**Làm thế nào để review tài liệu đặc tả yêu cầu (SRS)   
và tạo kịch bản kiểm thử (Test Scenario).**

Bài viết được tham khảo từ nguồn:

<http://www.softwaretestinghelp.com/rview-srs-document-and-create-test-scenarios-software-testing-training-course-day-2/>

Hôm nay chúng ta cùng nhau đi tìm hiểu về vấn đề làm thế nào để viết test scenarios từ tài liệu đặc tả yêu cầu.

Trước hết chúng ta hãy cùng tìm hiểu về khái niệm:

Test scenario là gì?

Test scenarion là một kịch bản trong đó có chứa các test case liên quan đến kịch bản đó.

Tài liệu đặc tả yêu cầu là gì? (SRS document).

Tài liệu đặc tả yêu cầu là những yêu cầu chính thức về những gì cần phải thực hiện của đội phát triển phần mềm. Tài liệu đặc tả yêu cầu nên bao gồm tất cả các định nghĩa về yêu cầu của người sử dụng và đặc tả yêu cầu của hệ thống. Tài liệu đặc tả yêu cầu không phải là tài liệu thiết kế hệ thống. Nó chỉ thiết lập những gì hệ thống phải làm chứ không phải việc mô tả rõ nó sẽ làm việc như thế nào?

Nào, bây giờ chúng ta hãy cùng bắt tay vào việc phân tích chi tiết về cách hướng một tài liệu đặc tả xảy ra, chúng ta cần xác định các bước cần phải làm trong giai đoạn này là gì, trước khi bắt đầu chúng ta cần phải thực hiện bước nào đầu tiên, những thách thức là gì khi chúng ta đối mặt...trong một vấn đề cụ thể.

**Pha Design trong vòng đời phát triển của phần mềm (SDLC-Software left cycle)**

Pha tiếp theo trong vòng đời phát triển của phần mềm là “Design” – đây là nơi yêu cầu chức năng được dịch đến kỹ thuật cụ thể. Đội phát triển, thiết kế, môi trường hay là dữ liệu đều tham gia vào pha này. Kết quả của bước này thường là một tài liệu kỹ thuật thiết kế (viết tắt là: TDD). Đầu vào là tài liệu mô tả yêu cầu của hệ thống cho cả quá trình tạo mới TDD và đội đảm bảo chất lượng để bắt đầu làm việc trên các khía cạnh đảm bảo chất lượng của dự án – đó là việc xem xét các tài liệu đặc tả và xác định đối tượng kiểm tra.

**Review tài liệu đặc tả là gì?**

Tài liệu đặc tả là một tài liệu được tạo bởi đội phát triển cùng với sự hợp tác của đội phân tích nghiệp vụ và đội môi trường/dữ liệu. Thông thường, tài liệu này sau khi được hoàn thành sẽ chia sẻ với đội đảm bảo chất lượng thông qua một cuộc họp nơi mà định hướng chi tiết đã được sắp xếp. Đôi khi, đối với một ứng dụng đã tồn tại, chúng ta có thể không cần đến một cuộc họp chính thức và chỉ cần có người hướng dẫn cho chúng ta thông qua tài liệu này. Từ đó chúng ta có thể có những thông tin cần thiết để làm điều này bởi chính mình.

*Review tài liệu đặc tả là không có gì nhưng hãy làm nó thông qua những tài liệu đặc tả chức năng và cố gắng hiểu mục đích của ứng dụng đang mong muốn là gì?*

Các định dạng chính thức và một ví dụ đã được chia sẻ với tất cả các bạn trong bài viết trước đó. Nó không nhất thiết có nghĩa rằng tất cả tài liệu mô tả đặc tả yêu cầu sẽ được ghi lại một cách chính xác. Hình thức luôn luôn là thứ yếu so với nội dung. Một số đội sẽ chỉ chọn cách viết một danh sách liệt kê, một số đội khác sẽ bao gồm các use case, một số đội khác thì lại bao gồm các mẫu ảnh (như các tài liệu đã có) và một số đội chỉ mô tả chi tiết trong đoạn văn.

**Từng bước để review tài liệu đặc tả yêu cầu của phần mềm.**

Step #1: Tài liệu qua nhiều lần sửa đổi, do đó hãy chắc chắn rằng chúng ta sẽ có phiên bản chuẩn của tài liệu tham khảo, tài liệu đặc tả yêu cầu.

Step #2: Xây dựng hướng dẫn về những gì sẽ được mong đợi ở cuối của quá trình review từ mỗi thành viên trong nhóm. Nói cách khác, quyết định về phân phối được mong đợi từ bước này – thông thường, đầu ra của bước này là xác định các kịch bản kiểm thử. Kịch bản kiểm thử sẽ không là gì nhưng một con trỏ dòng “Cái gì được test” cho một chức năng nhất định.

Step #3: Ngoài ra những hướng dẫn về cách chuyển giao này là để trình bày - ý của tôi nghĩa là, các bản mẫu.

Step #4: Quyết định về việc mỗi thành viên của nhóm là làm việc trên toàn bộ tài liệu hoặc chia nhau. Điều này khuyến khích tất cả mọi người nên đọc tất cả mọi thứ vì nó sẽ ngăn chặn sự tập chung kiến thức với các thành viên trong đội. Nhưng trong trường hợp của một dự án khổng lồ, với tài liệu đặc tả yêu cầu chạy đến 1000 trang, cách tiếp cận là chia nhỏ tài liệu ra thành từng module thông minh và phân chia cho các thành viên trong nhóm là điều thực tế nhất.

Step #5: Review tài liệu đặc tả yêu cầu cũng giúp ích trong việc hiểu biết tốt hơn nếu có bất kỳ điều kiện tiên quyết cụ thể cần thiết nào cho việc kiểm tra phần mềm.

Step #6: Là một sản phẩm phụ, một danh sách các truy vấn mà một số chức năng khó để hiểu hoặc nếu có nhiều thông tin cần thiết cần phải được đưa vào chức năng yêu cầu hoặc nếu có lỗi phát sinh trong quá trình làm tài liệu đặc tả yêu cầu đã được định nghĩa.

**Chúng ta cần những gì để bắt đầu?**

• Phiên bản tài liệu mô tả đặc điểm yêu cầu chính xác.

• Hướng dẫn rõ ràng về những người sẽ làm việc và bao nhiêu thời gian mà họ có thể tham gia.

• Một bản mẫu để tạo kịch bản kiểm thử.

• Thông tin khác như: Những người liên hệ trong trường hợp một câu hỏi hoặc người để báo cáo trong trường hợp có mâu thuẫn trong tài liệu.

**Ai sẽ cung cấp những thông tin này?**

Test leader có trách nhiệm hướng dẫn chung để cung cấp tất cả các yếu tố được liệt kê ở phần trên. Tuy nhiên, đầu vào của các thành viên trong team luôn luôn là yếu tố quan trọng cho sự thành công của toàn bộ sự nỗ lực này.

Team lead thường hỏi – Những kiểu nguyên liệu đầu vào là gì? Nó không thể tốt hơn để gán một module nào đó với một ai quan tâm đến nó hơn là một thành viên trong team hay không? Nó sẽ không tốt để quyết định đưa ra ngày cuối cùng dựa trên ý kiến của nhóm nghiên cứu. Ngoài ra, đối với sự thành công của một dự án, các templates là quan trọng. Như một quy luật trung, templates có tỷ lệ cao hơn hiệu quả khi chúng được thiết kế riêng để thuận tiện cho các nhóm cụ thể và thoải mái.

Do đó, cần chú ý rằng, team leads là bất kỳ điều gì là thành viên trong đội. Đưa nhóm của mình gắn cùng với những quyết định từ ngày đến ngày là rất quan trọng cho các hoạt động trơn tru của dự án.

**Tại sao một template cho các kịch bản kiểm thử - nó là không đủ nếu chúng ta chỉ cần thực hiện một danh sách?**

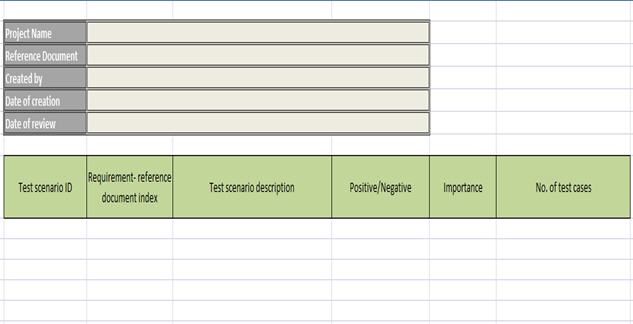
Chắc chắn rồi. Tuy nhiên, các dự án phần mềm không phải là “một người”. Họ tham gia làm việc theo nhóm. Hãy tưởng tượng trong một nhóm bốn người nếu mỗi người trong số họ quyết định để review một module của mỗi đặc tả yêu cầu của phần mềm. Thành viên nhóm 3 được sử dụng một phần mềm notepad. Nhóm team 4 được sử dụng phần mềm word. Làm thể nào để chúng ta có thể củng cố tất cả các công việc được thực hiện cho dự án vào cuối ngày? Ngoài ra, làm thế nào chúng ta có thể quyết định cái nào là tiêu chuẩn và làm thế nào chúng ta có thể khẳng định những gì là đúng và những gì là không đúng nếu chúng ta không tạo ra những nguyên tắc để bắt đầu?

*Đó là những mẫu gì – Một tập hợp các hợp các hướng dẫn và một mẫu thống nhất về tính đồng nhất cho toàn đội.*

**Làm thế nào để tạo một template cho các kịch bản kiểm thử chất lượng phần mềm?**

Templates không phức tạp và phải linh hoạt.

Tất cả cần phải làm là một cơ chế hiệu quả để tạo ra một quá trình kiểm thử hữu ích. Một cái gì đó đơn giản giống như những gì chúng tôi trình bày dưới đây:



Các tiêu đề của những template này có chứa các không gian cần thiết để nắm bắt thông tin cơ bản về dự án, tài liệu hiện tại và tài liệu tham khảo.

Bảng dưới đây sẽ cho chúng ta tạo ra các kịch bản thử nghiệm. Các cột bao gồm:

*Column #1: Test scenario ID*

Mỗi thực thể trong quá trình test phải được định danh (tức là phải có yếu tố để phân biệt với các thực thể khác mà không trùng nhau). Vì vậy, mỗi kịch bản kiểm thử phải được định danh bằng ID. Các quy tắc để tuân theo trong khi gán ID này phải được định nghĩa. Vì lợi ích của bài viết này chúng ta sẽ thực hiện theo các quy ước đặt tên như sau:

* Tiền tố viết tắt cho kịch bản kiểm thử là: TS
* Tiếp theo bởi dấu “\_”
* Tên module: MI
* Tiếp theo bởi dấu “\_”
* Và sau đó là các phần phụ (Ví dụ: MIM cho Module My info, P cho hình ảnh).
* Tiếp theo bởi dấu “\_”
* Theo sau cùng là số serial.

Một ví dụ sẽ là: “TS\_MI\_MIM\_01”.

*Column #2: Requirement*

Nó giúp chúng ta trong việc tạo một kịch bản kiểm thử, chúng ta có thể làm cho nó phù hợp trở lại phần của taid liệu SRS nơi mà chúng ta đã lựa chọn để base trên đó. Nếu yêu cầu có ID chúng ta sẽ sử dụng chúng. Nếu không phần số thậm chí số trang của tài liệu SRS từ nơi mà chúng ta xác đinh được yêu cầu có thể được kiểm thử sẽ làm.

*Column #3: Test scenario description*

Một lớp đệm đặc biết “Cái gì dùng để kiểm thử”. Chúng tôi sẽ đề cập đến nó như là một mục tiêu kiểm thử.

*Column #4: Importance*

Điều này để cung cấp cho một ý tưởng về tầm quan trọng của chức năng nhất định cho giai đoạn AUT. Những giá trị như cao, trung bình và thấp có thể được gán cho lĩnh vực này. Bạn cũng có thể chọn một hệ thống điểm như từ 1 đến 5, trong đó 5 là quan trọng nhất, 1 là ít quan trọng. Dù giá trị lĩnh vực này có thể mất, nhưng nó phải được quyết định trước.

*Column #5: No. of Test cases*

Một ước tính sơ vào có bao nhiêu test case cá nhân chúng ta có thể kết thúc bằng văn bản cho một kịch bản kiểm thử. Ví dụ: Để test chức năng login – tôi thiết lập bao gồm các tình huống: Tên người dùng và mật khẩu chính xác. Tên người dùng đúng và mật khẩu sai. Mật khẩu đúng và tên người dùng sai.

=> Vì vậy, việc chứng thực các chức năng đăng nhập sẽ cho kết quả trong 3 test case.

Note: Bạn có thể mở rộng template này hoặc xóa một số trường mà bạn thấy phù hợp.

Ví dụ:

Bạn có thể thêm “Reviewed by” vào tiêu đề hoặc loại bỏ các ngày tạo…Ngoài ra, trong bảng này có thể bao gồm một trường “Created by” để chỉ định tên người kiểm thử chịu trách nhiệm cho một kịch bản kiểm thử nhất định hoặc loại bỏ cột “No. of Test cases”. Sự lựa chọn là của bạn. Đi đến mục tiêu là những gì tốt nhất cho toàn team đảm bảo chất lượng của phần mềm.

*Bây giờ chúng ta cùng review về một tài liệu đặc tả yêu cầu cụ thể đó là dự án: “Orange HRM” và việc tạo kịch bản kiểm thử.*

Mẹo: Kiểm tra các bảng nội dung trong template tài liệu đặc tả yêu cầu mà chúng tôi đã cung cấp trong hướng dẫn trước để có được một ý tưởng tốt về cách để flow tài liệu và bao nhiêu công việc nó có thể liên quan.

Phần 1 là mục đích của tài liệu. Yêu cầu kiểm thử là không có ở đây.

Phần 2.1 – Cái nhìn chung về dự án – Audience – yêu cầu không thể được kiểm thử ở đây.

Phần 2.2. Phần cứng và hosting.

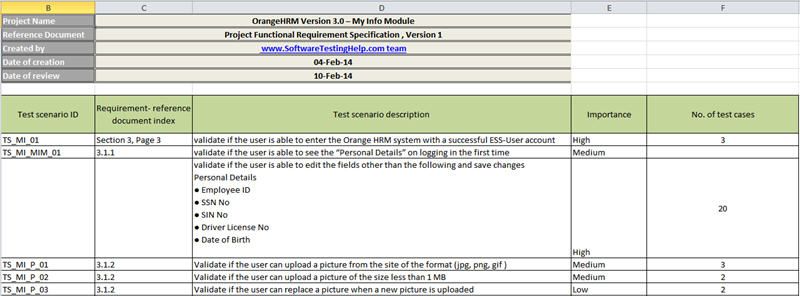
Phần này nói về cách các trang Orange HRM sẽ được tổ chức như thế nào. Bây giờ chúng ta hãy tìm hiểu và hỏi đâu là những thông tin quan trọng mà chúng ta cần kiểm tra?

Một câu trả lời là Yes hoặc No. Yes, bởi vì khi test chúng ta cần có môi trường cái mà như môi trường thời gian thực. Điều này cho chúng ta một ý tưởng về việc làm thế nào nó cần phải giả lập được. Không có lý do gì, bởi vì nó không có thể kiểm thử về yêu cầu – một dạng điều kiện tiên quyết cho hoạt động kiểm thử sẽ xảy ra.

Phần 3: Có một màn hình login ở đây và có chi tiết về các loại tài khoản mà chúng ta cần phải đăng nhập đến trang này. Đây là một yêu cầu có thể kiểm tra. Vì vậy nó là một phần cần thiết cho kịch bản kiểm thử của chúng ta.

Vui lòng xem các tài liệu kịch bản kiểm thử mà các kịch bản kiểm thử cho một vài phần của SRS được thêm vào. Đối với thực hành, vui lòng thêm phần còn lại của kịch bản kiểm thử một cách tương tự. Tuy nhiên, tôi sẽ để vào phần 4 của tài liệu.

Phần 4: Những yêu cầu Aesthetic/ HTML và hướng dẫn – Phần này là để giải thích tại sao một vài yêu cầu có thể không có ý nghĩa với nhóm thử nghiệm trong giai đoạn review SRS, nhưng team nghiên cứu nên thực hiện một lưu ý mà họ yêu cầu để có thể kiểm chứng tất cả là như nhau. Làm thể nào để test và nếu chúng ta cần tập hợp cụ thể lên/ sự giúp đỡ của một vài team để xác nhận nó là cụ thể chúng ta có thể không biết tại thời điểm này. Nhưng chúng một phần là phạm vi kiểm thử của chúng ta và là bước đầu tiên để đảm bảo rằng chúng không thiếu.



*Một số quan sát quan trọng liên quan đến đến review SRS.*

1. Không có thông tin sai nào được phát hiện.
2. Thực hiện việc phân tích tính khả thi về việc có một yêu cầu nào đó là đúng hay không và nó có thể được kiểm tra hay không.
3. Trừ khi có một hiệu suất riêng/ bảo mật hay bất kỳ hình thức khác đã tồn tại trong team test – đó là công việc của chúng tôi để đảm bảo rằng tất cả các yêu cầu phi chức năng cần phải được xem xét.
4. Không phải tất cả các thông tin đều là mục tiêu của những người kiểm thử, vì vậy điều quan trọng là phải hiểu những gì cần lưu ý và những gì không.
5. Tầm quan trọng và “No. of test case” cho một kịch bản kiểm thử không cần phải chính xác và có thể lấp đầy với một giá trị gần đúng hoặc có thể để trống.

**Tóm lại, kết quả review SRS như sau:**

• Danh sách các kịch bản kiểm thử.

• Kết quả review – lỗi tài liệu/ yêu cầu tìm thấy /xác minh các tài liệu SRS.

• Một danh sách các câu hỏi cho việc hiểu tốt nhất – trong bất kỳ trường hợp nào.

• Ý tưởng sơ bộ về môi trường test được cho là giống nhau.

• Xác định phạm vi kiểm thử và một ý tưởng thô trên việc có bao nhiêu test case là đủ để chúng ta có thể kết thúc – như vậy chúng ta có thể xác định được có bao nhiêu lần chúng ta cần cho tài liệu và việc thực hiện cuối cùng.

**Những điểm chú ý quan trọng:**

1. Kịch bản kiểm thử không mở rộng ra bên ngoài (không được chia sẻ với đội phân tích nghiệp vụ hoặc đội Dev) nhưng rất quan trọng cho nội bộ của đội QA. Vì họ là những người đầu tiên để hướng tới mục tiêu của việc test bao phủ 100%. Kịch bản kiểm thử một khi hoàn thành nó trải qua một cuộc đánh giá ngang hàng và một khi đã được thực hiện chúng sẽ được củng cố. Để biết thêm chi tiết về cách các tài liệu được review như thế nào, bạn hãy xem qua bài viết: How to Perform Test Documentation Reviews in 6 Simple Steps (<http://www.softwaretestinghelp.com/test-documentation-reviews/>)
2. Chúng ta có thể sử dụng một công cụ kiểm tra quản lý như HP ALM hoặc qTest để tạo kịch bản kiểm thử. Tuy nhiên, việc tạo ra các kịch bản kiểm thử trong thời gian thực là một hoạt động bằng tay. Theo ý kiến của tôi, phương pháp bằng tay là thuận tiện hơn. Vì nó là bước đầu tiên nên chúng ta không cần phải đi tìm các truy vấn lớn nào cả. Sheet excel là cách đơn giản và hữu ích nhất mà chúng ta nên làm.

Phần tiếp theo tôi sẽ trình bày một ví dụ cụ thể và chi tiết để tạo một kịch bản kiểm thử từ tài liệu đặc yêu cầu.