

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG



MAI THỊ NHI

**NGHIÊN CỨU VÀ ỨNG DỤNG CÔNG CỤ KIỂM THỬ
TỰ ĐỘNG TRONG KIỂM THỬ PHẦN MỀM**

Chuyên ngành: Truyền dữ liệu và mạng máy tính
Mã số: 60.48.15

TÓM TẮT LUẬN VĂN THẠC SĨ

HÀ NỘI - 2012

Luận văn được hoàn thành tại:
HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

Người hướng dẫn khoa học: PGS.TS. HUỲNH QUYẾT THẮNG
(Ghi rõ học hàm, học vị)

Phản biện 1:

Phản biện 2:

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: giờ ngày tháng năm

Có thể tìm hiểu luận văn tại:

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

MỞ ĐẦU

Lý do chọn đề tài:

Ngày nay, tự động hóa được ứng dụng ở rất nhiều lĩnh vực, mục đích thường rất đa dạng và tùy theo nhu cầu đặc thù của từng lĩnh vực, tuy nhiên điểm chung nhất vẫn là giảm nhân lực, thời gian và sai sót. Ngành công nghệ thông tin mà cụ thể là phát triển phần mềm cũng không ngoại lệ. Như chúng ta biết, để tạo ra sản phẩm công nghệ thông tin hay phần mềm có chất lượng thì hoạt động kiểm thử phần mềm đóng vai trò rất quan trọng, trong khi đó hoạt động này lại tiêu tốn và chiếm tỷ trọng khá lớn công sức và thời gian trong một dự án. Do vậy, nhu cầu tự động hoá quy trình kiểm thử phần mềm cũng được đặt ra.

Qua thực tế cho thấy, việc áp dụng kiểm thử tự động hợp lý sẽ mang lại thành công cho hoạt động kiểm thử phần mềm. Kiểm thử tự động giúp giảm bớt công sức thực hiện, tăng độ tin cậy, giảm sự nhầm lẫn và rèn luyện kỹ năng lập trình cho cán bộ kiểm thử.

Đó là lý em chọn đề tài “**Nghiên cứu và ứng dụng công cụ kiểm thử tự động trong Kiểm thử phần mềm**” làm luận văn tốt nghiệp.

Mục đích của luận văn: Đề tài tìm hiểu cơ sở lý thuyết về kiểm thử cũng như cách triển khai công cụ kiểm thử phần mềm tự động để giảm nhân lực kiểm thử và đảm bảo chất lượng phần mềm hơn với công việc kiểm thử bằng tay.

Mục tiêu chính của đề tài là nghiên cứu giai đoạn nào cần áp dụng công cụ kiểm thử tự động, các yếu tố nào cần kiểm thử hiệu năng.

Đối tượng và phạm vi nghiên cứu: Luận văn nghiên cứu lý thuyết kiểm thử phần mềm; bên cạnh đó nghiên cứu công cụ kiểm thử tự động và áp dụng các dự án tại Trung tâm phần mềm Viettel.

Phương pháp nghiên cứu: Nghiên cứu tổng quan về kiểm thử phần mềm, các kỹ thuật kiểm thử, nghiên cứu công cụ kiểm thử phần mềm tự động QuickTest Professional, LoadRunner và ứng dụng vào một dự án phát triển tại Trung tâm phần mềm viễn thông Viettel – Tập đoàn viễn thông quân đội.

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

Chương 1: Tổng quan về kiểm thử.

Chương 2: Nghiên cứu các công cụ kiểm thử phần mềm tự động.

Chương 3: Ứng dụng.

Phần kết luận đưa ra những đánh giá về những kết quả đạt được và thảo luận về hướng nghiên cứu tiếp của luận văn.

Trong quá trình thực hiện Luận văn, do thời gian cũng như trình độ của tác giả còn có những hạn chế nhất định nên không thể tránh khỏi những sai sót. Rất mong nhận được sự góp ý của các thầy, cô giáo và các bạn để Luận văn hoàn thiện hơn. Tôi xin chân thành cảm ơn sự hướng dẫn, và giúp đỡ tận tình của PGS.TS. **Huỳnh Quyết Thắng**, các thầy trong khoa Quốc tế và Đào tạo sau đại học – Học viện Công nghệ BC-VT đã giúp đỡ tôi trong quá trình học tập cũng như trong quá trình làm Luận văn.

CHƯƠNG 1

TỔNG QUAN VỀ KIỂM THỬ PHẦN MỀM

1.1. Kiểm thử phần mềm

Kiểm thử phần mềm là quy trình được sử dụng để đánh giá, kiểm tra chất lượng phần mềm ở nhiều khía cạnh khác nhau dựa trên các yêu cầu của người sử dụng đối với sản phẩm phần mềm, nhằm đảm bảo phần mềm hoạt động tốt trong các môi trường, các trường hợp khác nhau.

1.2. Phân loại và các kỹ thuật kiểm thử

Ta phân loại kiểm thử dựa vào các yếu tố: Chiến lược kiểm thử, phương pháp kiểm thử và kỹ thuật kiểm thử.

Dựa vào chiến lược kiểm thử ta có thể phân chia kiểm thử thành hai loại: kiểm thử thủ công và kiểm thử tự động.

Theo phương pháp tiến hành kiểm thử ta chia kiểm thử làm hai loại: kiểm thử tĩnh và kiểm thử động.

Dựa vào kỹ thuật kiểm thử ta có thể phân chia kiểm thử thành ba loại: kiểm thử hộp đen, kiểm thử hộp trắng và kiểm thử hộp xám.

1.3. Kiểm thử tĩnh và kiểm thử động

1.3.1. Kiểm thử tĩnh

1.3.2. Kiểm thử động

1.4. Kiểm thử hộp trắng, kiểm thử hộp đen và kiểm thử hộp xám

1.4.1. Kiểm thử hộp trắng – White box testing

Trong phương pháp kiểm thử hộp trắng, ta đi vào tìm hiểu các kỹ thuật kiểm thử hộp trắng đó là:

- Kiểm thử bao phủ chu trình cơ sở
- Kiểm thử cấu trúc điều khiển

1.4.1.1. Kiểm thử đường dẫn cơ sở (Basic Path Testing)

1.4.1.1.1. Đồ thị chu trình (Flow Graph)

1.4.1.1.2. Độ phức tạp cyclomat

1.4.1.1.3. Phát sinh các trường hợp kiểm thử theo đường dẫn cơ sở

1.4.1.2. Kiểm thử cấu trúc điều khiển

1.4.1.2.1. Kiểm thử điều kiện

1.4.1.2.2. Kiểm thử luồng dữ liệu

1.4.1.2.3. Kiểm thử vòng lặp

1.4.1.3. Kiểm thử hộp đen – Black box testing

1.4.1.4. Phân lớp tương đương

1.4.1.5. Phân tích giá trị biên (BVA – Boundary Value Analysis)

1.4.1.6. Kiểm thử so sánh

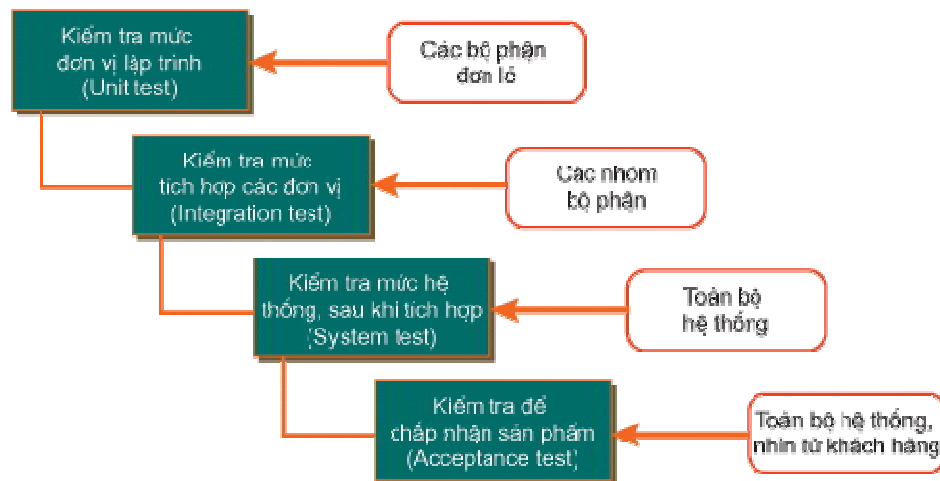
1.4.1.7. Đoán lỗi

1.4.2. Kiểm thử hộp xám – Gray box testing

1.5. Quy trình kiểm thử phần mềm

1.5.1. Các giai đoạn kiểm thử phần mềm

Trong một dự án, kiểm thử thường trải qua các giai đoạn: Kiểm thử đơn vị, kiểm thử tích hợp, kiểm thử hệ thống và kiểm thử chấp nhận.



Hình 1.12 - Các giai đoạn kiểm thử

1.5.1.1. Kiểm thử đơn vị (Unit Testing)

1.5.1.2. *Kiểm thử tích hợp (Intergration testing)*

1.5.1.3. *Kiểm thử hệ thống (System Test)*

1.5.1.4. *Kiểm thử chấp nhận sản phẩm (Acceptance Test)*

1.5.1.5. *Một số cấp độ kiểm thử khác*

1.6. Kết luận chương

Trong chương này, ta đi vào tìm hiểu để có cái nhìn tổng quan về kiểm thử phần mềm, phân loại kiểm thử dựa vào các yếu tố: Chiến lược kiểm thử, phương pháp kiểm thử và kỹ thuật kiểm thử.

- Dựa vào chiến lược kiểm thử ta có thể phân chia kiểm thử thành hai loại: kiểm thử thủ công và kiểm thử tự động.
- Theo phương pháp tiến hành kiểm thử ta chia kiểm thử làm hai loại: kiểm thử tĩnh và kiểm thử động.
- Theo phương pháp tiến hành kiểm thử ta chia kiểm thử làm hai loại: kiểm thử tĩnh và kiểm thử động.
- Dựa vào kỹ thuật kiểm thử ta có thể phân chia kiểm thử thành ba loại: kiểm thử hộp đen, kiểm thử hộp trắng và kiểm thử hộp xám.

Luận văn cũng tìm hiểu về quy trình phân phần mềm, các khái niệm về trường hợp kiểm thử (testcase), kịch bản kiểm thử (testscript). Quy trình phần mềm gồm các giai đoạn kiểm thử: kiểm thử đơn vị, kiểm thử tích hợp, kiểm thử hệ thống, kiểm thử chấp nhận sản phẩm và các phương pháp kiểm thử con người. Chương 2 sẽ đi tìm hiểu khái quát về kiểm thử tự động, quy trình kiểm thử tự động, mục đích của việc kiểm thử tự động và các công cụ (tool) kiểm thử tự động chức năng, công cụ kiểm thử hiệu năng cho các ứng dụng phần mềm.

CHƯƠNG 2

NGHIÊN CỨU CÔNG CỤ KIỂM THỬ PHẦN MỀM TỰ ĐỘNG

2.1. Khái quát về kiểm thử phần mềm tự động

Kiểm thử phần mềm tốn nhiều chi phí nhân công, thời gian. Trong một số dự án, chi phí kiểm thử phần mềm chiếm 50% tổng giá trị của dự án. Nếu cần ứng dụng an toàn hơn, chi phí kiểm thử còn cao hơn nữa.

Do đó một trong các mục tiêu của kiểm thử là tự động hóa nhiều, nhờ đó mà giảm thiểu chi phí, giảm lỗi, đặc biệt giúp việc kiểm thử hồi qui dễ dàng và nhanh chóng hơn.

Tự động hóa việc kiểm thử là dùng phần mềm điều khiển việc thi hành kiểm thử, so sánh kết quả có được với kết quả mong muốn, thiết lập các điều kiện đầu vào, các kiểm soát kiểm thử và các chức năng báo cáo kết quả...

2.1.1. Kiểm thử tự động là gì?

Kiểm thử tự động là quá trình thực hiện một cách tự động các bước trong một kịch bản kiểm thử. Kiểm thử tự động bằng một công cụ nhằm rút ngắn thời gian kiểm thử.

2.1.2. Tại sao phải kiểm thử tự động?

Kiểm thử phần mềm tự động với mục đích:

- Giảm bớt công sức và thời gian thực hiện quá trình kiểm thử
- Tăng độ tin cậy.
- Giảm sự nhàm chán cho con người
- Rèn luyện kỹ năng lập trình cho kiểm thử viên
- Giảm chi phí cho tổng quá trình kiểm thử.

Khi nào cần kiểm thử tự động?

- Không đủ tài nguyên: Khi số lượng TestCase quá nhiều mà kiểm thử viên không thể hoàn tất trong thời gian cụ thể
- Kiểm tra hồi quy: Nâng cấp phần mềm, kiểm tra lại các tính năng đã chạy tốt và những tính năng đã sửa. Tuy nhiên, việc này khó đảm bảo về mặt thời gian
- Kiểm tra khả năng vận hành phần mềm trong môi trường đặc biệt:

- ✓ Đo tốc độ trung bình xử lý một yêu cầu của Web server
- ✓ Xác định số yêu cầu tối đa được xử lý bởi Web Server
- ✓ Xác định cấu hình máy thấp nhất mà PM vẫn có thể hoạt động tốt

2.1.3. Quy trình kiểm thử tự động

Bảng sau mô tả rõ hơn các bước thực hiện kiểm thử tự động:

Bảng 2.1 – Các bước thực hiện kiểm thử tự động

STT	Bước thực hiện	Mô tả
1	Tạo kịch bản kiểm thử	Giai đoạn này dùng công cụ kiểm thử để ghi lại các thao tác lên phần mềm cần kiểm tra và tự động sinh ra kịch bản kiểm thử.
2	Chỉnh sửa kịch bản kiểm thử	Chỉnh sửa để kịch bản kiểm thử thực hiện kiểm tra theo đúng yêu cầu đặt ra. Cụ thể, làm theo trường hợp kiểm thử cần thực hiện.
3	Chạy kịch bản kiểm thử	Chạy kịch bản kiểm thử để kiểm tra phần mềm có đưa ra đúng như kết quả mong muốn không
4	Đánh giá kết quả	Đánh giá kết quả sau khi chạy kịch bản kiểm thử

2.1.4. Công cụ kiểm thử tự động

2.2. Nghiên cứu công cụ kiểm thử tự động chức năng bằng QuickTest

Professional

Phần mềm HP Quick Test Professional là phần mềm kiểm soát việc test tự động những chức năng của một sản phẩm phần mềm khác. Phần mềm QTP là một bộ phận của hệ thống Mercury Quality Center bao gồm nhiều module phần mềm phối hợp với nhau để quản lý toàn bộ quy trình đảm bảo chất lượng sản phẩm phần mềm. Trước đây, do hãng Mercury phát hành. Sau này, tập đoàn HP đã mua lại toàn bộ hãng Mercury, nên tên gọi của nó là: HP QuickTest Professional.

Quick Test Professional là công cụ kiểm thử dùng để kiểm tra chức năng (functional test) và cho phép thực hiện kiểm tra hồi qui (regression test) một cách tự động.

2.2.1. Loại phần mềm hỗ trợ

2.2.2. Đặc điểm

2.2.3. Các thành phần quan trọng của QuickTest Professional

a. Action

Giống như hàm hoặc thủ tục trong các ngôn ngữ lập trình khác, Action ghi lại các bước thực hiện kiểm thử tự động và nó có thể được sử dụng lại nhiều lần. Trong một test script có thể có nhiều Action.

b. Data Table

Nơi lưu dữ liệu phục vụ cho kiểm thử tự động. Một test script sẽ có một Data Table được dùng chung cho tất cả các Action. Bên cạnh đó, mỗi Action cũng có một Data Table cho riêng mình.

c. Object Repository (OR)

Cấu trúc theo dạng cây, mô tả các đối tượng trong phần mềm được kiểm tra. Đây được xem là cầu nối để test script tương tác với phần mềm được kiểm tra.

Khi ra lệnh cho QTP ghi lại các thao tác người dùng lên phần mềm thì trong OR sẽ tự động phát sinh thành phần đại diện cho những đối tượng trên phần mềm vừa được thao tác.

OR có thể tổ chức thành 2 loại, một loại dùng chung cho nhiều test script, loại khác dùng theo từng Action.

Để xem OR, chọn menu Tools > Object Repository.

d. Checkpoint

Có thể hiểu là nơi kiểm tra trong test script, khi chạy nó sẽ thực hiện so sánh kết quả thực tế khi kiểm tra phần mềm với kết quả mong đợi. Sau khi tiến hành so sánh QTP sẽ tự động ghi lại kết quả vào Test Results (nơi lưu kết quả khi chạy test script).

2.2.4. Ngôn ngữ sử dụng viết script

2.3. Nghiên cứu công cụ kiểm thử hiệu năng

2.3.1. Khái quát về kiểm thử hiệu năng

Các mục tiêu cần kiểm tra hiệu năng:

- Thời gian phản hồi (Response time), là thời gian từ lúc client gửi request tới server cho đến khi client nhận được response từ server trả về.
 - Throughput: Thông lượng hệ thống, tính bằng số giao dịch (transaction) hệ thống đáp ứng được trong một khoảng thời gian. Đơn vị tổng quát là transaction per time_period vd: transactions per second, calls per day...
 - Concurrency: Số giao dịch đồng thời được thực hiện, tính bằng số giao dịch đồng thời hệ thống đáp ứng được. Đơn vị là transaction, vd: 200 transactions đồng thời, 300 transactions đồng thời...
 - CPU usage: Hiệu suất sử dụng CPU. Đơn vị là %.
 - RAM usage: Hiệu suất sử dụng RAM. Đơn vị là %.
 - Fail rate: Tỷ lệ lỗi, tính bằng số giao dịch không thực hiện thành công trên tổng số giao dịch đã thực hiện. Giá trị này dùng để làm điều kiện cần cho các mục tiêu trên. Đơn vị là %.
- Tiêu chuẩn thành công: Response time, Throughput, Concurrency, CPU usage, RAM usage, Fail rate có thể chấp nhận được

2.3.2. Quy trình kiểm thử hiệu năng

2.3.3. Giới thiệu về công cụ Load Runner

2.3.3.1. Đặc điểm

LoadRunner giả lập nhiều người dùng ảo thực hiện các giao dịch cùng một lúc. Sau đó LoadRunner giám sát các thông số xử lý của phần mềm được kiểm tra. Kết quả thống kê sẽ được lưu lại và cho phép kiểm thử viên thực hiện phân tích.

Bảng 2.5 - Các thành phần của LoadRunner

Thành phần	Chức năng
Virtual User Generator	Tự động tạo ra VuGen script để lưu lại các thao tác người dùng tương tác lên phần mềm. VuGen script này còn được xem là hoạt động của một người ảo mà LoadRunner giả lập.
Controller	Tổ chức, điều chỉnh, quản lý và giám sát hoạt động kiểm tra tải. Thành phần này có chức năng tạo ra những tình huống (scenario) kiểm tra.

Load Generator	Cho phép giả lập hàng ngày người dùng, hoạt động của từng người sẽ được thực hiện theo VuGen script. Kết quả thực hiện sẽ được thông báo cho Controller.
Analysis	Cung cấp việc xem, phân tích và so sánh các kết quả PT.
Launcher	Nơi tập trung tất cả các thành phần của LoadRunner cho người dùng.

2.3.3.2. Các bước thực hiện

1. Recording (Ghi nhận):

Cho phép tự động phát sinh script ghi lại thao tác người dùng lên ứng dụng cần kiểm tra. Cách tổ chức script của LoadRunner được chia ra thành 3 thành phần chính.

- vuser_init: mỗi người dùng ảo sẽ thực hiện một lần trước khi chạy PT.
- Run: bao gồm một hoặc nhiều hàm (action), và cho phép người dùng ảo chạy lặp lại nhiều lần. Dựa trên action chúng ta có thể tổ chức các nhóm chứa các action khác nhau và theo thứ tự tùy ý.
- vuser_end: mỗi người dùng ảo thực hiện một lần cuối cùng khi chạy PT.

2. Replay (Phát lại):

Đây là bước cho phép chạy thử để kiểm tra script đã chạy đúng yêu cầu của một người dùng ảo hay chưa, qua đó có sự chỉnh sửa nếu cần. Bên cạnh đó LoadRunner còn cho phép tổ chức thứ tự, số lần lặp lại các action hiện đang có bằng cách chọn Open Run-Time Settings, và thiết lập thời gian tương tác giữa người dùng ảo và web server...

3. Enhancements (Nâng cao):

- Tạo transaction: transaction là một số hành động do chúng ta chọn từ quá trình tự động phát sinh script. Mục tiêu là giám sát thông số hoạt động của một số hành động trong transaction đó. Thông số giám sát sẽ được thể hiện sau khi chúng ta thực hiện kiểm thử hiệu năng
- Tham biến hóa: thay thế các giá trị cố định trong script bằng các biến.
- Kiểm tra nội dung: cho phép thêm các điểm kiểm tra nội dung, trong LoadRunner gọi là content check, có thể hiểu giống như một thuật ngữ khác đã được đề cập trong bài viết trước là checkpoint.

4. Prepare For Load (Chuẩn bị thực thi):

- Thiết lập sự lặp lại của các action, ở giai đoạn Replay chúng ta cũng có thể làm điều này.

- Thiết kế tình huống: thiết lập số người dùng ảo tối đa thực hiện cùng một lúc, thời gian chạy bao lâu, số lượng người dùng ảo tăng như thế nào (Ramp Up) hoặc giảm như thế nào (Ramp Down).

- Thực hiện tình huống: thực thi các tình huống kiểm tra, phân tích kết quả dựa trên các thông số của môi trường kiểm tra, ví dụ: số yêu cầu gửi tới web server xử lý trong 1 giây, số hồi đáp từ server trong 1 giây, số trang mà người dùng có thể mở trong 1 giây, ...

5. Finish (Kết thúc):

- Upload script lên Performance Center server, đây là một phần trong việc thực hiện giải pháp chia sẻ tài nguyên PT qua Internet.

2.4. Kết luận chương

Trong chương này ta tập trung vào tìm hiểu khái quát về kiểm thử tự động là gì? Tại sao phải kiểm thử tự động? Quy trình kiểm thử tự động? Các kỹ thuật và phương pháp kiểm thử tự động cũng được trình bày một cách cơ bản nhất.

Về kiểm thử tự động chức năng, luận văn cũng đánh giá khi nào cần kiểm thử tự động có hiệu quả nhất cả về chi phí và thời gian, đồng thời đề xuất ra quy trình kiểm thử tự động tự chức năng. Từ cơ sở lý thuyết trên, luận văn đã tìm hiểu chi tiết công cụ kiểm thử chức năng QuickTest Professional, gồm: Loại phần mềm hỗ trợ, đặc điểm về công cụ và ngôn ngữ sử dụng để viết script.

Về kiểm thử hiệu năng, luận văn đã đưa ra mục đích của việc kiểm thử hiệu năng và đề xuất các yếu tố về hiệu năng cần kiểm thử cho các ứng dụng phần mềm như: Số giao dịch đồng thời, thời gian đáp ứng cho mỗi giao dịch dựa trên các thông số về server CPU, RAM và có thể để đánh giá được hiệu suất sử dụng CPU, hiệu suất sử dụng RAM.

Từ cơ sở lý thuyết về kiểm thử hiệu năng, luận văn đã tìm hiểu chi tiết công cụ kiểm thử hiệu năng Load Runner, gồm: Đặc điểm, các thành phần và các bước thực hiện của LoadRunner.

Chương 3 sẽ dựa vào các khái niệm, quy trình kiểm thử chung trong chương 1 và quy trình kiểm thử tự động được mô tả trong chương 2 để áp dụng một ứng dụng cụ thể, đồng thời phân tích và đánh giá các kết quả mô phỏng cho ứng dụng này.

CHƯƠNG 3

ỨNG DỤNG

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

Kiểm thử phần mềm là một trong những hoạt động quan trọng trong quy trình phát triển phần mềm. Nó góp một phần rất lớn trong việc đánh giá chất lượng một phần mềm và là quy trình bắt buộc trong các dự án phần mềm trên thế giới cũng như trong nước. Tuy nhiên, hoạt động kiểm thử thường gặp nhiều khó khăn. Thứ nhất, kiểm thử các hệ thống phức tạp đòi hỏi rất nhiều nguồn tài nguyên và chi phí cao. Thứ hai, quy trình phát triển phần mềm luôn trải qua nhiều hoạt động biến đổi thông tin, sự mất mát thông tin trong quá trình biến đổi là yếu tố chính làm cho hoạt động kiểm thử khó khăn. Thứ ba, kiểm thử chưa được chú trọng trong đào tạo con người. Cuối cùng, không tồn tại kỹ thuật kiểm thử cho phép khẳng định một phần mềm hoàn toàn đúng đắn hay không còn lỗi.

Chính vì thế, 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. Chương 3 đi vào tìm hiểu một ứng dụng cụ thể và nghiên cứu xây dựng các bộ dữ liệu kiểm thử cho một số chức năng cần kiểm thử tự động.

3.2. Đề xuất áp dụng các công cụ kiểm thử chức năng QuickTest Professional và kiểm thử hiệu năng Load Runner tại Trung tâm phần mềm Viettel (TTPM Viettel)

3.2.1. Hiện trạng kiểm thử phần mềm tại TTPM Viettel

Trung tâm phần mềm Viettel đã và đang phát triển các ứng dụng quản lý cho các mạng như hỗ trợ văn phòng, các dịch vụ viễn thông, các dịch vụ VAS,...vv... Ban đầu, triển khai tại các đơn vị thuộc Tập đoàn viễn thông quân đội tại Việt Nam, sau đó triển khai các ứng dụng này ở một số quốc gia khác. Ở các nước mà Viettel triển khai mạng, các ứng dụng chỉ tập trung vào nâng cấp một số chức năng cho phù hợp với nghiệp vụ ở các nước sở tại. Việc thay đổi một số chức năng có thể ảnh hưởng đến các chức năng khác của cả hệ thống hoặc chỉ thay đổi ở tầng kiến trúc, không thay đổi trên giao diện màn hình, dẫn đến phải kiểm thử lại toàn bộ ứng dụng. Tuy nhiên, việc kiểm thử các ứng dụng nâng cấp này vẫn chủ yếu làm theo cách thủ công (manual) - kiểm thử bằng mắt, điều 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 giá hiệu năng cũng mới chỉ thực hiện bằng cách ghi lại lịch sử (log) hoặc giám sát các hệ thống đang chạy ở thời điểm thường và thời điểm cao tải, sau đó phân tích và đánh giá hiệu năng. Các công việc này thường được kiểm tra khi đã triển khai ở môi trường thật (khi khách hàng đã sử dụng phần mềm) nên tìm các lỗi hiệu năng thường được phát hiện muộn, làm mất uy tín của khách hàng.

3.2.2. Đề xuất quy trình kiểm thử tự động tại TTPM Viettel

Từ các khái niệm cơ bản, quy trình kiểm thử chung 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 các công cụ QuickTest Professional và LoadRunner tại TTPM Viettel như sau:

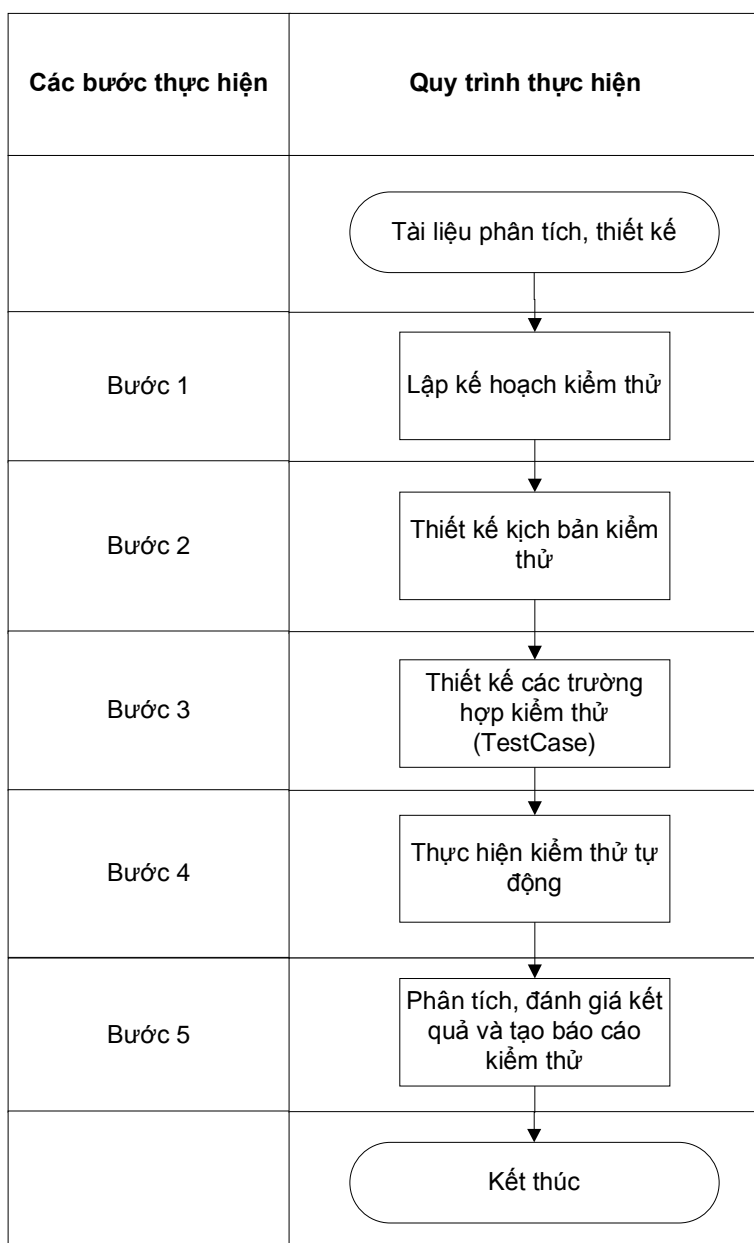
3.2.2.1. Quy trình kiểm thử chức năng tự động

Các bước thực hiện:

Bước 1: Sau khi tìm hiểu nghiệp vụ từ các tài liệu phân tích, thiết kế, nhóm kiểm thử **Lập kế hoạch kiểm thử** xác định các chức năng cần kiểm tra tự động.

Bước 2: Nhóm kiểm thử **Thiết kế kịch bản kiểm thử** các chức năng, mô tả các luồng nghiệp vụ chính, phụ cần kiểm tra.

Bước 3: Nhóm kiểm thử **Thiết kế các trường hợp kiểm thử**. Các trường hợp kiểm thử đủ các thông tin: Mục đích, điều kiện kiểm thử, các bước thực hiện và kết quả mong muốn.

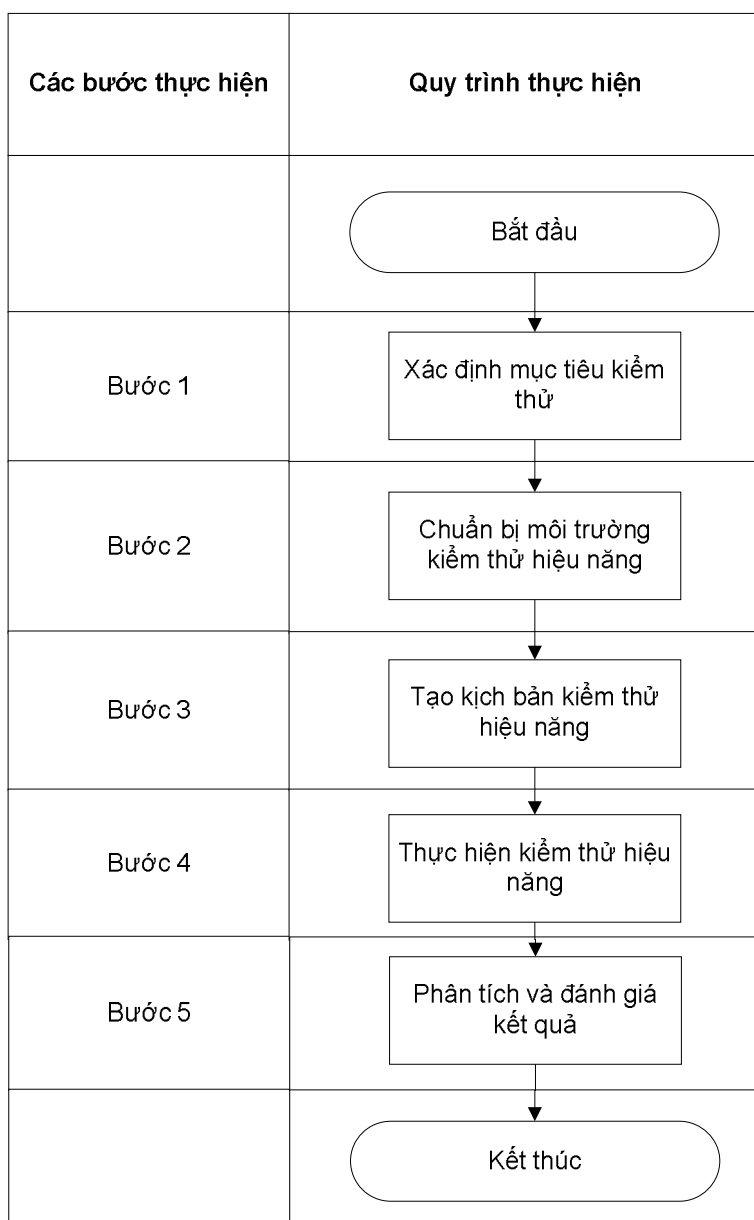


Hình 3.1 - Quy trình kiểm thử chức năng tự động

Bước 4: Từ các trường hợp kiểm thử được thiết kế trong bước 3, nhóm kiểm thử phát triển các Test Script bằng công cụ Quick Test Professional, cấu hình các dữ liệu đầu vào cho từng trường hợp và thực hiện kiểm thử tự động các Test Script đã tạo.

Bước 5: Kết quả thực hiện các test script ở Bước 4, đánh giá và tạo báo cáo kiểm thử.

3.2.2.2. Quy trình kiểm thử hiệu năng



Hình 3.2 - Quy trình kiểm thử hiệu năng

Các bước thực hiện:

Bước 1: Xác định các thông số về môi trường, mục tiêu và các chức năng kiểm thử để chuẩn bị cho hoạt động kiểm thử hiệu năng:

- Môi trường kiểm thử: phần cứng, phần mềm, cơ sở dữ liệu, công cụ sử dụng để đo hiệu năng
- Các mục tiêu kiểm thử hiệu năng: Thời gian xử lý trung bình (Response time), Thông lượng hệ thống (Throughput), Số người dùng đồng thời.

Bước 2: Chuẩn bị môi trường kiểm thử hiệu năng, từ các thông số hiệu năng được khảo sát, phân tích ở môi trường thật, ta xác định tỷ lệ tương ứng trên môi trường test.

Bước 3: **Tạo kịch bản kiểm thử hiệu năng** gồm các thông tin: Chức năng, thông số người dùng, mục tiêu kiểm thử, tiêu chuẩn thành công.

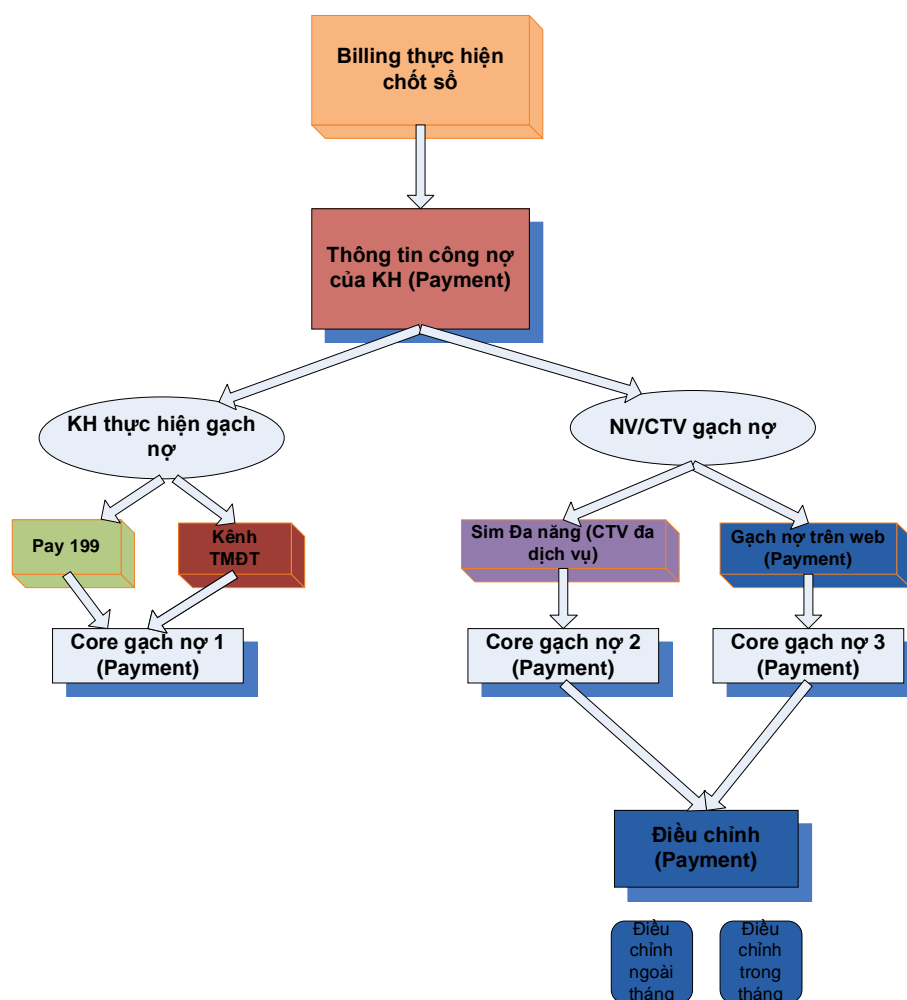
Bước 4: **Thực hiện kiểm thử hiệu năng:**

- Phát triển các Test Script trên công cụ Load Runner
- Cấu hình, cài đặt các tham số cần đo hiệu năng và chạy các script

Bước 5: Từ kết quả thực hiện ở bước 4, ta phân tích và đánh giá để xác định các ngưỡng cho các thông số cần đo.

3.3. Ứng dụng quy trình kiểm thử tự động cho một Dự án cụ thể

Để triển khai thử nghiệm kiểm thử tự động các chức năng bằng công cụ QuickTest Professional và kiểm thử hiệu năng bằng công cụ LoadRunner, ta thực hiện trên phần mềm **Hệ thống thanh toán cước phí** được viết bằng ngôn ngữ Java, sử dụng hệ quản trị cơ sở dữ liệu SQL Navigator.



Hình 3.3 - Luồng thanh toán cước phí dịch vụ trả sau

3.3.1. Thử nghiệm kiểm thử tự động các chức năng

Các bước kiểm thử tự động các chức năng hệ thống Thanh toán cước phí dịch vụ viễn thông trả sau như sau:

3.3.1.1. Lập kế hoạch kiểm thử

3.3.1.2. Thiết kế kịch bản kiểm thử

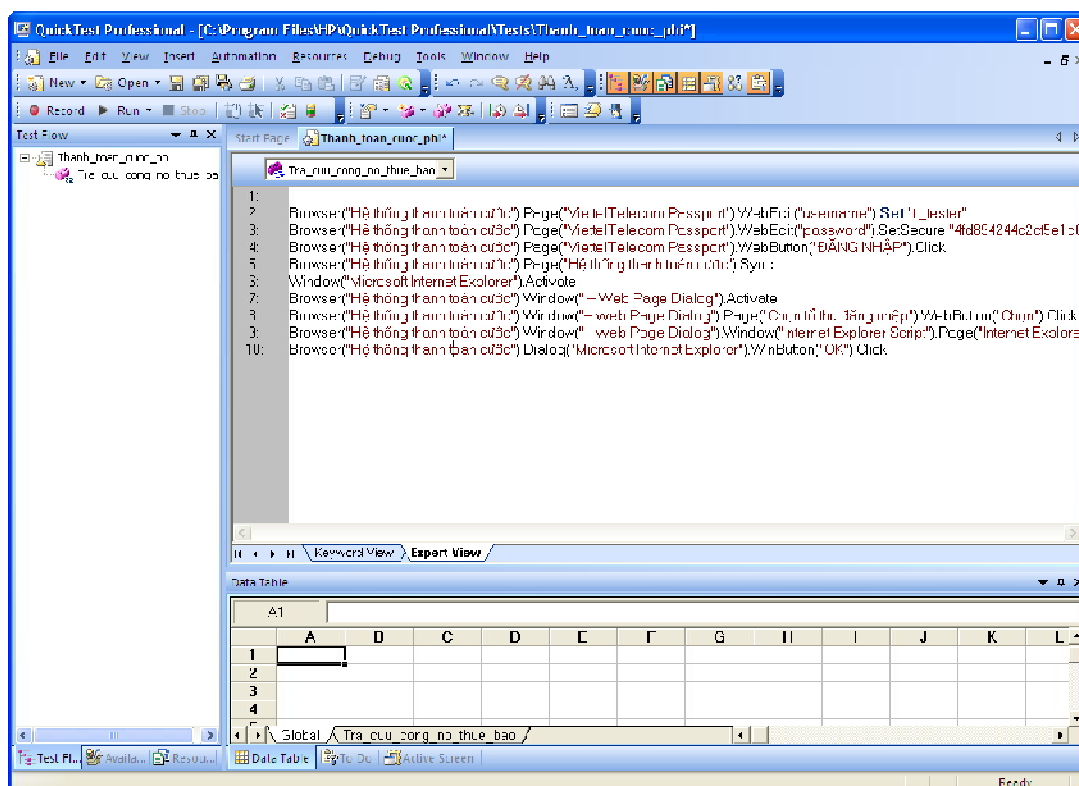
Hệ thống Thanh toán cước phí của dịch vụ viễn thông trả sau hỗ trợ những hoạt động sau đây:

- Tra cứu công nợ thuê bao
- Thực hiện gạch nợ
- Điều chỉnh công nợ thuê bao

3.3.1.3. Thiết kế các trường hợp kiểm thử

3.3.1.4. Thực hiện kiểm thử tự động

Tạo các Test Script cho mỗi trường hợp kiểm thử được mô tả ở mục 3.3.1.3 trên công cụ QuickTest Professional:



Hình 3.9 – Tạo Test Script trong QuickTest Professional

3.2.1.5. Phân tích, đánh giá và tạo báo cáo kiểm thử

Sau khi chạy Test Script, ta có thể xem kết quả của quá trình test thông qua một cửa sổ Test Result. Test Result bao gồm các mô tả từng bước thực hiện của quá trình. Đối với các test không chứa các bảng tham số thì Test Result chỉ thể hiện một quá trình test duy nhất. Ngược lại, với các test bao gồm các bảng tham số, test sẽ được cấu hình để lặp lại nhiều bước và Test Result sẽ hiển thị rõ từng bước lặp của quá trình Test. Các kết quả là một nhóm các hành động trong quá trình test.

Nội dung của cửa sổ Test Result sẽ được thể hiện như sau :

- Một báo cáo tổng quan về Test bao gồm các trạng thái pass, fail, done.
- Dữ liệu được sử dụng trong quá trình chạy test.
- Một cây mở rộng các bước trong quá trình test, chỉ rõ chính xác nơi nào ứng dụng

hay thao tác xuất hiện lỗi cùng với các diễn tả về lỗi đó.

- Các hình ảnh minh họa bất cứ sai sót nào trong mọi trạng thái test.

+ Từ việc kết quả chạy được ở trên ta tạo báo cáo tổng hợp kết quả kiểm thử như sau:

Bảng 3.4 – Báo cáo kết quả kiểm thử các chức năng

STT	Chức năng	Số trường hợp kiểm thử (Done)	Số trường hợp kiểm thử thành công (Pass)	Số trường hợp kiểm thử thất bại (Fail)
1	Tra cứu công nợ thuê bao	7	5	2
2	Thanh toán cước cho thuê bao trả sau	13	12	1
3	Điều chỉnh công nợ thuê bao	8	8	0

3.3.2. Thử nghiệm kiểm thử hiệu năng

Các bước kiểm thử hiệu năng cho ứng dụng “Thanh toán cước phí dịch vụ viễn thông trả sau” như sau:

3.3.2.1. Xác định các mục tiêu kiểm thử

Để đáp ứng nhu cầu nhiều người dùng thực hiện chức năng gạch nợ cho các thuê bao thành công trong khoảng thời gian chấp nhận được, ta tiến hành xác định các tham số sau cần đo hiệu năng:

- Tổng số người dùng đồng thời tại thời điểm cao tải
- Thời gian đáp ứng cho mỗi yêu cầu

3.3.2.2. Chuẩn bị môi trường kiểm thử hiệu năng

3.3.2.3. Tạo kịch bản kiểm thử hiệu năng

3.3.2.4. Thực hiện kiểm thử hiệu năng

3.3.2.5. Phân tích và đánh giá kết quả

3.4. Kết luận chương

Trên cơ sở lý thuyết về kiểm thử đã nghiên cứu và trình bày ở các chương trên, luận văn đã phân tích cụ thể nhu cầu của việc kiểm thử tự động, sử dụng công cụ kiểm thử tự động đem lại lợi ích cả về mặt thời gian và chi phí cho việc phát triển phần mềm. Từ đó, luận văn đã đề xuất đưa ra quy trình kiểm thử chức năng tự động, quy trình kiểm thử hiệu năng và áp dụng thử nghiệm tại đơn vị đang công tác – Trung tâm phần mềm Viettel, đồng thời áp dụng cho một ứng dụng cụ thể, dự án **Thanh toán cước phí dịch vụ viễn thông trả sau** với các luồng nghiệp vụ chính: Tra cứu công nợ, thanh toán cước, điều chỉnh giao dịch thanh toán. Trong đó:

Việc áp dụng kiểm thử chức năng tự động bằng công cụ QuickTest Professional đã được mô phỏng bằng các thuật toán sắp xếp chọn.

Việc kiểm thử hiệu năng tập trung vào 2 thông số sau:

- Số người dùng truy cập đồng thời
- Thời gian đáp ứng cho mỗi giao dịch

Trên cơ sở đó ta sử dụng công cụ LoadRunner thực hiện kiểm thử lại ứng dụng với các thông số thu được. Trong LoadRunner ta có thể kết hợp nhiều kịch bản tải, mô phỏng trạng thái, hành vi người dùng tạo nên kịch bản tải gần giống với thực tế giúp ta đánh giá hệ thống một cách đúng đắn, đây chính là điểm nổi bật của Load Runner. Bên cạnh đó, Load Runner còn có chức năng Analysis thực hiện phân tích kết quả kiểm thử hiệu năng giúp cho ta có cái nhìn tổng quát và chi tiết về kết quả kiểm thử hiệu năng một cách nhanh chóng. Nếu kết quả cho thấy hệ thống đã quá tải thì cần phải có các điều chỉnh như: mở rộng băng thông, tối ưu code, áp dụng cơ chế load balancing (phân tải),..

KẾT LUẬN VÀ KIẾN NGHỊ

1. Kết luận

Qua thời gian làm việc, với sự nỗ lực của bản thân và được sự hướng dẫn tận tình của PGS.TS Huỳnh Quyết Thắng, tôi đã hoàn thành Luận văn của mình.

Những nội dung chính đã được giải quyết trong luận văn:

- Nêu rõ kiểm thử phần mềm là một hoạt động quan trọng nhằm đảm bảo chất lượng phần mềm. Việc nghiên cứu lựa chọn các kỹ thuật và chiến lược kiểm thử phần mềm phù hợp giúp cho việc kiểm thử có hiệu quả, giảm chi phí và thời gian. Việc xây dựng tài liệu kiểm thử phần mềm hợp lý sẽ giúp cho việc tổ chức, quản lý và thực hiện kiểm thử có hiệu quả.
- Đề xuất quy trình kiểm thử chức năng tự động sử dụng công cụ QuickTest Professional, quy trình kiểm thử hiệu năng sử dụng công cụ Load Runner.
- Ứng dụng quy trình đề xuất trên với một số dự án cụ thể.
- Kết quả nghiên cứu được áp dụng thực tế cho các dự án phát triển phần mềm tại đơn vị công tác cho các dự án nâng cấp, có thể tái sử dụng để triển khai tại các nước khác

Hiện nay, vấn đề kiểm thử phần mềm hầu như vẫn chưa được đầu tư và quan tâm đúng mức. Và Việt Nam đang trong quá trình xây dựng một ngành công nghiệp phần mềm thì không thể xem nhẹ việc kiểm thử phần mềm vì xác suất thất bại sẽ rất cao, hơn nữa, hầu hết các công ty phần mềm có uy tín đều đặt ra yêu cầu nghiêm ngặt là nếu một phần mềm không có tài liệu kiểm thử đi kèm thì sẽ không được chấp nhận.

2. Kiến nghị

Việc ứng dụng công cụ phần mềm tự động chắc chắn sẽ giúp giảm công sức của cán bộ kiểm thử đồng thời làm tăng chất lượng phần mềm. Trong điều kiện Việt Nam hiện nay, với tỷ trọng gia công phần mềm ngày càng lớn tại các công ty phần mềm, tôi cho rằng kiểm tra phần mềm tự động với những công cụ như LoadRunner, QuickTest Professional rất đáng để các doanh nghiệp phần mềm quan tâm nghiên cứu, đầu tư và ứng dụng.