



# Một ví dụ

Bởi:

Khoa CNTT ĐHSP KT Hưng Yên

Một ví dụ về mô tả kiểu test:

*Kỹ thuật:*

- Functional Test

Đối với chu trình sự kiện của mỗi UC, sẽ xác định một tập các giao dịch đại diện cho mỗi hành động của tác nhân khi thực hiện UC.

Tối thiểu phải có 2 TC cho mỗi giao dịch, một TC để phản ánh điều kiện tích cực và một phản ánh điều kiện tiêu cực (không được chấp nhận)

Trong giai đoạn đầu tiên, các UC 1-4 và 12 sẽ được test, theo hình thức sau:

UC 1 bắt đầu với tác nhân đã truy cập thành công vào ứng dụng và tại cửa sổ chính, và kết thúc khi người dùng xác định SAVE.

Mỗi TC sẽ được tiến hành và thực hiện bằng cách sử dụng Rational Robot.

Việc kiểm tra và đánh giá việc thực hiện mỗi TC sẽ được thực hiện theo phương pháp sau:

*Thực hiện Test script (Mỗi test script có được thực hiện thành công như mong muốn không?)*

Tình trạng Window hoặc phương pháp kiểm tra Object Data (tiến hành trong các test script) sẽ được dùng để kiểm tra sự hiển thị của các màn hình chính và dữ liệu được xác định được nắm bắt/hiển thị bởi mục tiêu test trong khi thực hiện test.

Cơ sở dữ liệu của các mục tiêu test (sử dụng Microsoft Access) sẽ được kiểm tra trước khi test và kiểm tra lại sau khi test để kiểm chứng rằng các thay đổi thực hiện trong quá trình test đã được phản ánh chính xác trong dữ liệu.

- Performance Test:

Một ví dụ

Với mỗi UC, xác định một tập các giao dịch, như định nghĩa trong tài liệu phân tích workload, sẽ được tiến hành và thực hiện bằng Rational Suite PerformanceStudio và Rational Robot (GUI scripts)

Ít nhất 3 workload được phản ánh trong test script và lịch trình thực hiện test, bao gồm:

Stressed workload: 750 người dùng (15 % quản lý, 50 % bán hàng, 35 % marketing)

Peak workload: 350 người dùng (10 % quản lý, 60 % bán hàng, 30 % marketing)

Nominal workload: 150 người dùng (2 % quản lý, 75% bán hàng, 23 % marketing)

Test script dùng để thực hiện mỗi giao dịch sẽ bao gồm bộ đếm thời gian tương tự để đo thời gian phản hồi, ví dụ tổng thời gian giao dịch (như định nghĩa trong tài liệu phân tích workload), và các hoạt động giao dịch chính hoặc thời gian xử lý.

Test script sẽ thực hiện các workload trong 1 giờ (trừ phi được ghi chú khác trong tài liệu phân tích workload).

Kiểm tra và đánh giá việc thực hiện mỗi thực hiện test (của một workload) bao gồm:

Thực hiện test được theo dõi bằng biểu đồ trạng thái (để xác định rằng việc test và workload được thực hiện như mong muốn)

Thực hiện test script (mỗi test script có được thực hiện thành công như mong đợi không?)

Ghi nhận và đánh giá thời gian phản hồi đã định nghĩa bằng các báo cáo sau:

- Performance Percentile
- Response Time

*Điều kiện hoàn thành :*

Tất cả các TC có trong kế hoạch đều đã được thực hiện

Tất cả các lỗi được xác định phải được ghi nhận vào một giải pháp đã thỏa thuận (All identified defects have been addressed to an agreed upon resolution)

Tất cả các TC có trong kế hoạch đã được thực hiện lại và toàn bộ các lỗi mở đã được ghi nhận như đã thỏa thuận và không có lỗi mới nào được phát hiện

Hoặc

Một ví dụ

Toàn bộ các TC đặt mức ưu tiên cao đều đã được thực hiện

Toàn bộ các lỗi tìm thấy đều được ghi nhận vào một giải pháp đã thỏa thuận

Toàn bộ các lỗi có trọng số 1 và 2 đều được giải quyết

Tất cả các TC có mức ưu tiên cao đều đã được thực hiện lại và toàn bộ các lỗi mở đã được ghi nhận như đã thỏa thuận và không có lỗi mới nào được phát hiện

Các vấn đề đặc biệt

- Cơ sở dữ liệu test yêu cầu người thiết kế hoặc quản trị CSDL hỗ trợ để tạo mới, cập nhật và làm tươi dữ liệu test
- Việc test hiệu suất hệ thống sử dụng máy chủ trong mạng hiện tại (có hỗ trợ cả các giao dịch khác không thuộc việc test). Việc test sẽ phải được lập lịch vào những giờ không còn các giao dịch khác trên mạng.
- Mục tiêu test phải đồng nhất với hệ thống hợp lệ (hoặc giả lập đồng bộ) để việc test chức năng có thể được tiến hành và thực hiện
- Việc test có thể bị dừng khi <số lỗi vượt quá norm, ...>
- Cán bộ test có thể dừng test khi lập trình viên không thực hiện unit test, ...

>

## Test chức năng (Function Testing)

<Mục đích của test chức năng là tập trung vào các yêu cầu test có thể được lưu vết trực tiếp trong các UC hoặc các chức năng và qui tắc nghiệp vụ. Mục tiêu của kiểu test này là kiểm tra tính đúng đắn của các dữ liệu, qui trình và báo cáo cũng như việc thực hiện đúng những qui tắc nghiệp vụ. Kiểu test này dựa vào kỹ thuật black box, tức là kiểm tra ứng dụng và các xử lý nội tại bằng cách tương tác với ứng dụng thông qua giao diện người sử dụng và phân tích các kết quả hoặc đầu ra. Bảng sau liệt kê một số gợi ý đối với mỗi ứng dụng:

Mục đích test:	<Đảm bảo mục tiêu test đúng đắn của chức năng, bao gồm định hướng, dữ liệu đầu vào, xử lý và dữ liệu nhận được>
Cách thực hiện:	<Thực hiện mỗi UC, chu trình UC hoặc chức năng, sử dụng dữ liệu hợp lệ và không hợp lệ để kiểm tra:- Kết quả mong đợi với dữ liệu hợp lệ.- Lỗi thích hợp hoặc thông báo hiển thị khi dữ liệu không hợp lệ.- Mỗi qui tắc nghiệp vụ đều được áp dụng đúng>
Điều kiện	- <Toàn bộ kế hoạch test đã được thực hiện.- Toàn bộ các lỗi phát hiện ra đã được ghi nhận.>

Một ví dụ

hoàn thành:	
Các vấn đề đặc biệt:	<Xác định hoặc mô tả các vấn đề (nội bộ hoặc bên ngoài) ảnh hưởng đến việc test chức năng>

>

### Test giao diện người sử dụng (User Interface Testing)

<Test giao diện người dùng (UI) kiểm tra các tương tác của người dùng với phần mềm. Mục tiêu của test UI là để đảm bảo rằng giao diện người dùng cung cấp cho người sử dụng cách truy cập và sử dụng thích hợp thông qua các chức năng trong mục tiêu test. Ngoài ra, test UI còn để đảm bảo rằng các đối tượng trong phạm vi chức năng UI giống như mong đợi và phù hợp với tổ chức hoặc chuẩn ngành.>

Mục đích test:	<Kiểm tra:? Việc sử dụng thông qua mục tiêu test phản ánh đúng các chức năng và yêu cầu nghiệp vụ, bao gồm màn hình đến màn hình, trường đến trường và sử dụng các phương pháp truy cập (phím tabs, di chuột, tổ hợp phím)? Các đối tượng và thuộc tính màn hình như menus, size, position, state, và tập trung vào việc tương thích với chuẩn>
Cách thực hiện:	<Tạo ra và chỉnh sửa test cho mỗi màn hình để kiểm tra việc sử dụng đúng cách và tình trạng các đối tượng cho mỗi màn hình và đối tượng của ứng dụng>
Điều kiện hoàn thành:	<Mỗi màn hình được kiểm tra thành công đúng với phiên bản kiểm tra hoặc phạm vi chấp nhận được>
Các vấn đề đặc biệt:	<Không phải toàn bộ các thuộc tính của các đối tượng đều truy cập được>

## Test dữ liệu và tích hợp dữ liệu (Data and Database Integrity Testing)

<Cơ sở dữ liệu và xử lý cơ sở dữ liệu phải được test như một hệ thống con trong dự án. hệ thống con này phải được test không cần thông qua giao diện người dùng để giao tiếp với dữ liệu. Nghiên cứu thêm về DBMS để xác định các công cụ và kỹ thuật có thể có giúp hỗ trợ cho việc test:>

Mục đích test:	<Đảm bảo rằng các phương pháp truy cập và chức năng xử lý là đúng và không có sai lệch dữ liệu>
Cách thực hiện:	? <Thực hiện từng phương pháp truy cập và xử lý, thử từng trường hợp với dữ liệu hợp lệ và không hợp lệ hoặc các yêu cầu dữ liệu.? Kiểm tra cơ sở dữ liệu để đảm bảo rằng dữ liệu được lưu trữ như mong đợi, toàn bộ các sự kiện với cơ sở dữ liệu xảy ra đều đúng, học xem xét các dữ liệu trả về để đảm bảo rằng đã nhận được dữ liệu đúng cho các lý do đúng>
Điều kiện hoàn thành:	<Tất cả các phương pháp truy cập và chức năng xử lý đều giống như thiết kế và không có sai lệch dữ liệu>
Các vấn đề đặc biệt:	? <Việc test có thể đòi hỏi phải môi trường phát triển DBMS hoặc drivers để truy cập hoặc sửa dữ liệu trực tiếp trong cơ sở dữ liệu.? Các xử lý phải được thực hiện bằng tay.? Cơ sở dữ liệu có kích thước nhỏ hoặc tối thiểu (giới hạn số bản ghi) phải được dùng để làm rõ thêm các sự kiện không được phép chấp nhận>

>

## Test chu trình nghiệp vụ (Business Cycle Testing)

<Test chu trình nghiệp vụ phải thực hiện các hoạt động trong dự án qua thời gian. Phải xác định một chu kỳ, ví dụ một năm, và các giao dịch và hoạt động có thể xảy ra trong chu kỳ của năm đó phải được thực hiện. Việc này bao gồm cả các chu kỳ hàng ngày, hàng tuần hoặc hàng tháng và các sự kiện là ảnh hưởng bởi ngày tháng, ví dụ như ứng dụng ngân hàng>

Mục đích test:	<Đảm bảo mục đích của test là đúng đắn và các tiến trình chạy ngầm thực hiện đúng yêu cầu về mô hình nghiệp vụ và lịch trình>
----------------	---

<p>Cách thực hiện:</p>	<p>&lt;Việc test sẽ giả lập vài chu trình nghiệp vụ bằng cách thực hiện các công việc sau:? Các test dùng cho việc test chức năng sẽ được sửa lại hoặc nâng cấp để tăng số lần mỗi chức năng được thực hiện để giả lập một số người dùng khác nhau trong chu kỳ đã định.? Toàn bộ các chức năng theo ngày tháng sẽ được thực hiện với dữ liệu hợp lệ và không hợp lệ hoặc chu kỳ thời gian? Toàn bộ các chức năng xảy ra trong lịch trình chu kỳ sẽ được thực hiện vào thời gian thích hợp? Việc test sẽ bao gồm cả dữ liệu hợp lệ và không hợp lệ để kiểm tra: - Kết quả xảy ra khi dữ liệu hợp lệ. - Lỗi tương tự hoặc cảnh báo hiển thị khi dữ liệu không hợp lệ.? Mỗi qui tắc nghiệp vụ đều được áp dụng.</p>
<p>Điều kiện hoàn thành:</p>	<p>? &lt;Toàn bộ kế hoạch test đã được thực hiện.? Toàn bộ các lỗi phát hiện ra đều được ghi nhận&gt;</p>
<p>Các vấn đề đặc biệt:</p>	<p>? &lt;Ngày và các sự kiện của hệ thống có thể đòi hỏi các hoạt động hỗ trợ đặc biệt? Mô hình nghiệp vụ đòi hỏi xác định các yêu cầu và thủ tục test thích hợp&gt;</p>

## Test hiệu suất (Performance testing)

### Performance Profiling

<Performance profiling là một dạng test hiệu suất trong đó thời gian phản hồi, tỷ lệ giao dịch và các yêu cầu phụ thuộc thời gian khác được đo đạc và đánh giá. Mục đích của Performance Profiling là kiểm tra các yêu cầu về hiệu suất có đạt được hay không. Performance profiling là tiến hành và thực hiện để mô tả sơ lược và điều chỉnh các hành vi hiệu suất của mục tiêu test như một hàm của các điều kiện ví dụ workload hoặc cấu hình phần cứng.

Chú ý: Các giao dịch dưới đây tham chiếu đến “các giao dịch nghiệp vụ logic”. Các giao dịch này được định nghĩa như xác định các UC mà tác nhân của hệ thống hy vọng được thực hiện bằng cách sử dụng mục tiêu test, như thêm mới hoặc sửa một hợp đồng>

<p>Mục đích test:</p>	<p>&lt;Kiểm tra các biểu hiện về hiệu suất cho các giao dịch hoặc chức năng nghiệp vụ đã thiết kế theo những điều kiện sau:? workload bình thường đã biết trước (normal anticipated workload)? workload xấu đã biết trước (anticipated worst case workload)&gt;</p>
-----------------------	---

Cách thực hiện:	? <Sử dụng các thủ tục test cho test chức năng và chu trình nghiệp vụ? Chỉnh sửa file dữ liệu để tăng số lượng các giao dịch hoặc scripts để tăng số tương tác xảy ra trong mỗi giao dịch? Scripts phải được chạy trên một máy (trường hợp tốt nhất để đánh giá người dùng đơn lẻ, giao dịch đơn lẻ) và phải lặp lại trên nhiều máy trạm (ảo hoặc thực, xem các vấn đề đặc biệt dưới đây)>
Điều kiện hoàn thành:	? <Giao dịch đơn lẻ hoặc người dùng đơn lẻ: Thực hiện thành công test script không có lỗi và trong phạm vi mong đợi hoặc thời gian phản hồi cho mỗi giao dịch>? <Nhiều giao dịch hoặc nhiều người dùng: Thực hiện thành công test script không có lỗi và trong thời gian chấp nhận được>
Các vấn đề đặc biệt:	<Việc test hiệu suất toàn diện bao gồm phải có một workload nền trên máy chủ. Có một số phương pháp để thực hiện, bao gồm:? “Drive transactions” trực tiếp đến máy chủ, thường trong các form gọi SQL.? Tạo các người dùng ảo để giả lập nhiều máy trạm, thường là vài trăm. Sử dụng công cụ Remote Terminal Emulation để thực hiện việc load này, kỹ thuật này còn được dùng để load giao dịch trên mạng? Sử dụng nhiều người dùng, mỗi người chạy một test script để load lên hệ thống Test hiệu suất phải được thực hiện trên máy chuyên dụng hoặc thời gian chuyên dùng. Điều đó cho phép việc tính toán được đầy đủ và chính xác. Cơ sở dữ liệu sử dụng để test hiệu suất phải có kích thước thực tế hoặc đo bằng nhau>

## Load Testing

<Load testing là một kiểu test hiệu suất mà mục tiêu là kiểm tra workload để tính toán và đánh giá hiệu suất và khả năng của mục đích test để tiếp tục thực hiện các chức năng thích hợp với các workload khác. Mục đích của load testing là xác định và đảm bảo các chức năng hệ thống thích hợp với nhiều nhất các workload. Ngoài ra load testing còn đánh giá các tính năng hiệu suất như thời gian phản hồi, tỉ lệ giao dịch và các vấn đề liên quan đến thời gian khác.>

<Chú ý: Các giao dịch dưới đây tham chiếu đến “các giao dịch nghiệp vụ logic”. Các giao dịch này được định nghĩa như các chức năng xác định mà người dùng cuối của hệ thống mong muốn thực hiện thông qua ứng dụng như thêm hoặc sửa các thông tin hợp đồng>

Mục tiêu test:	<Kiểm tra hiệu suất về thời gian cho các giao dịch hoặc tình huống nghiệp vụ đã thiết kế với nhiều điều kiện workload>
----------------	--

Cách thực hiện:	? <Sử dụng các test đã xây dựng cho test chức năng và chu trình nghiệp vụ.? Sửa lại file dữ liệu để tăng số lượng giao dịch hoặc test nhằm tăng thêm số lần thực hiện mỗi giao dịch>
Điều kiện hoàn thành:	<Nhiều giao dịch hoặc nhiều người dùng: Thực hiện thành công việc test không có lỗi và trong thời gian chấp nhận được>
Các vấn đề đặc biệt:	? <Load testing phải được thực hiện trên máy chuyên dụng hoặc vào những giờ chuyên biệt. Nó cho phép đo đạc đầy đủ và chính xác.? Cơ sở dữ liệu dùng cho load testing phải có kích thước thực tế hoặc đo bằng nhau>

## Stress Testing

<Stress testing là một kiểu test hiệu suất được thực hiện để tìm ra các lỗi trong trường hợp thiếu tài nguyên hoặc cạnh tranh về tài nguyên. Bộ nhớ hoặc dung lượng đĩa ít có thể làm xuất hiện lỗi trong mục đích test mà nó không xuất hiện dưới điều kiện bình thường. Các lỗi khác có thể là kết quả của việc cạnh tranh hoặc chia sẻ tài nguyên như khóa cơ sở dữ liệu hoặc băng thông mạng. Stress testing cũng được dùng để xác định wordload tối đa mà mục đích test có thể điều khiển được.>

<Chú ý: Tham khảo các giao dịch dưới đây tham chiếu đến các giao dịch nghiệp vụ logic>

Mục đích test:	<Kiểm tra các chức năng của mục đích test là đúng đắn và không có lỗi với những điều kiện sau:? Có ít hoặc không có bộ nhớ phù hợp trên máy chủ (RAM và DASD)? Số lượng máy trạm tối đa trong thực tế hoặc giả lập kết nối vào máy chủ? Nhiều người dùng thực hiện cùng một giao dịch với cùng dữ liệu hoặc account? Độ lớn các giao dịch xấu hoặc hỗn hợp (xem phần Performance Testing ở trên).Chú ý: Mục đích của Stress Testing có thể được phát biểu rõ và ghi ra các điều kiện mà hệ thống có thể lỗi, không thể tiếp tục thực hiện các chức năng một cách thích hợp>
Cách thực hiện:	? <Sử dụng các test đã xây dựng để thực hiện Performance Profiling hoặc Load Testing.? Để test việc hạn chế tài nguyên, test phải chạy trên máy đơn lẻ và RAM và DASD trên máy chủ phải giảm đi hoặc hạn chế? Để thực hiện các stress tests khác phải sử dụng nhiều người dùng cùng chạy một TC hoặc bổ sung các test để thực hiện độ lớn giao dịch xấu hoặc hỗn hợp.>



Điều kiện hoàn thành:	<Toàn bộ kế hoạch test được thực hiện và các hạn chế của hệ thống được xác định thỏa mãn các điều kiện tối thiểu đã đặt ra hoặc chỉ sai trong trong hợp các điều kiện không nằm trong điều kiện đã xác định>
Các vấn đề đặc biệt:	? <Việc test Stressing mạng có thể đòi hỏi những công cụ để load mạng với nhiều thông báo hoặc gói dữ liệu.? DASD dùng cho hệ thống phải tạm thời giảm xuống để hạn chế khả năng chỗi trống cho tăng trưởng cơ sở dữ liệu.? Đồng bộ hóa các máy trạm đồng thời truy cập vào cùng một bản ghi hoặc các account dữ liệu>

## Volume Testing

<Mục tiêu của Volume Testing là để kiểm tra giới hạn của độ lớn của dữ liệu có thể làm phần mềm bị sai. Volume Testing cũng xác định load lớn nhất liên tục hoặc độ lớn mà mục đích test có thể điều khiển được trong chu kỳ đã cho. Ví dụ, nếu mục đích test là xử lý một tập các bản ghi để tạo báo cáo, Volume Test có thể dùng một cơ sở dữ liệu test lớn và kiểm tra xem phần mềm có chạy bình thường và cho ra báo cáo đúng không>

Mục đích test:	<Kiểm tra xem mục tiêu test có thực hiện thành công các chức năng theo những điều kiện sau không:? Số máy trạm lớn nhất kết nối (thực tế hoặc vật lý – có thể), hoặc giả lập, tất cả đều thực hiện cùng một chức năng nghiệp vụ trong một chu kỳ mở rộng.? Kích thước cơ sở dữ liệu lớn nhất có thể (thực tế hoặc đo được) và nhiều query hoặc giao dịch báo cáo được thực hiện đồng thời.>
Cách thực hiện:	? <Sử dụng các test đã xây dựng cho Performance Profiling hoặc Load Testing.? Có thể dùng nhiều người dùng, chạy cùng một test hoặc bổ sung các test để thực hiện trường hợp giao dịch volume hoặc hỗn hợp xấu nhất (xem Stress Testing ở trên) trong một chu kỳ mở rộng.? Tạo ra cơ sở dữ liệu lớn nhất (thực tế, qui đổi, hoặc lọc các dữ liệu đại diện) và nhiều người dùng chạy các query và giao dịch báo cáo đồng thời trong một chu kỳ mở rộng>
Điều kiện hoàn thành:	? <Toàn bộ kế hoạch test được thực hiện và các giới hạn của hệ thống được xác định là đạt tới hoặc xử lý mà không có lỗi>
Các vấn đề	<Chu kỳ thời gian như thế nào là chấp nhận được cho điều kiện cơ sở dữ liệu lớn, như đã nói ở trên?>

đặc biệt:	
-----------	--

### Test Bảo mật và Kiểm soát truy cập (Security and Access Control Testing)

<Test bảo mật và kiểm soát truy cập tập trung vào hai lĩnh vực bảo mật chính:

- Bảo mật ở mức ứng dụng, bao gồm truy cập dữ liệu và các chức năng nghiệp vụ
- Bảo mật ở mức hệ thống, bao gồm truy cập vào hệ thống hoặc truy cập từ xa

Bảo mật mức ứng dụng đảm bảo rằng, dựa trên bảo mật đã yêu cầu, người dùng bị hạn chế sử dụng một số chức năng hoặc tình huống sử dụng, hoặc bị hạn chế trong giới hạn dữ liệu phù hợp với họ. Ví dụ, mọi người có thể được phép nhập dữ liệu để tạo account nhưng chỉ có người quản lý có thể xóa chúng. Nếu là bảo mật ở mức dữ liệu, việc test đảm bảo rằng “người dùng nhóm 1” có thể nhìn thấy các thông tin khách hàng, bao gồm dữ liệu tài chính, tuy nhiên “người dùng nhóm 2” chỉ nhìn thấy các thông tin chung chung cho cùng một khách hàng.

Bảo mật mức hệ thống đảm bảo rằng chỉ những người dùng được cho quyền truy cập vào hệ thống mới có khả năng truy cập vào ứng dụng và chỉ bằng các cổng thích hợp

>

Mục đích test:	? Bảo mật mức ứng dụng: Đảm bảo rằng một người dùng chỉ có thể truy cập vào những chức năng hoặc dữ liệu mà nhóm người dùng đó được phép? Bảo mật mức hệ thống: Đảm bảo rằng chỉ những người được phép truy cập hệ thống và ứng dụng được phép truy cập chúng
Cách thực hiện:	? Bảo mật ứng dụng: Xác định và liệt kê từng nhóm người dùng và các chức năng hoặc dữ liệu mà họ được phép truy cập? Tạo test case cho mỗi nhóm người dùng và kiểm tra từng quyền bằng cách tạo các giao dịch xác định cho mỗi nhóm? Sửa lại nhóm người dùng và chạy lại tình huống test cho cùng những người dùng. Với mỗi trường hợp, kiểm tra các chức năng thêm vào hoặc dữ liệu có đúng không hay bị từ chối.? Truy cập mức hệ thống: tham khảo các điều kiện đặc biệt dưới đây
Điều kiện hoàn thành:	<Với mỗi nhóm người dùng đều có các chức năng hoặc dữ liệu thích hợp, và toàn bộ các chức năng giao dịch đều như dự kiến và chạy trong các test chức năng ứng dụng trước đó>

## Một ví dụ

Các vấn đề đặc biệt:	<Truy cập vào hệ thống phải được xem xét hoặc thảo luận với quản trị hệ thống hoặc quản trị mạng, có thể không cần nếu nó là chức năng của quản trị mạng hoặc quản trị hệ thống>
----------------------	--

## Test hồi qui (Regression Testing)

<Test hồi qui là một hoạt động cần thiết để chỉ ra rằng việc thay đổi code không gây ra những ảnh hưởng bất lợi>

Mục đích test:	Test hồi qui dùng để kiểm tra các phần được sửa chữa trong phần mềm, để đảm bảo rằng những sự thay đổi đó không gây ra lỗi trong những phần khác
Cách thực hiện:	? <Tái sử dụng các TC từ những phần test trước để test các module đã được sửa chữa>.? <Sử dụng công cụ Rational Robot: Tạo một số test script về chức năng. Định nghĩa lịch thực hiện tự động cho chúng>? <80% các TC được chọn ngẫu nhiên>? <Xây dựng một chương trình phân tích sơ sở hạ tầng. Chúng ta dựng một cơ sở hạ tầng có thể mở rộng được để thực hiện và đánh giá chương trình phân tích. Dựa vào kết quả phân tích chúng ta xác định phạm vi cần test hồi qui.>
Điều kiện hoàn thành:	? <Toàn bộ các TC được thực hiện và đạt yêu cầu>? <Toàn bộ các TC được chọn được thực hiện và đạt yêu cầu>
Các vấn đề đặc biệt:	