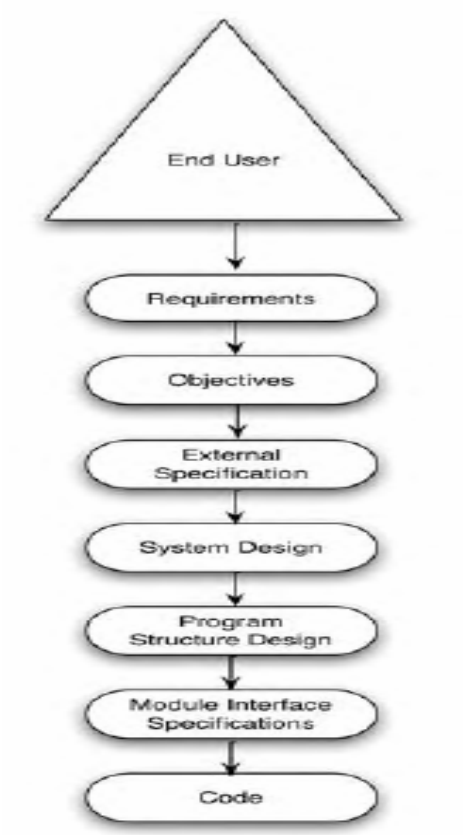
Chương 9

Các hoạt động kiểm thử khác

9.1 Giới thiệu

Sau khi kiểm thử mọi đơn vị chức năng phần mềm và sửa lỗi hoàn chỉnh cho chúng, ta cũng không thể đảm bảo là đã tìm hết lỗi trong phần mềm. Thật vậy, còn nhiều lỗi khác mà kiểm thử đơn vị chưa phát hiện được. Tại sao vậy ?

Như chúng ta biết trong qui trình phát triển phần mềm, ta đã thực hiện 1 số workflows như :

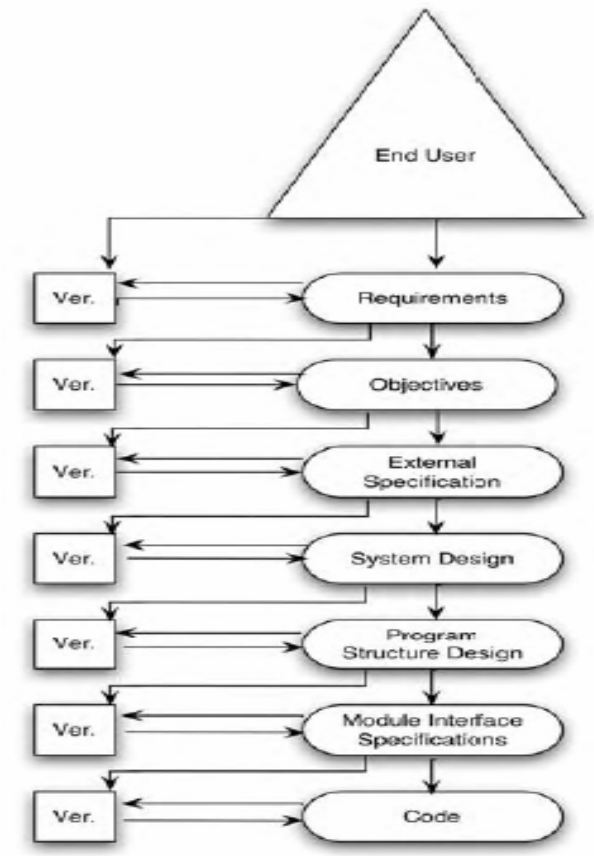


1. Xác định các yêu cầu để biết rõ tạo sao phần mềm là cần thiết.
2. Xác định các mục tiêu của phần mềm để biết rõ những gì phần mềm phải thực hiện và mức độ thực hiện chúng như thế nào ?

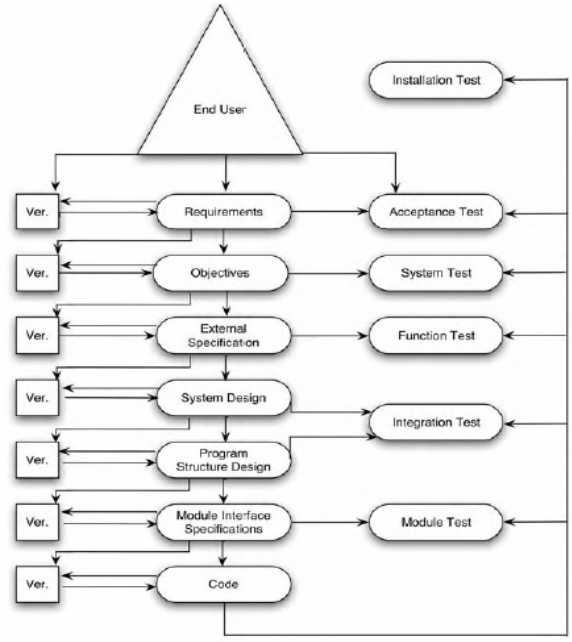
3. Đặc tả các chức năng mà người dùng thấy về phần mềm.
4. Thiết kế hệ thống và thiết kế cấu trúc cụ thể và chi tiết của phần mềm.
5. Đặc tả giao tiếp của từng module chức năng.
6. Hiện thực chi tiết các chức năng của từng module.

Về nguyên tắc, con người có những hạn chế nhất định, kết quả của 1 công việc nào đó đều có thể có lỗi, và nếu dùng kết quả này làm dữ liệu đầu vào cho hoạt động kế tiếp thì kết quả của hoạt động kế cũng sẽ bị lỗi,... Ta thường dùng tổ hợp 2 biện pháp sau đây để hạn chế/ngăn ngừa các lỗi :

- Xác định lại cho rõ ràng và chi tiết hơn từng workflows của qui trình phát triển phần mềm.
- Ở cuối việc thực hiện 1 workflows bất kỳ, cần thêm 1 hoạt động được gọi là "thanh kiểm tra kết quả" để đảm bảo chất lượng kết quả này trước khi dùng nó để thực hiện workflow kế tiếp.



Ứng với mỗi workflow khác nhau, ta xác định và dùng chiến lược kiểm thử phù hợp để dễ dàng xác định các loại lỗi đặc thù của workflow đó.



Mục đích của kiểm thử đơn vị là phát hiện sự khác biệt giữa đặc tả giao tiếp của đơn vị và thực tế mà đơn vị này cung cấp.

Mục đích của kiểm thử chức năng là chỉ ra rằng chương trình không tương thích với các đặc tả bên ngoài của nó.

Mục đích của kiểm thử hệ thống là chỉ ra rằng chương trình không tương thích với các mục tiêu ban đầu của nó.

Các lợi ích :

- tránh kiểm thử dư thừa.
- ngăn chặn sự quan tâm nhiều vào quá nhiều loại lỗi tại từng thời điểm.

Chú ý : trình tự các hoạt động kiểm thử trong hình ở slide trước không nhất thiết ám chỉ trình tự thời gian kiểm thử tương ứng.

9.2 Kiểm thử chức năng

Qui trình cố gắng tìm ra các khác biệt giữa đặc tả bên ngoài của phần mềm và thực tế mà phần mềm cung cấp.

Đặc tả bên ngoài của phần mềm là đặc tả chính xác về hành vi của phần mềm theo góc nhìn của người dùng thấy.

Kiểm thử chức năng thường sử dụng 1 kỹ thuật kiểm thử hộp đen nào đó :

- Kỹ thuật phân lớp tương đương (Equivalence Class Partitioning).
- Kỹ thuật dùng các bảng quyết định (Decision Tables)
- Kỹ thuật kiểm thử các bộ n thần kỳ (Pairwise)
- Kỹ thật phân tích vùng miền (domain analysis)
- Kỹ thuật dựa trên đặc tả Use Case (Use case)
- ...

Các cách tiếp cận để kiểm thử chức năng phần mềm :

- User Navigation Testing
- Transaction Screen Testing

- Transaction Flow Testing
- Report Screen Testing
- Report Flow Testing
- Database Create/Retrieve/Update/Delete Testing

1. Kiểm thử khả năng duyệt chức năng của người dùng (User Navigation Test)

Các màn hình phục vụ duyệt thực hiện chức năng là màn hình log on/log off, menu bar và hệ thống cây phân cấp các option để thực hiện chức năng, toolbar, tất cả các mối liên kết từ màn hình này tới màn hình khác để thể hiện sự liên tục của hoạt động nghiệp vụ đang cần thực hiện.

Kiểm thử khả năng duyệt chức năng của người dùng tập trung trên :

- khả năng người dùng login vào hệ thống với quyền hạn thích hợp.
- di chuyển qua các màn hình "giao tác" mong muốn 1 cách đúng đắn và logout khỏi phần mềm.

2. Kiểm thử màn hình giao tác (Transaction screen Test)

Màn hình giao tác có các field nhập liệu, list chọn option, các options, các button chức năng (Add, Change, Delete, Submit, Cancel, OK...).

Một vài loại kết quả có thể được hiển thị trên màn hình giao tác sau khi người dùng click button chức năng nào đó.

Công việc của người kiểm thử :

- Thiết kế testcase để xác thực hoạt động của mỗi field dữ liệu, list, option và button trên màn hình giao tác theo các yêu cầu nghiệp vụ, tài liệu người dùng và tài liệu người quản trị.

- Nếu kết quả được hiển thị trên màn hình giao tác, thì kỹ thuật kiểm thử hộp đen với testcase gồm (data input, ouput kỳ vọng) sẽ được dùng để xác thực kết quả hiển thị.

3. Kiểm thử luồng giao tác (Transaction Flow Test)

Kiểm tra kết quả tổng hợp của nhiều màn hình giao tác theo thứ tự duyệt đúng có hoàn thành hoạt động nghiệp vụ tương ứng không ?

Thí dụ nghiệp vụ cập nhật profile khách hàng gồm các màn hình giao tác sau :

- màn hình 1 cập nhật tên, địa chỉ, contact. Màn hình 2 cập nhật credit. Màn hình 3 cập nhật thông tin thanh toán và khuyến mãi. Màn hình 4 tổng kết profile và thực hiện cập nhật. Màn hình 5 để xem kết quả profile đã cập nhật.
- Kết quả cuối cùng của trình tự các màn hình là file hay database sẽ được cập nhật để chứa các thông tin mà người dùng đã cập nhật thông qua các màn hình giao tác.

Nhiệm vụ của người kiểm thử :

- Xác thực rằng nếu người dùng thực hiện đúng trình tự các màn hình giao tác và hoàn tất được chúng thì hệ thống se cung cấp kết quả đúng.
- Ngược lại, nếu người dùng không tuân thủ bất kỳ 1 qui luật nghiệp vụ nào trong 1 màn hình giao tác nào thì hệ thống sẽ không cung cấp kết quả gì cho người dùng.

4. Kiểm thử màn hình báo biểu (Report screen Test)

màn hình báo biểu cho phép tìm kiếm dữ liệu và hiển thị kết quả (không cần nhập dữ liệu như màn hình giao tác).

Khó khăn trong kiểm thử màn hình bào biểu nằm ở chỗ có nhiều cách mà người dùng có thể đặc tả dữ liệu cần được tìm kiếm (tiêu chuẩn) và cách thức dữ liệu này được hiển thị (sắp xếp và định dạng).

Công việc của người kiểm thử là chú ý đặc biệt vào dữ liệu tìm kiếm và hiển thị vì người dùng có thể chọn sai dữ liệu hay tệ hơn là không có kết quả nào được hiển thị.

5. Kiểm thử luồng báo biểu (Report Flow Test)

Kiểm thử các khác biệt giữa kết quả hiển thị trong màn hình báo biểu và các phương thức báo biểu khác (như máy in, file,..).

Nhiệm vụ của người kiểm thử :

- Xác định xem phần mềm gởi cùng kết quả ra màn hình report và máy in ?
- Xác thực kết quả báo biểu trên tất cả phương thức báo cáo khác nhau được hỗ trợ bởi phần mềm.
- Xác định xem khả năng máy in có hỗ trợ font, vùng chọn được người dùng xác định trong màn hình báo biểu ?

6. Kiểm thử việc Create/Retrieve/Update/Delete database

Thường được thực hiện thông qua 2 bước :

- Kiểm thử việc thiết kế, khởi tạo database ban đầu thông qua tiện ích bên ngoài phần mềm ứng dụng cần kiểm thử.
- Kiểm thử việc phần mềm sử dụng database đã được thiết kế và khởi tạo đúng.

Đòi hỏi sự hợp tác và cộng tác giữa người kiểm thử và người quản trị database.

9.3 Kiểm thử hệ thống

Kiểm thử hệ thống không phải là qui trình kiểm thử toàn bộ chức năng của 1 chương trình hay của 1 hệ thống phần mềm đầy đủ.

Mục đích của kiểm thử hệ thống là so sánh hệ thống hay chương trình với các mục tiêu ban đầu của nó.

Kiểm thử hệ thống không bị hạn chế với các hệ thống phần mềm. Nếu sản phẩm cần kiểm thử là 1 chương trình, kiểm thử hệ thống là qui trình cố gắng chứng minh cách mà toàn bộ phần mềm không thỏa mản các mục tiêu của nó.

Theo định nghĩa trên, kiểm thử hệ thống không thể xảy ra được nếu ta không viết ra rõ ràng các thông tin đo đạt được về các mục tiêu của chương trình.

Thí dụ về đặc tả mục tiêu của chương trình :

- Hãy hiện thực 1 hàng lệnh để từ cửa sổ text-mode, người dùng xem các nội dung chi tiết về các ô nhớ trong bộ nhớ chính của phần mềm.
- Cú pháp của hàng lệnh nên nhất quán với cú pháp của các lệnh khác mà hệ thống cung cấp.
- Người dùng nên có thể đặc tả vùng nhớ thông qua 2 địa chỉ đầu cuối hay thông qua địa chỉ đầu và số lượng ô nhớ cần xem.
- Các toán hạng của lệnh nên có nhiệm ý gợi nhớ.
- Kết quả nên xuất trên nhiều hàng, nội dung của từng ổ nhớ ở dạng hex cách nhau bởi 1 hay nhiều khoảng trắng.
- Mỗi hàng nên chứa địa chỉ của ô nhớ đầu hàng đó.
- Lệnh là bình thường, nghĩa là nếu máy đang chạy bình thường, nó sẽ bắt đầu xuất kết quả trong vài giây và kết quả xuất không có thời gian chờ giữa các ô nhớ trong hàng hay giữa các hàng.
- Lỗi lập trình ít nhất nên làm cho lệnh bị dừng, hệ thống và session người dùng không bị ảnh hưởng gì hết.
- Sau khi hệ thống bắt đầu xuất kết quả, bộ xử lý lệnh không nên có hơn 1 lỗi do người dùng phát hiện được.

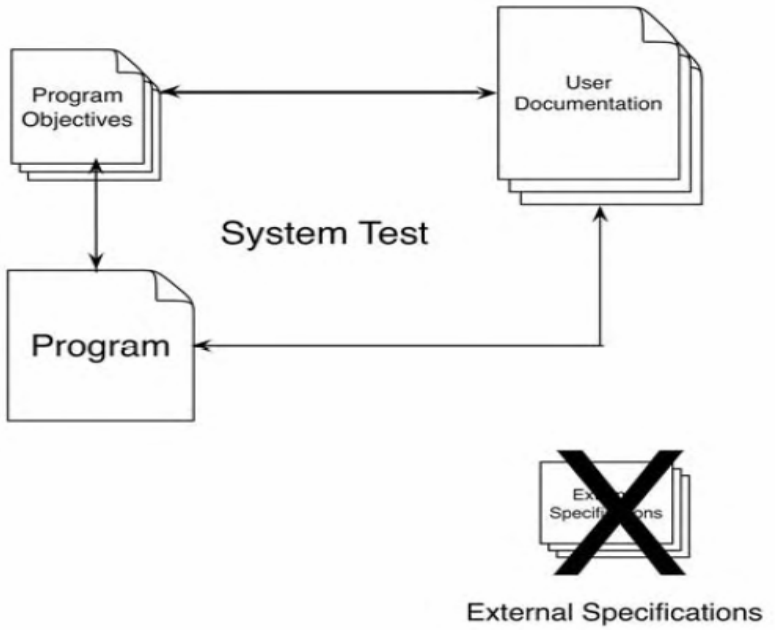
Kiểm thử hệ thống là qui trình kiểm thử quan trọng.

Các chất liệu để tạo testcase cho kiểm thử hệ thống :

- Chúng ta không chỉ dùng đặc tả theo góc nhìn người dùng để suy ra các testcase.
- Tài liệu về mục tiêu chương trình, tự nó cũng không thể được dùng để tạo ra các testcase, vì theo định nghĩa, nó không chứa các miêu tả chính xác, rõ ràng về giao tiếp từ ngoài vào chương trình.
- ta sẽ dùng tài liệu sử dụng và những công bố cho người dùng.

Thiết kế testcase bằng cách phân tích các mục tiêu.

Tạo các testcase bằng cách phân tích tài liệu dành cho người dùng.



Kiểm thử hệ thống là hoạt động kiểm thử khó khăn nhất

Phải so sánh chương trình với các mục tiêu ban đầu : nên không có phương pháp luận thiết kế testcase tường minh.

Dùng cách tiếp cận khác để thiết kế testcase :

- Thay vì miêu tả phương pháp luận, các loại testcase riêng biệt sẽ được đề cập.

- Do không có phương pháp luận, kiểm thử hệ thống đòi hòi rất nhiều sự năng động và sáng tạo.

Kiểm thử hệ thống tập trung vào kiểm thử các yêu cầu không chức năng. Có 15 yêu cầu không chức năng sau đây có thể cần kiểm thử (nhưng không phải phần mềm nào cũng đòi hỏi đủ 15 yêu cầu này) :

- Facility Testing
- Volume Testing
- Stress Testing
- Usability Testing
- Security Testing
- Performance Testing
- Storage Testing
- Configuration Testing
- Compatibility/Configuration/Conversion Testing
- Installability Testing
- Reliability Testing
- Recovery Testing
- Serviceability Testing
- Documentation Testing
- Procedure Testing

1. Kiểm thử phương tiện (Facility Test)

xác định xem mỗi phương tiện được đề cập trong phần mục tiêu của chương trình đã được hiện thực thực sự chưa.

Qui trình kiểm thử :

- Dò nội dung, từng câu một, miêu tả mục tiêu.

- Khi 1 câu miêu tả cái gì, xác định chương trình đã thỏa mãn cái đó chưa.

Thường ta có thể thực hiện kiểm thử phương tiện mà không cần chạy máy tính, so sánh bằng trí óc các mục tiêu với tài liệu sử dụng đôi khi đủ rồi.

2. Kiểm thử dung lượng (Volume Test)

Mục đích của kiểm thử dung lượng là chỉ ra rằng chương trình không thể xử lý khối lượng dữ liệu lớn được đặc tả trong bảng đặc tả mục tiêu chương trình.

Thí dụ :

- Chương trình dịch không thể dịch file mã nguồn dài 10MB.
- Trình liên kết không thể liên kết 1000 module chức năng khác nhau lại.
- Trình xem phim không thể chiếu file film dài 15GB.

Kiểm thử dung lượng thường đòi hỏi rất nhiều tài nguyên, con người lẫn máy tính.

3. Kiểm thử tình trạng căng thẳng (StressTest)

Mục đích của kiểm thử tình trạng căng thẳng là chỉ ra rằng chương trình sẽ không thể hoạt động được hay hoạt động không tốt trong tình huống căng thẳng : quá nhiều yêu cầu đồng thời, quá nhiều chương trình khác đang cạnh tranh tài nguyên,...

Thí dụ :

- web server sẽ bế tắc nếu có 100000 yêu cầu truy xuất trang web đồng thời.
- HĐH không thể quản lý 1000 process chạy đồng thời.
- Trình chiếu phim sẽ không chiếu phim mượt và tốt nếu có nhiều chương trình khác cần rất nhiều tài nguyên đang chạy.

4. Kiểm thử độ khả dụng (UsabililyTest)

Mục đích của kiểm thử độ khả dụng là chỉ ra các phương tiện/kết quả nhập/xuất không phù hợp, thân thiện với người dùng :

- Mỗi đối tượng giao diện có thân thiện, tự nhiên và dễ dùng không ?
- Kết quả xuất có ngắn gọn, trong sáng, nghĩa dễ hiểu không ?
- Các cảnh báo có dễ hiểu không ? "IEK022A OPEN ERROR ON FILE 'SYSIN' ABEND CODE=102?"
- Nói chung tất cả các kết quả, các cảnh báo đều phải nhất quán, đồng nhất về cú pháp, về định dạng, ngay cả các từ viết tắt được dùng.

Một số chú ý :

- Khi độ chính xác là rất quan trọng như trong hệ thống quản lý ngân hàng, thì thông tin nhập có tính dư thừa đủ không ?
- Hệ thống có quá nhiều nhiệm ý hay các nhiệm ý được người dùng thích dùng không ?
- Hệ thống có trả về đủ đáp ứng với mọi hoạt động nhập ?
- Chương trình có dễ dùng và thân thiện ?

5. Kiểm thử các dịch vụ cộng thêm (Serviceability Test)

Trong mục tiêu của phần mềm có thể đề cập đến 1 số dịch vụ cộng thêm, thí dụ như :

- Chương trình chẩn đoán và xuất nội dung thô của bộ nhớ chương trình.
- Thời gian trung bình để debug 1 vấn đề rõ ràng.
- Các thủ tục bảo trì.
- Chất lượng của tài liệu luận lý bên trong.

Các mục tiêu trên, nếu có đề cập trong mục tiêu chương trình thì cần phải được kiểm thử.

6. Kiểm thử tính an ninh (Security Test)

An ninh phần mềm gồm 3 vấn đề chính là bảo mật, tính toàn vẹn dữ liệu và độ sẵn sàng đáp ứng.

Nghiên cứu các vấn đề liên quan đến an ninh trong các hệ thống tương tự rồi tạo các testcase để chứng minh rằng các vấn đề này cũng tồn tại trong chương trình cần kiểm thử.

Các ứng dụng mạng và ứng dụng theo công nghệ Web hiện nay cần được kiểm thử tính an ninh ở mức độ cao hơn nhiều so với phần mềm truyền thống trên máy đơn. Điều này đặc biệt đúng cho các website thương mại, ngân hàng...

7. Kiểm thử hiệu xuất làm việc (Performance Test)

Mục đich của kiểm thử hiệu xuất làm việc là chỉ ra rằng phần mềm không đạt được hiệu xuất được đặc tả trong mục tiêu chương trình.

Thí dụ :

- trình chiếu phim full HD không chiếu kịp 20 frame/sec.
- trình nén dữ liệu không nén dữ liệu kịp với tốc độ đề ra.
- trình soạn thảo văn bản không nhận và xử lý kịp các ký tự được nhập bởi người dùng.
- trình ghi DVD không tạo dữ liệu ghi kịp tốc độ mà ổ DVD yêu cầu...

8. Kiểm thử độ sử dụng bộ nhớ (Storage Test)

Mục đich của kiểm thử độ sử dụng bộ nhớ là chỉ ra rằng phần mềm không tuân thủ về dung lượng bộ nhớ tối thiểu/tối đa được đặc tả trong mục tiêu chương trình.

Thí dụ :

- kích thước tối thiểu 128KB không đủ để chạy chương trình.
- chương trình không dùng hết kích thước bộ nhớ tối đa là 4GB.
- chương trình không chạy được khi đĩa còn dung lượng trống tối thiểu là 4MB.
- chương trình không quản lý được dung lượng đĩa là 1TB...

9. Kiểm thử cấu hình làm việc (Configuration Test)

Nhiều chương trình như HĐH, hệ quản trị CSDL, Website,... thường sẽ làm việc được trên nhiều cấu hình phần cứng/phần mềm cấp thấp. Số lượng các cấu hình khác nhau có thể quá lớn, nhưng ta nên chọn 1 số cấu hình phổ dụng nhất để kiểm thử xem chương trình có chạy tốt trên các cầu hình này không.

10. Kiểm thử tính tương thích/chuyển đổi/cấu hình (Compatibility/ Configuration/Conversion Test)

Đời sống của 1 phần mềm thường dài, nhất là phần mềm thương mại của các hãng lớn. Trong cuộc đời của mình, phần mềm được phát triển tăng dần theo từng release, từng version. Về nguyên tắc, version mới sẽ tương thích ngược với version đã có.

Mức độ tương thích, khả năng chuyển đổi định dạng file dữ liệu từ cũ sang mới hay ngược lại, khả năng cấu hình version mới để có thể làm việc như version cũ,... có thể được đặc tả trong mục tiêu của chương trình.

Nếu có thì ta phải kiểm thử các đặc tả này xem version cần kiểm thử có đáp ứng được không.

Thí dụ : Word 2003 có thể cấu hình để chạy y như Word 97 không ?

11. Kiểm thử khả năng cài đặt (Installability Test)

Một số hệ thống phần mềm có thủ tục cài đặt khá phức tạp.

Chương trình cài đặt chạy sai có thể ngăn chận người dùng không dùng được phần mềm được cài đặt.

Nhiệm vụ của kiểm thử khả năng cài đặt là kiểm thử chương trình cài đặt có hoạt động đúng không ?

12. Kiểm thử độ tin cậy (Reliability Test)

Mục tiêu của mọi loại kiểm thử đều hướng đến việc cải tiến độ tin cậy của chương trình.

Nếu mục tiêu của chương trình chứa các phát biểu đặc biệt về độ tin cậy, ta cũng cần phải thực hiện hoạt động kiểm thử độ tin cậy đặc thù.

Việc kiểm thử các mục tiêu về độ tin cậy có thể khó khăn. Thí dụ, 1 hệ thống online hiện đại như WAN hay ISP thường có thời gian làm việc thực tế bằng 99.97% thời gian sống của nó.

Chưa có cách để ta có thể kiểm thử mục tiêu này với thời gian kiểm thử hàng tháng hay hàng năm.

13. Kiểm thử độ phục hồi sau lỗi (Recovery Test)

Các chươnng trình như hệ điều hành, hệ quản trị database, các chương trình xử lý từ xa thường có các mục tiêu về phục hồi sau lỗi để miêu tả cách hệ thống phục hồi sau khi lỗi dữ liệu, lỗi phần mềm hay lỗi phần cứng xảy ra.

Mục tiêu của kiểm thử độ phục hồi sau lỗi:

- chỉ ra rằng các chức năng phục hồi không làm việc đúng.
- chỉ ra rằng hệ thống không thỏa sự thỏa thuận về thời gian trung bình để phục hồi sau lỗi (MTTR).

14. Kiểm thử tài liệu (Documentation Test)

Kiểm thử hệ thống cũng có liên quan đến độ chính xác của tài liệu dành cho người dùng.

Cách chính yếu để thực hiện điều này là dùng tài liệu để xác định các testcase hệ thống có độ ưu tiên cao.

Tài liệu dành cho người dùng nên là chủ đề của 1 hoạt động thanh tra (tương tự như khái niệm thanh tra mã nguồn), hãy kiểm tra nó để biết được độ chính xác và tính trong sáng.

15. Kiểm thử thủ tục (Procedure Test)

Nhiều phần mềm là thành phần của hệ thống lớn hơn nhưng chưa được tự động hóa hoàn toàn liên quan đến nhiều thủ tục mà con người cần thực hiện.

Bất kỳ thủ tục của con người nào được kê ra, như thủ tục dành cho người quản trị hệ thống, quản trị database, người dùng đầu cuối nên được kiểm thử trong suốt hoạt động kiểm thử hệ thống.

9.4 Kiểm thử độ chấp nhận của user (Acceptance)

là qui trình so sánh chương trình thực tế với các yêu cầu ban đầu của nó và với các nhu cầu hiện hành của người dùng đầu cuối.

Thường được thực hiện bởi khách hàng hay người dùng đầu cuối và thường không được coi như là 1 trách nhiệm của tổ chức phát triển phần mềm.

Trong trường hợp chương trình làm theo hợp đồng, bên đặt hàng thực hiện kiểm thử độ chấp nhận bằng cách so sánh hoạt động của chương trình với các điều khoản trong hợp đồng.

Trong trường hợp chương trình thương mại như HĐH, trình biên dịch, hệ quản trị CSDL, khách hàng nhạy cảm sẽ thực hiện kiểm thử độ chấp nhận để xác định xem sản phẩm có thỏa mãn các yêu cầu của họ không ?

9.5 Kiểm thử việc cài đặt (Installation)

Mục đích của kiểm thử việc cài đặt không phải là tìm lỗi của phần mềm mà là tìm lỗi xảy ra trong quá trình cài đặt phần mềm. Hiện nay, hầu hết các chương trình đều có chương trình cài đặt kèm theo.

Có nhiều sự kiện xảy ra trong quá trình cài đặt hệ thống phần mềm :

- Người dùng phải chọn 1 trong nhiều options.
- Các file và thư viện phải được phân phối và tải về.
- Các cấu hình phần cứng hợp lệ phải có sẵn.
- Chương trình có thể cần nối mạng để giao tiếp với các phần mềm trên các máy khác.

Các testcase có thể kiểm tra để đảm bảo rằng :

- 1 tập các option tương thích nhau đã được chọn.
- tất cả các thành phần của hệ thống phần mềm đã có sẵn.
- tất cả các file đã được tạo ra và có nội dung cần thiết.
- Cấu hình phần cứng phù hợp.

Nên được phát triển bởi đơn vị tạo hệ thống phần mềm, được phân phối như là 1 thành phần của hệ thống phần mềm và chạy sau khi hệ thống được cài đặt.

9.6 Kiểm thử hồi qui (Regression)

Kiểm thử hồi qui là chạy lại các testcase đã có trên TPPM đã được hiệu chỉnh nâng cấp để đảm bảo rằng những thay đổi của TPPM không có ảnh hưởng lề nào, rằng TPPM vẫn đáp ứng tốt với những testcase trước đây.

Thường được bắt đầu khi code mới đang viết và code hiện hành đang được cập nhật.

Cũng được thực hiện từ version này sang version khác phần mềm.

Qui trình lý tưởng là tạo 1 tập testcase bao quát và chạy nó sau mỗi lần có thay đổi phần mềm.

Các kỹ thuật chọn kiểm thử hồi qui :

- Hiệu suất không tiên lượng

- Các giả định về process không tương thích
- Các mô hình đánh giá không thích hợp
- Có thể được tự động hóa

9.6 Kết chương

Chương này đã giới thiệu lý do cần nhiều hoạt động kiểm thử khác nhau, trong đó kiểm thử đơn vị mà ta đã giới thiệu trong các chương trước chỉ là 1 hoạt động kiểm thử đầu tiên.

Chúng ta cũng đã giới thiệu các vấn đề cơ bản liên quan đến các hoạt động kiểm thử khác như kiểm thử chức năng của chương trình, kiểm thử các yêu cầu không chức năng, kiểm thử độ chấp thuận của người dùng, kiểm thử việc cài đặt phần mềm, kiểm thử hồi qui.