

Gợi ý: *PHP* | *WordPress* | *Laravel* | *Javascript* | *NodeJS* | *FullStack*

Cách viết Test Cases

Đăng bởi: TuoiPham - Vào ngày: 01-01-2019 - View: 3654

Test Case là tập hợp các hành động được thực thi để xác minh một tính năng hoặc chức năng cụ thể của ứng dụng phần mềm. Bài viết này mô tả **cách thiết kế Test Cases** và tầm quan trọng của các **thành phần trong Test Cases**.

Nội dung chính

1. Giới thiệu về Test Case
2. Định dạng của Test Cases tiêu chuẩn.
3. Cách để viết Test Cases tốt nhất.
 1. Các Test Cases cần phải đơn giản và minh bạch:
 2. Tạo Test Cases với vai trò như người dùng cuối
 3. Tránh lặp lại Test Cases.
 4. Không phỏng đoán
 5. Đảm bảo bao phủ 100%
 6. Các Test Cases phải được xác định.
 7. Thực hiện các kỹ thuật kiểm thử
 8. Làm sạch môi trường kiểm thử
 9. Lặp lại và không thay đổi



10. Đánh giá ngang hàng.
4. Công cụ quản lý Test Cases
5. Tài nguyên

1. Giới thiệu về Test Case

Hãy xem kịch bản Kiểm thử chức năng đăng nhập, có nhiều trường hợp có thể xảy ra như sau:

Test Cases 1: Kiểm thử kết quả khi nhập User Id & Password hợp lệ.

Test Cases 2: Kiểm thử kết quả khi nhập User Id & Password không hợp lệ.

Test Cases 3: Kiểm thử phản hồi khi User Id trống, click nút đăng nhập và nhiều thông tin khác.

Các kịch bản kiểm thử thường không rõ ràng và bao gồm nhiều trường hợp. Để kiểm thử tất cả các trường hợp, chúng ta cần có Test Cases.

2. Định dạng của Test Cases tiêu chuẩn.

Dưới đây là định dạng chuẩn của một Test Cases đăng nhập.

Test Case ID	Kịch bản kiểm thử	Các bước kiểm thử	Dữ liệu Kiểm thử	Kết quả dự kiến	Kết quả thực tế	Pass/Fail
--------------	-------------------	-------------------	------------------	-----------------	-----------------	-----------



TU01	Kiểm thử đăng nhập của khách hàng với dữ liệu hợp lệ.	<p>Bước 1: Truy cập trang web http://demo.guru99.com.</p> <p>Bước 2: Nhập tên người dùng và mật khẩu</p> <p>Bước 3: Click vào Submit</p>	Tên người dùng: guru99 Mật khẩu: pass99	Người dùng đăng nhập vào website thành công	Như mong đợi	Pass
TU02	Kiểm thử đăng nhập của khách hàng với dữ liệu không hợp lệ	<p>Bước 1: Truy cập trang web http://demo.guru99.com</p> <p>Bước 2: Nhập tên người dùng và mật khẩu</p> <p>Bước 3: Click vào Submit</p>	Tên người dùng: guru99 Mật khẩu: glass99	Người dùng đăng nhập vào website thành công	Như mong đợi	Pass

Một Test Cases bao gồm các thông tin sau:

- Mô tả về những yêu cầu được kiểm thử.
- Giải thích về cách hệ thống sẽ được kiểm thử.
- Các thiết lập khi kiểm thử như: phiên bản ứng dụng đang kiểm thử, phần mềm, những tệp dữ liệu, hệ điều hành, phần cứng, truy cập bảo mật, ngày thực hiện, các điều kiện tiên đề và bất kỳ thông tin thiết



lập nào khác phù hợp với các yêu cầu đang được kiểm thử.

- Đầu vào và đầu ra hoặc hành động và kết quả mong đợi.
- Bất kỳ bằng chứng hoặc tệp đính kèm.
- Sử dụng ngôn ngữ hoạt động.
- Test Case không được quá 15 bước.
- Kịch bản kiểm thử tự động được chú thích đầu vào, mục đích và kết quả mong đợi.
- Thiết lập những thay đổi cho các kiểm thử cần thiết trước.

3. Cách để viết Test Cases tốt nhất.

1. Các Test Cases cần phải đơn giản và minh bạch:

Tạo các Test Cases đơn giản nhất có thể. Test Cases phải rõ ràng và ngắn gọn vì tác giả của Test Cases có thể không thực hiện chúng.

Sử dụng ngôn ngữ dễ hiểu như: đi đến trang chủ, nhập dữ liệu, click vào Submit... Điều này làm cho việc hiểu các bước kiểm thử dễ dàng và thực hiện kiểm thử nhanh hơn.

2. Tạo Test Cases với vai trò như người dùng cuối

Mục tiêu cuối cùng của bất kỳ dự án phần mềm nào là tạo ra phần mềm đáp ứng yêu cầu của khách hàng và dễ sử dụng cũng như dễ vận hành. Người kiểm thử phải tạo Test Cases dựa trên quan điểm người dùng cuối.

3. Tránh lặp lại Test Cases.

Không lặp lại các Test Cases. Nếu một Test Cases là cần thiết để thực hiện một số Test Cases khác, hãy nêu Test Case Id trong cột điều kiện tiên đề.



4. Không phỏng đoán

Không phỏng đoán các chức năng và tính năng của ứng dụng phần mềm trong khi chuẩn bị Test Cases.
Phải bám sát các Tài liệu kỹ thuật.

5. Đảm bảo bao phủ 100%

Hãy chắc chắn rằng bạn viết các Test Cases để kiểm thử tất cả các yêu cầu phần mềm được đề cập trong tài liệu đặc tả. Sử dụng Ma trận truy xuất nguồn gốc (Traceability Matrix) để đảm bảo không có chức năng hay điều kiện nào chưa được kiểm thử.

6. Các Test Cases phải được xác định.

Đặt tên các Test Cases sao cho chúng được xác định dễ dàng khi theo dõi lỗi hoặc xác định yêu cầu phần mềm ở giai đoạn sau.

7. Thực hiện các kỹ thuật kiểm thử

Không thể kiểm thử mọi điều kiện có thể có trong ứng dụng phần mềm. Kỹ thuật kiểm thử sau đây giúp bạn chọn những Test Cases với khả năng tìm thấy lỗi tối đa.

Phân tích giá trị biên (Boundary Value Analysis - BVA): Đây là kỹ thuật kiểm thử xác định các ranh giới cho phạm vi giá trị được chỉ định.

Phân vùng tương đương (Equivalence Partition - EP): Kỹ thuật này phân vùng phạm vi thành các phần giống nhau hoặc nhóm có xu hướng có cùng kết quả.

Kỹ thuật chuyển trạng thái (State Transition Technique): Phương pháp này được sử dụng khi hệ thống có xử lý thay đổi từ trạng thái này sang trạng thái khác sau hành động cụ thể.

Kỹ thuật đoán lỗi (Error Guessing Technique): Dự đoán lỗi có thể phát sinh trong khi kiểm thử. Đây không phải là kỹ thuật chính thức và kỹ thuật này tận dụng trải nghiệm của tester với ứng dụng.

8. Làm sạch môi trường kiểm thử



Test Cases bạn tạo phải đảm bảo môi trường kiểm thử đúng theo yêu cầu, phải phục hồi môi trường kiểm thử về trạng thái trước khi kiểm thử khi môi trường kiểm thử đã bị thay đổi và môi trường kiểm thử phải sử dụng được. Điều này đặc biệt đúng đối với kiểm thử cấu hình.

9. Lặp lại và không thay đổi

Test Cases sẽ tạo ra kết quả giống nhau mỗi lần bất kể ai kiểm thử nó

10. Đánh giá ngang hàng.

Sau khi tạo các Test Cases, hãy để các đồng nghiệp của bạn review Test Cases. Đồng nghiệp của bạn có thể phát hiện ra các lỗi trong thiết kế Test Cases của bạn mà bạn có thể dễ dàng bỏ qua.

4. Công cụ quản lý Test Cases

Các công cụ quản lý kiểm thử là các công cụ tự động hóa giúp quản lý và duy trì các Test Cases. Các tính năng chính của một công cụ quản lý Test Cases là:

Đề ghi lại các test cases: Với các công cụ, bạn có thể tiến hành tạo test cases bằng cách sử dụng các templates.

Thực hiện Test Case và Ghi lại kết quả: Test Case có thể được thực thi thông qua các công cụ và kết quả thu được có thể dễ dàng được ghi lại.

Tự động theo dõi lỗi: Các kiểm thử failed được tự động liên kết với trình theo dõi lỗi, lần lượt có thể được chỉ định cho các developer và có thể được theo dõi bằng thông báo email.

Truy xuất nguồn gốc: Yêu cầu, test cases, thực thi Test Cases đều được liên kết với nhau thông qua các công cụ và mỗi trường hợp có thể được liên kết với nhau để kiểm tra phạm vi kiểm thử.

Lưu trữ các Test Cases: Các test cases nên được sử dụng lại và cần được lưu trữ để không bị mất hoặc bị hỏng. Công cụ quản lý Test Cases cung cấp các tính năng như:



- Quy ước đặt tên và đánh số
- Phiên bản
- Lưu trữ dưới dạng chỉ đọc
- Kiểm soát truy cập
- Sao lưu ngoài trang web

Các công cụ quản lý kiểm thử phổ biến là: Quality Center và JIRA

5. Tài nguyên

Xin lưu ý rằng template được sử dụng sẽ thay đổi theo từng dự án.

Tải xuống template test case dạng Excel (.xls)

-----#####-----

Chuỗi bài viết được tham khảo hoặc dịch lại từ nhiều nguồn như: TutorialsPoint, Guru99.

Khóa học nên xem

- Luyện thi lấy chứng chỉ quốc tế ISTQB Foundation cho Tester

» BÀI KẾ SAU

BÀI KẾ TIẾP »

Nguồn: *freetuts.net*

Những việc làm hấp dẫn



AI Data Science (Python)

Base Enterprise Ha Noi Up to \$1,600



Python

AI

Data Science



Ruby On Rails Developer

B&K Software Co., Ltd 📍 *Ho Chi Minh, Ha Noi* 💰 *Up to \$2,000*

Ruby on Rails



Solution Architect (.NET)

Wize Solutions 📍 *Ha Noi* 💰 *Negotiable*

.NET

Architect

BÀI GẦN ĐÂY

» Giám sát và kiểm soát kiểm thử

» Tài liệu kiểm thử

» Cách tạo Test Plan

» Tổ chức nhóm kiểm thử

» Phân tích rủi ro dự án và giải pháp trong quản lý kiểm thử

» Quy trình quản lý kiểm thử

» Vai trò và Trách nhiệm của Test Manager

» Kiểm thử Use Case

» Kỹ thuật kiểm thử chuyển đổi trạng thái

» Kỹ thuật kiểm thử bảng quyết định

