Chapter 4: Software Testing

**Giới thiệu**

Kiểm thử phần mềm bao gồm việc kiểm chứng động (dynamic), đó là một chương trình cung cấp các hành vi dự kiến trên một tập hữu hạn các trường hợp thử nghiệm, phù hợp được lựa chọn từ các miền thực hiện thường là vô hạn.

Các vấn đề quan trọng trong việc mô tả các kiến thức kiểm thử phần mềm (KA):

- Dynamic: Thuật ngữ này có nghĩa là kiểm thử luôn luôn bao gồm thực thi chương trình khi lựa chọn đầu vào (input). Để được chính xác, giá trị đầu vào đơn (alone) không phải lúc nào cũng đủ để xác định 1 bài test, vì một hệ thống không xác định phức tạp có thể phản ứng với các đầu vào cùng với các hành vi khác nhau, tùy thuộc vào trạng thái hệ thống. Tuy nhiên trong KA này, thuật ngữ đầu vào sẽ phải được duy trì, với quy ước bao hàm rằng ý nghĩa của nó cũng bao gồm một trạng thái đầu vào quy định trong những trường hợp quan trọng. Các kỹ thuật tĩnh khác nhau và bổ trợ cho kiểm thử động. Các kỹ thuật tĩnh được bao gồm trong Software Quality KA. Điều đáng chú ý là thuật ngữ không thống nhất giữa các cộng đồng khác nhau và một số sử dụng thuật ngữ “kiểm thử” trong mối liên hệ với các kỹ thuật tĩnh.

- Finite: Ngay cả trong các chương trình đơn giản, rất nhiều trường hợp kiểm thử về mặt lý thuyết được kiểm tra toàn diện có thể yêu cầu nhiều tháng hoặc nhiều năm để thực hiện. Đó là lý do tại sao trong thực tế, một tập đầy đủ các kiểm thử nhìn chung có thể là vô hạn, và kiểm thử được tiến hành trên một tập hợp tất cả các kiểm thử có khả thi, có thể được xác định theo các tiêu chí rủi ro và ưu tiên. Kiểm thử luôn bao hàm sự cân bằng giữa các tài nguyên hạn chế và tiến độ trên một mặt và yêu cầu kiểm thử vốn dĩ không giới hạn trên các mặt khác.

- Selected: Nhiều đề xuất kiểm thử khác nhau về cơ bản là cách một tập kiểm thử được lựa chọn, và các kỹ sư phần mềm phải được nhận thức rằng các tiêu chí lựa chọn khác nhau có thể mang lại mức độ khác nhau về tính hiệu quả. Làm thế nào để xác định các tiêu chí lựa chọn phù hợp nhất trong điều kiện nhất định là một vấn đề phức tạp; trong thực tế, các kỹ thuật phân tích rủi ro và chuyên môn công nghệ phần mềm được áp dụng.

- Expected: Phải có khả năng, mặc dù không phải lúc nào cũng dễ dàng, để quyết định xem các kết quả quan sát của chương trình thử nghiệm được chấp nhận hay không; nếu không, các nỗ lực thử nghiệm là vô ích. Các hành vi quan sát có thể được so sánh với nhu cầu của người sử dụng (thường được gọi là thử nghiệm để xác nhận), chống lại một đặc điểm kỹ thuật (thử nghiệm để xác minh), hoặc, có lẽ, đối với các hành vi được mong đợi từ các yêu cầu bao hàm hoặc mong đợi (xem thử nghiệm Chấp nhận trong các yêu cầu phần mềm KA ).

Trong những năm gần đây, kiểm thử không còn được coi là một hoạt động mà chỉ bắt đầu sau khi giai đoạn coding được hoàn thành với mục đích hạn chế của việc phát hiện lỗi. Kiểm thử phần mềm trở nên phổ biến trong suốt quá trình phát triển và bảo trì. Kế hoạch kiểm thử phần mềm nên bắt đầu với giai đoạn đầu của quy trình yêu cầu phần mềm, các kế hoạch và quy trình thử nghiệm phải được phát triển một cách có hệ thống và liên tục - và có thể tinh chế - như tiến hành phát triển phần mềm. Những hoạt động lập kế hoạch kiểm thử và thử nghiệm thiết kế cung cấp các đầu vào hữu ích cho các nhà thiết kế phần mềm và giúp họ làm nổi bật những khuyết điểm, thiết sót.

Đối với nhiều tổ chức, phương pháp tiếp cận đến chất lượng phần mềm là một trong những sự ngăn chặn: tức là ngăn chặn vấn đề tốt hơn là sửa chữa chúng. Kiểm thử có thể được nhìn thấy, sau đó, được dùng như một phương tiện để cung cấp thông tin về các tính năng và chất lượng các thuộc tính của phần mềm và cũng để xác định lỗi trong những trường hợp ngăn chặn lỗi không có hiệu quả. Sự thật hiển nhiên là phần mềm có thể chứa lỗi, thậm chí sau khi hoàn thành việc kiểm thử bao quát. Các lỗi phần mềm có sau đó sẽ được giải quyết bằng bảo trì sửa chữa. Mục bảo trì phần mềm có trong phần Software Mainteance KA (chương 5).

Trong Kỹ thuật quản lý chất lượng phần mềm, được phân loại thành 2 kỹ thuật đáng chú ý là Tĩnh (không thực thi mã) và động (thực thi mã). KA này tập trung và các kỹ thuật động. Kiểm thử phần mềm cũng có liên quan đến xây dựng phần mềm.

**BREAKDOWN OF TOPICS FOR SOFTWARE TESTING**

**1. Software Testing Fundamentals (kiểm thử phần mềm cơ bản)**

1.1. Thuật ngữ liên quan đến kiểm thử

1.1.1. Định nghĩa về kiểm thử và thuật ngữ liên quan

*1.1.2. Faults vs. Failures (Lỗi và chịu lỗi)*

1.2. Key Issues (Các vấn đề chính)

1.2.1. Tiêu chí lựa chọn kiểm thử/Kiểm tra mức độ đầy đủ tiêu chí

Tiêu chí lựa chọn thử nghiệm có nghĩa là lựa chọn trường hợp kiểm thử hoặc xác định một tập các trường hợp kiểm thử là đủ cho một mục đích cụ thể.

1.2.2. Hiệu quả kiểm thử/Mục tiêu kiểm thử

Hiệu quả kiểm thử được xác định bằng cách phân tích một tập hợp các chương trình thực thi. Lựa chọn kiểm thử được thực hiện có thể được hướng dẫn bởi các mục tiêu khác nhau.

1.2.3. Kiểm thử phát hiện khiếm khuyết

Trong kiểm thử để phát hiện khiếm khuyết, một thử nghiệm thành công là một trong những nguyên nhân dẫn đến hệ thống thất bại.

1.2.4. The Oracle Problem

Oracle là bất kỳ người hoặc máy móc dùng để quyết định xem liệu một chương trình có thực hiện một cách chính xác trong một thử nghiệm nhất định và phù hợp kết quả là đạt hoặc thất bại.

1.2.5. Giới hạn lý thuyết và thực tiễn của kiểm thử

1.2.6. Vấn đề về đường dẫn không khả thi

Đường dẫn không khả thi là các đường dẫn luồng điều khiển không thể được thực thi bởi bất kỳ dữ liệu đầu vào.

1.2.7. Khả năng kiểm thử

Thuật ngữ "khả năng kiểm thử phần mềm" có 2 nghĩa: Một mặt nó đề cập đến sự dễ dàng mà một tiêu chuẩn thử nghiệm nhất định có thể được thỏa mãn; Mặt khác, nó được định nghĩa là khả năng một tập các trường hợp kiểm thử sẽ làm phơi bày lỗi nếu phần mềm bị lỗi.

1.3. Mối quan hệ của kiểm thử với các hoạt động khác

- Kiểm thử vs kỹ thuật quản lý chất lượng phần mềm tĩnh

- Kiểm thử vs Bằng chứng về tính đúng đắn và xác minh hình thức

- Kiểm thử vs Gỡ lỗi

- Kiểm thử vs Xây dựng chương trình

2. Test Levels

Kiểm thử phần mềm thường được thực hiện ở các cấp độ khác nhau trong suốt quá trình phát triển và bảo trì. Mức có thể được phân biệt dựa trên các đối tượng thử nghiệm, được gọi là các target, hoặc về mục đích, được gọi là các objective (từ cấp thử nghiệm).

2.1. Mục tiêu của kiểm thử (Target of the Test)

Mục tiêu của thử nghiệm có thể khác nhau: một mô-đun duy nhất, một nhóm các mô-đun như (liên quan theo mục đích, sử dụng, hành vi, hoặc cơ cấu), hay toàn hệ thống. Ba giai đoạn thử nghiệm có thể được phân biệt: đơn vị, tích hợp và hệ thống.

2.1.1. Kiểm thử đơn vị

Đơn vị kiểm thử xác minh các chức năng một cách riêng biệt của các yếu tố phần mềm mà có thể kiểm chứng một cách riêng biệt. Tùy thuộc vào hoàn cảnh, đây có thể là các chương trình con riêng lẻ hoặc một bộ phận lớn các đơn vị có liên kết chặt chẽ.

2.1.2. Kiểm thử tích hợp

Kiểm thử tích hợp là quá trình xác minh sự tương tác giữa các thành phần phần mềm. Thường được sử dụng với các phần mềm có cấu trúc phân cấp. Kiểm thử tích hợp thường diễn ra ở từng giai đoạn của sự phát triển.

2.1.3. Kiểm thử hệ thống

Kiểm thử hệ thống là có liên quan với kiểm thử hành vi của toàn bộ hệ thống. Kiểm thử đơn vị và kiểm thử tích hợp thường đã xác định nhiều lỗi phần mềm. Kiểm thử hệ thống thường được coi là thích hợp cho việc đánh giá phi chức năng hệ thống yêu cầu - chẳng hạn như bảo mật, tốc độ, độ chính xác và đột in cậy. Giao diện bên ngoài cho các ứng dụng khác, tiện ích, các thiết bị phần cứng, hoặc các môi trường hoạt động cũng thường được đánh giá ở mức này.

2.2. Mục đích của kiểm thử

Kiểm thử được tiến hành trong bối cảnh mục tiêu cụ thể, trong đó được ghi nhận nhiều hơn hoặc ít hơn một cách rõ ràng và có độ chính xác khác nhau. Trong đó nêu rõ mục đích của kiểm thử một cách chính xác, hỗ trợ về mặt định lượng đo lường và kiểm soát trong quá trình kiểm thử.

Kiểm thử có thể được dùng để xác minh các tính chất khác nhau. Trường hợp kiểm thử có thể được thiết kế để kiểm tra các thông số kỹ thuật chức năng được thực hiện một cách chính xác, hiệu suất, độ tin cậy, khả dụng...

2.2.1. Nghiệm thu/Năng lực kiểm thử

Nghiệm thu/Năng lực kiểm thử xác định một hệ thống đáp ứng tiêu chuẩn nghiệm thu của nó, thường là bằng cách kiểm tra các hành vi hệ thống mong muốn đối với các yêu cầu của khách hàng.

2.2.2. Cài đặt kiểm thử

Thông thường, sau khi hoàn thành hệ thống và chấp nhận kiểm thử, các phần mềm được xác minh sau khi cài đặt trong môi trường mục tiêu. Cài đặt kiểm thử có thể được xem như là hệ thống thử nghiệm được tiến hành trong môi trường hoạt động cùa cấu hình phần cứng và hoạt động hạn chế khác.

2.2.3. Kiểm thử Alpha và Beta

Trước khi phần mềm được phát hành, nó đôi khi được trao cho một nhóm nhỏ, nhóm người dùng tiềm năng được chọn để sử dụng (alpha test) và một nhóm lớn hơn của bên đại diện (beta test).

2.2.4. Độ tin cậy và đánh giá

Kiểm thử cải thiện độ tin cậy bằng cách xác định và sửa lỗi. Ngoài ra, các biện pháp thống kế về độ tin cậy có thể được bắt nguồn bằng cách tạo ra một cách ngẫu nhiên trường hợp kiểm tra theo hoạt động của phần mềm.

2.2.5. Kiểm thử hồi quy

Kiểm thử hồi quy là kiểm thử lại chọn lọc của một hệ thống hay thành phần để xác minh rằng những sửa đổi không gây ra hiệu ứng ngoài ý muốn và hệ thống hay thành phần vẫn tuân thủ các yêu cầu của quy định.

2.2.6. Kiểm thử hiệu suất

Kiểm thử hiệu suất xác nhận rằng các phần mềm đáp ứng được yêu cầu thực hiện quy định và đánh giá các đặc tính hiệu suất - VD, năng lực và thời gian đáp ứng.

2.2.7. Kiểm thử bảo mật

Kiểm thử bảo mật tập trung vào việc xác minh rằng phần mềm được bảo vệ khỏi các cuộc tấn công từ bên ngoài. Đặc biệt, kiểm thử bảo mật xác minh tính bảo mật, tính toàn vẹn và tính sẵn sàng của hệ thống và dữ liệu của nó.

2.2.8. Kiểm thử Stress

Kiểm thử stress thử nghiệm các bài tập phần mềm thiết kế tải trọng tối đa, hoặc vượt ra ngoài nó, với mục tiêu xác định các giới hạn hành vi.

2.2.9. Kiểm thử Back-to-Back

IEEE/ISO/IEC Standard 24765 định nghĩa kiểm thử back-to-back là "kiểm thử, trong đó hai hoặc nhiều hơn các biến thể của một chương trình được thực hiện với các đầu vào tương tự, các kết quả được so sánh, và các lỗi được phân tích trong trường hợp có sự khác biệt".

2.2.10. Kiểm thử phục hồi

Kiểm thử phục hồi nhằm mục đích kiểm tra khả năng khởi động lại của phần mềm sau khi hệ thống bị treo hoặc crash.

2.2.11. Kiểm thử giao diện

Các nhược điểm giao diện rất phổ biến trong các hệ thống phức tạp. Kiểm thử giao diện nhằm xác minh các thành phần giao diện chính xác để cung cấp những trao đổi chính xác của dữ liệu và kiểm soát thông tin. Thông thường các trường hợp thử nghiệm được tạo ra từ các đặc tả giao diện. Mục tiêu cụ thể của thử nghiệm giao diện là để mô phỏng việc sử dụng các API của các ứng dụng của người dùng cuối. Điều này liên quan đến việc tạo ra các thông số của các cuộc gọi API, thiết lập các điều kiện môi trường bên ngoài, và định nghĩa của dữ liệu nội bộ có ảnh hưởng đến các API.

2.2.12. Kiểm thử cấu hình

Trong trường hợp phần mềm được xây dựng để phục vụ người dùng khác nhau, kiểm thử cấu hình dùng để kiểm chứng các phần mềm theo quy định cấu hình khác nhau.

2.2.13. Kiểm thử tính khả dụng và tương tác người - máy

Nhiệm vụ chính của phần này là đánh giá như thế nào là dễ dàng cho người dùng cuối để tìm hiểu và sử dụng phần mềm. Nói chung, nó có thể liên quan đến việc thử nghiệm các chức năng phần mềm hỗ trợ việc sử dụng, tài liệu hướng dẫn nhằm hỗ trợ người sử dụng, và khả năng của hệ thống để phục hồi từ lỗi người sử dụng