



**Nguyễn Thị Hồng Nhung** @nguyen.thi.hong.nhung

Follow

★ 887 👤 66 ✎ 36

Published May 17th, 2016 2:24 AM - 11 min read

👁 26.3K 💬 0 🔖 12

# Hướng dẫn tạo Test Case (cơ bản)

QA

test

Tester

Testcase

TCs

...

## 1. Khái niệm Test Cases (TCs) là gì?

- Test Cases là 1 tập hợp các trường hợp điều kiện theo đó mà Tester có thể dựa vào nó để xác định liệu 1 ứng dụng, hệ thống phần mềm hoặc là 1 trong các tính năng của nó có hoạt động như mong muốn cần làm hay không?
- Các cơ chế để xác định liệu một chương trình phần mềm hoặc hệ thống đã được thông qua hay không, một trường hợp kiểm thử (1 case) được biết đến như một định hướng kiểm thử (nghĩa là bạn cần phải xác định được trường hợp các trường hợp có thể xảy ra).

## 2. Các bước xác định TCs

### B1: Xác định mục đích test.

Trước tiên, bạn cần hiểu rõ đặc tả yêu cầu của khách hàng. Khi bắt đầu viết TCs cho các tính năng của 1 hệ thống phần mềm, việc đầu tiên cần xác định đó là cần hiểu và xác định được yêu cầu của hệ thống.

### B2: Xác định hiệu suất testing. (Dựa trên sự hiểu biết về hệ thống phần mềm)

Để viết kịch bản thử nghiệm tốt, bạn nên được cần với quen thuộc với các yêu cầu chức năng. Bạn cần phải biết làm thế nào phần mềm được sử dụng bao gồm các hoạt động, tổ chức chức năng khác nhau.

### B3: Xác định các yêu cầu phi chức năng.

Bước thứ ba là để hiểu những khía cạnh khác của phần mềm liên quan đến các yêu cầu phi chức năng (non-function) như yêu cầu phần cứng, hệ điều hành, các khía cạnh an ninh phải được xem xét và điều kiện tiên quyết khác như các tập tin dữ liệu hoặc chuẩn bị dữ liệu thử nghiệm.

Thử nghiệm các yêu cầu phi chức năng là rất quan trọng. Ví dụ, nếu các phần mềm đòi hỏi người dùng phải điền vào các form, hợp lý thời gian cần được xác định bởi các nhà phát triển để đảm bảo rằng nó sẽ không time-out khi chờ submit form. Đồng thời, cũng cần check thời gian log-in hệ thống để đảm bảo rằng phiên làm việc đó của người dùng (session) không bị hết hạn, đây được gọi là trường hợp kiểm thử security.

Các mẫu , các trường hợp thử nghiệm nên bao gồm giao diện UI, chức năng, khả năng chịu lỗi, khả năng tương thích và hiệu suất của một số chức năng. Mỗi thể loại nên được xác định phù hợp với logic của ứng dụng phần mềm.

### B5: Xác định tính ảnh hưởng giữa các nguyên tắc mô-đun.

Trước tiên cần hiểu rõ chức năng thực hiện của 1 mô-đun và sự tương tác của mô-đun đó với các mô-đun khác để xác định được follow, sự khớp nối của hệ thống

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. Muốn biết được vấn đề đó, bạn cần xác định rõ được tính năng của từng mô-đun riêng biệt, cách thức nó hoạt động tương tác với mô-đun khác như thế nào? Ví dụ: Khi test chức năng giỏ hàng của 1 website thương mại điện tử để kiểm tra bạn cũng cần phải xem xét quản lý hàng tồn kho và xác nhận nếu cùng số lượng của sản phẩm mua được khấu trừ từ các cửa hàng. Tương tự như vậy, trong khi thử nghiệm trở lại, chúng ta cần phải kiểm tra ảnh hưởng của nó trên một phần tài chính của ứng dụng cùng với cửa hàng tồn kho.

## 3. Cấu trúc của TCs

**Format của 1 TCs thông thường bao gồm:**

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

**Chức năng (Function):** Dựa theo chức năng của hệ thống có thể chia nhỏ các functions ra để tạo TCs rõ ràng hơn. Ví dụ: Check form đăng nhập gmail được coi là 2 function lớn.

**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

**A result:** Thông thường sẽ là pass, fail, và pending. Đây là kết quả thực tế khi thực hiện test theo test case trên môi trường của hệ thống

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

**\*\*Ngoài ra, có thể thêm 1 số cột như: \*\*** Tester ( người thực hiện test), Execute Date ( ngày thực hiện test),...

## 4. Xác định trường hợp kiểm tra.

Với 1 giá trị cần kiểm tra luôn luôn có 3 trường hợp lớn cần kiểm tra có thể xảy ra.

- 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.

Từ các trường hợp lớn trên tùy theo từng giá trị cần kiểm tra sẽ xác định và chia ra thành các case nhỏ hơn tương ứng.

## 5. Ví dụ : Thực hiện viết TCs cho chức năng đăng nhập facebook trên máy tính.

B1: Xác định yêu cầu.

Yêu cầu là test form login của facebook theo đường link: <https://www.facebook.com/>



- Mục đích test: Kiểm tra việc đăng nhập thành công vào hệ thống facebook, không test chức năng đăng ký ở cùng trên form, chỉ test trên môi trường web ,ko test trên môi trường điện thoại, browser trên điện thoại.
- Xác định hiệu suất testing: Chức năng form login của facebook cũng thực hiện tương tự như hầu hết chức năng của các hệ thống khác. Form login bao gồm: 2 text box email/điện thoại và mật khẩu, 1 button đăng nhập, 1 link quên mật khẩu.
- Xác định non-function yêu cầu: Cần check tính bảo mật với những trường hợp email chưa đăng ký vào hệ thống trước đó, lưu mật khẩu vào trình duyệt, loại trình duyệt: firefox, chrome, safari, IE, ..., Ngoài ra, cần kiểm tra hệ thống mạng, phần cứng máy tính,
- Xác định biểu mẫu cho TCs: Yêu cầu sẽ bao gồm test các phần: UI, chức năng đăng nhập, tốc độ đăng nhập.
- Xác định tính ảnh hưởng giữa các nguyên tắc mô-đun: Có thể check account đăng nhập của người dùng có là 1 account thực hay không so với DB của hệ thống (giả sử ta có DB đó). Sau khi đăng nhập thành công sẽ chuyển hướng tới trang chủ của người dùng.

B2: Xây dựng TCs.

- Xác định các case UI: Bao gồm UI chung của cả form: màu sắc, font, size, color của label, chiều dài, rộng, cao, loại của các textbox, button, vị trí của form, textbox, button, link trên trang... Nếu mỗi một UI tách ra thành 1 case thì TCs sẽ quá dài vì vậy chúng ta có thể gộp lại thành 1 case test UI chung hoặc tách nhỏ ra theo phân nhóm UI.
- Xác định case test chức năng: Ở đây chức năng là đăng nhập gồm 2 text box email/điện thoại và mật khẩu, 1 button đăng nhập, 1 link quên mật khẩu. Cho nên sẽ có những case như sau

Đối với email/ điện thoại textbox:

- Normal case sẽ gồm: đăng nhập với đúng sdt, địa chỉ email đã đăng ký với hệ thống facebook trước đó và đăng nhập với blank, sai sdt, địa chỉ email đã đăng ký với hệ thống facebook trước đó.
- Abnormal case sẽ gồm: Đăng nhập với số điện thoại mà thêm mã vùng, mã nước vào trước đó (ví dụ: +849....) hoặc email mà không nhập tổ email cho nó ( email đúng là: nguyen.thi.hong.nhung). Ngoài ra, còn có các cases: ngắt mạng, mạng yếu, sử dụng 3G, wi-fi, mạng lan, ăn cắp cookie, session, đăng nhập trên nhiều trình duyệt ...

- Boundary case sẽ gồm: Text sẽ nhập là từ tối thiểu và tối đa mà hệ text cho nhập. Có thể tạo ra 1 email với

VIBLO

Search Viblo



Sign In/Sign up

của follow đang nhập mà chỉ kiểm tra khả năng cho nhập tối thiểu và tối đa của ô text.)

Đối với button đăng nhập:

- Normal case sẽ gồm: Nhập giá trị vào text, click button đăng nhập, bấm phím enter từ bàn phím.
- Abnormal case sẽ gồm: Click liên tục or enter liên tục vào button
- Boundary case sẽ gồm: không cần check trường hợp này
- Testcase mẫu:

1	Project Name:	Facebook Test										
2	Screen:	Login										
3	Creator:	NhưngNTH										
4	Created date:	5/20/2016										
5												
6												
7	TC ID	Big Object	Medium Object	Small Object	Procedure / Step	Test Data	Expected Result	Testing Result	Remarks	System Test		
8										Phase	Defect/Bug ID	Comments
9	1	UI			Check màu sắc, font, size, color của label, chiều dài, rộng, cao, loại của các textbox, button, vị trí của form, textbox, button, link trên form...	Link tới server design: https://www.facebook.com/	Hiện thị giống với design	Pass				
10	2											
11												
12												
13												
14												
15												
16												
17												
18												



Related

Các loại test case trong kiểm thử phần mềm

[Nguyen Thi Thu Ha](#)

14 min read

6106

2

1

0

Positive Vs Negative testing.(P1)

[Nguyen Thi Phuong](#)

2 min read

654

1

0

0

Sự khác nhau giữa Test Case và Test Scenario

[Nguyễn Thu Mai](#)

4 min read

9766

1

1

5

Kiểm thử phần mềm các cấp độ (.phần 2)

[N guyen Minh Ngoc](#)

7 min read

1011

1

0

0

More from Nguyễn Thị Hồng Nhung

30 phút mỗi ngày với Jmeter Tool - Khái niệm performance, load test, test hiệu năng và các kiến thức...

[Nguyễn Thị Hồng Nhung](#)

0 min read

444

1

0

0

Sử dụng Test Techniques để khám phá bài toán : Một QA đi vào một quán bar

[Nguyễn Thị Hồng Nhung](#)

9 min read

428

0

0

0

Một số phương pháp đơn giản để giữ thái độ tích cực trong công việc của 1 QA

[Nguyễn Thị Hồng Nhung](#)

23 min read

703

0

3

0

Quy trình phát triển phần mềm ( SDLC ).

[Nguyễn Thị Hồng Nhung](#)

11 min read

3514

1

0



0

Login to comment

RESOURCES

- [Posts](#)
- [Questions](#)
- [Videos](#)
- [Discussions](#)
- [Tools](#)
- [System Status](#)
- [Organizations](#)
- [Tags](#)
- [Authors](#)
- [Recommend System](#)
- [Machine Learning](#)

SERVICES

-  [Viblo Code](#)
-  [Viblo CV](#)

MOBILE APP



GET IT ON  
Google Play



Download on the  
App Store

LINKS

