**KIỂM THỬ CƠ BẢN (SOF3031)**

Các Test Level trong kiểm thử phần mềm là gì?

* 4 loại:
* Unit Testing (Kiểm thử đơn vị)
* Integration Testing (Kiểm thử tích hợp)
* System Testing (Kiểm thử hệ thống)
* Acceptance Test (Kiểm thử chấp nhận)

**CÂU HỎI**

**LƯU Ý:**

**QUAN TRỌNG:**

***Câu 1:* SRS hoặc HLD là loại tài liệu gì và dùng để làm gì? Nó có quan trọng đối với kiểm thử không? Test Case là gì, tại sao phải dùng nó?**

* SRS và HLD đều là loại tài liệu trong quy trình phát triển phần mềm:

- SRS (Software Requirements Specification) là tài liệu mô tả yêu cầu của phần mềm một cách chi tiết và rõ ràng. Nó mô tả các chức năng, hiệu suất, giao diện người dùng, và các yêu cầu khác đối với phần mềm được phát triển.

- HLD (High-Level Design) là tài liệu mô tả thiết kế tổng quan của phần mềm, bao gồm cấu trúc tổng thể của hệ thống, các thành phần chính và cách chúng tương tác với nhau.

Cả hai loại tài liệu này đều rất quan trọng trong quy trình phát triển phần mềm. SRS giúp định nghĩa rõ ràng các yêu cầu của hệ thống, trong khi HLD giúp định hình cách thức triển khai các yêu cầu đó vào một kiến trúc cụ thể.

Đối với kiểm thử, SRS và HLD đều cực kỳ quan trọng. SRS cung cấp cơ sở để xác định các kịch bản kiểm thử và tạo ra các ca kiểm thử. HLD giúp kiểm thử viên hiểu cách hệ thống hoạt động và tương tác với các thành phần khác nhau.

* Test Case là một bộ các bước cụ thể để kiểm tra tính năng hoặc yêu cầu của phần mềm. Mỗi test case mô tả một trường hợp cụ thể mà phần mềm phải xử lý và kết quả mong đợi của nó. Test case được sử dụng để đảm bảo rằng phần mềm hoạt động đúng và đáp ứng được các yêu cầu đã được xác định trong SRS và HLD.

***Câu 2:* Kiểm thử hộp đen là gì? Điều kiện kiểm thử hộp đen? Kiểm thử hộp trắng là gì?**

* **Kiểm thử hộp đen (Black Box Testing):**

- Là phương pháp kiểm thử tập trung vào chức năng của phần mềm mà không cần biết về cấu trúc nội bộ của mã nguồn.

- Kiểm thử này tập trung vào việc kiểm tra xem phần mềm hoạt động như thế nào dựa trên đầu vào và đầu ra của nó.

- Tester không quan tâm đến cách mã nguồn được triển khai hoặc cách nó hoạt động bên trong.

- Một số phương pháp kiểm thử hộp đen bao gồm kiểm thử hồi quy, kiểm thử chức năng, kiểm thử tương tự, và kiểm thử thác nước.

* **Điều kiện kiểm thử hộp đen:**

- Phải có mô tả chức năng hoặc yêu cầu của phần mềm.

- Phải có dữ liệu đầu vào được sử dụng để thực hiện kiểm thử.

- Cần phải xác định được kết quả mong đợi của mỗi ca kiểm thử.

* **Kiểm thử hộp trắng (White Box Testing):**

- Là phương pháp kiểm thử tập trung vào kiểm tra cấu trúc nội bộ của mã nguồn.

- Kiểm thử này thường được thực hiện bởi các nhà phát triển hoặc người kiểm thử có kiến thức về mã nguồn.

- Tester kiểm tra các luồng điều khiển, điều kiện, vòng lặp, và các thành phần khác trong mã nguồn để đảm bảo rằng chúng hoạt động đúng.

- Một số phương pháp kiểm thử hộp trắng bao gồm kiểm thử dòng điều khiển, kiểm thử điều kiện, kiểm thử dữ liệu, và kiểm thử đường dẫn.

* Điều kiện kiểm thử hộp trắng:

- Cần có kiến thức về cấu trúc nội bộ của mã nguồn.

- Phải có công cụ phân tích mã nguồn hoặc trình biên dịch để kiểm tra mã.

- Cần xác định được các điểm kiểm thử trong mã nguồn và tạo các ca kiểm thử dựa trên chúng.

***Câu 3:* Kỹ thuật BVA (Boundary Value Analysis – Phân tích giá trị biên) là gì? Ví dụ?**

* **Kỹ thuật Boundary Value Analysis (BVA)** là một kỹ thuật kiểm thử phần mềm được sử dụng để tìm ra lỗi trong các biên của dữ liệu đầu vào. Ý tưởng cơ bản của BVA là chọn các giá trị biên cho các biến đầu vào để kiểm tra xem phần mềm có hoạt động đúng trong các biên này không, vì thường lỗi thường xảy ra ở các giá trị biên này.
* **Ví dụ:** Giả sử có một trường dữ liệu phần mềm yêu cầu nhập một số nguyên từ 1 đến 10. Để sử dụng kỹ thuật BVA, ta có các giá trị biên như sau:

- Giá trị biên nhỏ nhất: 1

- Giá trị biên lớn nhất: 10

Khi kiểm tra phần mềm, ta sử dụng các giá trị đầu vào như sau:

- Giá trị biên nhỏ nhất: 1 => Phần mềm báo hợp lệ

- Giá trị biên lớn nhất: 10 => Phần mềm báo hợp lệ

- Giá trị nhỏ hơn biên nhỏ nhất: 0 => Phần mềm báo lỗi

- Giá trị lớn hơn biên nhỏ nhất: 2 => Phần mềm báo hợp lệ

- Giá trị nhỏ hơn biên lớn nhất: 9 => Phần mềm báo hợp lệ

- Giá trị lớn hơn biên lớn nhất: 11 => Phần mềm báo lỗi

***Câu 4:* Trình bày các loại kiểm thử mà ta đã được học? Trình bày 4 Test Level trong kiểm thử phần mềm? 4 loại (Trình bày 4 loại nói ngắn gọn và lấy ví dụ). Quy trình kiểm thử như thế nào?**

* Các loại kiểm thử mà chúng ta đã học bao gồm:

1. **Kiểm thử hộp đen (Black Box Testing)**: Kiểm thử dựa trên các yêu cầu và chức năng của phần mềm mà không cần biết về cấu trúc nội bộ của mã nguồn. Ví dụ: Kiểm thử chức năng, kiểm thử hồi quy.
2. **Kiểm thử hộp trắng (White Box Testing)**: Kiểm thử tập trung vào cấu trúc nội bộ của mã nguồn và logic của chương trình. Ví dụ: Kiểm thử dòng điều khiển, kiểm thử điều kiện.
3. **Kiểm thử hợp pháp (Regression Testing)**: Kiểm tra để đảm bảo rằng các thay đổi mới không ảnh hưởng đến các tính năng đã tồn tại của phần mềm. Ví dụ: Khi bạn thêm một tính năng mới vào phần mềm, bạn cần kiểm tra lại các tính năng đã có để đảm bảo rằng chúng vẫn hoạt động đúng.
4. **Kiểm thử tương tự (Exploratory Testing)**: Kiểm thử mà không cần có kế hoạch cụ thể trước. Tester sẽ tự do khám phá và kiểm tra phần mềm từ quan điểm người dùng cuối. Ví dụ: Một tester khám phá một ứng dụng di động mới và kiểm tra các tính năng một cách tự nhiên, giống như một người dùng cuối.

* **Bốn cấp độ kiểm thử trong phần mềm là:**

1. **Unit Testing (Kiểm thử đơn vị)**: Kiểm thử từng đơn vị nhỏ nhất của mã nguồn, như hàm hoặc phương thức.
2. **Integration Testing (Kiểm thử tích hợp)**: Kiểm thử các module hoặc thành phần được kết hợp lại với nhau để đảm bảo rằng chúng hoạt động một cách hợp lý.
3. **System Testing (Kiểm thử hệ thống)**: Kiểm thử toàn bộ hệ thống như một phần mềm hoàn chỉnh để đảm bảo rằng nó đáp ứng các yêu cầu được đặt ra.
4. **Acceptance Testing (Kiểm thử chấp nhận)**: Kiểm thử được thực hiện để đảm bảo rằng phần mềm đáp ứng được các yêu cầu của người dùng cuối và sẵn sàng để triển khai vào môi trường sản xuất.

**Ví dụ cụ thể**:

* **Unit Testing (Kiểm thử đơn vị)**:

Ví dụ: Trong một ứng dụng web, một phần của mã nguồn xử lý việc đăng nhập người dùng. Một unit test có thể được viết để kiểm tra xem hàm xác thực đăng nhập hoạt động đúng cách hay không. Unit test này có thể kiểm tra các trường hợp như đúng tên đăng nhập và mật khẩu, tên đăng nhập không hợp lệ, mật khẩu không đúng và các trường hợp khác.

* **Integration Testing (Kiểm thử tích hợp)**:

Ví dụ: Trong một hệ thống quản lý bán hàng, một module quản lý sản phẩm có thể được tích hợp với module quản lý giỏ hàng. Integration testing sẽ kiểm tra xem việc thêm sản phẩm vào giỏ hàng hoạt động đúng cách và thông tin sản phẩm được cập nhật đúng trong giỏ hàng sau mỗi thay đổi.

* **System Testing (Kiểm thử hệ thống)**:

Ví dụ: Trong một ứng dụng di động đặt vé máy bay, kiểm thử hệ thống sẽ kiểm tra tính năng đặt vé từ đầu đến cuối. Tester sẽ kiểm tra xem người dùng có thể tìm kiếm chuyến bay, chọn chuyến bay, nhập thông tin hành khách và thanh toán một cách thành công không.

* **Acceptance Testing (Kiểm thử chấp nhận)**:

Ví dụ: Trong một ứng dụng giao diện người dùng, acceptance testing sẽ được thực hiện để đảm bảo rằng giao diện người dùng đáp ứng được các yêu cầu của khách hàng. Tester sẽ kiểm tra xem giao diện người dùng có thân thiện, dễ sử dụng và hiệu quả không, và xác định xem nó có đáp ứng được các mong đợi của khách hàng hay không.

**Câu hỏi chi tiết**

## 1. Kiểm thử phần mềm là gì? Mục tiêu của kiểm thử phần mềm để làm gì?

## 2. Test Case là gì, tại sao phải dùng nó?

## 3. Testing và Debugging là gì? Ví dụ?

## 4. Kế hoạch kiểm thử bao gồm 4 thứ là gì?

## 5. Cấu trúc của tài liệu kế hoạch kiểm thử là gì (Test Plan)? Mục Tiêu của việc lập kế hoạch kiểm thử?

## 6. Kỹ thuật BVA (Boundary Value Analysis-Phân tích giá trị biên) là gì? Ví dụ?

## 7. Kỹ thuật EP (Equivalent Partition-Phân vùng tương đương) là gì? Ví dụ?

## 8. Báo cáo kiểm thử là gì? Khi nào thực hiện báo cáo kiểm thử?

## 9. Kiểm thử theo trạng thái là gì (State Transition)? Ví dụ?

## 10. Kiểm thử hộp đen? Điều kiện kiểm thử hộp đen?

## 11. Kiểm thử hộp trắng?

## 12. Static và Dynamic Testing là gì?

## 13. Cấu trúc 1 Test Case cần bao gồm các thành phần chính nào?

## 14. 3 nhân tố tạo nên chất lượng sản phẩm phần mềm là gì? Phần mềm phát sinh lỗi có ảnh hưởng gì không hãy nêu ra một vài ảnh hưởng?

## 15. Quy trình kiểm thử như thế nào?

## 16. Giới thiệu các tài liệu liên quan đến kiểm thử mà em đã được học?

## 17. SRS hoặc HLD là loại tài liệu gì và dùng để làm gì? Nó có quan trọng đối với kiểm thử không?

## 18. Test Plan là gì? Nội dung trong Test Plan hãy liệt kê ra các mục quan trọng?

## 19. Test Design là gì? Ví dụ? GUI Test Case là gì?

## 20. Test report dùng để làm gì? Error lỗi này đến từ đâu? Defect là gì?

## 21. Kiểm thử Functional là gì? Kiểm thử Non-functional?

## 22. Test Progress Report là gì?

## 23. Quy trình xử lý lỗi như nào?

## 24. Mức độ nghiêm trọng của lỗi được đánh giá như nào? Trạng thái của lỗi có mấy trạng thái?

## 25. Test planning, Test control, Test closure là gì hãy giải thích?

26. Kể tên 7 nguyên tắc kiểm thử.

**ĐÁP ÁN**

**1. Kiểm thử phần mềm là gì? Mục tiêu của kiểm thử phần mềm để làm gì?**

**a. Kiểm thử phần mềm là gì?**

Kiểm thử hay còn gọi là testing, là quá trình đánh giá một hệ thống hay các thành phần của nó với mục đích tìm xem liệu hệ thống có đáp ứng các yêu cầu được đã được chỉ định hay không. Nói một cách đơn giản, kiểm thử được thực hiện trên một hệ thống để xác định bất kỳ lỗ hổng, các lỗi hoặc các yêu cầu đang bị thiếu hay trái ngược với các yêu cầu thực tế đã được đề ra.

**b. Mục tiêu của kiểm thử phần mềm để làm gì?**

Tìm ra các (defects – thiếu sót/ errors – sai sót / bugs – lỗi) trong kết quả thực tế với kết quả mong muốn.

**2. TEST CASE là gì tại sao phải dùng nó?**

Test Case là một bộ các bước cụ thể để kiểm tra tính năng hoặc yêu cầu của phần mềm. Mỗi test case mô tả một trường hợp cụ thể mà phần mềm phải xử lý và kết quả mong đợi của nó. Test case được sử dụng để đảm bảo rằng phần mềm hoạt động đúng và đáp ứng được các yêu cầu đã được xác định trong SRS và HLD.

**3. Testing và Debugging là gì? Ví dụ?**

**a. Testing:**

Hoạt động tìm lỗi được thực hiện bởi kiểm thử viên và tìm được càng nhiều lỗi càng tốt.

**b. Debugging:**

Hoạt động sửa lỗi được thực hiện bởi DEV – Loại bỏ các lỗi hiện hình và và những lỗi tìm được.

**4. Kế hoạch kiểm thử (Test Plan) bao gồm 4 thứ là gì?**

1. **Phạm vi (Scope)**: Xác định rõ ràng phạm vi của quá trình kiểm thử. Điều này bao gồm các tính năng, modules hoặc thành phần cụ thể của sản phẩm phần mềm sẽ được kiểm thử và định rõ các yếu tố không nằm trong phạm vi của kiểm thử để tránh nhầm lẫn và hiểu lầm.
2. **Chiến lược (Strategy)**: Chiến lược kiểm thử mô tả cách tiếp cận tổng thể cho quá trình kiểm thử, cần xác định các phương pháp, kỹ thuật và tiêu chí đánh giá sẽ được sử dụng trong quá trình kiểm thử. Và mô tả cụ thể các loại kiểm thử sẽ được thực hiện, chẳng hạn như kiểm thử chức năng, kiểm thử phi chức năng, kiểm thử tích hợp, và nhiều loại khác.
3. **Nguồn lực (Resources)**: Cần xác định các tài nguyên cần thiết để thực hiện kế hoạch kiểm thử, bao gồm con người, thiết bị, phần mềm, và môi trường kiểm thử và đảm bảo rằng tất cả các nguồn lực đã được cung cấp và sẵn sàng trước khi bắt đầu kiểm thử.
4. **Lịch trình (Schedule)**: Cần xác định thời gian bắt đầu và kết thúc của mỗi hoạt động, cũng như các milestone quan trọng trong quá trình kiểm thử và đảm bảo rằng lịch trình kiểm thử phù hợp với lịch trình phát triển tổng thể của dự án và đảm bảo rằng các hoạt động kiểm thử không làm chậm tiến độ của dự án.

**5. Cấu trúc của tài liệu kế hoạch kiểm thử là gì (Test Plan)? Mục Tiêu của việc lập kế hoạch kiểm thử?**

**a. Cấu trúc:**

* Giới thiệu (Mục đích, thông tin chung, phạm vi test, ràng buộc, liệt kê các mạo hiểm)
* Các yêu cầu cho Test
* Chiến Lược Test
* Tài nguyên
* Các mốc kiểm soát của giai đoạn
* Các sản phẩm phải bàn giao.

**b. Mục tiêu của việc lập kế hoạch kiểm thử:**

* Xác định phạm vi kiểm thử
* Định nghĩa các rủi ro
* Xác định các tiêu chí để hoàn thành việc kiểm thử
* Mang lại hợp tác hiệu quả giữa các nhóm
* Xác định chiến lược
* Xác định nguồn lực
* Định nghĩa các chỉ số để đánh giá hoạt động kiểm thử
* Xác định tài liệu được sử dụng
* Xác định được thời gian thực hiện

**6. Kỹ thuật BVA (Boundary Value Analysis - Phân tích giá trị biên) là gì?**

**- Kỹ thuật Boundary Value Analysis (BVA)** là một kỹ thuật kiểm thử phần mềm được sử dụng để tìm ra lỗi trong các biên của dữ liệu đầu vào. Ý tưởng cơ bản của BVA là chọn các giá trị biên cho các biến đầu vào để kiểm tra xem phần mềm có hoạt động đúng trong các biên này không, vì thường lỗi thường xảy ra ở các giá trị biên này.

**- Ví dụ:** Giả sử có một trường dữ liệu phần mềm yêu cầu nhập một số nguyên từ 1 đến 10. Để sử dụng kỹ thuật BVA, ta có các giá trị biên như sau:

- Giá trị biên nhỏ nhất: 1

- Giá trị biên lớn nhất: 10

Khi kiểm tra phần mềm, ta sử dụng các giá trị đầu vào như sau:

- Giá trị biên nhỏ nhất: 1 => Phần mềm báo hợp lệ

- Giá trị biên lớn nhất: 10 => Phần mềm báo hợp lệ

- Giá trị nhỏ hơn biên nhỏ nhất: 0 => Phần mềm báo lỗi

- Giá trị lớn hơn biên nhỏ nhất: 2 => Phần mềm báo hợp lệ

- Giá trị nhỏ hơn biên lớn nhất: 9 => Phần mềm báo hợp lệ

- Giá trị lớn hơn biên lớn nhất: 11 => Phần mềm báo lỗi

**7. Kỹ thuật EP (Equivalent Partition - Phân vùng tương đương) là gì?**

**- Kỹ thuật EP (Equivalent Partition - Phân vùng tương đương)** là một phương pháp thiết kế ca kiểm thử trong kiểm thử phần mềm. Ý tưởng cơ bản của kỹ thuật này là chia các dữ liệu đầu vào hoặc điều kiện kiểm thử thành các nhóm tương đương sao cho mỗi nhóm chỉ chứa một số lượng hữu hạn các trường hợp kiểm thử.

**- Ví dụ:** Giả sử có một trường dữ liệu phần mềm yêu cầu nhập mật khẩu từ 6 đến 10 kí tự. Để sử dụng kỹ thuật EP, ta có các vùng tương đương như sau:

- Phân vùng 1: < 6 kí tự

- Phân vùng 2: > 6 và < 10 kí tự

- Phân vùng 3: > 10 kí tự

Khi kiểm tra phần mềm, ta sử dụng các giá trị đầu vào như sau:

* Phân vùng 1: Nhập 5 kí tự
* Phân vùng 2: Nhập 8 kí tự
* Phân vùng 3: Nhập 13 kí tự

Đối với:

- Phân vùng 1: => Phần mềm báo lỗi

- Phân vùng 2: => Phần mềm báo hợp lệ

- Phân vùng 3: => Phần mềm báo lỗi

**8. Báo cáo kiểm thử là gì? Khi nào thực hiện báo cáo kiểm thử?**

**a. Test Report**

Là tài liệu, một sản phẩm chứa các thông tin tổng hợp với các chỉ số kiểm thử nhằm cung cấp cho người đọc biết về trạng thái các nhiệm vụ, các công việc kiểm thử đã thực hiện.

**b. Khi nào thực hiện báo cáo kiểm thử?**

Hàng ngày, Hàng tuần, Kết thúc một giai đoạn, Kết thúc một mốc kiểm thử

**9. Kiểm thử theo trạng thái là gì (State Transition)?**

- Kiểm thử chuyển đổi trạng thái (State Transition Testing) được định nghĩa là kỹ thuật kiểm thử phần mềm trong đó thay đổi điều kiện đầu vào gây ra thay đổi trạng thái trong Ứng dụng được kiểm thử.

- Là phương pháp quan sát theo dõi sự chuyển từ trạng thái này sang trạng thái khác khi có tác động gì đó tới chương trình phần mềm.

**10. Kiểm thử hộp đen?**

Kiểm thử chức năng là một loại kiểm thử hộp đen (black box) và test case của nó được dựa trên đặc tả của ứng dụng phần mềm/thành phần đang test. Các chức năng được test bằng cách nhập vào các giá trị và kiểm tra kết quả đầu ra, và ít quan tâm đến cấu trúc bên trong của ứng dụng. Điều kiện kiểm thử hộp đen? Phần mềm phải chạy được.

**11. Kiểm thử hộp trắng?**

- Kiểm thử Hộp Trắng (còn gọi là Clear Box Testing, Open Box Testing, Glass Box Testing, Transparent Box Testing, Code-Based Testing hoặc Structural Testing) là một phương pháp kiểm thử phần mềm trong đó tester biết về cấu trúc nội bộ / thiết kế.

- Người kiểm tra chọn đầu vào để thực hiện các đường dẫn thông qua mã và xác định đầu ra thích hợp. Kiến thức lập trình và kiến thức thực hiện là rất cần thiết trong kiểm thử hộp trắng.

- Kiểm thử hộp trắng bao gồm phân tích dòng dữ liệu, điều khiển dòng, dòng thông tin, mã thực hành, ngoại lệ và những lỗi trình bày trong hệ thống để kiểm tra những hành động của phần mềm không được định hướng trước.

**12. Static và Dynamic Testing là gì?**

**a. Static Testing- Thử nghiệm tĩnh**

- Là loại kiểm tra trong đó code không được thực hiện. Nó có thể được thực hiện bằng tay hoặc bằng một bộ công cụ.

- Loại kiểm tra này thực hiện kiểm tra code, tài liệu yêu cầu và tài liệu thiết kế và đưa ra nhận xét, lưu nhận xét vào tài liệu công việc.

- Khi phần mềm không thực thi và không làm gì, chúng ta thực hiện kiểm tra trạng thái an toàn để phân tích phần mềm trong môi trường không chạy.

- Với thử nghiệm tĩnh, chúng ta cố gắng tìm ra lỗi, các lỗi code và mã độc tiềm ẩn trong ứng dụng phần mềm.

- Static test bắt đầu sớm hơn trong vòng đời phát triển, và do đó nó còn được gọi là thử nghiệm xác minh (verification testing).

- Kiểm tra tĩnh có thể được thực hiện trên các tài liệu làm việc như yêu cầu kỹ thuật (requirement), tài liệu thiết kế (documents), mã nguồn (Source code), kế hoạch kiểm tra (Test plan), kịch bản thử nghiệm (test script) và các trường hợp thử nghiệm (test case), nội dung trang web.

**b. Dynamic Testing- Thử nghiệm động**

- Được thực hiện khi code đang ở chế độ thực thi.

- Thử nghiệm động được thực hiện trong môi trường thực thi chạy chương trình ứng dụng. Khi code được thực thi, thì đầu vào được truyền một giá trị, kết quả hoặc đầu ra của việc thực hiện được so sánh với kết quả dự kiến ban đầu đã đưa ra.

- Với việc này chúng ta có thể quan sát được các hành vi chức năng của phần mềm, giám sát hệ thống bộ nhớ, thời gian phản hồi của CPU, hiệu suất của hệ thống.

- Thử nghiệm dynamic còn được gọi là thử nghiệm xác nhận (Validation testing), đánh giá sản phẩm.

- Thử nghiệm động gồm hai loại: Kiểm tra chức năng và Kiểm tra phi chức năng.

**13. File Test case là gì? Cấu trúc 1 Test Case cần bao gồm các thành phần chính nào?**

**a. Định nghĩa**

Là file tiến hành thực hiện kiểm thử dựa trên các Test Case đã được thiết kế trên Test Design.

b. **Cấu trúc 1 Test Case:**

· ID (Mã chức năng)

· Test Title

· Test Case Description (Mô tả trường hợp)

· Test Case Procedure (Thủ tục thực hiện hay có thể dùng là Test Case Steps - Các bước thực hiện)

· Test Data (Dữ Liệu Truyền Vào Test)

· Expected Output (Kết quả mong muốn)

· Actual Result (Kết quả thực tế)

· Inter-test case Dependence (Những Test case liên quan để có thể thực hiện được test case này)

· Result (Pass/Fail)

· Test date (Ngày thực hiện)

· Note (Ghi Chú)

**14. 3 nhân tố tạo nên chất lượng sản phẩm phần mềm là gì?**

**a. 3 nhân tố**

CON NGƯỜI - CÔNG CỤ - QUY TRÌNH

**b. Phần mềm phát sinh lỗi có ảnh hưởng gì không hãy nêu ra một vài ảnh hưởng?**

* Mất tiền để xử lý
* Mất thời gian để xử lý
* Gây ra tổn thương với con người: như gây chết người và thiệt hại lớn về kinh tế.
* Ảnh hưởng đến uy tín của công ty

**15. Quy trình kiểm thử như thế nào?**

Tập hợp tất cả các hoạt động, các phương thức mà con người phải thực hiện để thực hiện kiểm thử một hệ thống hay một sản phẩm phần mềm.

* Lập kế hoạch và kiểm soát việc kiểm thử
* Phân tích yêu cầu và Thiết kế testcase
* Thực thi – Chạy test
* Đánh giá tiêu chí dừng test và làm báo cáo
* Đóng hoạt động kiểm thử

**16. Giới thiệu các tài liệu liên quan đến kiểm thử mà em đã được học?**

1. **SRS (Software Requirements Specification)**: SRS là tài liệu mô tả chi tiết các yêu cầu chức năng và phi chức năng của hệ thống phần mềm.
2. **HLD (High-Level Design)**: HLD là một tài liệu tóm tắt mô tả cách mà hệ thống sẽ được thiết kế.
3. **Test Design**: Test Design là quá trình lập kế hoạch và tạo ra các ca kiểm thử và dữ liệu kiểm thử.
4. **Test Case**: Test Case là một tài liệu chi tiết mô tả các bước cần thực hiện để kiểm tra một tính năng hoặc hành vi cụ thể của hệ thống.
5. **Test Plan**: Test Plan là một tài liệu tổng quan về kế hoạch kiểm thử, bao gồm phạm vi, mục tiêu, tài nguyên và lịch trình.
6. **Test Report**: Test Report là một tài liệu tổng hợp kết quả của quá trình kiểm thử, bao gồm kết quả kiểm thử và các vấn đề phát hiện.

**17. SRS hoặc HLD là loại tài liệu gì và dùng để làm gì? Nó có quan trọng đối với kiểm thử không?**

* SRS và HLD đều là loại tài liệu trong quy trình phát triển phần mềm:

- SRS (Software Requirements Specification) là tài liệu mô tả yêu cầu của phần mềm một cách chi tiết và rõ ràng. Nó mô tả các chức năng, hiệu suất, giao diện người dùng, và các yêu cầu khác đối với phần mềm được phát triển.

- HLD (High-Level Design) là tài liệu mô tả thiết kế tổng quan của phần mềm, bao gồm cấu trúc tổng thể của hệ thống, các thành phần chính và cách chúng tương tác với nhau.

Cả hai loại tài liệu này đều rất quan trọng trong quy trình phát triển phần mềm. SRS giúp định nghĩa rõ ràng các yêu cầu của hệ thống, trong khi HLD giúp định hình cách thức triển khai các yêu cầu đó vào một kiến trúc cụ thể.

Đối với kiểm thử, SRS và HLD đều cực kỳ quan trọng. SRS cung cấp cơ sở để xác định các kịch bản kiểm thử và tạo ra các ca kiểm thử. HLD giúp kiểm thử viên hiểu cách hệ thống hoạt động và tương tác với các thành phần khác nhau.

* Test Case là một bộ các bước cụ thể để kiểm tra tính năng hoặc yêu cầu của phần mềm. Mỗi test case mô tả một trường hợp cụ thể mà phần mềm phải xử lý và kết quả mong đợi của nó. Test case được sử dụng để đảm bảo rằng phần mềm hoạt động đúng và đáp ứng được các yêu cầu đã được xác định trong SRS và HLD.

**18. Test Plan là gì?**

**a. Kế hoạch kiểm thử**

là một tài liệu mô tả chi tiết các mục tiêu, thị trường đích, và các quy trình cho một thử nghiệm cụ thể cho một sản phẩm phần mềm hay phần cứng. Thông thường kế hoạch này chứa đựng hiểu biết chi tiết về luồng làm việc sẽ xảy ra.

**b. Nội dung trong Test Plan hãy liệt kê ra các mục quan trọng?**

* Các yêu cầu cho Test.
* Chiến lược Test.
* Tài nguyên.
* Các mốc kiểm soát của giai đoạn.
* Các sản phẩm phải bàn giao.

**19. Test Design là gì?**

**a. Định nghĩa**

Test Design (thiết kế kiểm thử) là quá trình lập kế hoạch và tạo ra các ca kiểm thử và các bộ dữ liệu kiểm thử để đảm bảo rằng sản phẩm phần mềm hoạt động đúng như mong đợi và đáp ứng các yêu cầu của khách hàng.

**b. Ví dụ? GUI Test Case là gì?**

- GUI Test Case (Graphical User Interface Test Case) là một ca kiểm thử được thiết kế để kiểm tra giao diện người dùng đồ họa của một ứng dụng phần mềm. Trong kiểm thử GUI, các nhà phát triển và kiểm thử viên kiểm tra các thành phần như các nút, hộp thoại, menu, thanh công cụ và các phần khác của giao diện người dùng để đảm bảo rằng chúng hoạt động đúng và phản hồi đúng cách với người dùng.

**20. Test report dùng để làm gì? Error lỗi này đến từ đâu? Defect là gì?**

**a. Test Report dùng để**

báo cáo lỗi, Đánh giá các kết quả kiểm thử, thống kê các yêu cầu thay đổi, Tính và phân phối các thông tin đo lường hoạt động kiểm thử, Tạo bảng tổng kết đánh giá hoạt động kiểm lỗi, Xác định xem đã đạt tiêu chí thành công và hoàn thành kiểm thử chưa.

**b. Error lỗi này đến từ đâu?**

Hành động của con người gây ra một kết quả không đúng. Bởi các lý do sau: Do một số nhầm lẫn trong việc tìm hiểu chức năng của phần mềm – tính toán sai giá trị khởi tạo.... (Viết chương trình xong không thể biên dịch)

**c. Defect là gì?**

mang ý nghĩa là bạn đã gặp trường hợp **không dự tính** trước được.

**21. Kiểm thử Functional là gì? Kiểm thử Non-functional?**

**a. Kiểm thử Functional**

Kiểm thử chức năng nhằm đảm bảo chức năng phần mềm được vận hành theo đúng mục đích (requirements) đưa ra.

**b. Kiểm thử Non-functional?**

Kiểm thử phi chức năng là một loại kiểm thử để kiểm tra các khía cạnh phi chức năng (hiệu suất, khả năng sử dụng, độ tin cậy, v.v.) của một ứng dụng phần mềm. Nó được thiết kế để kiểm tra sự sẵn sàng của một hệ thống theo các tham số không chức năng không bao giờ được giải quyết bằng thử nghiệm chức năng.

**Liệt kê ra 5 Non-functional mà em biết?**

* Hiệu suất (Performance)
* Bảo mật (Security)
* Usability (Tính khả dụng)
* Functionality (Chức năng)
* UI (giao diện)

**22. Test Progress Report là gì?**

Báo cáo tiến độ test. Phản hồi từ nhóm kiểm thử, Cung cấp kết quả kiểm thử, Xác định thời gian hoàn thành kiểm thử.

**23. Quy trình xử lý lỗi như nào?**

* Bước 1: Ghi nhận lỗi (Log)
* Bước 2: Phân tích lỗi (Analyze)
* Bước 3: Sửa lỗi (Fix)
* Bước 4: Kiểm thử lại nội dung. (Re-test)

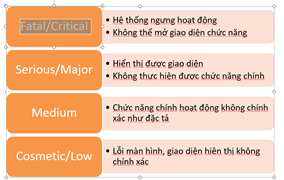
**Defect log cần phải bao gồm những gì?**

- Defect ID - Description - Defect title - Severity - Status- Assigned to

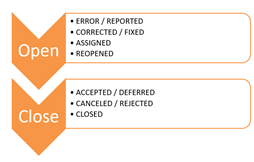


**24. Mức độ nghiêm trọng của lỗi được đánh giá như nào? Trạng thái của lỗi có mấy trạng thái?**

**a. Mức độ nghiêm trọng của lỗi - Severity**



**b. Trạng thái của lỗi có mấy trạng thái?**



**25. Test planning, Test control, Test closure là gì hãy giải thích?**

**a. Test planning**

Bản kế hoạch kiểm thử bao gồm Phạm vi kiểm thử, Phương pháp kiểm thử, Nguồn lực, Kế hoạch thực hiện

**b. Test Control**

Kiểm soát được trạng thái của quá trình kiểm thử Đưa ra được các hành động kịp thời để kiểm soát được và đạt được mục tiêu công việc hay dự án.

**c. Test closure**

Hoạt động thu thập dữ liệu từ các hoạt động kiểm thử, tổng hợp các kinh nghiệm dựa trên việc kiểm tra và hoàn thiện bộ sản phẩm kiểm thử.

**26. 7 nguyên tắc trong kiểm thử là gì?**

