**MỤC TIÊU:**

Kết thúc bài thực hành này bạn có khả năng

* Trình bày được các khái niệm trong Kiểm thử phần mềm
* Trình bày được 7 nguyên lý căn bản trong kiểm thử phần mềm
* Áp dụng được quy trình và giai đoạn kiểm thử phần mềm
* Vận dụng được tâm lý học trong kiểm thử phần mềm

# PHẦN I

## Bài 1 (2 điểm)

Sau khi đã học và hiểu vì sao phải kiểm thử phần mềm thì trước đây trong quá trình làm việc với những dự án hay bài tập lớn do nhà trường yêu cầu bạn đã không áp dụng khái niệm, nguyên lý, quy trình gì khi thực hiện phát triển phần mềm ?

Khi bạn đã học và hiểu về tầm quan trọng của kiểm thử phần mềm, bạn sẽ nhận ra rằng không áp dụng các khái niệm, nguyên lý và quy trình trong quá trình phát triển phần mềm có thể dẫn đến nhiều vấn đề và rủi ro. Dưới đây là một số khía cạnh mà bạn có thể đã bỏ qua:

1. **Thiết kế Trước**: Một phần quan trọng của quy trình phát triển phần mềm là việc lập kế hoạch và thiết kế trước khi bắt đầu viết mã. Thiết kế đảm bảo rằng bạn đã hiểu được yêu cầu của dự án và đã xác định cách thức hiện chúng một cách hiệu quả.
2. **Quản lý Phiên Bản**: Quản lý phiên bản là một khía cạnh quan trọng để theo dõi sự thay đổi trong mã nguồn và duy trì phiên bản ổn định của phần mềm.
3. **Kiểm Định Yêu Cầu**: Đảm bảo rằng bạn đã kiểm định yêu cầu từ người dùng và đã biết cách kiểm tra xem phần mềm có đáp ứng chúng hay không.
4. **Kiểm Thử Đơn Vị**: Kiểm thử đơn vị là việc kiểm tra từng phần của mã nguồn để đảm bảo tính chính xác và hoạt động như mong muốn.
5. **Kiểm Thử Hệ Thống**: Trước khi phát hành sản phẩm, kiểm thử hệ thống giúp đảm bảo rằng toàn bộ hệ thống làm việc đúng cách và tương tác hợp lý với nhau.
6. **Kiểm Thử Tự Động**: Sử dụng các công cụ kiểm thử tự động để giảm thiểu thời gian kiểm thử và đảm bảo tính ổn định của phần mềm trong quá trình phát triển.
7. **Kiểm Thử Bảo Mật**: Đảm bảo rằng bạn đã xem xét vấn đề bảo mật và đã thực hiện các biện pháp để bảo vệ phần mềm khỏi các lỗ hổng bảo mật.
8. **Tài Liệu và Ghi Chú**: Lập tài liệu về mã nguồn, cài đặt, và quy trình kiểm thử để dễ dàng theo dõi và hiểu quy trình phát triển phần mềm.
9. **Quản Lý Lỗi**: Theo dõi và quản lý lỗi trong quá trình phát triển để đảm bảo chúng được sửa lại kịp thời.
10. **Kiểm Tra Hiệu Năng**: Kiểm tra hiệu năng của phần mềm để đảm bảo rằng nó hoạt động ổn định trong điều kiện tải cao.

Việc áp dụng các khái niệm, nguyên lý và quy trình này giúp đảm bảo rằng phần mềm được phát triển một cách chất lượng và đáng tin cậy, đồng thời giảm thiểu rủi ro và sự cố sau khi sản phẩm được triển khai.

## Bài 2 (2 điểm)

Công ty XY muốn làm ra phần mềm quản lý nhân sự có tất cả đầy đủ tính năng và đảm bảo 95% các tính năng đều vận hành tốt không xảy ra lỗi. Bạn hãy đánh giá xem khi nào là Xác nhận(Validation) và Xác minh(Verification)

L

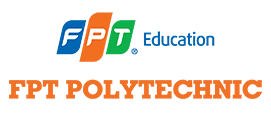
AB

1:

G

IỚI THIỆU VÀ KHÁI QUÁT CÁC KHÁI NIỆM VỀ KIỂM

THỬ PHẦN MỀM



Trong ngữ cảnh của dự án phần mềm quản lý nhân sự, "Xác nhận (Validation)" và "Xác minh (Verification)" là hai khái niệm quan trọng liên quan đến kiểm tra tính năng và chất lượng của sản phẩm. Dưới đây là cách bạn có thể đánh giá khi nào là Xác nhận và khi nào là Xác minh:

1. **Xác minh (Verification)**:

Xác minh là quá trình kiểm tra xem bạn đang xây dựng sản phẩm đúng cách theo yêu cầu không. Nó liên quan đến việc thực hiện kiểm tra bằng cách sử dụng tài liệu, quy trình, và công cụ kiểm tra để đảm bảo rằng phần mềm đang được phát triển theo đúng quy trình và đáp ứng yêu cầu thiết kế.

Trong trường hợp này, việc xác minh sẽ bao gồm kiểm tra xem phần mềm quản lý nhân sự đang được phát triển theo các yêu cầu được đặt ra ban đầu, như tính năng, giao diện, và quy trình làm việc có được triển khai đúng cách hay không.

1. **Xác nhận (Validation)**:

Xác nhận là quá trình kiểm tra xem sản phẩm cuối cùng có đáp ứng mong đợi của khách hàng không. Nó liên quan đến việc đảm bảo rằng sản phẩm đã được xây dựng đúng cách để đáp ứng nhu cầu và mong muốn thực tế của người dùng.

Trong trường hợp này, việc xác nhận sẽ bao gồm thử nghiệm sản phẩm quản lý nhân sự thực tế bằng cách đưa nó vào môi trường hoạt động và kiểm tra xem tính năng của sản phẩm có hoạt động một cách đáng tin cậy (vận hành tốt) và không xảy ra lỗi trong tình huống thực tế không.

Vì vậy, bạn có thể xác minh trong quá trình phát triển để đảm bảo rằng bạn đang xây dựng sản phẩm đúng cách, sau đó sau khi hoàn thành sản phẩm, bạn sẽ xác nhận bằng cách thử nghiệm nó trong môi trường thực tế để đảm bảo tính năng vận hành tốt và không gặp lỗi.

**Bài 3 (1 điểm)**

Giảng viên cho thêm

# PHẦN II

## Bài 4 (2 điểm)

Tuần 1: Bạn tìm thấy khiếm khuyết với mức độ nghiêm trọng 1 vào ngày 1 và việc kiểm thử hoàn chỉnh bị chặn trong 3 ngày. Do đó bạn không thể thực hiện

## KIỂM THỬ CƠ BẢN TRANG 1

bất kỳ kịch bản cho đến khi khuyếm khuyết mức độ nghiêm trọng 1 đó được giải quyết. Sau khi mất 3 ngày giải quyết nó, bạn tiếp tục với thực hiện việc kiểm thử của mình. Vào cuối tuần, bạn hoàn thành 20 kịch bản với nhiều khiếm khuyết hơn. Tuần này vẫn giống như kịch bản 1.

Tuần 2: Bạn tiếp tục có một vài khiếm khuyết mức độ nghiêm trọng 2 và mức độ nghiêm trọng 3 trong tuần thứ hai, do đó bạn không thể bao phủ hết kịch bản để đã bị tồn đọng từ tuần 1.

Với trường hợp trên hãy liệt kê các tiêu chí kết thúc chưa hoàn thành.

### Bài 5 (2 điểm)

Một nhóm kiểm thử luôn tìm ra số lượng lớn lỗi trong suốt quá trình phát triển, kể cả kiểm thử hệ thống. Mặc dù người quản lý kiểm thử hiểu rằng việc phát hiện lỗi này khá tốt trong phạm vi ngân sách của nhóm kiểm thử cô phụ trách, song lãnh đạo cấp cao vẫn chưa hài lòng, phàn nàn rằng nhóm kiểm thử đã bỏ sót vài lỗi mà người dùng phát hiện ra sau khi chuyển giao sản phẩm. Giả dụ người dùng nhìn chung là hài lòng với hệ thống và hỏng hóc chỉ có ít ảnh hưởng, thì nguyên lý kiểm thử nào có thể giúp người quản lý kiểm thử giải thích với lãnh đạo về lý do bỏ sót một số lỗi?

L

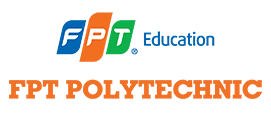
AB

1:

G

IỚI THIỆU VÀ KHÁI QUÁT CÁC KHÁI NIỆM VỀ KIỂM

THỬ PHẦN MỀM



guyên lý kiểm thử quan trọng có thể giúp người quản lý kiểm thử giải thích về lý do bỏ sót một số lỗi sau khi chuyển giao sản phẩm là "Nguyên lý bất ổn (Heisenbug)". Nguyên lý này có thể được sử dụng để giải thích tại sao một số lỗi không được phát hiện trong quá trình kiểm thử và chỉ xuất hiện sau khi sản phẩm đã được triển khai trong môi trường thực tế. Dưới đây là mô tả chi tiết về nguyên lý này:

**Nguyên lý Bất ổn (Heisenbug):**

Nguyên lý này xuất phát từ tên của nhà vật lý Werner Heisenberg và áp dụng vào ngữ cảnh kiểm thử phần mềm. Nguyên lý Bất ổn ám chỉ rằng quá trình kiểm thử có thể làm thay đổi hành vi của phần mềm. Cụ thể, khi một nhóm kiểm thử thực hiện kiểm thử, họ có thể tạo ra môi trường kiểm thử khác với môi trường thực tế, làm cho một số lỗi không xuất hiện trong quá trình kiểm thử ban đầu.

Nguyên lý Bất ổn giúp giải thích tại sao một số lỗi chỉ xuất hiện sau khi sản phẩm được triển khai trong môi trường thực tế, nơi mà các yếu tố bên ngoài như dữ liệu thực tế, tải trọng sử dụng, môi trường hệ thống thực tế, và các yếu tố khác có thể tác động lên hệ thống một cách không thể dự đoán được trong quá trình kiểm thử. Nhóm kiểm thử có thể không phát hiện những lỗi này trong môi trường kiểm thử của họ vì chúng không phản ánh môi trường hoạt động thực tế.

Khi giải thích với lãnh đạo về việc bỏ sót một số lỗi, người quản lý kiểm thử có thể sử dụng nguyên lý Bất ổn để thể hiện rằng các lỗi này là khó dự đoán và không thể phát hiện trong môi trường kiểm thử cụ thể, và chúng chỉ xuất hiện khi hệ thống hoạt động trong môi trường thực tế. Điều này có thể giúp làm rõ tại sao một số lỗi vẫn tồn tại sau khi sản phẩm được triển khai.

### Bài 6 (1 điểm)

Giảng viên cho thêm

## KIỂM THỬ CƠ BẢN TRANG 2