Một số kỹ thuật thiết kế test case

April 1, 2015 *Nguyen Thi Hang*

Like 6 Tweet Save Share 41

Các bạn có bao giờ tự hỏi khi các lập trình viên làm ra một phần mềm, một ứng dụng nào đó thì ai sẽ là người kiểm tra những sản phẩm này hay tại sao phải kiểm tra những sản phẩm này?

Theo suy nghĩ của nhiều người để tìm lỗi thì quá trình kiểm tra sẽ diễn ra trong giai đoạn phát triển và tổng quan lại mã code. Vì thế hoạt động kiểm tra từ các đối tượng khác có thể không cần thiết và không quan trọng nữa.

Albert Einstein đã từng nói “Một người thông minh giải quyết vấn đề, người khôn ngoan thì tránh nó”.

YOURSKILL test
Kiểm tra kỹ năng IT của bạn

Categories

Chắc chắn ai cũng muốn mình là người thông minh. Và các nhà lãnh đạo cũng không nằm ngoài phạm vi đó. Họ lựa chọn những đối tượng độc lập với phần mềm để thực hiện kiểm tra chúng. Đó chính là Tester.

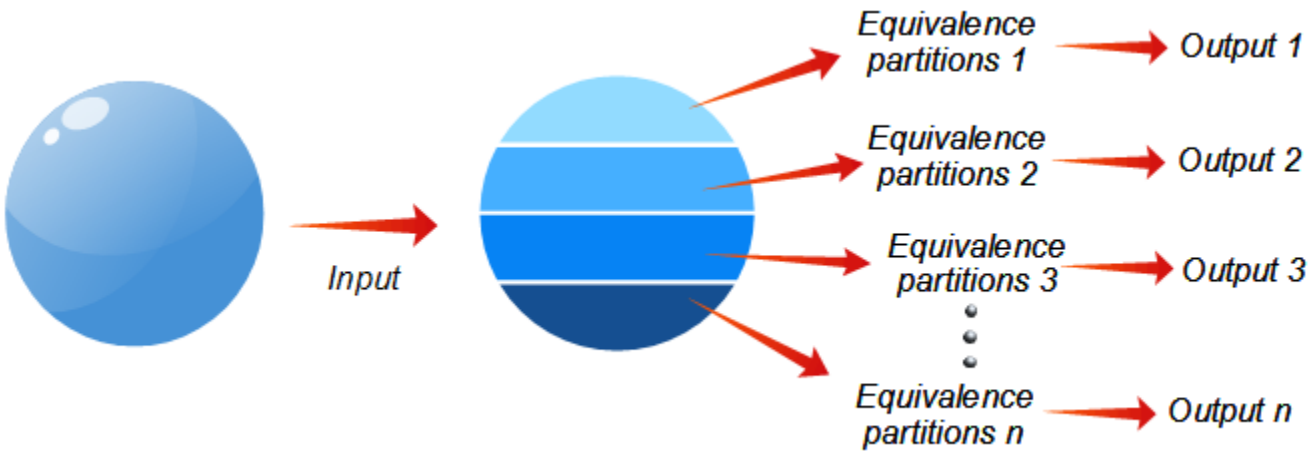
Thế nhưng người lãnh đạo luôn mong muốn Tester sẽ thực hiện việc kiểm tra phần mềm với thời gian ngắn nhất mà vẫn đạt chất lượng cao nhất. Để làm được điều đó Tester không chỉ phải hiểu sâu về nghiệp vụ của phần mềm mà còn cần linh hoạt trong việc thiết kế test case.

Hôm nay, tôi sẽ giới thiệu một số kỹ thuật thiết kế test case khá phổ biến trong phương pháp kiểm thử hộp đen. Nếu biết cách vận dụng những kỹ thuật này một cách linh hoạt thì sẽ giảm thiểu rất nhiều số lượng case thừa, tiết kiệm thời gian kiểm thử mà vẫn đảm bảo chất lượng phần mềm:

1. Kỹ thuật phân vùng tương đương (Equivalence Partitioning)

1.1. Ý tưởng:

Phân vùng tương đương là phương pháp chia các điều kiện đầu vào thành những vùng tương đương nhau. Tất cả các giá trị trong một vùng tương đương sẽ cho một kết quả đầu ra giống nhau. Vì vậy chúng ta có thể test một giá trị đại diện trong vùng tương đương.



Hình 1. Kỹ thuật phân vùng tương đương

Thiết kế test case bằng kỹ thuật phân vùng tương đương tiến hành theo 2 bước:

(1) Xác định các lớp tương đương

(2) Xác định các ca kiểm thử

1.2. Ví dụ:

(*) Form login bao gồm:

User: Text-box

PassWord: Text-box

Yêu cầu:

Thiết kế test case sao cho người dùng nhập vào ô text-box user chỉ cho nhập ký tự chữ với độ dài trong khoảng [6-20]

Nếu nhập giá trị với số ký tự không nằm trong khoảng [6-20] => hiển thị lỗi “Bạn chỉ được phép nhập chuỗi từ 6 => 20 ký tự”

Nếu để trống ô hoặc nhập ký tự khác ký tự chữ => hiển thị lỗi “Tên người dùng chưa hợp lệ! Vui lòng nhập ký tự chữ”

(*) Dựa vào yêu cầu bài toán ta có thể có các lớp tương đương(phân vùng) sau:

+ Phân vùng 1: Nhập giá trị hợp lệ từ 6 => 20

+ Phân vùng 2: Nhập giá trị không hợp lệ < 6 ký tự

+ Phân vùng 3: Nhập giá trị không hợp lệ > 20 ký tự

+ Phân vùng 4: Trường hợp để trống không nhập gì hay nhập ký tự không phải dạng chữ

Sau khi áp dụng phân vùng tương đương có thể chọn được các ca kiểm thử (test case) sau:

+ Case 1: Nhập giá trị từ 6 => 20 => pass

+ Case 2: Nhập giá trị < 6 ký tự (có thể chọn nhập 1, 2, 3, 4 hoặc 5 ký tự) => hiển thị lỗi "Bạn chỉ được phép nhập chuỗi từ 6 => 20 ký tự"

+ Case 3: Nhập giá trị > 20 ký tự (có thể chọn nhập 21, 22, 23,… ký tự) => hiển thị lỗi "Bạn chỉ được phép nhập chuỗi từ 6 => 20 ký tự"

+ Case 4: Để trống không nhập gì hay nhập ký tự không phải dạng chữ => hiển thị lỗi "Tên người dùng chưa hợp lệ! Vui lòng nhập ký tự chữ"

1.3. Ưu/ nhược điểm:

(*) Ưu điểm:

Vì mỗi vùng tương đương ta chỉ cần test trên các phần tử đại diện nên số lượng test case được giảm đi khá nhiều nhờ đó mà thời gian thực hiện test cũng giảm đáng kể.

(*) Nhược điểm:

Không phải với bất kỳ bài toán nào đều có thể áp dụng kỹ thuật này. Có thể bị lack lỗi ở biên nếu chỉ chọn giá trị ở khoảng giữa của miền tương đương.

Scrum
Security
Security Testing
SQA
TechNote
Tips
vue js

Popular Posts

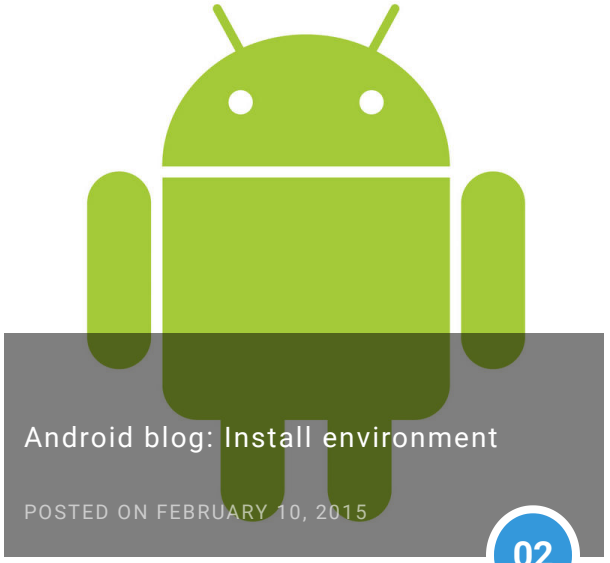
Vì vậy việc kết hợp linh hoạt giữa kỹ thuật phân vùng tương đương và phân tích giá trị biên dưới đây sẽ mang lại hiệu quả cao hơn để vừa tối ưu số lượng test case và vẫn đảm bảo đươc chất lượng phần mềm.

2. Kỹ thuật phân tích giá trị biên (Boundary-value Analysis)

2.1. Ý tưởng:

Hầu hết các lỗi được tìm thấy khi kiểm tra ở các giá trị biên. Vì vậy phương pháp này tập trung vào việc kiểm thử các giá trị biên này.

Phân tích giá trị biên là trường hợp đặc biệt của phân vùng tương đương, dựa trên những phân vùng tương đương tester sẽ xác định giá trị biên giữa những phân vùng này và lựa chọn test case phù hợp.



Các case chuẩn được lựa chọn dựa vào quy tắc sau:

Giá trị biên nhỏ nhất – 1

Giá trị biên nhỏ nhất

Giá trị biên lớn nhất

Giá trị biên lớn nhất + 1

Nhưng nếu bạn muốn kiểm tra sâu hơn thì bạn cũng có thể lựa chọn theo quy tắc:

Giá trị biên nhỏ nhất – 1

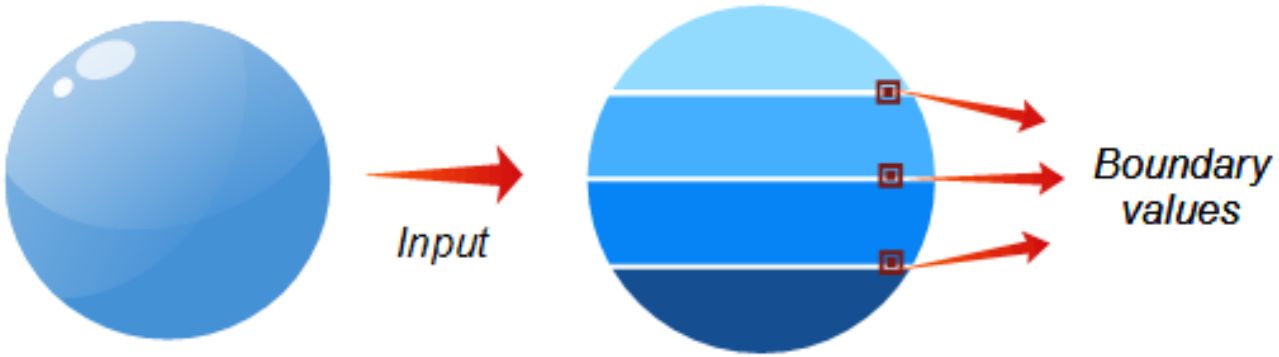
Giá trị biên nhỏ nhất

Giá trị biên nhỏ nhất + 1

Giá trị biên lớn nhất – 1

Giá trị biên lớn nhất

Giá trị biên lớn nhất + 1



Hình 2. Kỹ thuật phân tích giá trị biên

2.2. Ví dụ:

(*) Form login bao gồm:

User: Text-box

PassWord: Text-box

Yêu cầu:

Thiết kế test case sao cho người dùng nhập vào ô text-box user chỉ cho nhập ký tự chữ với độ dài trong khoảng [6-20]

Archives

Nếu nhập giá trị với số ký tự không nằm trong khoảng [6-20] => hiển thị lỗi “Bạn chỉ được phép nhập chuỗi từ 6 => 20 ký tự”

Nếu để trống ô hoặc nhập ký tự khác ký tự chữ => hiển thị lỗi “Tên người dùng chưa hợp lệ! Vui lòng nhập ký tự chữ”

(*)Theo phương pháp phân vùng tương đương ở trên ta xây dựng được các miền tương đương:

+ Phân vùng 1: Nhập giá trị hợp lệ từ 6 => 20

+ Phân vùng 2: Nhập giá trị không hợp lệ < 6 ký tự

+ Phân vùng 3: Nhập giá trị không hợp lệ > 20 ký tự

+ Phân vùng 4: Trường hợp để trống không nhập gì hay nhập ký tự không phải dạng chữ

Áp dụng kỹ thuật phân tích giá trị biên ta chọn được các case sau:
+ Case 1: Nhập giá trị với 5 ký tự => hiển thị lỗi “Bạn chỉ được phép nhập chuỗi từ 6 => 20 ký tự”

+ Case 2: Nhập giá trị với 6 ký tự => pass

+ Case 3: Nhập giá trị với 20 ký tự => pass

+ Case 4: Nhập giá trị với 21 ký tự => hiển thị lỗi “Bạn chỉ được phép nhập chuỗi từ 6 => 20 ký tự”

+ Case 5: Để trống không nhập gì hay nhập ký tự không phải dạng chữ => hiển thị lỗi “Tên người dùng chưa hợp lệ! Vui lòng nhập ký tự chữ

2.3. Ưu/ nhược điểm:

► 2017 (63)

► 2016 (80)

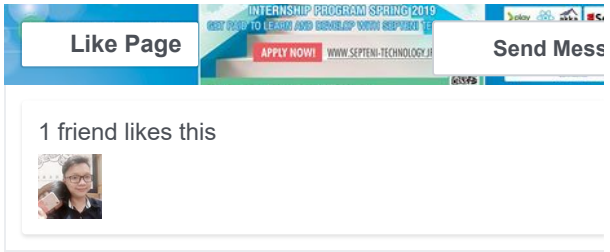
► 2015 (61)

► 2014 (54)

► 2013 (52)

FIND US ON FACEBOOK

Septeni Technology CO., L...
1,396 likes



1 friend likes this

(*) Ưu điểm:

Thay vì phải test hết toàn bộ các giá trị trong từng vùng tương đương, kỹ thuật phân tích giá trị biên tập trung vào việc kiểm thử các giá trị biên của miền giá trị inputs để thiết kế test case do “lỗi thường tiềm ẩn tại các ngõ ngách và tập hợp tại biên”.

Tiết kiệm thời gian thiết kế test case và thực hiện test.

(*) Nhược điểm:

Phương pháp này chỉ hiệu quả trong trường hợp các đối số đầu vào (input variables) độc lập với nhau và mỗi đối số đều có một miền giá trị hữu hạn.

3. Đoán lỗi (Error Guessing)

3.1. Ý tưởng:

Phương pháp này không có quy trình cụ thể vì có tính trực giác cao và không thể dự đoán trước.

Phương pháp chỉ phù hợp với những Tester có kinh nghiệm. Họ được đưa cho 1 chương trình, họ phỏng đoán dựa vào trực giác, dựa vào kinh nghiệm, dữ liệu lịch sử về các lỗi đã từng xảy ra với chương trình trước đó… và sau đó viết các ca kiểm thử để đưa ra các lỗi đó.

Hình 3: Kỹ thuật đoán lỗi

3.2. Ví dụ:

Ở màn hình login, đôi khi developer code thì gán user name là “admin” và pass là rỗng hoặc “123” vì vậy khi thực hiện test mình nên test cả case này

3.3. Ưu/ nhược điểm:

(*) Ưu điểm:

Sử dụng phương pháp này có thể giúp tester tìm ra những lỗi điển hình thường xảy ra trong phần mềm hoặc những lỗi không thể tìm thấy khi thiết kế test case theo hình thức formal

(*) Nhược điểm:

Kỹ thuật này thường được thực hiện bởi các Tester có kinh nghiệm và không theo một quy tắc nhất định, thiết kế test case dựa nhiều vào cảm tính

Tổng kết:

Ngoài 3 kỹ thuật thiết kế test case đã nói ở trên, bạn cũng có thể tìm hiểu và sử dụng rất nhiều các kỹ thuật khác nữa như: thiết kế test case dựa trên đồ thị nguyên nhân – kết quả(Cause-Effect Diagram), dựa trên bảng quyết định(Decision Tables)…

Để giảm thiểu số case đến mức tối ưu mà vẫn đảm bảo chất lượng phần mềm, mỗi tester cần linh hoạt trong việc lựa chọn các kỹ thuật thiết kế test case.

Chúc các bạn thành công trong việc vận dụng các kỹ thuật này để đạt được hiệu quả cao nhất!

Like 6 Tweet Save Share 41

RELATED POSTS

Selenium-WebDriver và Headless test tích hợp trong Cucumer

Giới thiệu: Selenium – nói 1 cách đơn giản, là...

Cài đặt ruby on rails trên windows

Xin chào bạn đang đọc bài viết ngắn này. Hôm...

Quality management system or the way to create higher quality products

Problem As a engineer, I see that the common problem that...

How to write good Test Cases ?

Every tester writes test cases however many times the test...

© 2019 SEPTENI TECHNOLOGY Developer's Blog

Back to Top ↑