**BÀI BÁO CÁO**

Mức kiểm thử nào lập trình viên thực hiện ?

Lập trình viên thực hiện các mức kiểm thử phần mềm để đảm bảo chất lượng và độ tin cậy của sản phẩm phần mềm. Dưới đây là một số mức kiểm thử chính mà lập trình viên thường thực hiện:

Kiểm thử đơn vị (Unit testing): Là mức kiểm thử tập trung vào việc kiểm tra từng đơn vị (hàm, module, lớp) của mã nguồn để xác định xem chúng hoạt động đúng như mong đợi hay không.

Kiểm thử tích hợp (Integration testing): Là mức kiểm thử nhằm đảm bảo rằng các đơn vị được kết hợp lại cùng nhau hoạt động một cách hợp lý. Kiểm thử tích hợp giúp đảm bảo rằng các liên kết và giao tiếp giữa các đơn vị là chính xác.

Kiểm thử hệ thống (System testing): Là mức kiểm thử nhằm kiểm tra phiên bản hoàn chỉnh của phần mềm để xem nó hoạt động đúng như mong đợi trong môi trường thực tế. Kiểm thử hệ thống kiểm tra toàn bộ hệ thống, bao gồm cả các tính năng, hiệu suất, bảo mật và tương tác với các thành phần khác.

Kiểm thử chấp nhận (Acceptance testing): Là mức kiểm thử nhằm xác nhận rằng phần mềm đáp ứng đầy đủ các yêu cầu và tiêu chuẩn của khách hàng. Kiểm thử chấp nhận thường được thực hiện bởi người dùng hoặc khách hàng để đánh giá liệu phần mềm đã sẵn sàng cho việc triển khai hay không.

Ngoài ra, còn có nhiều loại kiểm thử khác như kiểm thử hạt nhân, kiểm thử tải, kiểm thử bảo mật, kiểm thử hiệu suất, kiểm thử tương thích và kiểm thử giao diện người dùng. Lập trình viên sẽ lựa chọn và thực hiện các mức kiểm thử phù hợp tùy thuộc vào tính chất và yêu cầu cụ thể của dự án.

Nêu vòng đời của kiểm thử thành phần. Đưa ra 1 ví dụ về Stub và Driver.

Vòng đời của kiểm thử thành phần (component testing) bao gồm các bước sau:

Xác định các thành phần cần được kiểm thử: Trước tiên, các thành phần của hệ thống cần được xác định để bắt đầu quá trình kiểm thử thành phần. Các thành phần này có thể là các hàm, module, lớp hoặc giao diện.

Lập kế hoạch kiểm thử: Sau khi xác định các thành phần cần kiểm thử, lập kế hoạch kiểm thử sẽ được thực hiện. Kế hoạch này sẽ xác định phạm vi, các ca kiểm thử, bộ dữ liệu kiểm thử và các tiêu chí thành công.

Chuẩn bị môi trường kiểm thử: Môi trường kiểm thử cần được chuẩn bị để triển khai các ca kiểm thử. Điều này bao gồm cài đặt phần mềm, tạo dữ liệu và các tài nguyên cần thiết để chạy kiểm thử.

Triển khai kiểm thử thành phần: Kiểm thử thành phần được triển khai bằng cách chạy các ca kiểm thử được định nghĩa trong kế hoạch kiểm thử. Dữ liệu kiểm thử được cung cấp và thực hiện hoạt động trên các thành phần được kiểm thử.

Kết quả kiểm thử: Các kết quả kiểm thử (pass/fail) được ghi lại và phân tích để xác định các lỗi và vấn đề có thể xảy ra trong thành phần. Nếu thành phần không vượt qua kiểm thử, các bước khắc phục và kiểm tra lại được thực hiện.

Đánh giá và báo cáo: Cuối cùng, kết quả kiểm thử được đánh giá và báo cáo cho các bên liên quan. Báo cáo này có thể bao gồm kết quả kiểm thử, lỗi phát hiện được và các giải pháp khắc phục.

Ví dụ về Stub và Driver:

Stub: Trong kiểm thử phần mềm, Stub là một thành phần đơn giản và giả định được sử dụng để xác định và thử nghiệm một thành phần khác. Ví dụ, nếu bạn đang kiểm thử một ứng dụng di động có tích hợp kết nối mạng, nhưng chưa thể thực hiện kết nối thực tế, bạn có thể sử dụng Stub để thay thế phần kết nối mạng và gửi các kết quả giả định để kiểm tra tính năng của ứng dụng khác.

Driver: Driver là một thành phần giả định được sử dụng để gọi và kiểm tra một thành phần khác, thường là một thành phần cao hơn trong cấu trúc phần mềm. Ví dụ, nếu bạn đang kiểm thử một ứng dụng web sử dụng cơ sở dữ liệu, nhưng chưa có tầng giao diện người dùng hoàn chỉnh, bạn có thể sử dụng Driver để gọi các chức năng thành phần cơ sở dữ liệu và kiểm tra dữ liệu đầu ra của nó.

Thời điểm diễn ra Kiểm thử tích hợp là khi nào ?

Kiểm thử tích hợp là một giai đoạn trong quá trình kiểm thử phần mềm, nó diễn ra sau khi kiểm thử đơn vị (unit testing) hoàn thành. Thời gian diễn ra kiểm thử tích hợp thường là ngay sau khi các thành phần độc lập đã được kiểm thử và được kết hợp lại với nhau để hình thành hệ thống hoàn chỉnh hoặc một phần của hệ thống.

Trong kiểm thử tích hợp, mục tiêu là kiểm tra tính tương thích, giao tiếp và hiệu suất giữa các thành phần đã được kết hợp. Kiểm thử tích hợp cũng giúp xác định rõ ràng xem các thành phần và các liên kết giữa chúng hoạt động như mong đợi hay không.

Thời điểm thực hiện kiểm thử tích hợp thường phụ thuộc vào quy trình phát triển phần mềm cụ thể. Nó có thể diễn ra sau giai đoạn kiểm thử đơn vị và trước khi tiến hành kiểm thử hệ thống hoặc có thể được thực hiện song song với các giai đoạn kiểm thử khác trong quy trình phát triển. Tùy thuộc vào tổ chức, dự án và phạm vi của phần mềm, thời gian kiểm thử tích hợp có thể kéo dài từ vài ngày đến vài tuần.

Ưu/Nhược điểm của phương pháp kiểm thử Bigbang ?

Ưu điểm của phương pháp kiểm thử Bigbang:

Tiết kiệm thời gian: Phương pháp Bigbang cho phép kiểm thử các thành phần được kết hợp ngay sau giai đoạn phát triển, giúp tiết kiệm thời gian so với các phương pháp kiểm thử khác như kiểm thử từng cấp.

Tích hợp nhanh chóng: Với phương pháp Bigbang, có thể nhanh chóng hình thành môi trường kiểm thử tích hợp bằng cách kết hợp các thành phần lại với nhau.

Nhược điểm của phương pháp kiểm thử Bigbang:

Khó xác định lỗi: Vì phương pháp Bigbang kiểm thử tất cả các thành phần kết hợp nhau ở giai đoạn cuối cùng, việc xác định chính xác lỗi và vị trí lỗi có thể trở nên khó khăn.

Không kiểm thử từng phần riêng biệt: Phương pháp Bigbang không kiểm thử các thành phần độc lập một cách chi tiết như phương pháp kiểm thử từng cấp, dẫn đến khả năng bỏ qua các lỗi nhỏ hoặc thuật toán của từng thành phần.

Khó quản lý: Với việc kết hợp tất cả các thành phần và kiểm thử cùng nhau, phương pháp Bigbang có thể trở nên khó quản lý và khó truy vết các lỗi cụ thể liên quan đến từng thành phần.

Tóm lại, phương pháp kiểm thử Bigbang có ưu điểm là tiết kiệm thời gian và tích hợp nhanh chóng, nhưng cũng có các nhược điểm như khó xác định lỗi, không kiểm thử từng phần riêng biệt và khó quản lý. Việc sử dụng phương pháp này cần xem xét cẩn thận dựa trên yêu cầu và tổ chức của dự án.

Ưu/Nhược điểm của phương pháp kiểm thử Top Down ?

Ưu điểm của phương pháp kiểm thử Top Down:

Xác định lỗi sớm: Phương pháp Top Down tập trung kiểm thử các thành phần quan trọng và cao cấp trước, giúp phát hiện lỗi và vấn đề sớm hơn trong quá trình phát triển.

Giúp phát hiện các lỗi giao tiếp: Phương pháp này cho phép kiểm thử và xác định các lỗi giao tiếp giữa các thành phần trong quá trình tích hợp, đảm bảo tính tương thích và liên kết giữa các phần của hệ thống.

Cải thiện quản lý kiểm thử: Phương pháp Top Down khả năng quản lý tốt hơn so với các phương pháp khác vì nó phân chia và xác định rõ ràng các thành phần có thể kiểm thử từ trên xuống dưới.

Nhược điểm của phương pháp kiểm thử Top Down:

Fake data: Khi các thành phần con chưa được triển khai hoặc chưa thể kiển quả thực, có thể cần sử dụng dữ liệu giả để thực hiện kiểm thử, có thể làm giảm tính chính xác và độ tin cậy của quá trình.

Không kiểm thử các thành phần thấp hơn sâu: Phương pháp Top Down thường thiên về kiểm thử các thành phần trên cấp cao hơn, điều này dẫn đến việc quá trình kiểm thử không tập trung đủ vào các thành phần thấp hơn trong hệ thống.

Độ phức tạp trong việc phối hợp: Việc phối hợp và kiểm soát giao tiếp giữa người phát triển và các thành phần cấp dưới có thể trở nên phức tạp và khó khăn trong phương pháp Top Down.

Tóm lại, phương pháp kiểm thử Top Down có ưu điểm là xác định lỗi sớm, phát hiện lỗi giao tiếp và cải thiện quản lý kiểm thử. Tuy nhiên, nó cũng có nhược điểm như việc sử dụng fake data, không kiểm thử các thành phần thấp hơn sâu và độ phức tạp trong việc phối hợp. Quyết định sử dụng phương pháp này nên dựa trên yêu cầu và tổ chức của dự án.

Ưu/Nhược điểm của phương pháp kiểm thử Bottom Up?

Ưu điểm của phương pháp kiểm thử Top Down:

Xác định lỗi sớm: Phương pháp Top Down tập trung kiểm thử các thành phần quan trọng và cao cấp trước, giúp phát hiện lỗi và vấn đề sớm hơn trong quá trình phát triển.

Giúp phát hiện các lỗi giao tiếp: Phương pháp này cho phép kiểm thử và xác định các lỗi giao tiếp giữa các thành phần trong quá trình tích hợp, đảm bảo tính tương thích và liên kết giữa các phần của hệ thống.

Cải thiện quản lý kiểm thử: Phương pháp Top Down khả năng quản lý tốt hơn so với các phương pháp khác vì nó phân chia và xác định rõ ràng các thành phần có thể kiểm thử từ trên xuống dưới.

Nhược điểm của phương pháp kiểm thử Top Down:

Fake data: Khi các thành phần con chưa được triển khai hoặc chưa thể kiển quả thực, có thể cần sử dụng dữ liệu giả để thực hiện kiểm thử, có thể làm giảm tính chính xác và độ tin cậy của quá trình.

Không kiểm thử các thành phần thấp hơn sâu: Phương pháp Top Down thường thiên về kiểm thử các thành phần trên cấp cao hơn, điều này dẫn đến việc quá trình kiểm thử không tập trung đủ vào các thành phần thấp hơn trong hệ thống.

Độ phức tạp trong việc phối hợp: Việc phối hợp và kiểm soát giao tiếp giữa người phát triển và các thành phần cấp dưới có thể trở nên phức tạp và khó khăn trong phương pháp Top Down.Ưu điểm của phương pháp kiểm thử Bottom Up:

Phát hiện lỗi nhanh chóng: Phương pháp Bottom Up cho phép kiểm thử các thành phản ưu tiên cao và ưu tiên ở mức thấp trước, giúp phát hiện lỗi sớm trong quá trình kiểm thử.

Kiểm thử chi tiết từng thành phần: Phương pháp này tập trung vào kiểm thử từng thành phần riêng lẻ, giúp xác định lỗi và điều chỉnh chúng một cách chi tiết.

Tính tương thích và liên kết: Kiểm thử Bottom Up giúp đảm bảo tính tương thích và liên kết giữa các thành phần, đồng thời tăng cường khả năng giao tiếp giữa các thành phần trong hệ thống.

Nhược điểm của phương pháp kiểm thử Bottom Up:

Không thể kiểm thử hệ thống ngay từ đầu: Vì phương pháp Bottom Up bắt đầu từ các thành phần cấp thấp và tiến lên đến các thành phần cấp cao hơn, việc kiểm thử hệ thống như một toàn thể có thể bị trì hoãn đến giai đoạn sau.

Cần thiết phần mềm giả để kiểm thử: Để kiểm thử các thành phần cấp cao hơn trong phương pháp Bottom Up, thời gian và nguồn lực cần thiết để phát triển các thành phần giả có thể là một thách thức.

Tóm lại, phương pháp kiểm thử Top Down có ưu điểm là xác định lỗi sớm, phát hiện lỗi giao tiếp và cải thiện quản lý kiểm thử. Tuy nhiên, nó cũng có nhược điểm như việc sử dụng fake data, không kiểm thử các thành phần thấp hơn sâu và độ phức tạp trong việc phối hợp. Quyết định sử dụng phương pháp này nên dựa trên yêu cầu và tổ chức của dự án.

Kiểm thử hệ thống thì sẽ kiểm thử những gì ?

Khi thực hiện kiểm thử hệ thống, chúng ta sẽ kiểm tra các khía cạnh khác nhau của hệ thống để đảm bảo rằng nó hoạt động như dự định và đáp ứng được yêu cầu của người dùng. Dưới đây là một số khía cạnh chính mà bạn có thể kiểm thử trong quá trình kiểm thử hệ thống:

Kiểm thử chức năng: Đảm bảo rằng tất cả các chức năng của hệ thống hoạt động đúng theo thiết kế và yêu cầu người dùng.

Kiểm thử hiệu năng: Đánh giá hiệu suất và tốc độ của hệ thống trong các tình huống tải cao và đảm bảo rằng nó đáp ứng được yêu cầu hiệu năng.

Kiểm thử giao diện người dùng: Kiểm tra tính tương tác và sự dễ sử dụng của giao diện người dùng, đảm bảo rằng người dùng có thể tương tác và sử dụng hệ thống một cách dễ dàng và thuận tiện.

Kiểm thử bảo mật: Đảm bảo rằng hệ thống được bảo mật và không có lỗ hổng nghiêm trọng có thể bị tấn công hoặc lợi dụng.

Kiểm thử tích hợp: Kiểm tra tích hợp giữa các thành phần khác nhau của hệ thống để đảm bảo rằng chúng làm việc cùng nhau một cách hợp lý và không gây ra xung đột hoặc lỗi.

Kiểm thử khôi phục sau sự cố: Kiểm tra khả năng khôi phục của hệ thống sau các sự cố như mất điện, mất kết nối, hoặc lỗi phần mềm.

Kiểm thử tương thích: Đảm bảo rằng hệ thống là tương thích với các phiên bản phần mềm, nền tảng, trình duyệt, và các thành phần khác liên quan.