* 1. **Kiểm thử phát hành**

Kiểm thử phát hành là quá trình kiểm tra một bản phát hành cụ thể của một hệ thống được dự định sử dụng bên ngoài nhóm phát triển. Thông thường, việc phát hành hệ thống là dành cho khách hàng và người dùng. Tuy nhiên, trong một dự án phức tạp, việc phát hành có thể dành cho các đội khác đang phát triển các hệ thống liên quan. Đối với các sản phẩm phần mềm, việc phát hành có thể dành cho quản lý sản phẩm, người sau đó chuẩn bị bán.

Có hai điểm khác biệt quan trọng giữa Kiểm thử phát hành và Kiểm thử hệ thống trong quá trình phát triển:

1. Nhóm phát triển hệ thống, không nên chịu trách nhiệm kiểm thử phát hành..
2. Kiểm tra phát hành là một quá trình kiểm tra xác nhận để đảm bảo rằng một hệ thống đáp ứng các yêu cầu của nó và đủ tốt sử dụng bởi hệ thống khách hàng. Kiểm tra hệ thống bởi nhóm phát triển nên tập trung vào việc phát hiện ra các lỗi trong hệ thống (kiểm tra lỗi).

Mục tiêu chính của quy trình kiểm thử phát hành là thuyết phục nhà cung cấp hệ thống rằng nó đủ tốt để sử dụng. Nếu vậy, nó có thể được phát hành dưới dạng sản phẩm hoặc phân định cho khách hàng. Do đó, kiểm thử phát hành phải chỉ ra rằng hệ thống cung cấp chức năng, hiệu suất và độ tin cậy được chỉ định và không bị lỗi trong quá trình sử dụng thông thường.

Kiểm tra phát hành thường là một quy trình kiểm thử hộp đen, theo đó các kiểm tra được lấy từ đặc tả hệ thống. Hệ thống được coi là một hộp đen mà hành vi của chúng chỉ có thể được xác định bằng cách nghiên cứu các đầu vào của nó và các đầu ra liên quan. Một tên khác cho việc này là kiểm thử chức năng, nó được gọi như vậy là vì người kiểm thử chỉ quan tâm đến chức năng chứ không liên quan đến việc triển khai phần mềm.

* + 1. **Kiểm thử dựa trên yêu cầu**

Một nguyên tắc chung của thực hành kỹ thuật yêu cầu tốt là các yêu cầu phải được kiểm tra. Đó là, yêu cầu phải được viết để có thể thiết kế một bài kiểm tra cho yêu cầu đó. Sau đó, người kiểm tra có thể kiểm tra xem yêu cầu đã được thỏa mãn chưa. Do đó, kiểm thử dựa trên yêu cầu là một cách tiếp cận có hệ thống đối với thiết kế trường hợp kiểm thử trong đó bạn xem xét từng yêu cầu và rút ra một bộ kiểm thử cho nó. Kiểm thử dựa trên yêu cầu là xác nhận thay vì kiểm tra lỗi, bạn đang cố chứng minh rằng hệ thống đã thực hiện đúng yêu cầu của mình..

Ví dụ, hãy xem xét các yêu cầu hệ thống Mentcare sau đây có liên quan đến việc kiểm tra dị ứng thuốc:

*Nếu một bệnh nhân được biết là dị ứng với bất kỳ loại thuốc cụ thể nào, thì việc kê đơn thuốc đó sẽ dẫn đến một thông điệp cảnh báo được đưa ra cho người sử dụng hệ thống.*

*Nếu một người kê đơn chọn bỏ qua một cảnh báo dị ứng, anh ta hoặc cô ta sẽ cung cấp một lý do tại sao điều này đã bị bỏ qua.*

Để kiểm tra xem các yêu cầu này đã được thỏa mãn chưa, bạn cần phát triển một số kiểm thử liên quan:

* + - * 1. Thiết lập hồ sơ bệnh nhân không có dị ứng được biết đến. Kê đơn thuốc cho dị ứng được biết là tồn tại. Kiểm tra xem thông điệp cảnh báo không được hệ thống đưa ra.
        2. Thiết lập hồ sơ bệnh nhân bị dị ứng đã biết. Kê toa thuốc mà bệnh nhân bị dị ứng và kiểm tra xem cảnh báo có được đưa ra bởi hệ thống không?
        3. Thiết lập hồ sơ bệnh nhân trong đó ghi nhận dị ứng với hai loại thuốc trở lên. Kê đơn cả hai loại thuốc này một cách riêng biệt và kiểm tra xem cảnh báo chính xác cho từng loại thuốc được đưa ra.
        4. Kê toa hai loại thuốc mà bệnh nhân bị dị ứng. Kiểm tra xem hai cảnh báo được đưa ra chính xác.
        5. Kê đơn thuốc đưa ra cảnh báo và ghi đè cảnh báo đó. Kiểm tra xem hệ thống yêu cầu người dùng cung cấp thông tin giải thích lý do cảnh báo bị ghi đè.

Bạn có thể thấy trong danh sách này rằng kiểm thử một yêu cầu không có nghĩa là chỉ viết một kiểm thử. Bạn thường phải viết một số kiểm thử để đảm bảo rằng bạn có phạm vi yêu cầu. Bạn cũng nên lưu giữ hồ sơ truy xuất nguồn gốc của kiểm dựa trên yêu cầu của mình, liên kết các kiểm thử với các yêu cầu cụ thể mà bạn đã kiểm thử.

* + 1. **Kiểm thử kịch bản**

Kiểm thử kich bản là một cách tiếp cận để kiểm thử phát hành, theo đó bạn nghĩ ra các kịch bản sử dụng điển hình và sử dụng các kịch bản này để phát triển các trường hợp kiểm thử cho hệ thống. Kịch bản là một câu chuyện mô tả một cách mà hệ thống có thể được sử dụng. Các kịch bản phải thực tế và người dùng hệ thống thực sẽ có thể liên quan đến chúng. Nếu bạn đã sử dụng các kịch bản hoặc câu chuyện người dùng như một phần của quy trình kỹ thuật yêu cầu (được mô tả trong Chương 4), thì bạn có thể sử dụng lại chúng làm kịch bản kiểm thử.

Trong một bài viết ngắn về kiểm tra kịch bản, Kaner (Kaner 2003) đề xuất rằng kiểm tra kịch bản nên là một câu chuyện kể đáng tin cậy và khá phức tạp. Nó nên tạo ra các bên liên quan; nghĩa là, họ nên liên quan đến kịch bản và tin rằng điều quan trọng là hệ thống vượt qua bài kiểm tra. Ông cũng gợi ý rằng nó nên dễ dàng để đánh giá. Nếu có vấn đề với hệ thống, thì nhóm thử nghiệm phát hành sẽ nhận ra chúng.

Như một ví dụ về kịch bản có thể xảy ra từ hệ thống Mentcare, Kịch bản này kiểm tra một số tính năng của hệ thống Mentcare:

1. Xác thực bằng cách đăng nhập vào hệ thống.
2. Tải xuống và tải lên hồ sơ bệnh nhân được chỉ định vào máy tính xách tay.
3. Lên lịch khám tại nhà.
4. Mã hóa và giải mã hồ sơ bệnh nhân trên thiết bị di động.
5. Ghi lại và sửa đổi.
6. Liên kết với cơ sở dữ liệu thuốc duy trì thông tin tác dụng phụ.
7. Hệ thống nhắc nhở cuộc gọi.

Nếu bạn là người kiểm thử phát hành, bạn chạy qua kịch bản này, đóng vai trò của George và quan sát cách hệ thống hành xử để đáp ứng với các đầu vào khác nhau. Là George, bạn có thể mắc lỗi cố ý, chẳng hạn như nhập sai cụm từ khóa để giải mã các bản ghi. Điều này kiểm tra phản ứng của hệ thống với lỗi. Bạn nên lưu ý với bất cứ vấn đề phát sinh nào, bao gồm các vấn đề về hiệu suất. Nếu một hệ thống quá chậm, điều này sẽ thay đổi cách sử dụng nó. Ví dụ: nếu mất quá nhiều thời gian để mã hóa bản ghi, thì người dùng không đủ thời gian có thể bỏ qua giai đoạn này. Nếu sau đó họ bị mất máy tính xách tay, một người không được ủy quyền có thể xem hồ sơ bệnh nhân.

Khi bạn sử dụng cách tiếp cận dựa trên kịch bản, thông thường bạn sẽ thử nghiệm một số yêu cầu trong cùng một kịch bản. Do đó, bạn vừa kiểm tra các yêu cầu riêng lẻ, vừa đang kiểm tra xem các kết hợp yêu cầu không gây ra vấn đề.

* + 1. **Kiểm thử hiệu suất**

Khi một hệ thống đã được tích hợp hoàn toàn, có thể kiểm tra các thuộc tính rõ nét, chẳng hạn như hiệu suất và độ tin cậy. Các kiểm thử hiệu suất phải được thiết kế để đảm bảo hệ thống có thể xử lý dự định nạp vào của nó. Điều này thường liên quan đến việc chạy một loạt các kiểm thử trong đó bạn tăng tải cho đến khi hệ thống hoạt động trở nên không thể chấp nhận được.

Cũng như các loại kiểm thử khác, kiểm thử hiệu năng có liên quan cả với việc giải thích rằng hệ thống đáp ứng các yêu cầu của nó và phát hiện ra các vấn đề và khiếm khuyết trong hệ thống. Để kiểm tra xem các yêu cầu về hiệu suất có đạt được hay không, bạn có thể phải xây dựng một hồ sơ hoạt động. Một hồ sơ hoạt động (xem Chương 11) là một tập hợp các bài kiểm tra phản ánh hỗn hợp công việc thực tế sẽ được hệ thống xử lý. Do đó, nếu 90% giao dịch trong một hệ thống thuộc loại A, 5% loại B và phần còn lại của loại C, D và E, thì bạn phải thiết kế hồ sơ hoạt động sao cho phần lớn các kiểm thử là thuộc loại A. Nếu không, bạn sẽ không có được một bài kiểm thử chính xác về hiệu suất hoạt động của hệ thống.

Tất nhiên, cách tiếp cận này, không nhất thiết là cách tiếp cận tốt nhất để kiểm thử nhược điểm. Kinh nghiệm đã chỉ ra rằng một cách hiệu quả để khám phá các khuyết điểm là thiết kế các kiểm thử xung quanh các giới hạn của hệ thống. Trong kiểm thử hiệu năng, điều này có nghĩa là nhấn mạnh hệ thống bằng cách đưa ra các yêu cầu nằm ngoài giới hạn thiết kế của phần mềm. Điều này được gọi là thử nghiệm áp lực.

Giả sử bạn đang kiểm tra một hệ thống xử lý giao dịch được thiết kế để xử lý tối đa 300 giao dịch mỗi giây. Bạn bắt đầu bằng cách thử nghiệm hệ thống này với ít hơn 300 giao dịch mỗi giây. Sau đó, bạn tăng dần tải trên hệ thống vượt quá 300 giao dịch mỗi giây cho đến khi vượt quá tải thiết kế tối đa của hệ thống và hệ thống bị lỗi.

Thử nghiệm áp lực giúp bạn làm hai việc:

* + - 1. Kiểm tra hành vi lỗi của hệ thống. Hoàn cảnh có thể phát sinh thông qua sự kết hợp bất ngờ của các sự kiện trong đó tải được đặt trên hệ thống vượt quá tải dự kiến tối đa. Trong những trường hợp này, lỗi hệ thống không được gây ra hỏng dữ liệu hoặc mất dịch vụ người dùng bất ngờ. kiểm thử áp lực kiểm tra rằng quá tải hệ thống khiến hệ thống không thành công, thay vì sụp đổ dưới tải.
      2. Hiển thị các lỗi chỉ hiển thị khi hệ thống được tải đầy đủ. Mặc dù có thể lập luận rằng những khiếm khuyết này không có khả năng gây ra lỗi hệ thống khi sử dụng, nhưng có thể có những kết hợp bất thường của tình huống mà thử nghiệm áp lực lặp lại.

Thử nghiệm áp lực đặc biệt liên quan đến các hệ thống phân tán dựa trên một mạng lưới các bộ xử lý. Các hệ thống này thường thể hiện sự xuống cấp nghiêm trọng khi chúng quá tải. Mạng tràng trong dữ liệu phối hợp mà các quy trình khác nhau phải trao đổi. Các quy trình trở nên chậm hơn và chậm hơn khi chúng chờ dữ liệu cần thiết từ các quy trình khác. Thử nghiệm áp lực giúp bạn khám phá khi sự xuống cấp bắt đầu để bạn có thể thêm séc vào hệ thống để từ chối các giao dịch vượt quá điểm này.