|  |  |  |
| --- | --- | --- |
| **Nhóm** | **Họ và tên** | **Mssv** |
| 01 | Vũ Hoàng Anh | 20110205 |

**Bài tập**

***Kiểm thử phần mềm***

**------o0o------**

**Bài tập 02: Quy trình kiểm thử phần mềm**

1. Thực hiện: Làm và nộp bài cá nhân
2. Thời lượng: 60 phút.
3. Cách thức nộp bài: nộp file Word.
4. Kết quả / Sản phẩm: file word .DOC hoặc .DOCX
5. Yêu cầu cụ thể:

Tìm hiểu và trình bày định nghĩa của các thuật ngữ sau (trình bày ngắn gọn nhưng đầy đủ, i.e khoảng 5 dòng cho mỗi thuật ngữ):

* 1. Unit Test (1)

Unit Test là một loại kiểm thử phần mềm, trong đó các đơn vị nhỏ nhất của mã nguồn, như hàm, phương thức hay lớp được kiểm tra độc lập với các phần khác của ứng dụng. Mục đích của Unit Test là giúp đảm bảo tính đúng đắn và độ tin cậy của các đơn vị nhỏ trong mã nguồn, từ đó giúp hạn chế lỗi và tăng chất lượng của phần mềm.

* 1. Integration Test (2)

Integration Test là một loại kiểm thử phần mềm, trong đó các phần khác nhau của ứng dụng được kết hợp và kiểm tra tính tương thích và tính đúng đắn của chúng khi hoạt động cùng nhau. Mục đích của Integration Test là giúp đảm bảo tính đúng đắn và tính tương thích của các phần khác nhau trong ứng dụng, từ đó giúp hạn chế lỗi và tăng chất lượng của phần mềm trước khi đưa ra sản phẩm. Integration Test thường được thực hiện sau khi đã hoàn thành Unit Test.

* 1. System Test (3)

System Test (Kiểm thử hệ thống) là một loại kiểm thử phần mềm, trong đó hệ thống được kiểm tra chức năng, hiệu suất, độ bảo mật và tính tương thích với môi trường và ứng dụng khác. Mục đích của System Test là đảm bảo rằng hệ thống hoạt động đúng đắn theo yêu cầu thiết kế và đáp ứng các yêu cầu chức năng, hiệu suất và bảo mật của khách hàng. System Test thường được thực hiện sau khi đã hoàn thành Integration Test và trước khi sản phẩm được triển khai ra thị trường.

* 1. Acceptance Test (4)

Acceptance Test là một loại kiểm thử phần mềm, trong đó hệ thống được kiểm tra xem nó có đáp ứng được các yêu cầu của khách hàng hay không. Mục đích của Acceptance Test là đảm bảo rằng sản phẩm hoàn thành đáp ứng yêu cầu và mong đợi của khách hàng. Acceptance Test thường được thực hiện bởi khách hàng hoặc người đại diện của khách hàng, và được thực hiện sau khi đã hoàn thành System Test và trước khi sản phẩm được chấp nhận và triển khai hoặc bán ra thị trường.

* 1. Stress / Load Test (5)

Stress / Load Test là một loại kiểm thử phần mềm, trong đó hệ thống được kiểm tra với các tải trọng và điều kiện khác nhau để đánh giá tối đa khả năng chịu tải và hiệu suất của nó. Mục đích của Stress / Load Test là giúp đảm bảo rằng hệ thống có thể xử lý được một lượng lớn người dùng và dữ liệu, đồng thời đảm bảo rằng hiệu suất của hệ thống không bị giảm sút trong quá trình hoạt động. Stress / Load Test thường được thực hiện trước khi sản phẩm được triển khai ra thị trường để đảm bảo rằng hệ thống có thể đáp ứng được nhu cầu sử dụng và không gặp sự cố trong quá trình hoạt động.

* 1. Performance Test (6)

Performance Test là một loại kiểm thử phần mềm, trong đó hệ thống được kiểm tra về khả năng xử lý và thời gian đáp ứng trong các tình huống khác nhau. Mục đích của Performance Test là đánh giá hiệu suất của hệ thống, đảm bảo rằng nó có thể xử lý một lượng lớn người dùng và dữ liệu trong thời gian ngắn nhất có thể. Performance Test thường được thực hiện trước khi sản phẩm được triển khai ra thị trường để đảm bảo rằng hệ thống có thể đáp ứng được nhu cầu sử dụng và không gặp sự cố trong quá trình hoạt động. Performance Test thường được thực hiện đồng thời với Stress / Load Test để đánh giá khả năng chịu tải và hiệu suất của hệ thống.

* 1. Install / Uninstall Test

Install / Uninstall Test là một loại kiểm thử phần mềm, trong đó sản phẩm được cài đặt và gỡ bỏ để đảm bảo rằng quá trình cài đặt và gỡ bỏ được thực hiện đúng cách và không gây ra sự cố nào cho hệ thống hoặc người dùng. Mục đích của Install / Uninstall Test là đảm bảo rằng sản phẩm có thể được cài đặt và gỡ bỏ một cách dễ dàng và hiệu quả, đồng thời không gây ra tác động tiêu cực đến hệ thống hoặc ứng dụng khác. Install / Uninstall Test thường được thực hiện trước khi sản phẩm được triển khai ra thị trường để đảm bảo rằng người dùng có thể cài đặt và sử dụng sản phẩm một cách thuận tiện và hiệu quả.

* 1. Recovery Test (1)

Recovery Test là một loại kiểm thử phần mềm, trong đó hệ thống được kiểm tra khả năng phục hồi sau khi xảy ra các sự cố, như mất điện, mất dữ liệu hoặc lỗi phần cứng. Mục đích của Recovery Test là đảm bảo rằng hệ thống có thể phục hồi một cách nhanh chóng và hiệu quả sau khi xảy ra sự cố, đồng thời đảm bảo tính khả dụng và tin cậy của hệ thống. Recovery Test thường được thực hiện để đánh giá tính ổn định và khả năng phục hồi của hệ thống trong các trường hợp khẩn cấp hoặc sự cố không mong muốn.

* 1. Compatibility Test (2)

Compatibility Test là một loại kiểm thử phần mềm, trong đó hệ thống được kiểm tra tính tương thích với các phần mềm, phần cứng và môi trường khác nhau. Mục đích của Compatibility Test là đảm bảo rằng sản phẩm có thể hoạt động đúng đắn và tương thích với các phần mềm, phần cứng và môi trường khác nhau mà người dùng có thể sử dụng. Compatibility Test thường được thực hiện trước khi sản phẩm được phát hành ra thị trường để đảm bảo rằng sản phẩm có thể hoạt động đúng đắn trên các nền tảng và môi trường khác nhau mà người dùng có thể sử dụng.

* 1. Comparison Test

Comparison Test là một loại kiểm thử phần mềm, trong đó sản phẩm được so sánh với các sản phẩm cùng loại khác trên thị trường để đánh giá tính năng, chức năng và hiệu suất của sản phẩm. Mục đích của Comparison Test là đảm bảo rằng sản phẩm có tính năng và hiệu suất tương đương hoặc tốt hơn các sản phẩm cùng loại trên thị trường. Comparison Test thường được thực hiện để đánh giá sự cạnh tranh của sản phẩm trên thị trường và tìm cách cải thiện sản phẩm để đáp ứng được nhu cầu và yêu cầu của khách hàng.

* 1. Conversion Test (3)

Conversion Test là một loại kiểm thử phần mềm, trong đó dữ liệu được chuyển đổi từ hệ thống cũ sang hệ thống mới. Mục đích của Conversion Test là đảm bảo rằng dữ liệu được chuyển đổi một cách chính xác và hoàn toàn từ hệ thống cũ sang hệ thống mới, đồng thời đảm bảo tính toàn vẹn của dữ liệu sau khi chuyển đổi. Conversion Test thường được thực hiện trước khi hệ thống mới được triển khai để đảm bảo rằng dữ liệu được chuyển đổi đúng đắn và không bị mất mát, và đảm bảo tính ổn định của hệ thống sau khi chuyển đổi dữ liệu.

* 1. Usability Test

Đây là phương pháp dùng để đánh giá khả năng sử dụng, dễ dàng và hiệu quả của một sản phẩm hoặc ứng dụng. Nó thường được sử dụng để đánh giá trải nghiệm người dùng với sản phẩm hoặc ứng dụng, đo lường độ hài lòng của người dùng và phát hiện các vấn đề về thiết kế, tính năng và trải nghiệm người dùng. Usability Test bao gồm quan sát người dùng, đánh giá và phỏng vấn người dùng, và thu thập phản hồi từ các nhà phát triển và người dùng để cải thiện sản phẩm hoặc ứng dụng.

* 1. Configuration Test (4)

Đây là phương pháp kiểm thử phần mềm nhằm kiểm tra tính năng và hiệu suất của ứng dụng trong các cấu hình khác nhau. Các cấu hình có thể bao gồm các môi trường khác nhau, hệ điều hành, trình duyệt web, các thiết bị đầu cuối khác nhau, các phiên bản phần mềm khác nhau và các cấu hình mạng khác nhau. Phương pháp kiểm thử này đảm bảo tính tương thích của sản phẩm hoặc ứng dụng với các hệ thống khác để đảm bảo tính ổn định và độ tin cậy của sản phẩm hoặc ứng dụng.

* 1. Black box / functional testing (5)

Đây là phương pháp kiểm thử phần mềm, trong đó các chức năng và tính năng của một ứng dụng được kiểm tra mà không cần biết về cấu trúc bên trong của chương trình. Trong Black box testing, các nhà kiểm thử xem ứng dụng như một hộp đen và chỉ quan tâm đến kết quả của các thao tác người dùng đầu vào và đầu ra của ứng dụng.

Đây là một phương pháp quan trọng để đảm bảo rằng ứng dụng hoạt động đúng đắn và đáp ứng được các yêu cầu của người dùng mà không cần biết chi tiết về bên trong của chương trình.

* 1. White box / structural testing (6)

Đây là một phương pháp kiểm thử phần mềm, trong đó các người kiểm thử có kiến thức về cấu trúc bên trong của chương trình và sử dụng kiến thức này để thiết kế và thực hiện các bài kiểm thử. Trong White box testing, các nhà kiểm thử xem xét cấu trúc bên trong của chương trình, bao gồm các đơn vị mã nguồn, các hàm và các lớp, để thiết kế các ca kiểm thử.

White box testing yêu cầu kiến thức về cấu trúc và mã nguồn của chương trình, do đó nó thường được sử dụng bởi các nhà phát triển phần mềm, kỹ sư phần mềm và các chuyên gia kiểm thử có kiến thức chuyên môn về lập trình.

* 1. Sanity testing

Đây là một loại kiểm thử phần mềm đơn giản để đảm bảo rằng các tính năng cơ bản của chương trình hoạt động một cách đúng đắn sau khi đã được thay đổi hoặc cập nhật. Nó thường được thực hiện sau khi đã thực hiện Unit Test hoặc Integration Test để đảm bảo rằng hệ thống không bị lỗi nghiêm trọng và có thể tiếp tục kiểm thử.

Sanity Testing thường chỉ kiểm tra các tính năng cơ bản của chương trình hoặc tính năng liên quan đến các thay đổi mới nhất, và không kiểm tra toàn bộ chương trình. Sanity Testing là một phương pháp kiểm thử nhanh và hiệu quả để đảm bảo tính ổn định và đáng tin cậy của phần mềm sau khi đã được thay đổi hoặc cập nhật.

* 1. Regression testing (1, 2)

Đây là một phương pháp kiểm thử phần mềm để đảm bảo rằng các thay đổi mới trong phần mềm không ảnh hưởng đến các tính năng và chức năng hiện có của chương trình. Nó bao gồm thực hiện các kiểm thử lại trên các tính năng và chức năng đã kiểm tra trước đó để đảm bảo rằng chúng vẫn hoạt động chính xác sau khi đã thay đổi.

Mục đích của Regression Testing là đảm bảo rằng các tính năng và chức năng đã được kiểm tra trước đó vẫn hoạt động chính xác và không bị ảnh hưởng bởi các thay đổi mới. Regression Testing yêu cầu nhiều thời gian và nguồn lực để thực hiện và không thể đảm bảo rằng phần mềm hoàn toàn không có lỗi.

* 1. Incremental testing (3, 4)

Incremental testing là một phương pháp kiểm thử phần mềm được thực hiện bằng cách kiểm tra từng phần của phần mềm một cách tách biệt, trước khi kết hợp chúng lại với nhau để tạo thành hệ thống hoàn chỉnh.

Phương pháp này thường được sử dụng trong quá trình phát triển phần mềm để đảm bảo rằng mỗi phần của phần mềm đáp ứng được yêu cầu và hoạt động đúng cách trước khi được tích hợp vào phần còn lại của hệ thống.

* 1. Thread testing

Thread testing là một kỹ thuật kiểm thử phần mềm được sử dụng để kiểm tra tính đúng đắn và hiệu suất của các luồng (threads) trong một ứng dụng đa luồng (multithreaded application). Thread testing được thực hiện bằng cách tạo và chạy nhiều luồng đồng thời trong ứng dụng và theo dõi các hành vi của chúng.

Trong thread testing, các lỗi thường được tìm thấy bao gồm các lỗi về đồng bộ hóa luồng, truy cập dữ liệu không đồng bộ và deadlock (một trường hợp khi hai hoặc nhiều luồng bị mắc kẹt khi đợi lẫn nhau để giải phóng tài nguyên).

* 1. Alpha Test

Alpha Test là một loại kiểm thử phần mềm được thực hiện trước khi phần mềm được phát hành cho người dùng chính thức. Trong giai đoạn này, nhóm phát triển sẽ thực hiện kiểm tra chức năng và hiệu suất của phần mềm bằng cách sử dụng một số trường hợp thử nghiệm được thiết kế để phát hiện và sửa chữa các lỗi và sự cố tiềm ẩn trước khi phần mềm được phát hành. Alpha Test thường được thực hiện bởi nhóm phát triển phần mềm hoặc các nhà phát triển thử nghiệm chuyên nghiệp. Sau khi hoàn thành và kiểm tra, phần mềm sẽ tiếp tục qua giai đoạn Beta Test để kiểm tra với một tập khách hàng mẫu thực tế.

* 1. Beta Test (5, 6)

Beta Test là quá trình thử nghiệm một sản phẩm hoặc dịch vụ mới trước khi chính thức ra mắt cho khách hàng sử dụng. Đây là giai đoạn cuối cùng trong quá trình phát triển sản phẩm, trong đó các lỗi và sự cố có thể được phát hiện và khắc phục trước khi sản phẩm được phát hành cho công chúng.

1. Tài liệu sử dụng:

[1] Bài giảng powerpoint Chương 1: Tổng quan về kiểm thử (từ slide 41)

[2] Phần 2: Tài liệu Software Testing Fundamental.

[3] Chương 1&2, tài liệu tiếng Việt của Thầy Nguyễn Văn Hiệp.

[4] Các nguồn trên Internet và wiki.