Nguyen Thi Trang @trangnguyen Follow

506 28 32

Published Sep 24th, 2016 9:01 AM - 12 min read

Các kỹ thuật kiểm thử hộp đen (Phần 1)

Testing QA

Mở đầu

Nếu bạn là một QA/Tester thì hẳn bạn chẳng xa lạ gì với các khái niệm **Kiểm thử hộp đen**, **Kiểm thử hộp trắng** cũng như các phương pháp cụ thể để áp dụng nó. Tuy nhiên bạn có đang áp dụng một cách tốt nhất các phương pháp này vào trong công việc hằng ngày hay không lại là một câu chuyện khác. Nội dung bài viết này được dịch từ **Chương 4. Test design techniques** trong cuốn **Foundations Of Software testing (ISTQB certification)** của các tác giả Rex Black, Erik Van Veenendaal, Dorothy Graham hi vọng sẽ giúp mọi người hiểu thêm về các phương pháp này cũng như áp dụng một cách tốt hơn trong công việc của mình.

Bài viết sẽ đề cập đến 4 phương pháp phổ biến nhất trong kiểm thử hộp đen bao gồm:

- **Phân vùng tương đương (Equivalence partition)**
- **Phân tích giá trị biên (Boundary value analysis)**
- **Bảng quyết định (Decision table)**
- **Chuyển đổi trạng thái (State transition testing)**

Ta sẽ đi lần lượt từng phương pháp một!

1.Phân vùng tương đương (Equivalence partition)

Đây là một kĩ thuật rất hiệu quả trong kiểm thử hộp đen, nó có thể áp dụng ở tất cả các test level. Phân vùng tương đương thường được áp dụng đầu tiên trong hoạt động thiết kế test case. Phương pháp này được nhận xét là “common sense approach” bởi có nhiều tester vẫn sử dụng phương pháp này trong công việc nhưng theo một cách informally đồng thời cũng không biết được mình đang dùng phương pháp này. Ý tưởng phía sau phương pháp Phân vùng tương đương là phân vùng tập các điều kiện test vào các nhóm mà các phần tử trong nhóm đó có thể xem là như nhau khi dùng để kiểm thử

Sử dụng phân vùng tương đương ta chỉ cần kiểm thử 1 điều kiện nằm trong mỗi vùng vì các điều kiện còn lại trong vùng đó cũng sẽ cho ta kết quả như nhau khi thực hiện. Nếu một điều kiện trong vùng cho output đúng, ta cũng có thể giả định rằng tất cả các điều kiện trong vùng đó cũng cho output đúng. Do đó áp dụng phân vùng tương đương sẽ có ít case cần test hơn. Ngược lại, nếu 1 điều kiện trong vùng không cho output đúng ta sẽ giả định rằng tất cả các điều kiện trong vùng cũng không cho output đúng. Giả định trên rất đơn giản do đó trong nhiều trường hợp nó có thể không đúng, do đó nếu có thời gian bạn hãy thử nhiều hơn 1 giá trị trong vùng, nhất là với dạng User Inputs

Ví du

Tỉ lệ lãi suất phụ thuộc vào số dư tài khoản (balance) trong ngân hàng. Phần mềm có chức năng tính tính toán và đưa ra tiền lãi thích hợp:

$0 - $100 : 3%

$100. 01 - $ 999.99: 5%

Lớn hơn hoặc bằng $ 1000 : 7%

Nhìn vào requirement trên chủ quan ta có thể thấy có 3 phân vùng được đề cập. Tuy nhiên áp áp dụng phân vùng tương đương ta sẽ có 4 phân vùng như sau:

Invalid partition	Valid (for 3% interest)		Valid (for 5%)		Valid (for 7%)
-$0.01	$0.00	$100.00	$100.01	$999.99	$1000.00

Việc nảy sinh thêm 1 phân vùng như trên đã cho ta thấy một điều rằng tester không chỉ test những gì có trong đặc tả yêu cầu, tester còn phải nghĩ về những điều kiện không được nhắc đến trong đặc tả yêu cầu. Trong bài toán trên, ta đã nghĩ thêm được vùng có balance < $0.00 và đó là vùng invalid. Ba vùng còn lại là 3 vùng valid. Tuy nhiên vùng cuối cùng vẫn chưa được xác định rõ. Cụ thể ta vẫn không biết được cận phải của vùng này và việc xác định cận phải này không hề đơn giản. Ngoài ra cũng cần lưu ý rằng partition có các input không phải là số (non- numeric) cũng là 1 vùng invalid của bài toán này, tuy nhiên phần này ta chỉ đề cập tới input là số (numberic partition).

Chúng ta đã giả định khoảng cách giữa hai giá trị liền kề là 2 chữ số thập phân. Ta cũng có thể giả định khoảng cách đó là 0 hoặc một chữ số thập phân. Trong nhiều trường hợp, bạn nên đưa ra ý kiến của mình về việc xác định khoảng cách này và mọi người sẽ review ý kiến đó

Cần chú ý rằng, ta cũng có thẻ áp dụng phân vùng tương đương cho các giá trị output và trong trường hợp này là 3%, 5%, 7% và error message. Trong ví dụ này, phân vùng tương đương cho output và input là như nhau.

Vậy nếu không áp dụng kĩ thuật này, tester sẽ test như thế nào? Tester có thể sẽ đưa ra các điều kiện đầu vào cách nhau một giá trị nào đó ví dụ 50$: $50.00, $100.00, $150.00, $200.00, $250.00, ... say up to $800.00 . Tuy nhiên số lượng case trên chỉ cover được 2 partition và người tester cũng không thể tìm được thêm lỗi, hơn nữa lại phải thực hiện nhiều case hơn so với việc áp dụng phân vùng tương đương. Như vậy áp dụng phân vùng tương đương giúp việc test của ta hiệu quả và năng suất hơn rất nhiều.

Chú ý rằng vùng gọi là INVALID không phải là vùng chứa các giá trị mà người dùng không bao giờ nhập vào hay không có ý định nhập vào. Gọi là vùng invalid là vì nó chứa các giá trị mà ta không mong muốn. Khi kiểm thử bạn phải chắc chắn rằng phần mềm đã handle đầy đủ và chính xác các giá trị trong vùng invalid.

Cũng chú ý rằng vùng invalid mà ta đưa ra ở trong ví dụ trên mới chỉ ở trong bối cảnh test chức năng tính lãi xuất, ngoài ra còn nhiều vùng invalid khác nữa ví dụ : tài khoản bị rút vượt quá số dư...

2. Phân tích giá trị biên (Boundary value anlalysis)

Phân tính giá trị biên dựa trên giá trị biên giữa các phân vùng. Cần phải chú ý rằng ta có cả giá trị bên cho cả 2 vùng valid và invalid

Invalid		Valid		Invalid
0	1	99		100

để có được các giá trị biên, ta lấy giá trị lớn nhất và nhỏ nhất của mỗi partition. Với ví dụ interest payment trên, ta có các giá trị biên như sau:

Invalid boundary value: -$0.01

Valid boundary value: $0.00, $100.00, $100.01, $999.99 and $1000.00

Như vậy ta có 6 case cho giá trị biên và như vậy, sau khi áp dụng cả 2 phương pháp phân vùng tương đương và phân tích giá trị biên ta đã có tổng cộng 10 testcase.

Ta có thể thấy rằng không có giá trị lớn nhất được định nghĩa trong vùng 7% và như vậy ta cũng không thể xác định được biên bên phải của khoảng này. Trường hợp này người ta gọi là **Open Boundary** vì chỉ có một phía của phân vùng không được định nghĩa. Nhưng không phải vì thế mà ta bỏ qua việc test trường hợp này. Ta vẫn phải test nhưng như thế nào?

Open boundary rất khó để test nhưng có các phương pháp để thực hiện việc test chúng. Giải pháp tốt nhất cho trường hợp này là tìm ra một giá trị biên cụ thể. Một cách tiếp cần nữa là quay lại specification để tìm ra giá trị biên này có được để cập đến ở chỗ nào không. Một cách tiếp cận khác nữa là ta đi khám phá các vùng liên quan trong hệ thống. Ví dụ trường cần test chỉ thực hiện với số dư tài khoản với 6 chữ số phần đơn vị và 2 chữ số phần thập phân. Như vậy ta có thể xác định được maximum account lúc này là $999999.99 và có thể coi đây như là boundary của phân vùng trên. Và cuối cùng nếu bạn không tìm ra được nên cho giá trị biên là gì, có thể sử dụng trực giác hoặc kinh nghiệm của mình dùng các giá trị lớn để dò và tìm lỗi.

Trong quá trình sử dụng Phân vùng tương đương và Phân tích giá trị biên, bạn nên trình bày ra bảng để quát sát tốt hơn!

Mở rộng cho Phân vùng tương đương và Phân tích giá trị biên

Nên test với một vài giá trị trong 1 phân vùng, và nếu kết quả trả về không không tương tự nhau, rất có vùng bạn xác định còn bao gồm 2 hoặc nhiều vùng khác nữa.

Phân vùng tương đương và phân tích giá trị biên có thể áp dụng ở tất cả các Test Level. Đôi khi ta còn phải dùng Phân vùng tương đương và Phân tích giá trị biên nhiều lần trong bài toán.

Phân tích giá trị biên có thể ứng dụng ở tất cả các đầu vào khác nhau. Ví dụ như đầu vào của 1 module nào đó truyền vào hệ thống, giá trị tham số giữa các interface (valid và invalid). Loại lỗi này rất khó để phát hiện khi có 1 interface được thêm vào hệ thống, chính vì thế ta cần phải áp dụng phân vùng tương đương khi với Integration Test (Bao gồm cả Component Intergration và System Intergration)

Phân tích giá trị biên có thể ứng dụng ở cả string input chứ không đơn thuần là các giá trị số. Ta dựa trên số lượng character trong string để phân vùng. Ví dụ: số lượng character nằm trong khoảng [1, 30] là một partition với 2 valid boundary là 1 và 30. Invalid boundary có thể là 0 character, null, return key và 31 character. Cả 2 trường hợp này khi thực thi kết quả trả về cần phải có message báo lỗi.

Partition cần được xác định trong khi thiết lập Data. Nếu có nhiều bản ghi khác nhau với các loại dữ liệu khác nhau, bạn có thể có thêm nhiều đối tượng test. Kích thước của bản ghi cũng là một đối tượng và ta có thể phân vùng, xác định giá trị biên cho nó. Ngoài ra nếu bạn nắm bắt và có hiểu biết về phần phía bên trong của data được tổ chức như thế nào, ta có thể xác định được các giá trị biên ẩn (hidden boundary). Nghe có vẻ như ta đang lấn sân sang Kiểm thử hộp trắng tuy nhiên nó không phải là vấn đề, miễn là việc tìm ra bug hiệu quả phải không 😃

ứng dụng các kỹ thuật kiểm thử trong quá trình tạo Testcase.

Thị Thoa Bùi

7 min read

4974 3 0 5

Sử dụng Decision table - Bảng quyết định trong kiểm thử phần mềm

Huongntt

12 min read

8636 3 0 0

Negative testing

Nguyen Thi Trang

11 min read

534 2 0 4

Tìm hiểu về kỹ thuật phân tích giá trị biên và phân vùng tương đương trong kiểm thử hộp đen

Tran Thi Huong Trang

15 min read

12901 1 1 3

More from Nguyen Thi Trang

APPIUM Tutorial cho kiểm thử Android & iOS Mobile App

Nguyen Thi Trang

11 min read

28 2 0 0

Project Risk Analysis & Solutions trong Test Management

Nguyen Thi Trang

10 min read

55 1 0 1

Quản lý issue trong dự án

Nguyen Thi Trang

10 min read

293 2 0 2

Tìm hiểu về IOT testing

Nguyen Thi Trang

7 min read

232 0 0 0

Comments

Login to comment