùy chỉnh. Việc thiết lập chỉ trong vài phút với giải pháp dựa trên điện toán đám mây.

Đặc trưng:

TestRail quản lý hiệu quả các trường hợp, kế hoạch và kiểm thử thủ công và tự động. Nhận thông tin chi tiết theo thời gian thực về tiến độ kiểm thử, chỉ số và báo cáo thông tin hoạt động. Tăng cường hiệu quả với các mốc quan trọng, danh sách việc cần làm và thông báo qua email. Tải lên các trường hợp kiểm thử với ảnh chụp màn hình và kết quả mong đợi. Sử dụng các mẫu có sẵn linh hoạt hoặc tạo các mẫu tùy chỉnh cho riêng người sử dụng.

Tích hợp với các công cụ trong kênh CI / CD / DevOps của bạn bao gồm JIRA, Bugzilla, Jenkins, TFS ...

TestRail được thiết kế cho các nhóm lớn và các dự án quan trọng.

Hỗ trợ cho vùng chứa Docker.

#### c. Công cụ kiểm thử Xray

Xray là ứng dụng quản lý kiểm thử tự động và thủ công số một dành cho QA. Nó là một công cụ đầy đủ tính năng và tích hợp liền mạch với Jira. Mục đích của Xray là

giúp các công ty cải thiện chất lượng sản phẩm của họ thông qua kiểm thử hiệu năng của chương trình.

Đặc trưng:

Truy xuất nguồn gốc giữa các yêu cầu, kiểm tra, lỗi, thực thi. Xác định các điều kiện tiên quyết có thể sử dụng lại và liên kết với các kiểm thử khác nhau. Tổ chức các cuộc kiểm thử trong các thư mục và toàn bộ test case. Có kế hoạch kiểm tra để theo dõi tiến độ công việc. Môi trường kiểm thử. Tích hợp với các khuôn khổ tự động kiểm thử (Selenium, Junit, Uunit, Robot...), API REST tích hợp, Tích hợp CI (Bamboo, JenKins).

#### d. Công cụ kiểm thử TestMonitor

TestMonitor là một công cụ quản lý kiểm tra đầu cuối cho mọi tổ chức. Một cách tiếp cận đơn giản, trực quan để kiểm tra. Cho dù bạn đang triển khai phần mềm doanh nghiệp, cần QA, xây dựng một ứng dụng chất lượng hay chỉ cần trợ giúp trong dự án thử nghiệm của mình, TestMonitor đều có bạn.

Đặc trưng:

Yêu cầu và kiểm thử dựa trên rủi ro. Thiết kế trường hợp thử nghiệm tiên tiến có khả năng hỗ trợ hàng nghìn trường hợp cùng một lúc. Các công cụ lập kế hoạch với các lần chạy nhiều người thử nghiệm và nhân bản tiêu chuẩn ban đầu. Theo dõi kết quả toàn diện. Quản lý vấn đề tích hợp. Báo cáo thông minh với nhiều tùy chọn bộ lọc và hình ảnh hóa. Giao diện người dùng đơn giản. Tích hợp của bên thứ ba có Jira, DevOps và Slack. Hỗ trợ chuyên nghiệp với thời gian phản hồi nhanh chóng.

#### e. Công cụ kiểm thử PractiTest

PractiTest là một công cụ quản lý kiểm tra đầu cuối. Một nền tảng chung cho tất cả các bên liên quan đến QA, nó cho phép hiển thị đầy đủ quy trình kiểm thử và hiểu sâu hơn về kết quả kiểm thử.

Đặc trưng:

Một loạt các tích hợp của bên thứ ba với các trình theo dõi lỗi phổ biến, các công cụ tự động hóa và API mạnh mẽ cho phần còn lại. Hoàn toàn có thể tùy chỉnh linh hoạt theo các nhu cầu luôn thay đổi của nhóm QA: tùy chỉnh các trường, chế độ

xem, quyền, quy trình làm việc .... Sử dụng lại các thử nghiệm và kết quả tương quan trên các bản phát hành và sản phẩm khác nhau. Cây lọc phân cấp độc đáo - sắp xếp mọi thứ và tìm kiếm mọi thứ một cách nhanh chóng. Không bao giờ làm việc hai lần - với các bản sao chống lỗi, hoán vị, tham số và lệnh gọi để kiểm tra. Trực quan hóa dữ liệu với các trang tổng quan và báo cáo nâng cao. Hỗ trợ chuyên nghiệp và nhanh chóng

f. Công cụ kiểm thử QTP

Quick Test Professional (QTP) hiện được gọi là Micro Focus UFT (Kiểm tra chức năng hợp nhất) là một công cụ kiểm tra GUI chức năng tự động cho phép tự động hóa các hành động của người dùng trên web hoặc ứng dụng máy tính dựa trên máy khách. QTP được sử dụng rộng rãi để tự động hóa kiểm thử hồi quy các chức năng. Nó là một trong những công cụ kiểm tra thủ công sử dụng ngôn ngữ kịch bản để thao tác các đối tượng và điều khiển của ứng dụng đang được kiểm tra.

Đặc trưng:

QTP là một trong những công cụ kiểm thử phần mềm phù hợp với người kiểm thử mới. Công cụ QTP rất dễ hiểu. Nó trình bày một trường hợp thử nghiệm trong một quy trình làm việc đơn giản cho người kiểm thử. Công cụ cho phép xác nhận hoàn toàn các ứng dụng thông qua bổ sung đầy đủ các điều kiện kiểm tra.

j. Công cụ kiểm thử Katalon Studio

Là sử dụng công cụ katalon để kiểm thử tự động các ứng dụng web/mobile/web services. Được xây dựng trên các khung selenium và Appium, Katalon Studio có lợi thế trong kiểm thử tự động tích hợp. Công cụ này hỗ trợ nhiều mức độ kiểm thử khác nhau, những người kiểm thử sẽ dễ dàng sử dụng, tiết kiệm thời gian, không phải xây dựng thư viện, bảo trì script

Đặc trưng:

Không yêu cầu giấy phép (license) và bảo trì (có sẵn các dịch vụ hỗ trợ chuyên dụng có tính phí). Tích hợp các bộ khung (framework) và tính năng cần thiết để tạo và thực hiện các trường hợp thử thử nhanh. Được xây dựng dựa trên bộ khung

Selenium nhưng đã bỏ được việc yêu cầu kỹ năng lập trình nâng cao cần thiết như Selenium.

## 2.4. Phân tích so sánh các công cụ kiểm thử tự động.

### So sánh công cụ kiểm thử katalon và selenium

**Bảng 2.5 So sánh công cụ kiểm thử tự động katalon và selenium**

<b>Đặc tính</b>	<b>Katalon Studio</b>	<b>Selenium</b>
Nền tảng	Đa nền tảng	Đa nền tảng
ứng dụng	Windows desktop, Web apps, Mobile apps, API/Web service	Web apps
Ngôn ngữ lập trình	Java/Groovy	Java, JavaScript, Ruby, PHP, C#, Perl, Python
Kỹ năng lập trình	Không yêu cầu có kỹ năng lập trình cao. Đối với các kịch bản nâng cao mới yêu cầu có kỹ năng lập trình	Cần phải có kỹ năng lập trình ở mức nâng cao để tích hợp các công cụ khác nhau
Thời gian tạo test script	Tạo test script nhanh	Tạo test script Chậm
Kiểm thử hình ảnh	Hỗ trợ tích hợp	Phải cài đặt thư viện bổ sung
Giá	Miễn phí với các tính năng cơ bản cho người dùng, và thu phí đối với các tính năng được bổ sung hoặc nâng cao	Miễn phí

## 2.5. Phân tích đánh giá và nêu ra phương pháp kiểm thử đối với phần mềm được kiểm thử.

Theo phân tích ở mục 2.2.2 Các mô hình kiểm thử hiện nay. Để đáp ứng được yêu cầu cũng như giúp cho việc kiểm thử được dễ dàng thì em lựa chọn Kiểm thử

module (đơn vị) để thực hiện kiểm thử cho web khai thác nhu cầu vay của CIC. Để thử module là quá trình kiểm thử chương trình con, từng thử tục nhỏ trong chương trình (tránh tình trạng sai sót)

Kiểm thử đơn vị quản lý phần tử cần kiểm thử, tập trung chú ý từng phần tử nhỏ của chương trình

Kiểm thử đơn vị giúp dễ dàng việc debug chương trình, tạo cơ hội nhất cho việc kiểm thử bởi nhiều người.

Mục đích của kiểm thử đơn vị: So sánh chức năng thực tế của từng Module với đặc tả chức năng hay đặc tả giao diện của modul đó. Không chỉ ra việc module có thỏa mãn đầy đủ đặc tả chức năng hay không? Chỉ ra việc module có làm điều khác biệt so với đặc tả của module.

Kiểm thử module sử dụng 2 phương án:

- Kiểm thử không tăng tiến hay kiểm thử độc lập (Non – incremental testing) kiểm thử các module chức năng độc lập nhau, sau đó kết hợp chung lại để tạo thành chương trình.

- Kiểm thử tăng tiến (Incremental testing): kết hợp modul cần kiểm thử vào bộ phận đã kiểm thử, kiểm thử ngữ cảnh.

## **KẾT LUẬN CHƯƠNG.**

Trong chương này, ta đã tổng hợp được các phương pháp kỹ thuật cho việc áp dụng vào kiểm thử tự động, đồng thời đưa ra các mô hình kiểm thử tự động trên các phần mềm và một số phần mềm kiểm thử tự động thường được sử dụng trong thực tế.

## **CHƯƠNG III: ỨNG DỤNG**

Chương này sẽ trình bày chi tiết các bước xây dựng quy trình kiểm thử tự động cho việc phát triển của Trung tâm Thông tin Tín dụng Quốc gia Việt Nam

### **3.1. Đặt vấn đề cho bài toán ứng dụng công cụ kiểm thử**

Kiểm thử phần mềm là những hoạt động quan trọng trong vòng đời phát triển phần mềm. Kiểm thử góp một phần rất lớn trong việc đánh giá chất lượng một phần mềm và là quy trình không thể thiếu trong tất cả các dự án phần mềm. Tuy nhiên, kiểm thử thường gặp phải nhiều khó khăn.

Thứ nhất, kiểm thử các ứng dụng, phần mềm phức tạp đòi hỏi nhiều nguồn tài nguyên và chi phí cao.

Thứ hai, quy trình phát triển phần mềm trải qua nhiều hoạt động thay đổi thông tin, sự mất mát thông tin trong quá trình thay đổi là yếu tố làm cho hoạt động kiểm thử gặp nhiều khó khăn.

Thứ ba, vì kiểm thử chưa được xem trọng trong việc đào tạo con người.

Cuối cùng, không có một kỹ thuật kiểm thử phần mềm nào có thể khẳng định một phần mềm hoàn toàn đúng đắn, không còn lỗi.

Vì vậy, luận văn này tập trung nghiên cứu các công cụ kiểm thử tự động để áp dụng vào giai đoạn kiểm thử cho các ứng dụng nhằm giảm tải bớt công việc thủ công lặp đi lặp lại. Chương 3 là chương tìm hiểu về một số ứng dụng cụ thể và nghiên cứu xây dựng chương trình kiểm thử cho một số chức năng có thể kiểm thử tự động.

### **3.2. Đề xuất áp dụng công cụ kiểm thử chức năng cho website.**

Trong luận văn này em đề xuất kiểm thử chức năng cho website khai thác nhu cầu vay tại Trung tâm Thông tin Tín dụng Quốc gia Việt Nam.

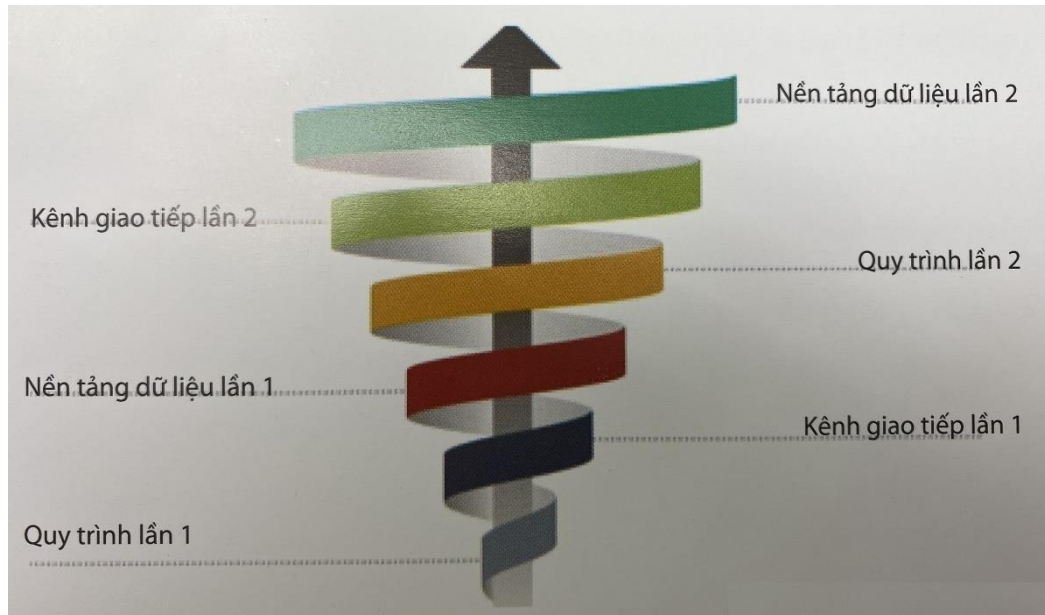
#### ***3.2.1. Hiện trạng kiểm thử phần mềm tại Trung tâm Thông tin tín dụng Quốc gia Việt Nam.***

CIC là đơn vị xây dựng được Hệ thống công nghệ thông tin tiên tiến, hiện đại, nâng cao năng lực thu thập, xử lý dữ liệu lớn, cung cấp sản phẩm dịch vụ có chất lượng đáp ứng được yêu cầu trong giai đoạn hiện nay.



Trong 20 năm hoạt động và phát triển, với sự dẫn dắt của lãnh đạo cùng với nỗ lực không ngừng của toàn thể cán bộ Trung tâm đã khiến cho quá trình chuyển đổi công nghệ diễn ra nhanh chóng, hiệu quả và đóng vai trò vô cùng quan trọng trong sự thành công của CIC hôm nay

Quá trình chuyển đổi công nghệ thường được thể hiện qua 3 bước [2]:



**Hình 3.1 Mô hình chuyển đổi kênh giao tiếp của CIC**

Chuyển đổi về quy trình (Business processes), chuyển đổi các kênh giao tiếp (Channels) và cuối cùng là chuyển đổi về nền tảng dữ liệu (Data foundation). Tại CIC, việc chuyển đổi công nghệ trong những năm qua đã trải qua đầy đủ cả 3 bước trên, trong đó từng bước của quá trình chuyển đổi là nguyên nhân và động lực để thực hiện bước tiếp theo. Thực tế đã chứng minh, việc chuyển đổi công nghệ theo mô hình xoáy ốc, trải qua lần lượt các bước với mức độ tăng dần là hướng đi đúng đắn, mang lại hiệu quả to lớn đối với hoạt động thông tin tín dụng của CIC nói riêng, góp phần không nhỏ vào hoạt động cấp tín dụng của các TCTD và hoạt động điều hành chính sách của NHNN nói chung.

CIC hiện nay, bước chuyển đổi về quy trình đã được thực hiện tương đối hoàn thiện, quy trình hoạt động thông tin tín dụng đã được xây dựng, phát triển phù hợp với mô hình, công nghệ mới. Phần lớn các quy trình làm việc tại CIC đã được tự động

hóa. Điều này không chỉ nâng cao năng suất công việc, đảm bảo thời gian, tốc độ xử lý của hồ sơ mà còn nâng cao tính chính xác, an toàn bảo mật của thông tin tín dụng ngay từ bước đầu vào đến khi hoàn thành báo cáo cuối cùng.

Kênh giao tiếp của CIC cũng ngày càng được mở rộng và hoàn thiện hơn, hướng tới những tiêu chuẩn, chuẩn mực mới nhằm đem lại trải nghiệm tốt nhất cho TCTD trong việc khai thác và cung cấp thông tin tín dụng.

Giai đoạn hiện nay, sau khi các bước cải tiến về quy trình và kênh giao tiếp cơ bản đã hoàn thành thì CIC cần tiếp tục đổi mới ở bước khai thác nền tảng dữ liệu (data). Trải qua nhiều giai đoạn phát triển, việc ứng dụng công nghệ trong việc khai thác nền tảng dữ liệu đã đưa CIC từ một đơn vị sử dụng dữ liệu giấy tờ trong những ngày đầu tiên đến việc khai thác, lưu trữ dữ liệu trên Cơ sở dữ liệu với công nghệ và công cụ hiện đại, tiên tiến hàng đầu trên thế giới, Kho dữ liệu của CIC từ khi còn “đếm được” cho đến hiện nay đã có đến 40 triệu hồ sơ khách hàng với hàng tỷ bản ghi về thông tin tín dụng của toàn bộ các khách hàng vay tại Việt Nam.

Khi nền kinh tế số ( Digital economy) đang có sự dịch chuyển dần sang nền kinh tế dữ liệu (Data economy), là một đơn vị hoạt động thông tin tín dụng, nếu CIC không thay đổi về nền tảng dữ liệu, bổ sung thêm các thách thức khai thác dữ liệu mới thì khó long đáp ứng được yêu cầu ngày càng cao của các TCTD và có thể sớm bị thay thế bởi các tổ chức tư nhân khác. Việc sở hữu kho dữ liệu không lồ về thông tin tín dụng của toàn bộ các khách hàng vay là một lợi thế mà khó tổ chức nào có thể so sánh, tuy nhiên điều này mang lại nhiều thách thức không hề nhỏ đối với mục tiêu chuyển đổi nền tảng dữ liệu và hệ thống công nghệ tự động.

Không những thế, bên cạnh việc mở rộng các nguồn dữ liệu thu thập được, phân hóa thông tin tín dụng thành hàng trăm chỉ tiêu thì với nhu cầu càng ngày càng cao của các TCTD, việc chuyển hướng cung cấp các sản phẩm sẵn có sang sản phẩm có tính cá biệt hóa, phù hợp với từng TCTD là một yếu tố cấp thiết mà CIC cần phải làm ngay, để tiếp tục duy trì vị thế đơn vị đứng đầu trong hoạt động thông tin tín dụng tại Việt Nam.

Tuy nhiên khi đặt trong bối cảnh thế giới đang chứng kiến những thành quả của cuộc cách mạng công nghệ 4.0 như học máy (machine learning), dữ liệu lớn (big data) và gần đây nhất là trí tuệ nhân tạo (AI) tạo ra sự bùng nổ về dữ liệu, thì chúng ta cũng có rất nhiều cách để cá biệt hóa dữ liệu sản phẩm thông tin tín dụng cho các TCTD, thay vì chỉ dựa vào thông tin lịch sử về tín dụng của khách hàng vay như trước. Trước đây, CIC cung cấp sản phẩm từ một nguồn dữ liệu chính là lịch sử tín dụng của khách hàng do TCTD cung cấp, với khoảng 300 chỉ tiêu thì trong thời gian tới, chúng ta có thể cung cấp sản phẩm từ hàng chục nguồn dữ liệu khác nhau với số lượng chỉ tiêu có thể gấp nhiều lần hiện nay.

Từ đó, ý tưởng sử dụng trí tuệ nhân tạo để phân loại sản phẩm cho một nhóm TCTD phù hợp đến mức từng TCTD một trên cơ sở dữ liệu ứng dụng công nghệ lớn đã được hình thành.

Đầu tiên là về AI – trí tuệ nhân tạo. Là công nghệ mô phỏng lại các quá trình suy nghĩ và học tập của con người cho máy móc thực hiện, đặc biệt nhất là các hệ thống máy tính. Trong hoạt động thông tin tín dụng của CIC, áp dụng công nghệ trí tuệ nhân tạo và các công cụ hiện có, cụ thể như áp dụng việc học máy vào trong công cụ: phần mềm tổng hợp phí, tổng đài chăm sóc khách hàng, quản lý người dùng, chatbot..., chúng ta sẽ xây dựng những mô hình tiên đoán phù hợp.

Không chỉ ứng dụng trong việc cung cấp sản phẩm đơn lẻ cho từng khách hàng, mà đối với cả các sản phẩm tổng hợp, các báo cáo dành cho Ngân hàng Nhà nước cũng có thể áp dụng trí tuệ nhân tạo. Từ đó CIC có thể xây dựng được mô hình cảnh báo sớm những rủi ro có thể xảy đến đối với hoạt động tín dụng của từng TCTD, toàn ngành Ngân hàng.

Với việc CIC đã, đang và sẽ tiếp tục mở rộng nguồn thông tin thu nhập, sự đa dạng về dữ liệu đặc biệt là các dữ liệu mang lại ở dạng không có cấu trúc, thì áp dụng trí tuệ nhân tạo sẽ không còn là lựa chọn mà sẽ trở thành yêu cầu tất yếu mà CIC cần sớm nghiên cứu và phát triển trong thời gian tới.

Thứ 2 là Big data – dữ liệu lớn. Big data thường được định nghĩa như là một tập hợp dữ liệu lớn, bao gồm nhiều loại dữ liệu có cấu trúc, không có cấu trúc hoặc được

cấu trúc không đầy đủ, mà mỗi dữ liệu trong đó đều có thể được sử dụng để khai thác thành các thông tin chi tiết đòi hỏi phải cần được xử lý với tốc độ nhanh với độ chính xác cao và giá trị thông tin lớn đáp ứng được nhu cầu sử dụng.

Khi ứng dụng dữ liệu lớn, có rất nhiều chiều dữ liệu phát sinh tại mỗi điểm tiếp xúc giữa TCTD và khách hàng vay, CIC có thể tiến hành thu thập và cung cấp theo thời gian thực (Real-time).

Các sản phẩm, dịch vụ của CIC cung cấp nhằm mục tiêu hỗ trợ cho việc ra quyết định cấp tín dụng hay không của TCTD cung cấp thông tin giúp việc điều hành, quản lý chính sách của Ngân hàng Nhà nước hiệu quả hơn,...

Với hệ thống lớn mang tầm cỡ quốc gia nên giai đoạn kiểm thử là rất quan trọng, nếu thay đổi một số chức năng có thể ảnh hưởng đến nhiều chức năng khác của cả hệ thống hoặc thay đổi ở tầng kiến trúc nhưng không thay đổi trên màn hình giao diện, dẫn đến phải kiểm thử lại toàn bộ ứng dụng, chương trình. Tuy nhiên, kiểm thử các ứng dụng thay đổi này chủ yếu làm theo cách thủ công (manual) - kiểm thử bằng mắt, việc này làm tăng chi phí cho dự án, dễ bị trễ tiến độ hoặc dễ bị lọt lỗi phần mềm khi triển khai cho khách hàng sử dụng.

Ngoài ra, việc đánh chức năng chỉ thực hiện bằng cách ghi lại lịch sử hoặc giám sát các hệ thống chạy ở trên nhiều trình duyệt khác nhau, sau đó xem xét tổng hợp lỗi. Các công việc này thường được kiểm tra khi phần mềm đã được triển khai ở môi trường thật đảm bảo ứng dụng, phần mềm khi bàn giao cho khách hàng luôn là sản phẩm tốt nhất.

### ***3.2.2. Đề xuất quy trình kiểm thử tự động tại Trung tâm Thông tin Tín dụng Quốc gia Việt Nam***

Từ các khái niệm, quy trình kiểm thử được mô tả trong chương 1 và quy trình kiểm thử tự động được mô tả trong chương 2, luận văn đề xuất áp dụng quy trình kiểm thử kiểm thử tự động sử dụng công cụ selenium cho kiểm thử website tại Trung tâm Thông tin Tín dụng Quốc gia Việt Nam như sau:

### **3.2.3. Các bước để tiến hành Kiểm Thử Tự Động**

Bước 1: Khả năng sử dụng kiểm thử tự động.

Một điều chắc chắn là chúng ta không thể tự động hóa mọi việc trong kiểm thử phần mềm. Công nghệ mới, phần mềm mới khi viết ra phần mềm mà những công cụ kiểm thử tự động hiện tại chưa hỗ trợ hoặc chỉ có thể hỗ trợ một phần.

Ví dụ khi chúng ta kiểm thử một trang Web trên một trình duyệt mới, lúc đó, công cụ kiểm thử tự động chưa có phiên bản mới hỗ trợ trên trình duyệt đó. Hay như về chương trình SAP, các nhà phát triển SAP đã đưa ra một lựa chọn ngăn chặn việc chạy script tự động trên nó, và để có thể kiểm thử tự động trên SAP, chúng ta cần yêu cầu các nhà phát triển chương trình SAP gỡ bỏ lựa chọn này.

Bước 2: Lựa chọn công cụ kiểm thử tự động thích hợp.

Sau khi xác định được sản phẩm hiện có thể làm Kiểm Thử Tự Động hay không, bước tiếp theo, chúng ta cần phải xác định nên sử dụng công cụ kiểm thử tự động nào cho phù hợp. Công cụ kiểm thử tự động cho công nghệ mà sản phẩm sử dụng. Ưu nhược điểm của các công cụ. Ngôn ngữ và kịch bản nào mà công cụ kiểm thử sử dụng. Nhân sự hiện tại có quen với công cụ đó hay không?

Bước 3: Xây dựng môi trường làm việc.

Môi trường làm việc bao gồm các chu trình, thủ tục, khái niệm và môi trường như kịch bản kiểm thử thiết kế. Bên cạnh đó, môi trường làm việc cũng bao gồm cấu trúc thư mục, lưu trữ các kịch bản kiểm thử, xây dựng các mối quan hệ logic giữa các thành phần.

Bước 4: Viết kịch bản kiểm thử, thực thi và phân tích kết quả.

Dựa trên kịch bản của kiểm thử thủ công, dựa vào những hỗ trợ của các công cụ kiểm thử tự động, đây là tiền đề để chúng ta viết các đoạn mã tương tác với sản phẩm phần mềm trên các môi trường và thực thi. Sau khi thực thi chúng ta cần phải phân tích các kết quả đạt được và ghi lại những vấn đề của sản phẩm.

### **3.3. Phân tích sử dụng công cụ.**

Qua tìm hiểu thì em lựa chọn công cụ kiểm thử tự động Selenium để sử dụng cho ứng dụng website của Trung tâm Thông tin Tin dụng Quốc gia Việt Nam

### 3.3.1. Công cụ kiểm thử tự động Selenium.

#### a. Khái quát về Selenium

Selenium là một công cụ hỗ trợ kiểm thử tự động cho các ứng dụng Web. Selenium hỗ trợ kiểm thử trên hầu hết các trình duyệt phổ biến hiện nay như Firefox, Internet Explorer, Safari, ... và các hệ điều hành chủ yếu như Windows, Linux, Mac,... Selenium cũng hỗ trợ một số lớn các ngôn ngữ lập trình Web phổ biến hiện nay như C#, Java, Perl, PHP, Python, Ruby,... Công cụ này có thể kết hợp thêm với một số công cụ khác như Junit và TestNG nhưng với người dùng thông thường chỉ cần chạy tự động mà không cần cài thêm công cụ hỗ trợ [10].

#### b. Đặc điểm của Selenium

Mã nguồn mở: Đây là điểm mạnh nhất của Selenium khi so sánh với các công cụ kiểm thử khác. Không phải lo lắng về phí bản quyền hay thời hạn sử dụng vì selenium sử dụng mã nguồn mở.

Vì selenium sử dụng mã nguồn mở nên có một cộng đồng hỗ trợ mạnh mẽ. Bên cạnh đó, Selenium phát triển bởi Google nên chúng ta hoàn toàn có thể yên tâm. Tuy nhiên, ngoài điểm mạnh thì đây cũng là một điểm yếu của Selenium. Vì công cụ này miễn phí, thì cộng đồng lại đông nên một vấn đề có thể nhiều giải pháp, và có thể một số giải pháp là không hữu ích. Mặt khác, không thể hối thúc hay ra thời hạn cho sự hỗ trợ.

Selenium hỗ trợ chạy trên nhiều hệ điều hành khác nhau, hỗ trợ nhiều ngôn ngữ lập trình với mức độ chỉnh sửa script hầu như là không có. Thực sự thì điều này phụ thuộc phần lớn vào khả năng viết script của người dùng.

Chạy kịch bản kiểm thử ở phần cuối. Khi thực thi một test script, chúng ta hoàn toàn có thể làm việc khác trên cùng một máy tính. Việc này hỗ trợ chúng ta không cần tốn quá nhiều tài nguyên máy móc, thiết bị khi chạy test script.

Không hỗ trợ ứng dụng. Selenium không hỗ trợ chúng ta làm việc với các ứng dụng mà chỉ hỗ trợ chúng ta tương tác với trình duyệt. Vì vậy, để giải quyết các trường hợp cần tương tác với hệ thống hoặc một ứng dụng thứ ba, thì chúng ta cần thư viện khác như ngôn ngữ lập trình AutoIt. Selenium là công cụ hỗ trợ kiểm tra tính

năng nên không có khả năng giả lập nhiều người dùng ảo cùng một lúc. Công việc của selenium là chạy kiểm thử tự động dựa trên kịch bản kiểm thử đã được thiết kế. Từ đó chúng ta có thể biết được đối tượng kiểm thử có hoạt động đúng như mong đợi hay không?

### c. Các thành phần của Selenium

Selenium là một bộ công cụ hỗ trợ kiểm thử tự động các tính năng của ứng dụng trên nền Web, bao gồm 4 thành phần: Selenium IDE, Selenium Grid, Selenium 1.0 ( hay Selenium Remote Control – Selenium RC) và Selenium 2.0 ( hay Selenium WebDriver). Mỗi loại có một vai trò cụ thể trong việc hỗ trợ sự phát triển của tự động hóa kiểm thử ứng dụng web.

- Môi trường phát triển tích hợp (Integrated Development Environment - Selenium IDE)

Là được phát triển dưới hình thức thêm vào của Firefox. Chúng ta chỉ có thể ghi lại trên trình duyệt FireFox, nhưng bù lại, chúng ta có thể phát lại trên các trình duyệt khác như là Internet Explorer, Chrome.... Selenium có thể sinh mã tự động hoặc nạp các đoạn mã viết tay. Công cụ này cung cấp chức năng ghi và phát lại. Sau đó chạy kiểm thử lại các câu lệnh này. Chức năng này hỗ trợ tiết kiệm thời gian viết kịch bản kiểm thử. Selenium IDE hỗ trợ lưu kịch bản dưới dạng nhiều loại ngôn ngữ lập trình khác nhau như Java, PHP, C#, Ruby...

Selenium Core: Là một công cụ chạy các test script. Điểm mạnh của công cụ này là có thể chạy test script trên gần như toàn bộ các trình duyệt, nhưng lại yêu cầu được cài đặt trên máy chủ của website được kiểm tra. Điều này là không thể khi người kiểm thử không có quyền truy cập đến máy chủ đó.

- Selenium RC (Selenium Remote Control)

Là một khung kiểm thử cho phép thực hiện nhiều hơn và tuyến tính các hành động trên trình duyệt. Nó cho phép cho phép các nhà phát triển tự động hóa kiểm thử sử dụng một ngôn ngữ lập trình cho tính linh hoạt tối đa và mở rộng trong việc phát triển hợp lý thử nghiệm. Selenium RC có thể nhận các test script được thu bởi Selenium IDE, cho phép sửa, cải tiến bằng nhiều ngôn ngữ lập trình khác nhau. Tiếp

theo là khởi động một trong các trình duyệt Web được chỉ định để thực hiện kiểm thử trực tiếp trên trình duyệt đó. Selenium RC còn cung cấp khả năng lưu lại kết quả kiểm thử; cung cấp một giao diện lập trình ứng dụng (Application Programming Interface - API) và thư viện cho mỗi ngôn ngữ được hỗ trợ: HTML, Java, C#, Perl, PHP, Python và Ruby. Khả năng sử dụng Selenium RC với một ngôn ngữ lập trình bậc cao để phát triển các trường hợp kiểm thử cũng cho phép kiểm thử tự động được tích hợp với một dự án xây dựng môi trường tự động.

- Selenium WebDriver

Selenium WebDriver là sự kế thừa từ Selenium Remote Control, làm việc trực tiếp với trình duyệt ở mức hệ điều hành, cho phép gửi lệnh trực tiếp đến trình duyệt và xuất ra kết quả.

- Selenium-Grid

Selenium – Grid Là một hệ thống hỗ trợ người dùng thực hiện test script trên nhiều trình duyệt cũng một lúc và song song với nhau mà không cần phải chỉnh sửa test script.

Thực hiện phương pháp kiểm tra phân bổ, phối kết hợp nhiều Selenium RC để thực thi được trên nhiều trình duyệt Web khác nhau trong cùng một lúc nhằm giảm thiểu thời gian thực hiện.

## **Selenium WebDriver**

### **Tiền thân của Selenium WebDriver**

Trước khi Selenium WebDriver ra đời và phát triển thì Selenium RC là công cụ chính trong suốt một thời gian dài. Hiện nay, Selenium RC không được sử dụng nhiều như Selenium WebDriver nữa, tuy nhiên người dùng vẫn có thể tiếp tục phát triển các kịch bản kiểm thử với Selenium RC.

Selenium RC là công cụ phục vụ cho các công việc kiểm thử đòi hỏi nhiều hơn việc thao tác với các website trên giao diện . Nó cho phép viết các kịch bản kiểm thử tự động ứng dụng Web với sự hỗ trợ của các ngôn ngữ lập trình như Java, C#, Python, Perl, PHP để tạo ra các trường hợp kiểm thử phức tạp hơn như đọc và viết các tệp tin, truy vấn cơ sở dữ liệu, gửi mail kết quả kiểm thử.



Các thành phần của Selenium RC gồm:

- **Máy chủ Selenium:** Thực hiện phân tích và chạy các lệnh được gửi đến từ ứng dụng cần kiểm thử và các thao tác như giao thức truyền tải proxy, phân tích và xác minh các thông điệp HTTP, giữa trình duyệt và ứng dụng cần kiểm tra.
- **Các thư viện máy khách:** cung cấp sự hỗ trợ lập trình cho phép chạy lệnh Selenium từ chương trình. Các thư viện máy khách hỗ trợ các ngôn ngữ lập trình khác nhau thì khác nhau. Giao diện lập trình là tập các chức năng chạy lệnh Selenium, trong mỗi giao diện có một chức năng lập trình hỗ trợ Selenium.

### **Đặc trưng của Selenium WebDriver**

Selenium WebDriver (hay còn gọi là Selenium 2.0) kế thừa và phát triển từ Selenium IDE, Selenium RC, Selenium Grid. Selenium WebDriver tương tác trực tiếp với trình duyệt mà không cần thông qua bất kỳ trung gian, không giống như Selenium RC phụ thuộc vào một máy chủ. Tính năng quan trọng trong Selenium 2.0 là tích hợp WebDriver API. Ngoài việc xử lý một số hạn chế trong Selenium RC API, Selenium WebDriver còn được thiết kế một giao diện lập trình đơn giản hơn. Nó cho phép sử dụng một số ngôn ngữ lập trình như HTML, Net, Java, Ruby, Perl... để tạo kịch bản kiểm thử kết hợp với sử dụng các điều kiện, vòng lặp... khiến cho test script trở nên chính xác hơn. Selenium WebDriver còn được phát triển tốt hơn để hỗ trợ cho các trang Web động do các phần tử trong một trang Web động có thể thay đổi bất cứ lúc nào, ngay cả khi trang đó không được tải lại. Vì vậy, Selenium WebDriver được phát triển để hỗ trợ quá trình kiểm thử mà không cần phải thực hiện lại khi có thay đổi xảy ra.

Selenium phù hợp với giai đoạn kiểm thử Hồi quy và những đợt hệ thống đã hoạt động nhưng có thay đổi và chỉnh sửa.

### **3.3.2. Một số tính năng nổi bật chung:**

Có thể kết hợp được với nhiều ngôn ngữ lập trình: Java, .Net, Ruby, Python, Perl, Selenium có thể giả lập thao tác người dùng trên web page và các web element, Có thể kiểm chứng và so sánh thông tin trên các web page, Script được base theo HTML,

dễ học và ứng dụng cũng rất nhanh, có thể tạo ra một bộ test bao gồm nhiều test case, có thể chạy các test suite thông qua Selenium IDE hoặc Selenium command line, selenium API hỗ trợ trên nhiều loại trình duyệt, do đó sẽ rất dễ dàng khi test các ứng dụng web với các trình duyệt khác nhau.

#### 3.3.2.1. Selenium phù hợp với giai đoạn test nào?

Selenium phù hợp với kiểm thử hồi quy, và những đợt hệ thống đã hoạt động nhưng có thay đổi, chỉnh sửa cần test lại

### 3.3.3. Các câu lệnh sử dụng trong Selenium WebDriver

#### a. Các câu lệnh trình duyệt

Một số câu lệnh thao tác của Selenium WebDriver như trình duyệt như mở, đóng và lấy tiêu đề của trang Web như dưới đây:

- Câu lệnh get

Mục đích: Sử dụng để mở một trang Web mới trong trình duyệt lựa chọn. Cú pháp: `drive.get(URL);`

Trong đó URL: Là dùng để tải trang, nên sử dụng một url đầy đủ

- Câu lệnh để lấy tiêu đề getTitle

Mục đích: Sử dụng để lấy tiêu đề trang web muốn lấy

Cú pháp: `driver.getTitle();`

- Câu lệnh lấy URL hiện tại getCurrentUrl

Mục đích: Câu lệnh này sử dụng để lấy URL của trang hiện tại.

Cú pháp: `driver.getCurrentUrl();`

- Câu lệnh lấy nguồn của trang Web getPageSource

Mục đích: Câu lệnh này dùng để lấy nguồn của trang được tải cuối cùng.

Cú pháp: `driver.getPageSource();`

#### b. Các câu lệnh WebElement

Để tương tác với trang Web thì việc đầu tiên cần phải xác định vị trí của các phần tử trên trang Web, WebDriver cung cấp 2 phương thức “Find Element” and “Find Elements” để xác định vị trí của các phần tử trên trang Web.

### Phương thức “Find Element” và “Find Elements”

Phương thức “Find Element” và “Find Elements” là trả về đối tượng WebElement, trả về một danh sách WebElement, có thể sẽ trả về danh sách rỗng nếu không có phần tử DOM phù hợp với truy vấn. Phương thức “Find” lấy vị trí hoặc đối tượng truy vấn gọi bằng phương thức “By”

#### - Tìm phần tử bằng ID: By ID

Mục đích: Tìm vị trí của phần tử bằng ID, nếu tìm được ID phù hợp sẽ trả về vị trí của phần tử, nếu không có phần tử phù hợp với id sẽ xuất hiện

NoSuchElementException Cú pháp: `driver.findElement(By.id(""))`;

#### - Tìm phần tử bằng Name (By Name)

Mục đích: Tìm vị trí của phần tử bằng name, nếu tìm được giá trị name phù hợp sẽ trả về vị trí của phần tử, nếu không có phần tử phù hợp với thuộc tính name sẽ xuất hiện NoSuchElementException.

Cú pháp: `driver.findElement(By.name(""))`;

#### - Tìm phần tử bằng Class Name

Mục đích: Tìm các phần tử dựa trên giá trị của thuộc tính chúng “class”.

Cú pháp: `driver.findElement(By.className(""))`;

#### - Tìm phần tử bằng Link Text

Mục đích: Tìm phần tử thẻ a bằng tên link ,

Cú pháp: `driver.findElement(By.tagName(""))`;

### **c. Các câu lệnh điều hướng trình duyệt**

#### - Câu lệnh forward

Mục đích: Dùng để đi đến trang tiếp theo.

Cú pháp: `driver.navigate().forward()`;

#### - Câu lệnh back: Lệnh này dùng để quay lại về trang trước.

Cú pháp: `driver.navigate().back()`;

#### - Câu lệnh refresh

Mục đích: Dùng để làm mới trang hiện tại .

Cú pháp: `driver.navigate().refresh()`;

#### **d. Các lệnh switch**

Một số trang Web có nhiều khung hoặc nhiều cửa sổ. Selenium WebDriver gán ID cho mỗi cửa sổ ngay khi đối tượng WebDriver được khởi tạo. ID này được gọi là cửa sổ xử lý.

Sau đây là một số câu lệnh switch:

- Câu lệnh `getWindowHandle`

Mục đích: Dùng gọi cửa sổ xử lý của một cửa sổ đang dùng hiện tại.

Cú pháp: `driver.getWindowHandle();`

- Câu lệnh `getWindowHandles`

Mục đích: Dùng để gọi cửa sổ xử lý (window handle) của nhiều cửa sổ hiện tại.

Cú pháp: `driver.getWindowHandles();`

- Câu lệnh `Switch To Window`

Mục đích: Dùng hỗ trợ di chuyển giữa các cửa sổ khác nhau thông qua tên của chúng bằng cách sử dụng phương thức “switchTo”.

Cú pháp: `driver.switchTo().window("windowName");`

#### **e. Các câu lệnh wait**

- Câu lệnh `implicitlyWait`

Mục đích: thời gian chờ khi không tìm thấy các phần tử trên trang web.

Cú pháp: `drive.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);`

- Câu lệnh `pageLoadTimeout`

Mục đích: Thiết lập thời gian chờ cho trang Web hoàn thành tải trước khi báo lỗi.

Cú pháp: `driver.manage().timeouts().pageLoadTimeout(100, SECONDS);`

- Câu lệnh `setScriptTimeout`

Mục đích: Thiết lập thời gian chờ một kịch bản(script) không đồng bộ để kết thúc việc thực hiện trước khi báo lỗi.

Cú pháp: `driver.manage().timeouts().setScriptTimeout(100,SECONDS);`

- Câu lệnh `sleep`

Mục đích: Câu lệnh này được sử dụng rất ít, vì nó luôn buộc các trình duyệt chờ đợi một thời gian cụ thể. Cú pháp: `thread.sleep(1000);`

### **3.4. Lý do để lựa chọn Selenium để kiểm thử ứng dụng web cho CIC.**

#### **a. Hỗ trợ ngôn ngữ và framework**

Selenium hỗ trợ tất cả các ngôn ngữ chính như Java, Python, JavaScript, C #, Ruby và ngôn ngữ lập trình Perl để kiểm thử tự động.

Có thể viết tập lệnh kiểm thử tự động bằng bất kỳ ngôn ngữ lập trình nào và Selenium sẽ chuyển đổi nó thành mã tương thích Selenium ngay lập tức. Vì vậy, không cần lo lắng về việc linh động trong các ngôn ngữ lập trình. Ngoài ra, mọi ngôn ngữ được hỗ trợ Selenium đều có các framework chuyên dụng giúp viết kịch bản kiểm thử cho tự động. Vì vậy, khi chọn Selenium như một công cụ để sử dụng trong automation test cho đơn vị, em không phải lo lắng về hỗ trợ ngôn ngữ và framework.

#### **b. Nguồn mở sẵn có**

Selenium là tính sẵn có của nguồn mở. Vì vậy, là một công cụ nguồn mở, Selenium là một khung tự động có thể truy cập công khai và miễn phí, không có chi phí trả trước. Vì vậy, CIC có thể tiết kiệm tiền ở đây và sử dụng chúng cho các mục đích khác.

Selenium giúp các lập trình viên và kỹ sư kiểm thử phần mềm tự động hóa các tính năng và chức năng của trình duyệt web. Selenium là nguồn mở cũng giúp tùy chỉnh mã để quản lý mã tốt hơn và tăng cường chức năng của các hàm và lớp được xác định trước. Selenium là công cụ kiểm thử tự động đáng tin cậy.

#### **c. Hỗ trợ nhiều trình duyệt**

Kịch bản của Selenium sử dụng cho tất cả các trình duyệt. Theo các trình duyệt StatCount, Safari, Internet Explorer, Chrome, Firefox, Opera và Edge là các trình duyệt được sử dụng nhiều nhất trên toàn thế giới và tập lệnh Selenium tương thích với tất cả các trình duyệt. Khi sử dụng không cần viết lại các tập lệnh cho mọi trình duyệt, chỉ một tập lệnh cho tất cả các trình duyệt.

#### **d. Hỗ trợ hoạt động trên nhiều hệ điều hành**

Vì nhiều người khác nhau nên sử dụng các hệ điều hành khác nhau. Selenium vẫn là một công cụ có tính di động cao hỗ trợ và có thể hoạt động trên các hệ điều hành khác nhau như Windows, Linux, Mac OS, UNIX...

Khi làm việc kiểm thử tự động có thể tạo bộ kiểm thử Selenium trên bất kỳ nền tảng nào như Windows và có thể thực thi cùng bộ kiểm thử trên nền tảng khác, ví dụ: Mac hoặc Linux. Điều này cho phép các lập trình viên và tester viết các kịch bản kiểm thử tự động dễ dàng hơn và không chú trọng nhiều đến nền tảng mà nó sẽ chạy.

#### **e. Dễ thực hiện**

Khung tự động Selenium là công cụ rất dễ sử dụng khi làm về kiểm thử tự động. Selenium cung cấp giao diện thân thiện với người dùng, giúp tạo và thực thi các tập lệnh kiểm thử một cách dễ dàng và hiệu quả.

#### **f. Tái sử dụng và tích hợp**

Bộ kiểm thử tự động Selenium có thể tái sử dụng và có thể được kiểm thử trên nhiều trình duyệt và hệ điều hành. Tuy nhiên, vấn đề khó khăn là Selenium không phải là một công cụ kiểm thử tự động toàn diện. Do đó, nó cần các framework và tiện ích của bên thứ ba để mở rộng phạm vi kiểm thử.

#### **j. Tính linh hoạt**

Quản lý kiểm thử là rất quan trọng trong vòng đời của kiểm thử phần mềm. Nó trở nên dễ dàng và hiệu quả hơn với các tính năng Selenium như tập hợp lại và tái cấu trúc các trường hợp kiểm thử. Điều này giúp các lập trình viên và tester thay đổi mẫu một cách nhanh chóng, giảm trùng lặp, giảm thiểu các biến chứng và cải thiện khả năng bảo trì. Các tính năng này làm cho Selenium linh hoạt hơn và có thể sử dụng được so với các công cụ kiểm thử tự động khác và do đó giúp Selenium giữ được lợi thế.

#### **k. Thực hiện kiểm thử song song và tiếp thị nhanh hơn**

Mục đích chính của kiểm thử tự động là để tiết kiệm thời gian và nỗ lực. Với sự trợ giúp của Selenium Grid, kiểm thử tự động có thể thực hiện song song nhiều kiểm thử, do đó giảm thời gian thực hiện kiểm thử. Với sự trợ giúp của các lưới điện toán đám mây để kiểm thử trình duyệt chéo, tester có thể kiểm thử trên hàng trăm trình duyệt song song bằng Selenium, do đó giúp tiết kiệm thời gian.

#### **l. Sử dụng phần cứng ít hơn**

Nếu so sánh Selenium với các công cụ kiểm thử tự động tập trung của nhà cung

cấp khác như QTP, UFT, SilkTest, thì chắc chắn chúng ta sẽ thấy Selenium yêu cầu ít phần cứng hơn so với các công cụ kiểm thử khác.

#### **m. Dễ tìm hiểu và sử dụng**

Các tập lệnh Selenium không giống như viết thuật toán phức tạp trăm trang. Viết tập lệnh Selenium không hơn là viết một vài đoạn mã để kiểm thử tự động các chức năng của trang web của bạn. Ngoài ra với tiện ích mở rộng Selenium IDE trên trình duyệt Firefox, bạn có thể sử dụng chức năng ghi và phát để tạo tập lệnh Selenium để tham khảo trong tương lai.

#### **n. Cập nhật liên tục**

Vì Selenium được hỗ trợ bởi một cộng đồng và cộng đồng Selenium cũng liên tục phát hành các bản cập nhật và nâng cấp liên tục. Phần tốt nhất để có một cộng đồng là những nâng cấp này có sẵn và dễ hiểu.

### **3.5. Xây dựng Testcase để thực hiện kiểm thử.**

#### **3.5.1. Test case là gì?**

Test case là mô tả một dữ liệu đầu vào (input), hành động (action) hoặc một sự kiện (event) và kết quả truy vấn (expected response). Mục đích của test case là kiểm tra từng chức năng của ứng dụng phần mềm hoạt động đúng hay không.

Test case có thể có nhiều phần khác nhau như mã test case, tên test case, mục tiêu kiểm thử, các điều kiện kiểm thử, dữ liệu đầu vào, các bước thực hiện và kết quả mong đợi.

Quá trình hình thành và phát triển test case có thể giúp tìm ra lỗi trong các yêu cầu hoặc trong thiết kế của ứng dụng, phần mềm, vì nó đòi hỏi phải tư duy hoàn toàn thông qua các hoạt động của ứng dụng, phần mềm. Vì vậy, việc chuẩn bị test case sớm là cần thiết trong quy trình phát triển phần mềm.

#### **3.5.2. Vai trò của test case**

Test case là rất quan trọng trong bất kỳ dự án phần mềm nào vì đây là bước đầu tiên trong quá trình kiểm thử và nếu có gì đó sai sót ở bước này sẽ kéo theo nhiều hệ quả ở các giai đoạn tiếp theo trong vòng đời kiểm thử.

Test case là những tình huống yêu cầu của khách hàng. Là việc quan trọng để xác định những thay đổi mà khách hàng mong muốn.

Một bản test case bắt buộc phải có từng trạng thái riêng để trưởng nhóm biết được chức năng nào của ứng dụng đã được test và chưa được test, những chức năng nào nhiều lỗi và ngược lại.

### ***3.5.3. Cấu trúc cơ bản của test case***

Tùy thuộc vào từng template của từng công ty nên cấu trúc của test case sẽ khác nhau. Tuy nhiên, về cơ bản thì test case của CIC bao gồm các thành phần chính sau đây

### ***3.5.4. Những bước quan trọng để bạn viết được một test case hiệu quả.***

Đầu tiên, phải xác định được phạm vi và mục đích của việc test là gì. Xác định những điểm có thể test, hiểu mục đích của việc test, bạn phải hiểu được nội dung yêu cầu của khách hàng.

Xác định việc bắt đầu test. Một file test case thường phải có: Các chức năng cơ bản, có khả năng chịu lỗi cao và hiệu năng của file test, giao diện trực quan. Dựa vào các yêu cầu của hệ thống để có hướng thiết kế file test case tốt nhất.


File test case cần có những bước đơn giản, minh bạch, dễ hiểu, các trường hợp thử nghiệm nên có giá trị, tóm tắt và ngắn, test case nên có sự liên kết, test case có thể bảo trì, chuẩn bị dữ liệu test. Dữ liệu test đa dạng ứng với các trường hợp kiểm thử. Các dữ liệu hợp lệ, không hợp lệ, dữ liệu lỗi.



### 3.5.5. Test case kiểm thử web CIC.

Dưới đây là một số test case mà em viết để phục vụ việc kiểm thử một phần đối với Trang web dành cho các cá nhân vào đăng ký người dùng khai thác báo cáo tại Trung tâm Thông tin Tín dụng Quốc gia Việt Nam

Website: <https://cic.gov.vn>

 <b>TRUNG TÂM THÔNG TIN TÍN DỤNG</b>	
Dự án	CIC
Tên màn hình/ Tên chức năng	Phân hệ khách hàng vay
Mã trường hợp kiểm thử	KHV
Tài liệu tham khảo	
Kiểu test:	Kiểm thử tự động
Ngày test:	25/03/2021
Người viết test case	Ngocpt
Dữ liệu test	

Mã TC	Tình huống thực hiện	Các bước thực hiện	Kết quả mong đợi	Tình trạng	Cán bộ test	Ghi chú
KHV_1	Truy cập trình duyệt	Truy cập bằng trình duyệt Chrome, Truy cập bằng trình duyệt Firefox,	Truy cập thành công trên tất cả các trình duyệt		ngocpt	
KHV_2	Đăng nhập	Vào đường link trang website: <a href="https://cic.gov.vn/">https://cic.gov.vn/</a> Click “Đăng nhập” Xuất hiện màn hình đăng nhập Điền thông tin: - Tên tài khoản - Mật khẩu Click “Đăng nhập”	Đăng nhập thành công. Màn hình xuất hiện thông báo “Về việc hỗ trợ TCTD và khách hàng do ảnh hưởng dịch covid - 19”		ngocpt	
KHV_3		Vào đường link trang website: <a href="https://cic.gov.vn/">https://cic.gov.vn/</a> Click “Đăng nhập” Xuất hiện màn hình đăng nhập Điền thông tin	Màn hình thông báo “Vui lòng nhập đủ thông tin tài khoản và mật khẩu		ngocpt	

		- Tên tài khoản: để trống - Mật khẩu: để trống Click “Đăng nhập”				
KHV_4		Vào đường link trang website: <a href="https://cic.gov.vn/">https://cic.gov.vn/</a> Click “Đăng nhập” Xuất hiện màn hình đăng nhập Điền thông tin - Tên tài khoản: nhập đúng thông tin - Mật khẩu: để trống Click “Đăng nhập”	Màn hình thông báo “Vui lòng nhập đủ thông tin tài khoản và mật khẩu”		ngocpt	
KHV_5		Vào đường link trang website: <a href="https://cic.gov.vn/">https://cic.gov.vn/</a> Click “Đăng nhập” Xuất hiện màn hình đăng nhập Điền thông tin - Tên tài khoản: nhập không đúng thông tin - Mật khẩu: nhập không đúng thông tin Click “đăng nhập”	Màn hình thông báo “Đăng nhập thất bại! Quý khách vui lòng kiểm tra thông tin đăng nhập và thử lại.		ngocpt	

KHV_6		<p>Đăng nhập thành công click chọn “đổi mật khẩu”</p> <p>Xuất hiện màn hình “đổi mật khẩu đăng nhập”</p> <p>Điền thông tin vào trường bắt buộc</p> <ul style="list-style-type: none"> <li>- Mật khẩu hiện tại: điền đúng mật khẩu hiện tại</li> <li>- Nhập mật khẩu: nhập mật khẩu mới theo đúng yêu cầu</li> <li>- Nhập lại mật khẩu: nhập lại mật khẩu trùng với phần nhập mật khẩu</li> </ul> <p>Click nút “lưu”</p>	Màn hình xuất hiện thông báo “Đổi mật khẩu thành công”		ngocpt	
KHV_7	Đổi mật khẩu	<p>Đăng nhập thành công click chọn “đổi mật khẩu”</p> <p>Xuất hiện màn hình “đổi mật khẩu đăng nhập”</p> <p>Điền thông tin vào trường bắt buộc</p> <ul style="list-style-type: none"> <li>- Mật khẩu hiện tại: điền không đúng mật khẩu</li> <li>- Nhập mật khẩu: nhập mật khẩu mới theo đúng yêu cầu</li> <li>- Nhập lại mật khẩu: nhập lại mật khẩu trùng với phần nhập mật khẩu</li> </ul> <p>Click nút “Lưu”</p>	Màn hình thông báo “Có lỗi xảy ra! không thể thay đổi mật khẩu”		ngocpt	
KHV_8		<p>Đăng nhập thành công click chọn “đổi mật khẩu”</p> <p>Xuất hiện màn hình “đổi mật khẩu đăng nhập”</p> <p>Điền thông tin vào trường bắt buộc</p>	Màn hình thông báo “Mật khẩu xác nhận không trùng khớp”		ngocpt	

		<ul style="list-style-type: none"> <li>- Mật khẩu hiện tại: điền đúng mật khẩu hiện tại</li> <li>- Nhập mật khẩu: nhập mật khẩu mới theo đúng yêu cầu</li> <li>- Nhập lại mật khẩu: nhập lại mật khẩu không trùng với trường nhập mật khẩu</li> </ul> <p>Click nút “Lưu”</p>				
KHV_9		<p>Đăng nhập thành công click chọn “đổi mật khẩu”</p> <p>Xuất hiện màn hình “đổi mật khẩu đăng nhập”</p> <p>Điền thông tin vào trường bắt buộc</p> <ul style="list-style-type: none"> <li>- Mật khẩu hiện tại: để trống</li> <li>- Nhập mật khẩu: để trống</li> <li>- Nhập lại mật khẩu: để trống</li> </ul> <p>Click nút “lưu”</p>	Nút “Lưu” không click được		ngocpt	
KHV_10	Khai thác báo cáo	<p>Sau khi đăng nhập click chọn mục khai thác báo cáo</p> <p>Màn hình xuất hiện mục “Khai thác báo cáo tín dụng”</p> <p>Click chọn nút “Tra cứu báo cáo”</p> <p>Màn hình xuất hiện mục “Xác nhận tra cứu báo cáo”</p> <p>Click nút “Xác nhận”</p> <p>Màn hình xuất hiện mục “Xác thực mã OTP” và thông báo “mã OTP được gửi vào điện thoại của quý khách”</p>	Màn hình xuất hiện báo cáo thông tin tín dụng khách hàng vay cá nhân		ngocpt	

		<p>Mở điện thoại nhập mã OTP đúng vào phần xác thực mã OTP</p> <p>Click vào nút “Thực hiện”</p> <p>Chương trình trở về màn hình “danh sách màn hình đã báo cáo”</p> <p>Click node “Xem”</p>				
KHV_11		<p>Sau khi đăng nhập click chọn mục khai thác báo cáo</p> <p>Màn hình xuất hiện mục “Khai thác báo cáo tín dụng”</p> <p>Click chọn nút “Tra cứu báo cáo”</p> <p>Màn hình xuất hiện mục “Xác nhận tra cứu báo cáo”</p> <p>Click nút “Xác nhận”</p> <p>Màn hình xuất hiện mục “xác thực mã OTP” và thông báo “Mã OTP được gửi vào điện thoại của quý khách”</p> <p>Mở điện thoại nhập mã OTP đúng vào phần xác thực mã OTP</p> <p>Click vào nút “Thực hiện”</p> <p>Chương trình trở về màn hình “danh sách màn hình đã báo cáo”</p> <p>Click node “Tải về”</p>	<p>Chương trình sẽ tải file PDF về máy</p>		ngocpt	

KHV_12		<p>Sau khi đăng nhập click chọn mục khai thác báo cáo</p> <p>Màn hình xuất hiện mục “Khai thác báo cáo tín dụng”</p> <p>Click chọn nút “tra cứu báo cáo”</p> <p>Màn hình xuất hiện mục “Xác nhận tra cứu báo cáo”</p> <p>Click nút “Hủy bỏ”</p>	<p>Màn hình trở về mục</p> <p>“Khai thác báo cáo tín dụng”</p>		ngocpt	
KHV_13		<p>Trường hợp khai thác báo cáo trong cùng một ngày (chưa có biến động)</p> <p>Sau khi đăng nhập click chọn mục khai thác báo cáo</p> <p>Màn hình xuất hiện mục “Khai thác báo cáo tín dụng”</p> <p>Click chọn nút “Tra cứu báo cáo”</p>	<p>Màn hình thông báo “Quý khách hàng đã thực hiện khai thác thành công báo cáo trong cùng ngày. Quý khách có chắc chắn muốn tiếp tục thực hiện khai thác báo cáo không?”</p> <p>Click “Đồng ý” chương trình xuất hiện “Xác nhận tra cứu báo cáo”</p>		ngocpt	
KHV_14		<p>Trường hợp khai thác báo cáo trong cùng một ngày (chưa có biến động)</p> <p>Sau khi đăng nhập click chọn mục khai thác báo cáo</p> <p>Màn hình xuất hiện mục “Khai thác báo cáo tín dụng”</p>	<p>Màn hình đóng thông báo</p>		ngocpt	

		Click chọn nút “Tra cứu báo cáo” Click nút “Đóng”				
--	--	--	--	--	--	--



### 3.6. Xây dựng phần mềm và ứng dụng

Ở phần này em đã viết chương trình phục vụ cho kiểm thử tự động trang web của Trung tâm Thông tin Tin dụng Quốc gia Việt Nam, sử dụng ngôn ngữ java để phát triển dựa trên công cụ selenium,

#### 3.6.1. Xây dựng chương trình kiểm thử tự động

Cấu trúc của phần mềm kiểm thử tự động gồm có các tầng (gói) chính sau

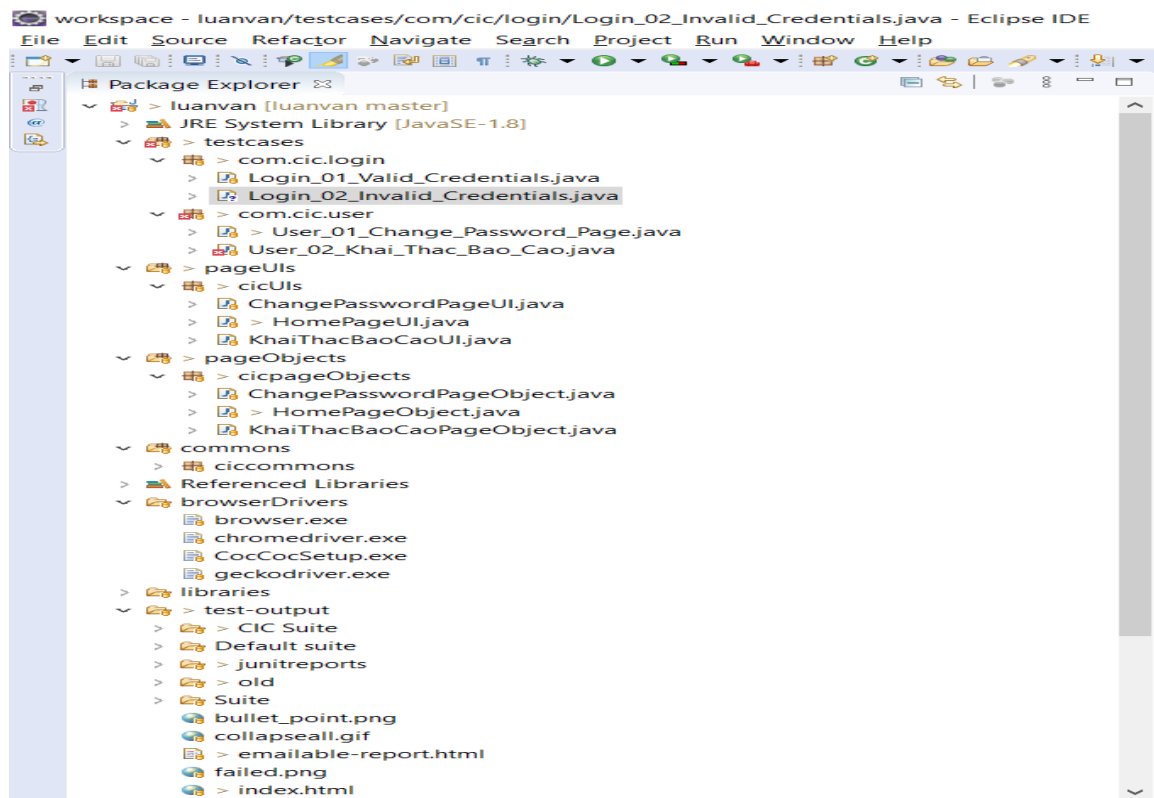
Tầng 1: Testcases: quản lý các Module / function trong dự án

Tầng 2: PageUIs: Quản lý UI, HTML, Locator

Tầng 3: PageObjects: Quản lý các actions trong tầng trang

Tầng 4: Commons: các function dùng chung

Hình [3.2] dưới đây là hình ảnh cấu trúc phần mềm được xây dựng

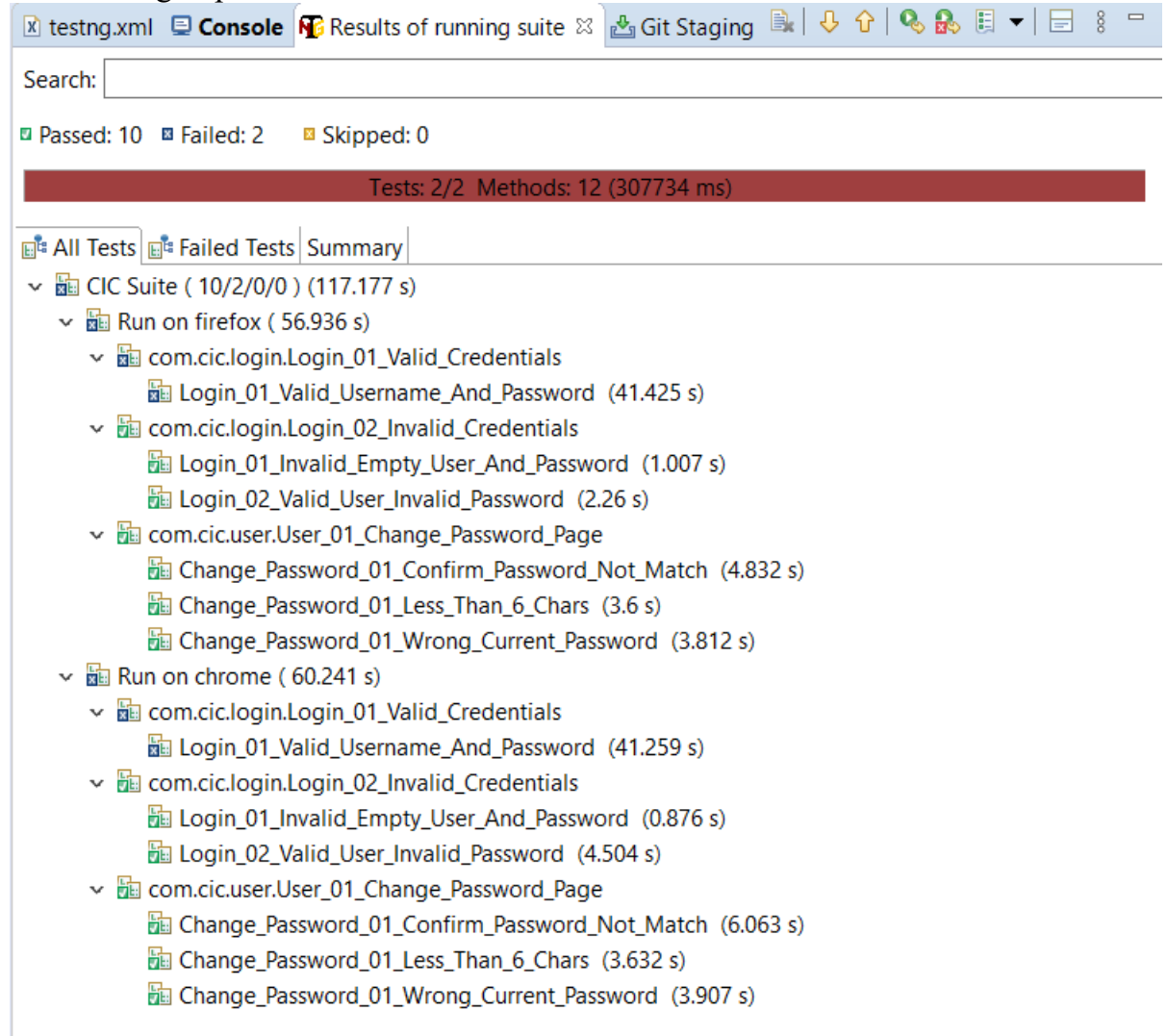


Hình 3.2 Các tầng cấu trúc của dự án

### 3.6.2. Xuất báo cáo kết quả

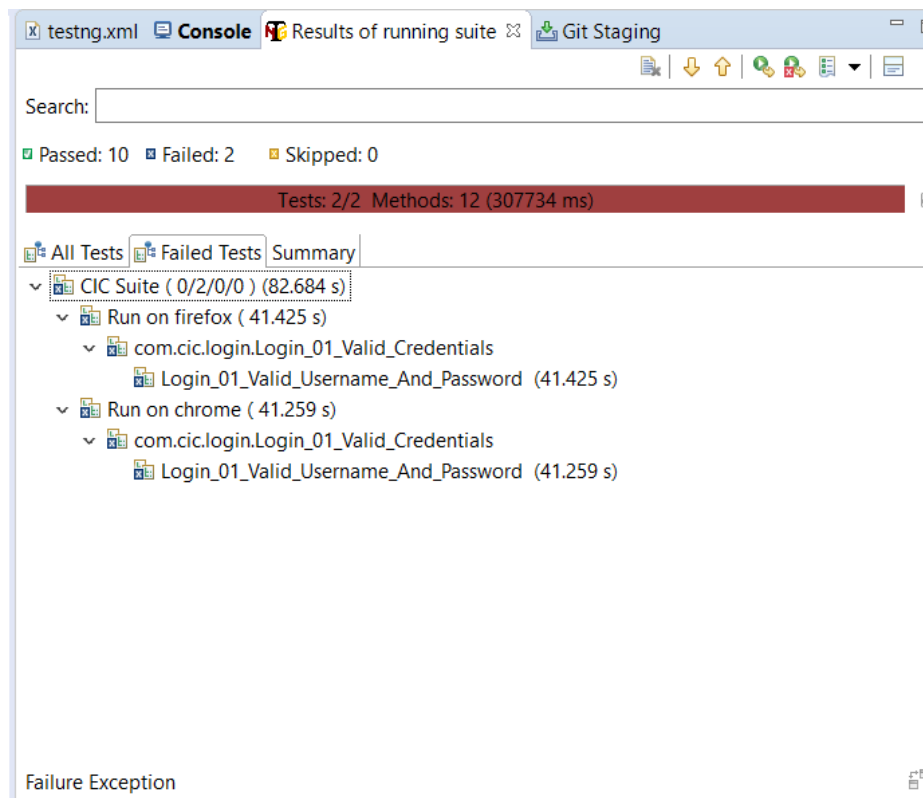
Để biết được việc chạy kiểm thử tự động thì người kiểm thử sẽ xem báo lỗi ở phần kết quả này, tổng số lượng case lỗi, số case không lỗi, tổng số case đã chạy ....

- Tổng hợp toàn bộ ca kiểm thử



All tests: thể hiện các testcase được chạy, những case có dấu X màu xanh dương là những case lỗi, còn những case dấu tích màu xanh lá cây là những case không bị lỗi. Khi nhìn vào bản này người xem sẽ biết được lỗi ở case nào, và sửa ở case đó.

- Tổng hợp những case lỗi (failed tests)



Failed tests: lọc những case bị lỗi, với dự án lớn có nhiều case thì việc tổng hợp các case lỗi sẽ giúp người kiểm thử nhìn ra các case lỗi dễ dàng hơn.

- Tóm tắt (summary)

Test name	Time (seconds)	Class count	Method count
Run on chrome	60.241	3	6
Run on firefox	56.936	3	6

Qua bảng tóm tắt chúng ta thấy chương trình chạy kiểm thử 2 tên kiểm thử (test name) chạy trên test name chrome (run on chrome) tổng thời gian chạy 60.241s số lớp (class count) 3 và số hàm được chạy (method count) là 6. Chương trình chạy trên test name firefox (run on firefox), tổng thời gian 56.936s, số lớp (class count) 3 và số hàm được chạy (method count) là 6.

## Kết quả kiểm thử (test results)

- Kết quả tổng thể

**CIC Suite**

**Info**

- F:\luanvan\eclipse\workspace\luanvan\testng.xml
- 2 tests
- 0 groups
- Times
- Reporter output
- Ignored methods
- Chronological view

**Results**

- 12 methods, 2 failed, 10 passed
- Failed methods (hide)
  - ✗ Login\_01\_Valid\_Username\_And\_Password
  - ✗ Login\_01\_Valid\_Username\_And\_Password
- Passed methods (hide)
  - ✓ Change\_Password\_01\_Confirm\_Password\_Not\_Match
  - ✓ Change\_Password\_01\_Confirm\_Password\_Not\_Match
  - ✓ Change\_Password\_01\_Less\_Than\_6\_Chars
  - ✓ Change\_Password\_01\_Less\_Than\_6\_Chars
  - ✓ Change\_Password\_01\_Wrong\_Current\_Password
  - ✓ Change\_Password\_01\_Wrong\_Current\_Password
  - ✓ Login\_01\_Invalid\_Empty\_User\_And\_Password
  - ✓ Login\_01\_Invalid\_Empty\_User\_And\_Password
  - ✓ Login\_02\_Valid\_User\_Invalid\_Password
  - ✓ Login\_02\_Valid\_User\_Invalid\_Password

- Thông tin
  - ✓ Đường link lưu trữ file kiểm thử :  
<F:\luanvan\eclipse\workspace\luanvan\testng.xml>
  - ✓ Các kiểm thử cho CIC suite: chạy trên 2 trình duyệt firefox, chrome
- Thời gian kiểm thử (time)

Bảng báo cáo kết quả kiểm thử.

Times for CIC Suite			
Total running time: 1 minutes			
Number	Method	Class	Time (ms)
0	Login_01_Valid_Username_And_Password	com.cic.login.Login_01_Valid_Credentials	41,425
1	Login_01_Valid_Username_And_Password	com.cic.login.Login_01_Valid_Credentials	41,259
2	Change_Password_01_Confirm_Password_Not_Match	com.cic.user.User_01_Change_Password_Page	6,063
3	Change_Password_01_Confirm_Password_Not_Match	com.cic.user.User_01_Change_Password_Page	4,832
4	Login_02_Valid_User_Invalid_Password	com.cic.login.Login_02_Invalid_Credentials	4,504
5	Change_Password_01_Wrong_Current_Password	com.cic.user.User_01_Change_Password_Page	3,907
6	Change_Password_01_Wrong_Current_Password	com.cic.user.User_01_Change_Password_Page	3,812
7	Change_Password_01_Less_Than_6_Chars	com.cic.user.User_01_Change_Password_Page	3,632
8	Change_Password_01_Less_Than_6_Chars	com.cic.user.User_01_Change_Password_Page	3,600
9	Login_02_Valid_User_Invalid_Password	com.cic.login.Login_02_Invalid_Credentials	2,260
10	Login_01_Invalid_Empty_User_And_Password	com.cic.login.Login_02_Invalid_Credentials	1,007
11	Login_01_Invalid_Empty_User_And_Password	com.cic.login.Login_02_Invalid_Credentials	876

Number: Số thứ tự

Method: các phương thức

Class: các lớp

Time: thời gian chạy các case

- Các kết quả
- Kết quả bao gồm:
  - ✓ Phương thức (methods): 12
  - ✓ Kết quả lỗi (failed): 2



Login\_01\_Valid\_Username\_And\_Password



Login\_01\_Valid\_Username\_And\_Password

- ✓ Kết quả đúng (passed): 10



Change\_Password\_01\_Confirm\_Password\_Not\_Match



Change\_Password\_01\_Confirm\_Password\_Not\_Match



Change\_Password\_01\_Less\_Than\_6\_Chars



Change\_Password\_01\_Less\_Than\_6\_Chars



Change\_Password\_01\_Wrong\_Current\_Password



Change\_Password\_01\_Wrong\_Current\_Password



Login\_01\_Invalid\_Empty\_User\_And\_Password



Login\_01\_Invalid\_Empty\_User\_And\_Password



Login\_02\_Valid\_User\_Invalid\_Password



Login\_02\_Valid\_User\_Invalid\_Password

- File tổng hợp báo cáo các ca kiểm thử và kết quả sau khi kiểm thử

STT	Test name	Các ca kiểm thử	Kết quả	Thời gian chạy	Lý do
1	Chạy trên trình duyệt firefox	Login_01_Valid_Username_And_Password	failed	41.425s	Nhập số liệu không đúng
2		Change_Password_01_Confirm_Password_Not_Match	pass	4.832s	
3		Change_Password_01_Less_Than_6_Chars	pass	3.6s	
4		Change_Password_01_Wrong_Current_Password	pass	3.812s	
5		Login_01_Invalid_Empty_User_And_Password	pass	1.007s	
6		Login_02_Valid_User_Invalid_Password	pass	2.26s	
7	Chạy trên trình duyệt chrome	Login_01_Valid_Username_And_Password	failed	41.259s	Nhập số liệu không đúng
8		Change_Password_01_Confirm_Password_Not_Match	pass	6.063s	
9		Change_Password_01_Less_Than_6_Chars	pass	3.632s	
10		Change_Password_01_Wrong_Current_Password	pass	3.907s	
11		Login_01_Invalid_Empty_User_And_Password	pass	0.876s	
12		Login_02_Valid_User_Invalid_Password	pass	4.504s	

**Bảng 3.1 Tổng kết lỗi**

## **KẾT LUẬN CHƯƠNG.**

Trong chương này học viên đã lựa chọn được công cụ, cũng như phương thức để thực hiện kiểm thử tự động đối với website của Trung tâm Thông tin Tín dụng Quốc gia Việt Nam, cũng trong phần này, học viên đã xây dựng được một số kịch bản kiểm thử, xây dựng hệ thống kiểm thử và đã chạy thành công trên môi trường thật của đơn vị.

## KẾT LUẬN

Sử dụng kiểm thử tự động tại Trung tâm thông tin tín dụng Quốc gia Việt nam là yêu cầu cần thiết và phù hợp trong sự phát triển của thời đại công nghệ kỹ thuật số. Ứng dụng / chương trình được xây dựng, cập nhật để đáp ứng những đòi hỏi nhu cầu của khách hàng, nâng cấp cơ sở dữ liệu của Trung tâm thông tin tín dụng Quốc gia Việt Nam để phục vụ nhu cầu của khách hàng cũng như giảm bớt những hoạt động thủ công, luận văn đã trình bày những nội dung sau:

*Thứ nhất*, nghiên cứu tổng quan về kiểm thử ứng dụng/phần mềm

*Thứ hai*, nghiên cứu về công nghệ kiểm thử tự động, các công cụ kiểm thử tự động hiện nay, lựa chọn công cụ và phương pháp kiểm thử phù hợp với yêu cầu của đơn vị.

*Thứ ba*, thực hiện xây dựng testcase để phục vụ việc tạo ra chương trình kiểm thử tự động đối với website khai thác nhu cầu vay của Trung tâm Thông tin Tín dụng Quốc gia Việt Nam.

Đề tài nghiên cứu về kiểm thử tự động đã được nhiều nhiều người nghiên cứu, nhưng với Trung tâm Thông tin Tín dụng Quốc gia Việt nam thì đây là một ứng dụng mới. Mặc dù đã cố gắng hoàn thiện một cách tốt nhất nhưng luận văn không thể tránh những sai sót. Em mong nhận được sự đóng góp ý kiến từ thầy cô và các bạn để hoàn thiện luận văn của mình.



## DANH MỤC TÀI LIỆU THAM KHẢO

### Tiếng Việt:

- [1].<http://123doc.org/document/1040712-nguyen-cuu-va-ung-dung-cong-cu-kiemthu-tu-dong-trong-kiem-thu-phanmem.htm>, truy cập ngày 23/03/2021
- [2] 20 năm xây dựng và phát triển Trung tâm Thông tin Tín dụng Quốc gia Việt Nam, (2020), 105 trang.
- [3] TS. Đỗ Thị Bích Ngọc và TS. Phan Hoài Phương (2020) “Bài giảng đảm bảo chất lượng phần mềm”
- [4] ThS. Nguyễn Thị Kiêm Ái (nhà xuất bản thanh niên), “kiểm thử phần mềm từng bước trở thành tester chuyên nghiệp”.
- [5] Trần Tường Thụy, Phạm Quang Hiễn (nhà xuất bản Thông tin và Truyền thông), “*kiểm thử phần mềm (testing)*”
- [6] <https://www.guru99.com/testing-tools.html>

### Tiếng Anh:

- [8] <https://www.softwaretestinghelp.com/test-automation-frameworks-selenium-tutorial-20/> , truy cập ngày 23/03/2021
- [9] <https://phoenixnap.com/blog/best-automation-testing-tools>
- [10] Kunal Chaudhari. (January 2014) “*Selenium WebDriver 3 Practical Guide second Edition: End-to-end automation testing for web and mobile browsers with selenium WebDriver*”.
- [11] Dorothy Graham Paperback. (2016) “Software Test Automation: Effective Use of Test Execution Tools”
- [12] J. Ross Publication Inc., (2011). Murali Chemuturi. *Mastering Software Quality Assurance: Best Practices, Tools and Techniques for Software Developers.*

**BẢN CAM ĐOAN VỀ KIỂM TRA MỨC ĐỘ TƯƠNG ĐỒNG**

Em cam đoan đã thực hiện việc kiểm tra mức độ tương đồng nội dung luận văn qua phần mềm DoIT một cách trung thực và đạt kết quả mức độ tương đồng là 18% toàn bộ nội dung luận văn. Bản luận văn kiểm tra qua phần mềm là bản cứng đã nộp để bảo vệ trước hội đồng. Nếu sai em xin chịu các hình thức kỷ luật theo quy định hiện hành của Học viện.

Hà Nội, ngày    tháng    năm 2021

**HỌC VIÊN CAO HỌC**

**Phan Thị Ngọc**