# QUY TRÌNH VIẾT TEST CASE

Contents

[QUY TRÌNH VIẾT TEST CASE 1](#_heading=h.gjdgxs)

[1.](#_heading=h.30j0zll) Quy trình viết TCs. 2

[1.1.. Chuẩn bị để viết một testcase 2](#_heading=h.1fob9te)

[1.2. Viết TCs theo format 2](#_heading=h.3znysh7)

[1.3. Review TCs 2](#_heading=h.2et92p0)

[2. Cấu trúc của Testcase 2](#_heading=h.tyjcwt)

[3. Các bước xác định TCs 3](#_heading=h.3dy6vkm)

[4. Testcase hiệu quả cần có 3](#_heading=h.1t3h5sf)

# Quy trình viết TCs.

## 1.1. Chuẩn bị để viết một testcase

* Cân nhắc testcase đó đã có hay chưa.

Nếu bạn đã tìm được testcase cho module đó thì hãy cân nhắc việc cập nhật nó thay vì viết một cái mới.

* Xem xét các kịch bản khác nhau trước khi viết. Tập trung vào điều gì có thể xảy ra với sản phẩm khi được sử dụng bởi khách hàng. Hãy suy nghĩ về điều này một cách cẩn thận và thiết kế các bài test của bạn cho phù hợp.
* Kịch bản nên ngắn gọn.
* Cho bản thân bạn đủ thời gian viết.

## 1.2. Viết TCs theo format

Tùy mục đích của đợt test mà lựa chọn format khác nhau.

Ex: Regression test, Ad-hoc testing,…

## 1.3. Review TCs

* Self-review
* Peer- review
* Supervisor review

# Cấu trúc của Testcase

•**Test Case ID** : Giá trị cần để xác định số lượng trường hợp cần để kiểm thử.

•**Test Items** : Dựa theo chức năng của hệ thống có thể chia nhỏ các functions ra để tạo TCs rõ ràng hơn.

•**Pre-condition** : Điều kiện tiền đề nếu có

•**Test Data** : Những dữ liệu cần chuẩn bị để test

•**Test Steps** : Mô tả các bước thực hiện test

•**Expected results**: Kết quả mong đợi từ các bước thực hiện trên

•**Actual result**: Thông thường sẽ là pass, fail

•**Comments** : Cột này dùng để note lại thông tin liên quan khi thực hiện test case.

# 3. Các bước xác định TCs

**B1**: Xác định mục đích test: cần hiểu rõ đặc tả yêu cầu của khách hàng.

**B2**: Xác định chức năng testing: cần phải biết làm thế nào phần mềm được sử dụng bao gồm các hoạt động , tổ chức chức năng khác nhau.

Các bước thực hiện chỉ mô tả các bước thực hiện đứng từ phía người dùng cuối bao gồm nhập dữ liệu, nhấn button.

Việc kiểm tra dữ liệu trong DB so với hiện thị trên màn hình nằm ở kết quả mong muốn. Thường được dùng cho các trường hợp kiểm thử kiểm tra lưu, cập nhật, xóa DB SELECT \* FROM … WHERE…

Ví dụ: Tạo 1 email đăng ký thành công

+Test tạo 1 email đăng ký thành công. Đã đăng nhập thành công bằng email mới trên giao diện

+Vào DB check xem email đó có được lưu vào DB hay không? ( tuy nhiên nhiều nơi ko yêu cầu tester vào CSDL để check)

**B3**: Xác định các yêu cầu phi chức năng: yêu cầu phần cứng, hệ điều hành, các khía cạnh an ninh

**B4**: Xác định biểu mẫu cho TCs: bao gồm giao diện UI, chức năng, khả năng tương thích và hiệu suất…

**B5**: Xác định tính ảnh hưởng giữa các nguyên tắc mô-đun: TCs nên được thiết kế để có thể che phủ được sự ảnh hưởng của các mô-đun với nhau ở mức độ cao nhất.

**B6:** Xác định trường hợp kiểm tra

•*Normal case*: Các trường hợp kiểm thử thông thường

•*Abnormal case*: Các trường hợp kiểm thử bất bình thường

•*Boundary case*: Các trường hợp kiểm tra boundary.

# 4. Testcase hiệu quả cần có

•Chính xác, đầy đủ nghiệp vụ hệ thống

•Độc lập (có thể thực hiện mà không phụ thuộc vào các testcase khác, dễ dàng chia cho nhiều người cùng kiểm thử)

•Nội dung đơn giản, có mục đích rõ ràng và ai đọc cũng hiểu theo một cách duy nhất. (đầu vào, đầu ra, các bước thực hiện rõ ràng) Trình bày mạch lạc thống nhất cho toàn bộ tài liệu.

•Có khả năng tái sử dụng (có thể dễ dàng cập nhật và sửa đổi).