

Các nguyên tắc kiểm thử

I. Tại sao cần phải kiểm thử

Không có định nghĩa kiểm thử được dùng chung trên thế giới. Thuật ngữ “kiểm thử phần mềm” được định nghĩa trong BS 7925-1 là "quá trình kiểm tra phần mềm để xác minh rằng nó đáp ứng các yêu cầu quy định và phát hiện các sai sót", được phát triển bởi một Nhóm chuyên gia trong kiểm thử phần mềm (BCS SIGIST), được chấp thuận bởi tổ chức ISEB/ISTQB (International Software Testing Qualifications Board). Những thiếu sót của định nghĩa trong BS 7925-1 là nó có phần thiên vị và nhấn mạnh đối với kiểm thử thành phần. Nó cũng là nguồn gốc phần định nghĩa của BS 7952-2 và như vậy đã bước đầu quan tâm và nhấn mạnh việc kiểm thử thành phần và từ đó đã được mở rộng để bao gồm cả kiểm thử phần mềm nói chung.

A. Các thuật ngữ khác

1. Error: Lỗi do con người tạo ra.
2. Bug: Là một khiếm khuyết trong một thành phần hoặc hệ thống mà nó có thể làm cho thành phần hoặc hệ thống này không thực hiện đúng chức năng yêu cầu của nó, ví dụ như thông báo sai hoặc định nghĩa dữ liệu không đúng
3. Fault: Một biểu hiện của lỗi (error) trong phần mềm
4. Failure: Độ sai khác của phần mềm khi dự kiến chuyển giao.
5. Tóm lại, con người gây ra lỗi (error, mistake) trong mã nguồn, tài liệu, dẫn đến có lỗi (bug, defect hoặc fault) trong mã nguồn, tài liệu, khi thực thi chương trình thì bắt gặp failure.

B. Độ tin cậy và lỗi

1. Độ tin cậy: xác suất phần mềm đó sẽ không gây ra lỗi trong một thời gian xác định dưới một điều kiện nhất định.
2. Một hệ thống không thể không có lỗi (chỉ có trường hợp chưa tìm ra lỗi trong lần đầu tiên)
3. Một hệ thống có thể đáng tin cậy nhưng vẫn có lỗi.
4. Một ứng dụng không có lỗi cũng không thể chắc chắn luôn luôn đáng tin cậy.

C. Tại sao lại xảy ra lỗi trong phần mềm

1. Phần mềm được viết bởi con người:
 - a) Người viết không thể kiểm soát được mọi vấn đề.
 - b) Người viết có kỹ năng nhưng không phải là hoàn hảo.

c) Người viết do vô ý tạo ra lỗi (error)

2. Dưới áp lực phải hoàn thành phần mềm kịp deadline.
3. Không có thời gian để kiểm tra toàn diện phần mềm.
4. Hệ thống có thể chưa hoàn thành.

D. Lỗi phần mềm tiêu tốn tiền của như thế nào

1. Những lỗi phần mềm lớn:

- a) Tên lửa Ariane 4 được sử dụng lại trong phiên bản tiếp theo Ariane 5. Tuy nhiên, rắc rối đã xảy ra khi mã này thực hiện quá trình chuyển đổi số chứa dấu phẩy động 64 bit sang ký hiệu số nguyên 16 bit, Ariane 5 đã phát nổ chỉ 40 giây sau khi phóng.
- b) Tàu thăm dò vũ trụ Mariner I (250 triệu đô).

2. Những lỗi nhỏ khó nhận ra:

- a) Chỉ gây ra một chút bất tiện, tác hại gây ra có thể không nhìn thấy được.
- b) Phần mềm không phải là 1 hệ thống tuyến tính, 1 lỗi nhỏ đầu vào có thể gây ra sai lệch, ảnh hưởng lớn.

3. Lỗi phần mềm có thể gây ra thiệt hại đến tính mạng con người như:

Điều trị bức xạ có thể giết chết bệnh nhân (Therac-25), tai nạn máy bay (Airbus & Korean Airlines), hệ thống ngân hàng thua lỗ dẫn đến tự tử.

E. Vì sao kiểm thử là cực kì quan trọng

1. Bởi vì phần mềm thì luôn luôn có lỗi
2. Để kiểm tra độ tin cậy của phần mềm
3. Các lỗi có thể gây ra tổn thất tốn kém
4. Tránh bị các khách hàng kiện cáo
5. Uy tín trong kinh doanh

F. Kiểm thử toàn diện:

Thực hiện test tất cả các kết hợp đầu vào và các khối lệnh điều khiển

Tuy nhiên, bạn không thể kiểm tra mọi thứ, và nếu không kiểm tra mọi trường hợp thì bạn sẽ bỏ sót lỗi. Sản phẩm phải được tung ra thị trường, vì vậy, bạn cần dừng việc kiểm tra, nhưng nếu dừng quá sớm thì một số vùng sẽ không được kiểm thử. Bạn phải làm như thế nào?

Một nội dung quan trọng mà tester cần phải tìm hiểu là làm thế nào để giảm số lượng các trường hợp kiểm thử rất lớn thành một tập các *test case* có thể thực thi được, và làm thế nào để sàng lọc lựa chọn những quyết định ít rủi ro nhất. Điều này buộc *tester* phải xác định được đâu là vấn đề quan trọng và đâu là vấn đề không quan trọng.

Cần kiểm thử với mức độ như thế nào ?

1. Phụ thuộc vào RỦI RO

- a) Rủi ro những lỗi quan trọng
- b) Rủi ro phát sinh chi phí do lỗi
- c) Rủi ro phát hành phần mềm chưa được kiểm tra
- d) Rủi ro mất uy tín và thị phần
- e) Rủi ro mất thị trường
- f) Rủi ro kiểm thử không hiệu quả

2. Do đó cần sử dụng độ RỦI RO để xác định:

- a) Kiểm thử gì trước
- b) Kiểm thử gì kĩ nhất
- c) Làm thế nào để kiểm thử từng thành phần
- d) Phân bổ thời gian để ưu tiên kiểm thử.

Ưu tiên kiểm tra rằng, bất cứ khi nào ngừng kiểm thử, bạn đã thực hiện các kiểm thử tốt nhất trong khoảng thời gian trước đó tức là kiểm thử các chức năng hệ thống, tính đúng đắn của nó, độ tin cậy, khả năng tái sử dụng, bảo trì. Tuy nhiên có nhiều yếu tố khác ảnh hưởng đến kiểm thử như: yêu cầu của hợp đồng, yêu cầu pháp lý, yêu cầu lĩnh vực cụ thể như dược phẩm, y tế, kiểm soát không lưu, an toàn đường sắt cần được tuân thủ tuyệt đối qui trình định ra.

II. Quy trình kiểm thử cơ bản

A. Quy trình kiểm thử là gì:

- 1. Chế độ kiểm thử được định nghĩa bởi tổ chức phát triển phần mềm là gì.
- 2. Cần có chiến lược kiểm thử và nó sẽ lý giải tại sao tổ chức phần mềm kiểm thử các thành phần mà mình tạo ra.
- 3. Cần nhận dạng cái gì là quan trọng đối với tổ chức (chi phí, chất lượng, thời gian, phạm vi...) và cách nào, bởi ai và khi nào việc kiểm thử sẽ được thực hiện.
- 4. Tất cả các thông tin trên sẽ được lập thành tài liệu cho hoạt động kiểm thử và ta có thể gọi qui trình tạo lập tài liệu này là qui trình lên kế hoạch kiểm thử phần mềm

B. Xây dựng kế hoạch kiểm thử :

- 1. Test Manager hoặc Test Leader sẽ xây dựng kế hoạch ban đầu về kiểm thử:
 - a) Định nghĩa phạm vi kiểm thử.
 - b) Định nghĩa các chiến lược kiểm thử.
 - c) Nhận dạng các rủi ro và yếu tố bất ngờ.

- d) Nhận dạng các hoạt động kiểm thử nào là thủ công, kiểm thử nào là tự động hay cả hai.
- e) Ước lượng chi phí kiểm thử và xây dựng lịch kiểm thử.
- f) Nhận dạng môi trường kiểm thử.

2. Kế hoạch cần được:

- a) Xem lại bởi QC team, Developers, Business Analys, PM và Customer
- b) Chấp thuận bởi : Project Manager and Customer.
- c) Hiệu chỉnh trong suốt chu kỳ kiểm thử để phản ánh các thay đổi nếu cần thiết

C. Phân tích và thiết kế kiểm thử:

1. Test Analyst hoặc Test Designer sẽ thiết kế các testcase từ các yêu cầu liên quan:

- a) Sẽ thiết kế các testcase từ các yêu cầu chức năng và các yêu cầu không chức năng của phần mềm.
- b) Các testcase cần bao phủ tất cả khía cạnh kiểm thử cho từng yêu cầu phần mềm.
- c) Các testcase cần bao phủ tất cả yêu cầu trong các chiến lược kiểm thử.
- d) Nếu cần kiểm thử tự động, Test Designer sẽ xây dựng các kịch bản dựa trên các Testcase/Test procedures.

2. Kiểm tra kế hoạch:

Bao gồm:

- a) Nhận dạng các chiến lược được dùng để kiểm tra và đảm bảo rằng sản phẩm thỏa mãn đặc tả thiết kế phần mềm và các yêu cầu khác về phần mềm.
- b) Định nghĩa các mục tiêu và phạm vi của nỗ lực kiểm thử. Test Planning Test Analysis & Design (Manual or Automation) Test Executing (Manual or Automation) Test Report & Evaluation
- c) Nhận dạng phương pháp luận mà đội kiểm thử sẽ dùng để thực hiện công việc kiểm thử.
- d) Nhận dạng phần cứng, phần mềm và các tiện ích cần cho kiểm thử.
- e) Nhận dạng các tính chất và chức năng sẽ được kiểm thử.
- f) Xác định các hệ số rủi ro gây nguy hại cho việc kiểm thử.
- g) Lập lịch kiểm thử và phân phối công việc cho mỗi thành viên tham gia.

3.Test Manager hoặc Test Leader sẽ xây dựng kế hoạch kiểm thử.

4.Đặc tả yêu cầu kiểm thử:

Bao gồm:

- a) Xác định điều kiện
- b) Thiết kế test case
- c) Xây dựng test case

5.Một test case tốt:

- a) Có hiệu quả: phải tìm được lỗi
- b) Làm bản mẫu: minh họa các test case khác
- c) Phát triển được: dễ bảo trì
- d) Tính kinh tế: chi phí thấp

6.Phân tích:

Xác định điều kiện: xác định cái gì cần được kiểm thử, cần được ưu tiên.

Liệt kê các điều kiện mà chúng ta muốn kiểm tra bằng cách:

- a) Sử dụng các kỹ thuật thiết kế kiểm thử được định nghĩa trong kế hoạch kiểm thử
- b) Có thể có nhiều điều kiện cho mỗi chức năng hoặc thuộc tính của hệ thống

7.Điều kiện ưu tiên:

- a) Phải đảm bảo các điều kiện quan trọng nhất được thực hiện trước
- b) Phải lựa chọn điều kiện kiểm thử

8.Thiết kế test case: xác định cách thức để kiểm thử chúng:

Thiết kế đầu vào và dữ liệu kiểm thử:

- a) Mỗi lần test kiểm tra một hoặc nhiều điều kiện kiểm thử
- b) Xác định kết quả mong đợi:
 - (1) Dự đoán kết quả từng test case, đầu ra là gì, cái gì thay đổi và cái gì không thay đổi
- c) Thiết kế các bộ kiểm thử:
 - (1) Mỗi bộ kiểm thử khác nhau cho mục tiêu khác nhau như đệ quy, xây dựng một cách tin cậy và phát hiện lỗi

9.Xây dựng test case: thực hiện test case:

Chuẩn bị kịch bản kiểm thử

- a) Ít thông tin hệ thống thì người kiểm thử có thêm nhiều chi tiết mà kịch bản sẽ phải làm

- b) Kịch bản cho các công cụ phải xác định từng chi tiết

10. Chuẩn bị dữ liệu kiểm thử

- a) Dữ liệu phải tồn tại trong tập tin và cơ sở khi bắt đầu kiểm thử

Chuẩn bị kết quả mong đợi

- b) Cần được xác định trước khi kiểm thử thi hành

D. Thực hiện kiểm thử:

1. Thực hiện truyền lệnh cho test cases:

- a) Truyền lệnh cái quan trọng nhất trước
- b) Không thể thi hành tất cả test cases nếu
 - (1) Kiểm tra chỉ sửa lỗi
 - (2) Quá nhiều lỗi được tìm thấy ở các test cases trước
 - (3) Áp lực về thời gian
- c) Có thể thực hiện thủ công hoặc tự động

E. Báo cáo kiểm thử:

1. Báo cáo kiểm thử gồm:

- a) Các đặc tính và các phiên bản (rõ ràng) của:
 - (1) Phần mềm đang thử nghiệm
 - (2) Đặc tả yêu cầu kiểm thử

2. Thực hiện theo kế hoạch:

- a) Phân biệt các quá trình trong kịch bản kiểm thử
- b) Ghi kết quả thực tế từ các thử nghiệm
- c) Nắm bắt bất kỳ ý tưởng nào bạn có cho các test case mới
- d) Chú ý rằng các báo cáo này được dùng để thiết lập tất cả hoạt động thử nghiệm đã thực hiện như quy định

3. So sánh kết quả thực tế với kết quả mong đợi. Log khác nhau cho nên:

- a) Lỗi phần mềm
- b) Lỗi kiểm thử (vd: kết quả mong đợi sai)
- c) Lỗi môi trường hoặc phiên bản
- d) Chạy thử nghiệm không chính xác
- e) Log bao phủ mức độ đạt được (với đơn vị đo lường xác định như tiêu chí hoàn thành kiểm thử)

- f) Sau khi lỗi được sửa, lặp lại các hoạt động kiểm thử bắt buộc (thực hiện, thiết kế, lập kế hoạch)

F. Kiểm tra hoàn thành:

1. Tiêu chí hoàn thành thử nghiệm được xác định trong kế hoạch kiểm thử
2. Nếu không được đáp ứng, cần lặp lại các hoạt động kiểm thử, ví dụ như kiểm tra đặc tả để thiết kế nhiều thử nghiệm khác.

3. Tiêu chí hoàn thành xác định khi nào dừng

- a) Phạm vi, dùng một kỹ thuật đo lường, ví dụ:
 - (1) Phạm vi nhánh cho kiểm thử đơn vị
 - (2) Yêu cầu người dùng
 - (3) Giao dịch thường xuyên nhất được sử dụng
- b) Lỗi được tìm thấy (ví dụ: kết quả khác với dự kiến)
 - (1) Chi phí hoặc thời gian
- c) Tạo sao cần phải thực hiện quy trình kiểm thử phần mềm ?
 - (1) Cần làm rõ vai trò và trách nhiệm của việc kiểm thử phần mềm.
 - (2) Cần làm rõ các công đoạn, các bước kiểm thử.
 - (3) Cần phải hiểu và phân biệt các tính chất kiểm thử (tạo sao phải kiểm thử), các bước kiểm thử (khi nào kiểm thử), và các kỹ thuật kiểm thử (kiểm thử bằng cách nào).

III. Quan niệm về kiểm thử

A. Tại sao cần phải kiểm thử:

1. Xây dựng sự tin cậy
2. Giải thích sự phù hợp với yêu cầu
3. Tìm lỗi
4. Giảm chi phí
5. Cho thấy hệ thống đáp ứng nhu cầu người dùng
6. Đánh giá chất lượng phần mềm

B. Theo thời gian tính tin cậy càng cao

1. Cách tiếp cận test truyền thống:
Cho biết rằng hệ thống:
 - a) Làm những gì nó nên làm
 - b) Không làm những gì nó không nên làm

2.Cách tiếp cận test hiện đại:

Cho hệ thống thấy rằng :

- a) Làm những gì nó không nên làm
- b) Không làm những gì nó nên làm

C. Kiểm thử nghịch lý

1.Mục đích kiểm thử:

- a) Tìm ra lỗi.
- b) Tìm kiếm lỗi và phá hủy độ tin cậy.
- c) Gia tăng độ tin cậy ???
=> Cách để gia tăng độ tin cậy chính là phá hủy nó. (Bạn sẽ được khách hàng tin tưởng hơn khi chính bạn tìm ra lỗi phần mềm của chính bạn viết ra.)

2.Đối tượng nào muốn trở thành một kiểm thử viên?

- a) Một người muốn phá hoại.
- b) Mang lại tin xấu.
- c) Dưới áp lực của deadline.
- d) Cần có một cái nhìn khác, một tư duy khác.
- e) Làm thế nào khi lỗi thông tin khi truyền đạt.

3.Kiểm thử viên cần làm:

- a) Thông tin chính xác về sự tiến độ và thay đổi.
- b) Hiểu biết sâu sắc từ các nhà phát triển về lĩnh vực phần mềm.
- c) Mã test theo một tiêu chuẩn thống nhất.
- d) Được coi như một chuyên gia.
- e) Tìm lỗi.
- f) Thách thức về kỹ thuật và kế hoạch kiểm thử.
- g) Báo cáo những lỗi nghiêm trọng.
- h) Đưa ra dự đoán về mức độ lỗi trong tương lai.
- i) Cải thiện quá trình, khả năng kiểm thử của mình.

4.Kiểm thử viên có trách nhiệm:

- a) Làm theo kế hoạch kiểm tra, kịch bản cũng như tài liệu.
- b) Báo cáo lỗi khách quan và chính xác.
- c) Kiểm tra lỗi có đúng không trước khi báo cáo s/w về lỗi.

- d) Ghi nhớ nó là phần mềm, bạn không phải là lập trình viên, và bạn đang kiểm thử nó.
- e) Đánh giá rủi ro khách quan.
- f) Ưu tiên quan tâm đến những gì bạn báo cáo.
- g) Báo cáo lại sự thật.

5. Tính độc lập:

- a) Kiểm tra công việc của chính bạn.
 - (1) Tìm ra 30%-50% lỗi của bản thân.
 - (2) Cùng một giả thiết và quá trình tư duy.
 - (3) Xem những gì có ý nghĩa hoặc cần xem, không phải những gì đang có.
 - (4) Cảm xúc chi phối:
 - (a) Không muốn tìm lỗi.
 - (b) Chủ động không muốn tìm lỗi.

6. Mức độ độc lập.

- a) Chủ quan: Test được thiết kế bởi những người viết phần mềm.
- b) Các test được thiết kế bởi một người khác
- c) Các test được thiết kế bởi một người nào đó từ một bộ phận khác nhau hoặc nhóm khác (ví dụ như nhóm test)
- d) Các test được thiết kế bởi một người nào đó từ một tổ chức khác nhau (ví dụ như cơ quan)
- e) Các test được tạo ra bởi một công cụ (Có thể tạo ra các test chất lượng thấp?)

IV. Kiểm thử lại và Kiểm thử hồi quy.

A. Kiểm thử lại sau khi lỗi đã được sửa:

1. Chạy test, nếu có lỗi, hãy báo cáo lỗi.
2. Phiên bản mới của phần mềm vẫn còn lỗi? Hay đã được sửa?
3. Kiểm thử lại tương tự (tức là tái kiểm thử)
 - a) phải được lặp lại chính xác.
 - b) Cùng một môi trường, phiên bản (trừ các phần mềm đã bị thay đổi có chủ ý!)
 - c) Cùng một đầu vào và điều kiện tiên quyết.
4. Nếu lúc này chương trình chạy đúng, lỗi đã được sửa một cách chính xác hay chưa - hoặc vẫn còn?

=> Nhược điểm: Lỗi mới được tạo ra bởi lần sửa lỗi đầu tiên không được tìm thấy trong quá trình kiểm thử lại.

B. Kiểm thử hồi quy:

1. Mục tiêu để tìm bất kỳ lỗi nào bất thường tuy nhiên không thể đảm bảo tìm được tất cả các lỗi.
2. Tên nhầm lẫn: "ngược hồi quy" hoặc "tiến trình"
3. Bộ tiêu chuẩn của kiểm thử - gói kiểm thử hồi quy
4. Thực hiện ở mọi cấp độ (đơn vị, tích hợp, hệ thống, chấp nhận)
5. Cũng có thể tự động hoá
6. Một nội dung đang phát triển nhưng cần phải được bảo trì.
7. Kiểm tra hồi quy được thực hiện khi:
 - a) sau khi thay đổi phần mềm, bao gồm cả lỗi đã được sửa.
 - b) khi môi trường thay đổi, kể cả khi ứng dụng không thay đổi.
 - c) các bản sửa lỗi khẩn cấp (có thể chỉ là một thay đổi nhỏ)
8. Các bộ kiểm thử hồi quy:
 - a) Phát triển theo thời gian
 - b) Chạy thường xuyên
 - c) Có thể tăng kích thước.
9. Bảo trì các gói kiểm thử hồi quy:
 - a) Loại bỏ các test lặp đi lặp lại (test mà kiểm thử cùng điều kiện thử nghiệm)
 - b) Kết hợp test (ví dụ như nếu các test luôn luôn chạy cùng nhau)
 - c) Chọn một tập hợp khác nhau của các bộ hồi qui đầy đủ để chạy mỗi khi kiểm thử hồi quy là cần thiết.
 - d) Loại bỏ test mà không tìm thấy một lỗi trong thời gian dài.
10. Kiểm thử hồi quy và tự động hóa:
 - a) Công cụ thực hiện kiểm thử (ví dụ capture replay) là công cụ kiểm thử hồi quy - nó thực hiện lại các bài kiểm tra đã được thực hiện.
 - b) Một khi tự động, kiểm tra hồi quy có thể được chạy thường xuyên như mong muốn (ví dụ như mỗi đêm)
 - c) Tự động kiểm thử không phải là đơn giản (thông thường phải mất 2-10 lần nữa để tự động hóa một thử nghiệm hơn khi chạy nó bằng tay)

- d) Đừng tự động hóa tất cả mọi thứ - hãy lập kế hoạch để kiểm thử tự động cái gì đầu tiên, chỉ kiểm thử tự động nếu đáng giá.

V. Kết quả mong đợi.

- A. Nên được dự đoán trước như là một phần của quá trình thiết kế thử nghiệm 'Oracle Assumption' giả định rằng kết quả chính xác có thể tiên đoán được.
- B. Tại sao không chỉ nhìn vào những gì các phần mềm thể hiện và đánh giá vào thời điểm đó?
1. Có mong muốn vượt qua các test - ít việc để làm, không có báo cáo sự việc.
 2. Nó trông có vẻ hợp lý, thì nó phải OK - ít nghiêm ngặt hơn trong với tính toán và so sánh.

VI. Kiểm thử ưu tiên:

- A. Lí do dùng kiểm thử ưu tiên:
1. Chúng ta không thể kiểm thử mọi thứ.
 2. Không bao giờ có đủ thời gian để làm tất cả các test mà bạn muốn.
 3. Vì vậy, những test nào bạn nên làm?
- B. Nguyên tắc quan trọng nhất:
1. Kiểm thử ưu tiên là khi nào bạn ngừng kiểm thử có nghĩa bạn đã thực hiện các kiểm thử tốt nhất trong thời gian có hạn.
- C. Lựa chọn độ ưu tiên:
1. Tiêu chí xếp hạng có thể (tất cả đều dựa theo mức độ rủi ro)
 - a) Kiểm thử nơi lỗi là nghiêm trọng nhất.
 - b) Kiểm thử nơi lỗi sẽ được nhìn thấy rõ nhất.
 - c) Kiểm thử nơi lỗi có khả năng xảy ra nhất.
 - d) Hỏi khách hàng để ưu tiên các yêu cầu.
 - e) Những gì là quan trọng nhất đối với các khách hàng, doanh nghiệp?
 - f) Khu vực thay đổi thường xuyên nhất.
 - g) Khu vực có nhiều lỗi nhất trong quá khứ.
 - h) Hầu hết các khu vực phức tạp, hoặc kỹ thuật quan trọng.

VII. Tóm tắt: Những điểm chính

- A. Kiểm thử là cần thiết bởi vì con người luôn tạo ra lỗi.
- B. Quá trình kiểm thử: lập kế hoạch, phân tích đặc điểm kỹ thuật, thực hiện, ghi lại, kiểm tra hoàn thành.

- C.** Độc lập và mối quan hệ rất quan trọng trong việc kiểm thử.
- D.** Kiểm thử lại những lỗi đã sửa; kiểm thử hồi quy cho những điểm bất thường.
- E.** Kết quả mong đợi từ một đặc điểm kỹ thuật trước.
- F.** Ưu tiên để làm các kiểm thử tốt nhất trong thời gian bạn có.

