

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

TÓM TẮT LUẬN VĂN THẠC SĨ

HÀ NỘI- NĂM 2021

Luận văn được hoàn thành tại:

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

Người hướng dẫn khoa học: **TS. Phan Thị Hà**

Phản biện 1: PGS. TS. Trần Đình Quế

Phản biện 2: PGS. TS. Đỗ Trung Tuấn

Luận văn sẽ được bảo vệ trước Hội đồng chấm luận văn Thạc sĩ tại
Học viện Công nghệ Bưu Chính Viễn thông

Vào lúc: 09h00 ngày 28 tháng 08 năm 2021

Có thể tìm hiểu luận văn tại:

- Thư viện của Học viện Công nghệ Bưu chính Viễn thông

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... 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.

Trung tâm Thông tin Tín dụng Quốc gia Việt Nam là một đơn vị trực thuộc Ngân hàng Nhà nước Việt Nam, kho dữ liệu 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.

Phần mềm được thực hiện bởi nhiều người kiểm thử. Năm 1980 "nhân viên kiểm thử phần mềm" đã được sử dụng, 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.

3. Mục tiêu nghiên cứu.

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

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.

Đối tượng và phạm vi nghiên cứu của luận văn bao gồm:

- 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 Tin 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 hệ thống kiểm thử tự động đối với chương trình đã lựa chọn.

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 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.

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 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.

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.

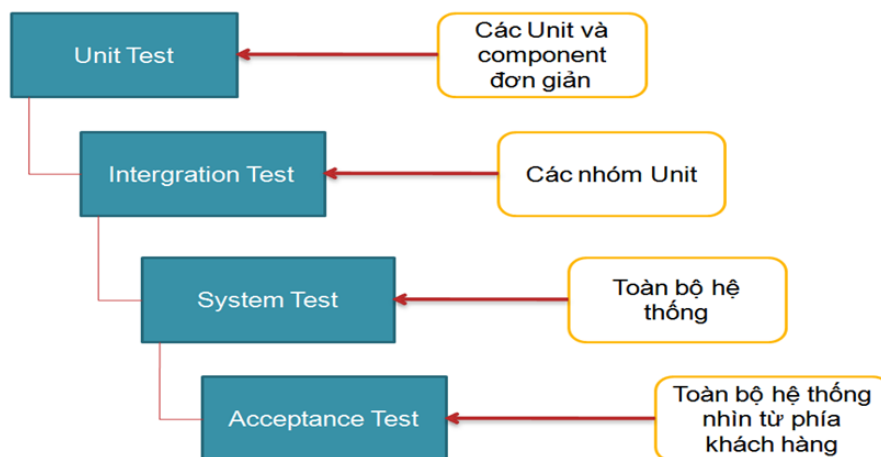
Sau khi sửa lỗi đã được phát hiện cần phải xác định và kiểm tra loại, giúp cho phần mềm được kiểm thử đạt đến mức độ chấp nhận được về chất lượng phần mềm.

Mục đích gián tiếp:

Biên kịch một bản ghi về lỗi phần mềm ddwwr 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

- Kiểm thử mức đơn vị (Unit Test)
- Kiểm thử tích hợp (Integration Test)

- c. Kiểm thử hệ thống (System test)
- d. Kiểm thử chấp nhận sản phẩm (Acceptance Test)

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 là phòng thông tin phòng ngừa rủi ro 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.

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

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

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 tiêu chuẩn cố định nào trên thế giới, nhưng về cơ bản quy trình kiểm thử bao gồm những giai đoạn sau:

- 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)

1.3.2. Phân tích quy trình kiểm thử phần mềm

- a. Phân tích yêu cầu (Test cycle closure)
- b. Lập kế hoạch kiểm thử (Test planning)
- c. Thiết kế kịch bản kiểm thử (Test case development)
- d. Thiết lập môi trường kiểm thử (Test environment set up)
- e. Thực hiện kiểm thử (Test execution)
- f. Đóng chu trình kiểm thử (Test cycle closure)

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). Hàng loạt nhà băng đều đang chú trọng ngân hàng số và muốn tận dụng cuộc cách mạng công nghệ 4.0 để phát triển và để chiếm lĩnh thị trường. Đây không chỉ là xu hướng mà còn là nền tảng giúp giảm thiểu chi phí cho ngân hàng. Điều này buộc các ngân hàng phải tính tới bài toán kiến tạo hệ thống công nghệ lõi, gia tăng giá trị dịch vụ, tính bảo mật và kiểm soát rủi ro.

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.

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

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.

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

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

	Kiểm thử thủ công (Kiểm thử thủ công)	Kiểm thử tự động (Automaiton Testing)
Định nghĩa	Kiểm thử thủ công là việc thử nghiệm phần mềm được làm thủ công bởi con người.	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.
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	Kiểm thử tự động được thực hiện bởi các công cụ kiểm thử nên thời gian thực hiện nhanh
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.	Kiểm thử tự động phù hợp với kiểm tra hồi quy, kiểm tra hiệu suất,

Hiệu suất và hàng loạt (Performance & Batch Testing)	Kiểm tra hiệu suất và hàng loạt không thể làm được với kiểm thử thủ công	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 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	Chi phí ban đầu cao, chi phí để thuê một người kiểm thử tự động với chi phí cao. 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
Con người (Human element)	Tốn rất nhiều người	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

	Kiểm thử thủ công (Kiểm thử thủ công)	Kiểm thử tự động (Kiểm thử tự động)
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.	<ul style="list-style-type: none"> - Phản hồi nhanh hơn - Kết quả tăng tốc - Giảm chi phí kinh doanh - Kiểm tra cải thiện hiệu quả - Mức độ kiểm tra tổng thể cao hơn

	<p>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 công không bị mất thời gian để thay đổi các trường hợp kiểm thử.</p>	<ul style="list-style-type: none"> - Sử dụng lại các thử nghiệm tự động - Phát hiện lỗi trước đó - Sự triệt để trong kiểm thử - Thời gian tiếp thị nhanh hơn - Bảo mật thông tin
Nhược điểm	<p>Kết quả kiểm thử ít tin cậy</p> <p>Quá trình thực hiện các ca kiểm thử không được ghi lại, 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>Việc sử dụng Kiểm thử thủ công sẽ giúp bạn phát hiện lỗi.</p> <p>Chi phí dành cho dự án có thể phải tăng lên do những đòi hỏi trong việc phải thuê nguồn nhân lực để tiến hành kiểm tra.</p> <p>Quá trình sử dụng công cụ tự động để kiểm thử sẽ cho bạn kết quả nhanh hơn cũng như chính xác hơn.</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,</p> <p>Nếu có một thay đổi nhỏ cũng sẽ mất thời gian để cập nhật kịch bản kiểm thử.</p> <ul style="list-style-type: none"> - Nhiều công cụ có chi phí rất cao, - Thường thì lương trả cho người kiểm thử tự động nhiều hơn người kiểm thử thủ công, vì công việc đòi hỏi họ có kỹ năng cao hơn. - Chi phí phát triển và bảo trì test script cao. - Đòi hỏi người kiểm thử phải có kinh nghiệm công nghệ và kỹ năng lập trình. - Đòi hỏi thời gian chuẩn bị dài hơn để thiết kế, cài đặt kỹ càng trước khi cần đưa dự án đi kiểm thử.

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)

Kiểm thử khả dụng (Usability Testing)

Kiểm thử Ad hoc (Ad-hoc Testing)

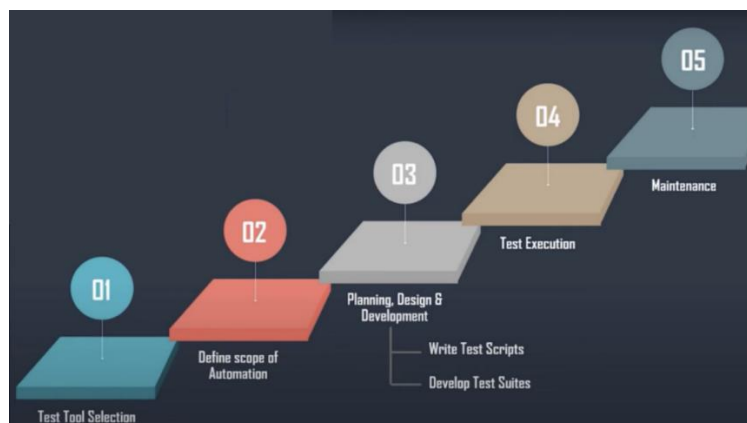
- Kiểm thử tự động phù hợp với:

Kiểm thử hồi quy (Regression Testing)

Kiểm thử khả năng chịu tải (Load Testing)

Kiểm thử hiệu năng (Performance Testing):

2.1.5. Quy trình kiểm thử phần mềm tự động (Automation testing)



Hình 2.1 Quy trình kiểm thử tự động

- Chọn công cụ kiểm thử (Test tool selection)
- Xác định phạm vi tự động (Define scope of automation)
- Lập kế hoạch thiết kế và phát triển (Planning, design and development)
- Thực hiện kiểm thử (Test Execution).
- Bảo trì (Maintainance)

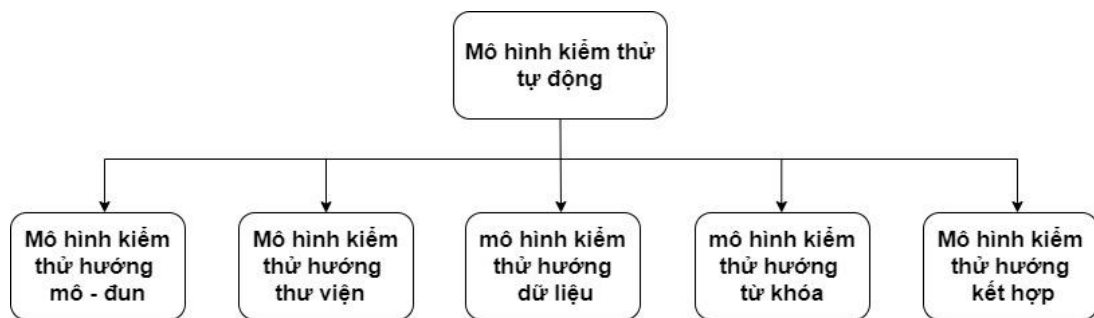
2.2. Tìm hiểu về các mô hình kiểm thử tự động hiện nay.

2.2.1. Mô hình kiểm thử tự động

Mô hình kiểm thử tự động là tập hợp các quy luật, nguyên tắc dùng trong quá trình viết mã kiểm thử.

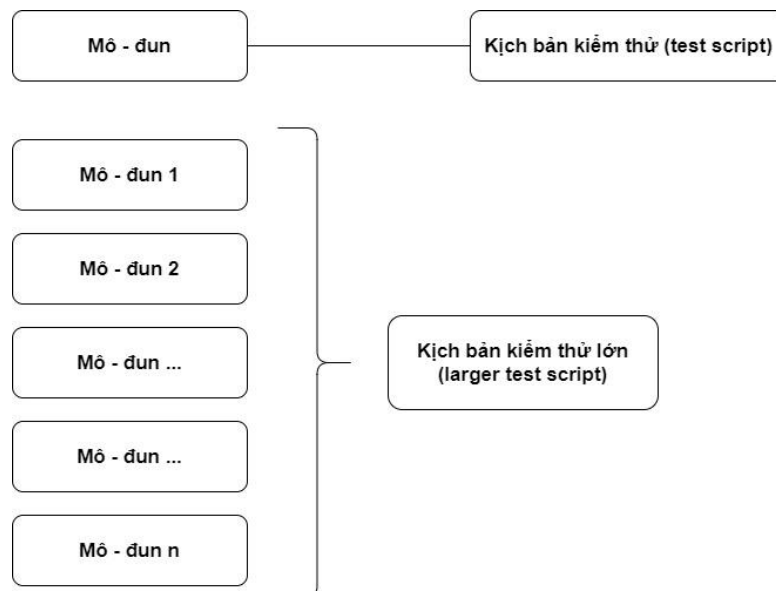
Mô hình kiểm thử tự động được sử dụng phổ biến hiện nay:

- 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



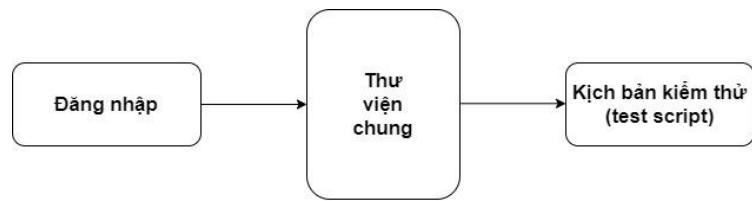
Hình 2.2 Mô hình kiểm thử tự động

a. Mô hình kiểm thử hướng Mô - đun



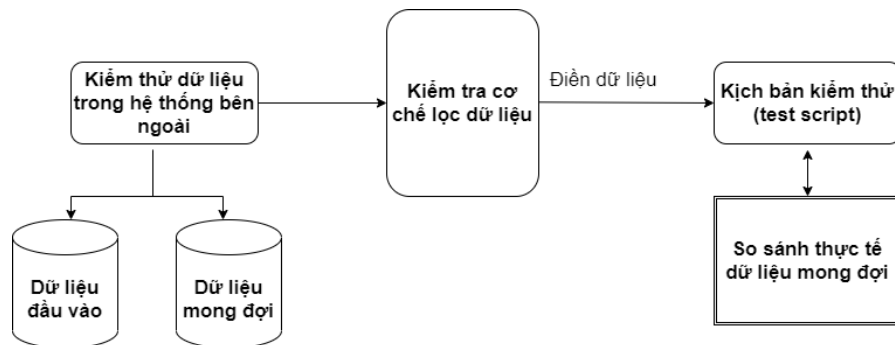
Hình 2.3 Mô hình kiểm thử hướng mô - đun

b. Mô hình kiểm thử kiến trúc thư viện



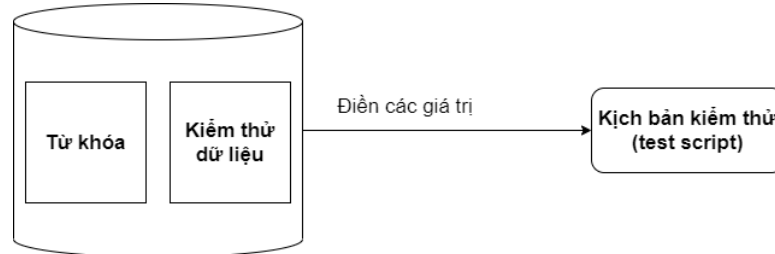
Hình 2.4 Mô hình kiểm thử hướng thư viện

c. Mô hình kiểm thử hướng dữ liệu



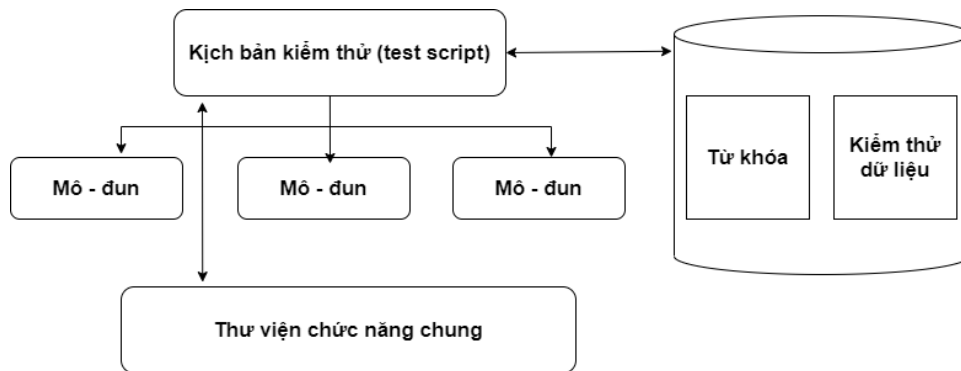
Hình 2.2 Mô hình kiểm thử hướng dữ liệu

d. Kiểm thử theo hướng từ khóa



Hình 2.3 Mô hình kiểm thử hướng từ khóa

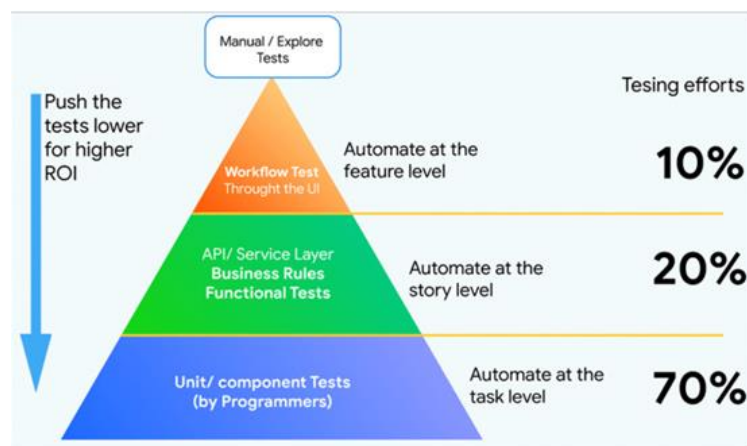
e. Kiểm thử hướng kết hợp



Hình 2.4 Mô hình kiểm thử hướng kết hợp

2.2.2. Cấp độ kiểm thử tự động.

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



Hình 2.5 Cấp độ kiểm thử tự động

- Kiểm tra quy trình hoạt động của phần mềm thông qua giao diện người dùng (Workflow test throught the - UI)
- Kiểm thử phương thức giao diện kết nối với thư viện và ứng dụng khác (Application Programming Interface – API)
- Kiểm thử đơn vị / Kiểm thử thành phần (Unit / component tests)

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 giúp thực hiện 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.
- Giúp kiểm thử viên rèn luyện kỹ năng lập trình.
- Giúp giảm chi phí cho quá trình kiểm thử.
- Giảm sự nhầm 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

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.

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.

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.

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.

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.

f. Công cụ kiểm thử QTP

Quick Test Professional (QTP) 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.

g. 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.

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

So sánh công cụ kiểm thử katalon và selenium

Bảng 2.3 So sánh công cụ kiểm thử tự động katalon và selenium

Đặc tính	Katalon Studio	Selenium
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.	Cần phải có kỹ năng lập trình ở mức nâng cao
Thời gian tạo test script	Tạo test script nhanh	Tạo test script Chậm
Kiểm thử hình ảnh	Hỗ trợ tích hợp	Yêu cầu cài đặt thư viện bổ sung
Giá	Có phí	Miễn phí

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

KẾT LUẬN CHƯƠNG.

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.

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.

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.

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

Bước 3: Xây dựng môi trường làm việc.

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

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 Tín dụng Quốc gia Việt Nam

3.3.1. Công cụ kiểm thử tự động Selenium.

Như đã trình bày ở chương II.

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.

3.3.3. Các câu lệnh sử dụng trong Selenium WebDriver

- a. Các câu lệnh trình duyệt
- b. Các câu lệnh WebElement

- c. Các câu lệnh điều hướng trình duyệt
- d. Các lệnh switch
- e. Các câu lệnh wait

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

- Hỗ trợ ngôn ngữ và framework
- Nguồn mở sẵn có
- Hỗ trợ nhiều trình duyệt
- Hỗ trợ hoạt động trên nhiều hệ điều hành
- Dễ thực hiện
- Tái sử dụng và tích hợp
- Tính linh hoạt
- Thực hiện kiểm thử song song và tiếp thị nhanh hơn
- Sử dụng phân cứng ít hơn
- Dễ tìm hiểu và sử dụng
- Cập nhật liên tục

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.

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. Một tester phải biết dữ liệu

nào cần thực hiện kiểm thử, thứ tự kiểm thử đây là điều kiện quyết định cho việc kiểm thử. Test case là những tình huống yêu cầu của khách hàng.

3.5.3. Cấu trúc cơ bản của test case

Tùy thuộc vào từng mẫu (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.

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.

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>

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 Tín 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

3.6.2. Xuất báo cáo kết quả

- File tổng hợp báo cáo

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

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 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