

Chương 3

KỸ THUẬT KIỂM THỬ PHẦN MỀM

3.1. Tổng quan về kỹ thuật kiểm thử phần mềm

3.1.1. Khái niệm kiểm thử

Như đã đề cập, Kiểm thử phần mềm là hoạt động khảo sát thực tiễn sản phẩm hay dịch vụ phần mềm trong đúng môi trường chúng dự định sẽ được triển khai. Nhằm cung cấp cho người có lợi ích liên quan những thông tin về chất lượng của sản phẩm hay dịch vụ phần mềm đó. Mục đích của kiểm thử phần mềm là tìm ra các lỗi cũng như các khiếm khuyết của phần mềm nhằm đảm bảo hiệu quả hoạt động tối ưu của phần mềm trong nhiều ngành khác nhau [5].

Có thể định nghĩa kiểm thử phần mềm một cách dễ hiểu như sau: kiểm thử phần mềm là một tiến trình hay một tập hợp các tiến trình được thiết kế để đảm bảo mã hóa máy tính thực hiện theo cái mà chúng đã được thiết kế để làm, và không thực hiện những cái không mong muốn. Đây là một hoạt động quan trọng trong quá trình phát triển phần mềm, giúp cho người xây dựng phần mềm cũng như khách hàng biết được phần mềm đã đáp ứng các yêu cầu đặt ra hay chưa? Đồng thời đây cũng là tiến trình nhằm phát hiện ra lỗi bằng việc xem xét lại các đặc tả, thiết kế và mã hóa, để nâng cao khả năng hạn chế và kiểm soát lỗi xảy ra trong quá trình phát triển phần mềm ngay từ những giai đoạn đầu.

3.1.2. Nguyên tắc kiểm thử

Để hoạt động kiểm thử đạt kết quả tốt khi kiểm thử cần tuân theo các nguyên tắc sau:

- Một phần quan trọng của ca kiểm thử là xác định đầu ra hoặc kết quả mong muốn.
- Lập trình viên nên tránh tự kiểm tra chương trình của chính mình.
- Nhóm lập trình viên không nên kiểm thử chương trình của chính họ. Nên để một nhóm độc lập thực hiện kiểm thử.
- Kiểm tra kỹ lưỡng kết quả của mỗi kiểm thử.

- Kiểm tra một chương trình để xem liệu chương trình có thực hiện cái mà nó cần thực hiện chỉ là một phần, phần còn lại là xem liệu chương trình có thực hiện cái mà nó không cần phải thực hiện hay không.

- Tránh các ca kiểm thử không rõ ràng trừ khi chương trình thực sự là một chương trình không rõ ràng.

- Không dự đoán kết quả của kiểm thử theo giả thiết ngầm là không tìm thấy lỗi.

- Xác suất tồn tại của các lỗi hơn nữa trong một đơn vị phần mềm tỉ lệ với số lỗi đã tìm thấy trong đoạn đó.

- Kiểm thử là một nhiệm vụ cực kì sáng tạo và có tính đầy thử thách.

3.1.3. Vai trò

Vì kiểm thử phần mềm thường chiếm 40% tổng công sức phát triển và $\geq 30\%$ tổng thời gian phát triển. Đối với các phần mềm có ảnh hưởng tới sinh mạng, chi phí của hoạt động này có thể gấp từ 3 đến 5 lần tổng các chi phí khác cộng lại. Chính vì vậy nếu hoạt động kiểm thử phần mềm được thực hiện tốt thì:

- Giảm chi phí phát triển phần mềm
- Tăng độ tin cậy của sản phẩm phần mềm.

3.1.4. Những khó khăn khi kiểm thử

Nâng cao chất lượng phần mềm nhưng không vượt quá chất lượng thiết kế. Chỉ phát hiện các lỗi tiềm tàng trong sản phẩm để sửa chữa và khắc phục chúng.

Dễ bị ảnh hưởng vấn đề tâm lí khi thực hiện quá trình kiểm thử. Vì người kiểm thử đồng thời cũng là người xây dựng phần mềm.

Khi kiểm thử chương trình ta không thể tạo ra tất cả các trường hợp kiểm thử vì đây là một số lượng vô hạn. Vì vậy khó đảm bảo tính đầy đủ cho quá trình kiểm thử.

Quá trình phát hiện lỗi bị hạn chế do hoạt động kiểm thử được thực hiện bằng thủ công là chủ yếu.

Quá trình kiểm thử sẽ khó khăn hơn khi chương trình có bộ nhớ như hệ điều hành hoặc các ứng dụng cơ sở dữ liệu. Ví dụ trong cơ sở dữ liệu ứng dụng của hệ thống đặt vé của hãng hàng không, việc thực hiện một giao dịch phụ thuộc vào những gì đã xảy ra trong các giao dịch

trước đó. Do đó, khi kiểm thử ta phải kiểm thử tất cả các trình tự của giao dịch. Chính vì vậy quá trình kiểm thử sẽ gặp nhiều khó khăn hơn.

3.1.5. Mục tiêu của hoạt động kiểm thử

Quá trình kiểm thử phần mềm gồm 2 mục tiêu cơ bản sau:

- Chứng minh cho người phát triển cũng như khách hàng thấy được các yêu cầu của phần mềm. Nghĩa là thử nghiệm tất cả các đặc tính của hệ thống được kết hợp trong các sản phẩm.
- Phát hiện các lỗi và khiếm khuyết trong phần mềm. Chính nhờ hoạt động này mà ta tìm ra được các thực hiện không như mong đợi của hệ thống như các tính toán sai, sai lệch dữ liệu .vv..

Tóm lại, mục tiêu của kiểm thử phần mềm là thuyết phục cho người phát triển phần mềm và khách hàng thấy rằng phần mềm là đủ tốt cho các thao tác sử dụng.

3.2. Các chiến lược kiểm thử

3.2.1. Kiểm thử đơn vị

3.2.1.1. Khái niệm

Kiểm thử đơn vị là quá trình kiểm thử các thành phần riêng biệt của hệ thống, như các hàm, thủ tục, phương thức hay các lớp. Kiểm thử đơn vị được thực hiện bởi người phát triển phần mềm, chứ không phải người kiểm thử phần mềm hay người sử dụng cuối.

3.2.1.2. Mục đích

Mục đích của kiểm thử đơn vị là so sánh các chức năng của một đơn vị với các đặc tả chức năng hoặc giao diện xác định các đơn vị. Mục tiêu ở đây không phải là để cho thấy rằng đơn vị đáp ứng được các đặc tả của nó, mà để thấy rằng đơn vị mâu thuẫn với đặc tả đó. Nhằm đảm bảo các thông tin được xử lý và xuất ra là chính xác trong mối tương quan giữa dữ liệu nhập vào và chức năng của đơn vị. Điều này thường đòi hỏi tất cả các nhánh bên trong đơn vị đều phải được kiểm tra để phát hiện các nhánh phát sinh lỗi. Một nhánh thường là một chuỗi các lệnh được thực thi trong một đơn vị.

3.2.1.3. Nội dung kiểm thử

• **Kiểm thử giao diện:** kiểm thử các dòng dữ liệu qua một giao diện đơn vị liên quan đến số lượng và định dạng của các biến và các đơn vị sử dụng trên giao diện. Các đặc trưng qua giao diện:

- Đơn vị của đối được truyền để gọi đơn vị có trùng với đơn vị tham số của đơn vị không?
- Số đối được truyền gọi đơn vị có bằng với số các tham số đầu vào của đơn vị không?

• **Kiểm thử cấu trúc dữ liệu cục bộ:** loại kiểm thử này được sử dụng để phát hiện ra các lỗi như:

- Kiểu dữ liệu không nhất quán
- Tên các biến không đúng
- Đánh máy không đúng hoặc không nhất quán

• **Kiểm thử các phép xử lý:** nhằm phát hiện các lỗi trong các phép xử lý như:

- Thứ tự ưu tiên của các phép tính số học không đúng
- Độ chính xác của kết quả bị sai lệch

• **Kiểm thử các điều kiện logic:** Phát hiện được các lỗi như:

- So sánh các kiểu dữ liệu khác nhau
- Sự ưu tiên của các toán tử logic không đúng đắn

• **Kiểm thử dòng điều khiển/ biên:** phát hiện được các lỗi như:

- Vòng lặp không kết thúc hoặc kết thúc không chính xác
- Các biến lặp bị cải biên không chính xác

• **Kiểm thử liên quan đến giá trị vào/ ra:** loại kiểm thử này tập trung vào việc tìm ra các điểm sai liên quan đến giá trị vào ra như:

- Các câu lệnh OPEN/ CLOSE có đúng không?
- Các file có được mở trước khi sử dụng không?

3.2.1.4. Các kỹ thuật được sử dụng trong kiểm thử đơn vị

- Kiểm thử đường thi hành cơ bản
- Kiểm thử giá trị biên
- Kiểm thử vòng lặp

3.2.2. Kiểm thử tích hợp

3.2.2.1. Khái niệm

Kiểm thử tích hợp kết hợp các thành phần của một ứng dụng và kiểm thử như một ứng dụng đã hoàn thành. Trong khi kiểm thử đơn vị kiểm tra các thành phần và đơn vị riêng lẻ thì kiểm thử tích hợp kết hợp chúng lại với nhau và kiểm tra sự giao tiếp giữa chúng.

Người thực hiện kiểm thử có thể là người lập trình, người thiết kế.

Các lỗi thường gặp khi thực hiện kiểm thử tích hợp

- Sự kết hợp các chức năng phụ có thể không sinh ra chức năng chính mong muốn.
- Dữ liệu bị mất khi đi qua một giao diện.
- Một modul lỗi có thể gây ra lỗi cho các modul khác.

3.2.2.2. Mục tiêu

Hai mục tiêu chính của kiểm thử tích hợp:

- Phát hiện lỗi giao tiếp xảy ra giữa các đơn vị.
- Tích hợp các đơn vị đơn lẻ thành các hệ thống nhỏ và cuối cùng là nguyên hệ thống hoàn chỉnh chuẩn bị cho kiểm thử ở mức hệ thống.

Trừ một số ít ngoại lệ, kiểm thử tích hợp chỉ nên thực hiện trên những đơn vị đã được kiểm tra cẩn thận trước đó bằng kiểm thử đơn vị, và tất cả các lỗi mức đơn vị đã được sửa chữa. Một số người hiểu sai rằng đơn vị một khi đã qua giai đoạn kiểm thử đơn vị với các giao tiếp giả lập thì không cần phải thực hiện kiểm thử tích hợp nữa. Thực tế việc tích hợp giữa các đơn vị dẫn đến những tình huống hoàn toàn khác. Một chiến lược cần quan tâm trong kiểm thử tích hợp là nên tích hợp dần từng đơn vị. Một đơn vị tại một thời điểm được tích hợp vào một nhóm các đơn vị khác đã tích hợp trước đó và đã hoàn tất các đợt kiểm thử tích hợp trước đó. Lúc này, ta chỉ cần

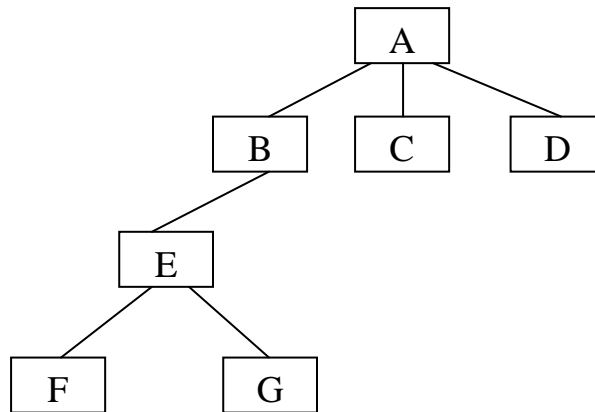
kiểm thử giao tiếp của đơn vị mới thêm vào với hệ thống các đơn vị đã tích hợp trước đó, điều này sẽ làm cho số lượng ca kiểm thử giảm đi rất nhiều, và sai sót sẽ giảm đáng kể.

3.2.2.3. Các phương pháp áp dụng cho kiểm thử tích hợp

1. Phương pháp tích hợp trên xuống

Là quá trình kiểm thử được thực hiện bằng cách tăng dần việc xây dựng cấu trúc chương trình. Các modul được tích hợp bằng cách đi dần xuống qua cấp bậc điều khiển, bắt đầu với modul điều khiển chính. Sau đó các modul phụ thuộc vào modul điều khiển chính sẽ được tổ hợp dần vào trong cấu trúc theo chiều sâu trước hoặc chiều rộng trước.

Ví dụ:



Hình 1. Sơ đồ cấu trúc điều khiển

Ở sơ đồ cấu trúc điều khiển trên nếu việc tích hợp theo chiều sâu được thực hiện trước thì ta sẽ tích hợp tất cả các modul trên đường điều khiển chính của cấu trúc. Việc chọn đường điều khiển chính này có thể tùy ý và phụ thuộc vào các đặc trưng của ứng dụng. Ví dụ nếu chọn theo tay trái thì ở đây các modul A, B, E, F sẽ được tích hợp trước. Sau đó các đường còn lại sẽ được tích hợp dần vào. Ngược lại nếu việc tích hợp theo chiều rộng được thực hiện trước thì ta sẽ thực hiện đi theo cấu trúc ngang. Ở ví dụ trên các modul B, C, D sẽ được tích hợp trước tiên, sau đó các modul còn lại sẽ được tích hợp dần vào theo quy tắc trên.

- Tiến trình tích hợp trên xuống bao gồm các bước sau:
 1. Modul điều khiển chính được dùng như bộ lái kiểm thử và tất cả các modul phụ trợ được thay thế bởi các cuống.
 2. Thay thế dần từng cuống bởi modul thực thi tương ứng.

3. Sau khi tích hợp modul đó tiến hành các kiểm thử tương ứng.
4. Sau khi hoàn thành đủ tập các kiểm thử này thì thay một cuống khác bằng modul thực(nghĩa là quay lại bước 2)
5. Có thể kiểm thử lại để đảm bảo rằng không có sai sót mới nào được sinh ra.
6. Tiếp tục lặp lại bước 2 cho tới khi toàn bộ cấu trúc chương trình được xây dựng xong.

- **Ưu điểm:**

- Phát hiện sớm các lỗi thiết kế: chính nhờ ưu điểm này nên khi phát hiện lỗi thì quá trình sửa lỗi sẽ được thực hiện dễ dàng hơn, đồng thời cũng làm giảm giá thành sửa đổi.
- Có phiên bản hoạt động sớm .

- **Nhược điểm:**

- Việc mô phỏng các modul cấp thấp khó thực hiện.
- Không kiểm thử đầy đủ các chức năng.

2. Phương pháp tích hợp dưới lên

Là quá trình kiểm thử được bắt đầu xây dựng và kiểm thử với modul ở mức thấp nhất trong cấu trúc chương trình. Lúc này vì các modul được tích hợp từ dưới lên trên nên việc xử lý yêu cầu đối với các modul phụ thuộc của một mức nào đó bao giờ cũng có sẵn.

- Tiến trình tích hợp dưới lên bao gồm các bước sau:

1. Các modul ở mức thấp được tổ hợp vào trong các cụm thực hiện cho một chức năng con phần mềm đặc biệt.
2. Một bộ lái được viết để phối hợp đầu vào và đầu ra của ca kiểm thử.
3. Kiểm thử cụm đó.
4. Loại bỏ các driver và các cụm tổ hợp ngược lên trong cấu trúc chương trình.

- **Ưu điểm**

- Có thể tiến hành ca kiểm thử dễ và không cần cuống.

- **Nhược điểm**

- Luôn chưa có 1 chương trình chỉnh thể cho đến khi modul cuối cùng được thêm vào.

3. Phương pháp tích hợp hồi quy

Là quá trình kiểm thử lại các phép thử đã thành công mỗi khi tích hợp thêm modul hoặc khi cập nhật mã nguồn chương trình.

Khi chúng ta tích hợp thêm modul vào hệ thống hoặc khi tiến hành nâng cấp chương trình thì sẽ tạo ra một số tổ hợp trạng thái mới. Các trạng thái mới này sẽ gây ra các vấn đề như:

- Xuất hiện lỗi ở modul mà trước đó nó chưa gây ra lỗi.
- Khắc phục một lỗi mới có thể làm ảnh hưởng tới lỗi mà chúng ta đã sửa trước đó.
- Sinh ra lỗi mới mà trước đây chưa có.

Chính vì những lí do trên mà kiểm thử hồi quy là một quá trình tốn kém và không khả thi nếu không có sự hỗ trợ tự động.

3.2.3 Kiểm thử hệ thống

3.2.3.1. Khái niệm

Một hệ thống bao gồm hai hoặc nhiều thành phần tích hợp với nhau nhằm thực hiện các chức năng hoặc đặc tính của hệ thống. Sau khi tích hợp các thành phần tạo nên hệ thống thành công thì quá trình kiểm thử hệ thống được tiến hành. Kỹ thuật kiểm thử này là kỹ thuật dễ gây hiểu nhầm và khó khăn nhất. Nó không phải là quá trình kiểm thử các chức năng của một hệ thống hay chương trình hoàn chỉnh mà là mức độ kiểm thử toàn bộ các chức năng của hệ thống phần mềm. Bao gồm tất cả các thành phần tương tác với nhau, và hoạt động trong môi trường giống như môi trường thực tế (hệ điều hành, cơ sở dữ liệu, kết nối mạng, khả năng tương thích với các phần mềm khác, .vv..). Bên cạnh đó, kiểm thử hệ thống cũng chú ý đến vấn đề bảo mật, thân thiện, khả năng đáp ứng, tốc độ thực hiện của hệ thống phần mềm.

3.2.3.2. Các đặc điểm cơ bản

Thông thường loại hình kiểm thử này thường tốn nhiều thời gian và công sức. Trong nhiều trường hợp, việc kiểm thử đòi hỏi một số thiết bị phụ trợ, phần mềm hoặc phần cứng đặc

thù. Ở mức độ hệ thống, người kiểm thử cũng tìm kiếm các lỗi, nhưng trọng tâm là đánh giá về hoạt động, thao tác, sự tin cậy và các yêu cầu khác liên quan đến chất lượng của toàn hệ thống.

Nhiều khi kiểm thử hệ thống được xem như kiểm thử tích hợp. Nhưng thực tế thì chúng khác nhau điểm cơ bản đó là kiểm thử hệ thống chú trọng vào các hành vi và lỗi trên toàn bộ hệ thống, còn kiểm thử tích hợp chủ yếu chú trọng tới sự giao tiếp giữa các đối tượng khi chúng làm việc với nhau. Thông thường ta phải thực hiện kiểm thử đơn vị và kiểm thử tích hợp trước để bảo đảm mọi đơn vị và sự tương tác giữa chúng hoạt động chính xác trước khi thực hiện kiểm thử hệ thống.

Sau khi hoàn thành quá trình kiểm thử tích hợp thì một hệ thống phần mềm đã được hình thành cùng với các thành phần đã được kiểm tra đầy đủ. Tại thời điểm này lập trình viên hoặc kiểm thử viên bắt đầu kiểm thử phần mềm như một hệ thống hoàn chỉnh. Việc lập kế hoạch cho quá trình kiểm thử hệ thống nên bắt đầu thực hiện từ giai đoạn hình thành và phân tích các yêu cầu của phần mềm.

Kiểm thử hệ thống là quá trình kiểm thử cả các hành vi chức năng của phần mềm lẫn các yêu cầu về chất lượng như độ tin cậy, tính tiện lợi khi sử dụng, hiệu năng và bảo mật. Mức kiểm thử này đặc biệt thích hợp cho việc phát hiện lỗi giao tiếp với phần mềm hoặc phần cứng bên ngoài, chẳng hạn các lỗi "tắc nghẽn" hoặc chiếm dụng bộ nhớ.

Đây là kĩ thuật kiểm thử đòi hỏi nhiều thời gian, công sức và tính chính xác. Nó thường được thực hiện bởi một nhóm kiểm thử viên hoàn toàn độc lập với nhóm phát triển dự án.

3.2.3.3. Các giai đoạn thực hiện trong kiểm thử hệ thống

Với hầu hết các hệ thống phức tạp quá trình kiểm thử hệ thống thường gồm 2 giai đoạn sau:

- **Kiểm thử tích hợp:** Đội kiểm thử nhận mã nguồn của hệ thống. Khi một vấn đề được phát hiện thì những người tích hợp sẽ tìm nguồn gốc của vấn đề và nhận biết các thành phần cần phải gỡ lỗi. Kiểm thử tích hợp hầu như liên quan với việc phát hiện các khiếm khuyết của hệ thống.
- **Kiểm thử phát hành:** Một phiên bản của hệ thống có thể được phát hành tới người dùng để kiểm thử. Đội kiểm thử tập trung vào việc hợp lệ các yêu cầu của hệ thống và đảm bảo

tính tin cậy của hệ thống. Kiểm thử phát hành thường tập trung vào mô tả các đặc tính hệ thống có thể làm được hoặc không làm được. Sau đó các vấn đề được phát hiện sẽ được báo cáo cho đội phát triển để gỡ lỗi chương trình. Nếu hệ thống phát hành đủ tốt thì khách hàng có thể chấp nhận nó để sử dụng.

3.2.3.4. Mục đích

Kiểm thử các thiết kế và toàn bộ hệ thống sau khi đã được tích hợp có thỏa mãn các yêu cầu đã đặt ra hay không.

- Kiểm thử chức năng:
 - Các chức năng mức hệ thống bao gồm các chức năng giao diện, các chức năng mức người dùng hay đầu ra cuối cùng khỏi hệ thống.
 - Các chức năng này thường mang tính tích hợp nên khi phát hiện lỗi thì phải quay lại kiểm thử từng phần trước nó.
 - Các giao diện được xem như là điểm phân định giữa các phần để kiểm thử.
 - Đây là kỹ thuật kiểm thử nhằm đảm bảo các hành vi của hệ thống thỏa mãn đúng yêu cầu thiết kế.
- Kiểm thử phục hồi:
 - Đây là kỹ thuật bắt phần mềm phải thất bại để xem khả năng phục hồi của nó đến đâu.
 - Độ tin cậy là một độ đo đánh giá khả năng phục hồi của hệ thống.
 - Có 2 cách phục hồi:
 1. Phục hồi tự động: bằng khởi động lại, sau khi phục hồi dữ liệu hệ thống tự tiếp tục thì được đánh giá là đúng đắn.
 2. Phục hồi có sự can thiệp của con người: Lúc này cần đánh giá thời gian trung bình để sửa chữa trong thời gian cho phép hay không?

Tóm lại đây là kỹ thuật kiểm thử nhằm đảm bảo hệ thống có khả năng khôi phục trạng thái ổn định trước đó trong tình huống mất tài nguyên hoặc mất dữ liệu.

- Kiểm thử an ninh:

- Là kĩ thuật kiểm tra mọi cơ chế bảo vệ được xây dựng trong hệ thống xem có được đạt hiệu quả đề ra trước các đột nhập hay không? Nhằm bảo đảm tính toàn vẹn, bảo mật của dữ liệu và của hệ thống.
- Khi thực hiện kĩ thuật này người kiểm thử sẽ đóng vai trò của kẻ đột nhập.
- Bài toán thiết kế hệ thống an ninh đặt ra là:
 1. Làm cho việc đột nhập tốn phí nhiều hơn giá trị thu được do đột nhập.
 2. Công sức bỏ ra xây dựng công cụ bảo vệ phải ít hơn giá trị mất đi nếu bị đột nhập.
- Kiểm thử thi hành:
 - Kĩ thuật này được sử dụng để kiểm thử việc vận hành của phần mềm khi hệ thống được tích hợp.
 - Xuất hiện trong mọi bước của quá trình kiểm thử, tuy nhiên chỉ khi tất cả các phần tử của hệ thống đã được tích hợp thì kiểm thử mới thực sự là chắc chắn.
 - Kĩ thuật này có thể làm bộc lộ các tình thế dẫn đến sự suy giảm hiệu năng hoặc thất bại hệ thống tiềm ẩn.
 - Thường gắn liền với kiểm thử áp lực vì cả hai thường đòi hỏi các thiết bị phần cứng và phần mềm chuyên dụng.
- Kiểm thử chịu tải:
 - Là kĩ thuật vận hành hệ thống khi sử dụng nguồn lực với số lượng, tần suất và cường độ dị thường.
 - Nó tập trung vào các trạng thái tới hạn, các điểm chết, cũng như các tình huống bất thường như đang giao dịch thì ngắt kết nối .vv..
 - Kĩ thuật này có thể làm bộc lộ các sự kiện mà có thể gây ra việc xử lí không ổn định hoặc không chính xác.

Tóm lại, tất cả các kĩ thuật kiểm thử trên đều đóng vai trò quan trọng trong quá trình kiểm thử phần mềm. Vì chúng đảm bảo cho hệ thống đủ khả năng làm việc trong môi trường thực. Tuy nhiên không nhất thiết phải thực hiện đầy đủ tất cả các loại kiểm thử trên. Tùy vào yêu cầu và đặc trưng của từng hệ thống, tùy khả năng và thời gian cho phép của dự án mà khi lập kế hoạch người quản lí dự án sẽ quyết định áp dụng những loại kiểm thử nào.