TRƯỜNG ĐẠI HỌC KINH TẾ

**KHOA THỐNG KÊ – TIN HỌC**

–––––––––––––––––––––––––––––––

**BÁO CÁO THỰC TẬP NGHỀ NGHIỆP**

**ỨNG DỤNG SENELIUM TRONG KIỂM THỬ TỰ ĐỘNG CHỨC NĂNG ĐẶT HÀNG CỦA WEBSITE FAST PRINT**

**Sinh viên thực hiện: Hồ Thanh Tâm**  **Lớp : 44k21.1**

**Giảng viên hướng dẫn:** Cao Thị Nhâm

**Đà Nẵng, 6/202**

**LỜI CẢM ƠN**

**LỜI CAM ĐOAN**

Quy định:

Soạn thảo trên trang A4 (trang dọc, lề trái: 3.5cm; trên, phải, dưới: 2.5 cm), Font Times New Roman, canh đều 2 bên, size 13, cách dòng 1.5, cách đoạn trên 6pt, cách đoạn dưới 3pt; hình và bảng soạn theo caption, chèn trích dẫn chéo (Cross-reference) cho bảng và hình; các danh mục hình, bảng, mục lục làm tự động; đánh số trang như file mẫu (bìa không có số trang, danh mục + mục lục số trang theo i, ii, iii…, nội dung chính theo 1,2,3…)

# MỤC LỤC

[MỤC LỤC ii](#_Toc74643748)

[LỜI MỞ ĐẦU 1](#_Toc74643749)

[CHƯƠNG 1. CƠ SỞ LÝ THUYẾT 2](#_Toc74643750)

[1.1. Tổng quan về .NET framework 2](#_Toc74643751)

[1.1.1. Mục 1.1.1 2](#_Toc74643752)

[1.1.2. Mục 1.1.2 2](#_Toc74643753)

[1.2. Tổng quan về Windows Forms 2](#_Toc74643754)

[1.3. Tổng quan về hệ quản trị cơ sở dữ liệu Microsoft SQL Server 2](#_Toc74643755)

[CHƯƠNG 2. phân tích và thiết kế ứng dụng 3](#_Toc74643756)

[2.1. Mô tả bài toán 3](#_Toc74643757)

[2.2. Thiết kế các chức năng 3](#_Toc74643758)

[2.3. Thiết kế cơ sở dữ liệu 3](#_Toc74643759)

[CHƯƠNG 3. PHÁT TRIỂN ỨNG DỤNG 4](#_Toc74643760)

[3.1. Sơ đồ chức năng 4](#_Toc74643761)

[3.2. Thông tin chi tiết các form làm việc 4](#_Toc74643762)

[3.2.1. Form X 4](#_Toc74643763)

[3.2.2. Form Y 4](#_Toc74643764)

[CHƯƠNG 4. KẾT QUẢ 5](#_Toc74643765)

[KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 6](#_Toc74643766)

[TÀI LIỆU THAM KHẢO 7](#_Toc74643767)

[PHỤ LỤC (nếu có) 8](#_Toc74643768)

DANH MỤC HÌNH ẢNH

[Hình 1.1 Ngôn ngữ lập trình Python 2](#_Toc74235469)

[Hình 4.1 Kiến trúc của mô hình RNN 5](#_Toc74235470)

DANH MỤC BẢNG BIỂU

[Bảng 4.1 Kiến trúc 5](#_Toc74235471)

DANH MỤC CÁC TỪ VIẾT TẮT

**AI** : Artificial Intelligence

**…**

# LỜI MỞ ĐẦU

1. **Mục tiêu của đề tài**

* Nghiên cứu ngôn ngữ lập trình C# (Windows Forms)
* Nghiên cứu hệ quản trị CSDL Microsoft SQL Server 2016
* Xây dựng ứng dụng desktop để quản lý XXXX

1. **Nhiệm vụ của đề tài**

* Nghiên cứu ngôn ngữ lập trình C# (Windows Forms)
* Nghiên cứu Microsoft SQL Server 2016
* Phân tích bài toán quản lý XXXX
* Phát triển ứng dụng để quản lý XXXX

1. **Kết cấu của đề tài**

Đề tài được tổ chức gồm 4 chương nội dung và phần kết luận.

* **Chương 1**: Cơ sở lý thuyết
* **Chương 2**: Phân tích và thiết kế ứng dụng
* **Chương 3**: Phát triển ứng dụng
* **Chương 4**: Kết quả
* Kết luận và hướng phát triển

# CƠ SỞ LÝ THUYẾT

## Tổng quan về phần mềm

### Phần mềm là gì

Phần mềm máy tính, hay đơn giản là phần mềm, cũng được mọi người gọi là tập hợp dữ liệu hoặc hướng dẫn cho máy tính biết cách làm việc. Điều này trái ngược với phần cứng vật lý, từ đó hệ thống được xây dựng và thực sự thực hiện công việc.

### Vòng đời phát triển phần mềm

Quy trình phát triển phần mềm (SDLC) là quá trình lập kế hoạch, tạo, thử nghiệm và triển khai một hệ thống thông tin. Một số mô hình SDLC hoặc các phương pháp đã được tạo ra, chẳng hạn như thác nước, xoắn ốc, phát triển phần mềm Agile, tạo mẫu nhanh, gia tăng và đồng bộ hóa và ổn định. SDLC được sử dụng trong quá trình phát triển một dự án CNTT, nó mô tả các giai đoạn khác nhau liên quan đến dự án từ bản vẽ, thông qua việc hoàn thành dự án.



**1.1.2.1. Lập kế hoạch, phân tích sơ bộ**

Ở giai đoạn 1 là tiến hành lập kế hoạch và phân tích sơ bộ, đưa ra các giải pháp thay thế, mô tả chi phí và lợi ích và đưa ra một kế hoạch sơ bộ.

• Tiến hành phân tích sơ bộ: trong bước này, cần phải tìm ra các mục tiêu của tổ chức và bản chất và phạm vi của vấn đề đang nghiên cứu. Sau đó, cần xem vấn đề đang được nghiên cứu phù hợp như thế nào.

• Đề xuất các giải pháp thay thế: Trong quá trình đào sâu vào các mục tiêu của tổ chức và các vấn đề cụ thể, có thể đã đề cập đến một số giải pháp. Các đề xuất thay thế có thể đến từ phỏng vấn nhân viên, khách hàng, nhà cung cấp và / hoặc chuyên gia tư vấn. Với dữ liệu này, sẽ có ba lựa chọn: Giữ nguyên hệ thống, cải thiện nó, hoặc phát triển một hệ thống mới.

• Mô tả chi phí và lợi ích.

* + - 1. **Phân tích hệ thống, định nghĩa yêu cầu**

Xác định các mục tiêu của dự án thành các chức năng và hoạt động của ứng dụng. Đó là quá trình thu thập và giải thích các sự kiện, chẩn đoán các vấn đề và đề xuất cải tiến hệ thống. Phân tích nhu cầu thông tin của người dùng cuối và cũng loại bỏ bất kỳ sự không nhất quán và không đầy đủ trong các yêu cầu này.

* Thu thập dữ liệu: Các yêu cầu của người dùng cuối có được thông qua tài liệu, phỏng vấn của khách hàng, quan sát và bảng câu hỏi,
* Kiểm tra hệ thống hiện tại: Xác định ưu và nhược điểm của hệ thống hiện, để chuyển các ưu và tránh những khuyết điểm trong hệ thống mới.
* Phân tích hệ thống được đề xuất: Giải pháp cho những thiếu sót trong bước hai được tìm thấy và bất kỳ đề xuất người dùng cụ thể được sử dụng để chuẩn bị các thông số kỹ thuật.

**1.1.2.3 . Thiết kế hệ thống**

Trong giai đoạn này, thiết kế hệ thống và phần mềm được chuẩn bị từ các thông số kỹ thuật yêu cầu đã được nghiên cứu trong giai đoạn đầu tiên. Kiến trúc hệ thống liên quan đến việc bảo đảm rằng hệ thống phần mềm sẽ đáp ứng đầy đủ các yêu cầu của sản phẩm, cũng như đảm bảo rằng các yêu cầu trong tương lai có thể được giải quyết. Nó cũng liên quan đến việc giao tiếp giữa các hệ thống phần mềm và các sản phẩm phần mềm khác, cũng như các phần cứng cơ bản hoặc các hệ điều hành chủ. Các thông số kỹ thuật thiết kế hệ thống phục vụ như đầu vào cho giai đoạn tiếp theo của mô hình.

* + - 1. **Phát triển**

Khi nhận được tài liệu thiết kế hệ thống, công việc được chia thành các mô-đun / đơn vị và việc mã hóa được bắt đầu. Vì, trong giai đoạn này mã được tạo ra nên nó là trọng tâm chính cho phát triển phần mềm. Đây là giai đoạn dài nhất của vòng đời phát triển phần mềm.

**1.1.2.5. Tích hợp và thử nghiệm**

Sau khi code xong, mã nguồn được kiểm thử dựa trên yêu cầu đã được thu thập trong giai đoạn đầu và dựa trên tính khả dụng người dùng. Trong giai đoạn này tất cả các loại kiểm thử chức, kiểm tra chấp nhận được thực hiện cùng với các kiểm thử phi chức năng cũng được thực hiện. Sau đây là các loại thử nghiệm thường dùng, tùy thuộc vào loại hệ thống đang được phát triển:

* Defect testing the failed scenarios, including defect tracking
* Path testing
* Data set testing
* Unit testing
* System testing
* Integration testing
* Black-box testing
* White-box testing
* Regression testing
* Automation testing
* User acceptance testing
* Software performance testing
  + - 1. **Cài đặt, triển khai**

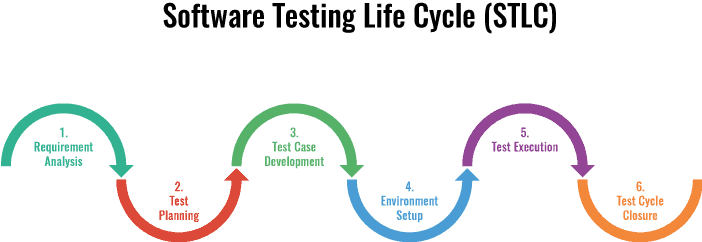
Ngay sau khi sản phẩm được cung cấp cho khách hàng, trước tiên họ sẽ thực hiện kiểm thử beta. Nếu có bất kỳ thay đổi nào được yêu cầu hoặc nếu bất kỳ lỗi nào tồn tại, thì sẽ báo cáo cho nhóm kỹ thuật. Một khi những lỗi đó được khắc phục hoặc các lỗi là cố định thì việc triển khai cuối cùng sẽ xảy ra.

* + - 1. **Bảo trì**

Sau khi đưa sản phẩm phần mềm cho khách hàng sử dụng nếu phát sinh lỗi và cần giải quyết theo thời gian hoặc bị lỗi thời thì sẽ được đưa lại bộ phận chăm sóc để khắc phục.

## Tổng quan về kiểm thử phần mềm

### Quy trình kiểm thử phần mềm



Hình 3‑2 Quy trình kiểm thử phần mềm

Quy trình kiểm thử phần mềm xác định các giai đoạn/ pha trong kiểm thử phần mềm. Tuy nhiên, không có STLC tiêu chuẩn cố định nào trên thế giới, nhưng về cơ bản quy trình kiểm thử bao gồm những giai đoạn sau:

* Requirenment analysis - Phân tích yêu cầu
* Test planning - Lập kế hoạch kiểm thử
* Test case development - Thiết kế kịch bản kiểm thử
* Test environment set up - Thiết lập môi trường kiểm thử
* Test execution - Thực hiện kiểm thử
* Test cycle closure - Đóng chu trình kiểm thử

Các giai đoạn kiểm thử được thực hiện một cách tuần tự. Mỗi giai đoạn sẽ có những mục tiêu khác nhau, đầu vào và kết quả đầu ra khác nhau nhưng mục đích cuối cùng vẫn là đảm bảo chất lượng sản phẩm phần mềm tốt nhất.

* + - 1. **Requirenment analysis - Phân tích yêu cầu**
* Đầu vào

Đầu vào của giai đoạn phân tích yêu cầu bao gồm các tài liệu như: tài liệu đặc

tả yêu cầu, tài liệu thiết kế hệ thống, tài liệu khách hàng yêu cầu về các tiêu chí chấp nhận của sản phẩm, bản prototype của khách hàng yêu cầu(nếu có),...

* Hoạt động

Phân tích yêu cầu là giai đoạn đầu tiên trong quy trình kiểm thử phần mềm.

QA team sẽ thực hiện đọc hiểu, nghiên cứu và phân tích cụ thể các yêu cầu trong tài liệu đặc tả của dự án hoặc tài liệu khách hàng. Qua hoạt động này, QA team sẽ nắm bắt được các yêu cầu mà dự án đưa ra bao gồm yêu cầu kiểm thử chức năng/ phi chức năng nào.

* Đầu ra

Đầu ra của giai đoạn phân tích yêu cầu bao gồm tài liệu chứa các câu hỏi và câu trả lời liên quan đến nghiệp vụ của hệ thống, tài liệu báo cáo tính khả thi, phân tích rủi ro của việc kiểm thử phần mềm.

* + - 1. **Test planning - Lập kế hoạch kiểm thử**
* Đầu vào

Đầu vào của giai đoạn lập kế hoạch kiểm thử là các tài liệu đặc tả đã được cập nhật thông qua các câu hỏi và trả lời được đưa ra trong giai đoạn phân tích yêu cầu, tài liệu báo cáo tính khả thi, phân tích rủi ro của việc kiểm thử phần mềm.

* Hoạt động

Dựa vào các tài liệu được cung cấp và cập nhật mới nhất, thông thường, test manager hoặc test leader sẽ là người lập kế hoạch kiểm thử cho cả QA team. Lập kế hoạch kiểm thử nhằm xác định một số yếu tố quan trọng sau:

Xác định phạm vi (Scope) dự án: Dự án thực hiện trong thời gian bao lâu? Bao gồm những công việc gì cho từng khoảng thời gian xác định? Từ đó đưa ra lịch trình thực hiện cho từng công việc nhỏ sao cho phù hợp với toàn bộ đội dự án.

Xác định phương pháp tiếp cận: Dựa vào nhiều tiêu chí: Thời gian cho phép test có phù hợp với con số ước lượng, nhiều hay ít, yêu cầu chất lượng từ phía khách hàng thế nào? Cao, thấp hay khắc khe hay sao cũng được? Công nghệ / kỹ thuật sử dụng để phát triển ứng dụng này là gì? Lĩnh vực của hệ thống/sản phẩm đang được test (domain) là gì? ...Từ đó, test manager có thể đưa ra những phương pháp và kế hoạch phù hợp nhất cho cả quá trình thực hiện dự án sao cho đúng với các tiêu chí chấp nhận của sản phẩm và kịp tiến độ với các mốc thời gian bàn giao, phát hành.

Xác định các nguồn lực

* Con người: Bao nhiêu người tham gia dự án, ai sẽ test phần nào, bao nhiêu tester tham gia?Tester và nhóm phát triển có kinh nghiệm về lĩnh vực này không?
* Thiết bị: số lượng server, version, máy tính, mobile để thực hiện test là bao nhiêu.
* Lên kế hoạch thiết kế công việc test: Bản kế hoạch kiểm thử sẽ bao gồm các nội dung:
* Liệt kê các chức năng cần kiểm thử. Để thực hiện test chức năng này thì cần làm những công việc gì, trong thời gian bao lâu, cái nào thực hiện trước, cái nào thực hiện sau, ai là người thực hiện.
* Xác định điều kiện bắt đầu: xác định những điều kiện tối thiểu để bắt đầu hoạt động kiểm thử cho từng chức năng.
* Xác định điều kiện kết thúc: khi có những điều kiện nào thì sẽ kết thúc việc kiểm thử.
* Đầu ra

Đầu ra của giai đoạn lập kế hoạch bao gồm các tài liệu như test plan, test estimation, test schedule.

* + - 1. **Test case development - Thiết kế kịch bản kiểm thử**
* Đầu vào

Đầu vào của giai đoạn thiết kế kịch bản kiểm thử là test plan, test estimation, test schedule, các tài liệu đặc tả đã được cập nhật.

* Hoạt động

Review tài liệu: Đầu tiên, các kiểm thử viên cần review lại tất cả các tài liệu để xác định công việc cần làm, các công việc có khác gì so với dự án trước khách hàng đưa cho, chức năng nào cần test, chức năng nào không cần test lại nữa. Từ đó, vừa có thể tiết kiệm thời gian mà vẫn đưa ra được một kịch bản kiểm thử đầy đủ và hiệu quả.

Viết test case/ check list: Tester bắt tay vào việc viết test case chi tiết dựa vào kế hoạch đã đưa ra và vận dụng các kỹ thuật thiết kế kịch bản kiểm thử. Test case cần bao phủ được tất cả các trường hợp kiểm thử có thể xảy ra cũng như đáp ứng đầy đủ các tiêu chí của sản phẩm. Đồng thời tester cũng cần đánh giá mức độ ưu tiên cho từng test case.

Chuẩn bị dữ liệu kiểm thử: Chuẩn bị trước các dữ liệu kiểm thử cho các trường hợp cần thiết như test data, test script.

Review test case/ check list: Sau khi hoàn thành, các thành viên trong đội kiểm thử hoặc test leader cũng cần review lại test case đã tạo để có thể bổ sung, hỗ trợ lẫn nhau nhằm tránh những sai sót trong thiết kế test case và rủi ro về sau.

* Đầu ra

Sau khi hoàn thành thiết kế kịch bản kiểm thử, đội kiểm thử sẽ có các tài liệu bao gồm: test design, test case, check list, test data, test automation script.

* + - 1. **Test environment set up - Thiết lập môi trường kiểm thử**
* Đầu vào

Đầu vào của giai đoạn cài đặt môi trường kiểm thử là test plan, smoke test case, test data.

* Hoạt động

Việc cài đặt môi trường kiểm thử là giai đoạn cũng rất quan trọng trong vòng đời phát triển phần mềm. Môi trường kiểm thử sẽ được quyết định dựa trên những yêu cầu của khách hàng, hay đặc thù của sản phẩm ví dụ như server/ client/ network, ...

Tester cần chuẩn bị một vài test case để kiểm tra xem môi trường cài đặt đã sẵn sàng cho việc kiểm thử hay chưa. Đây chính là việc thực thi các smoke test case.

* Đầu ra

Đầu ra của giai đoạn này là môi trường đã được cài đặt đúng theo yêu cầu, sẵn sàng cho việc kiểm thử và kết quả của smoke test case.

* + - 1. **Test execution - Thực hiện kiểm thử**
* Đầu vào

Tài liệu đầu vào của giai đoạn này là test plan, test design, test case, check list, test data, test automation script.

* Hoạt động

Thực hiện các test case như thiết kế và mức độ ưu tiên đã đưa ra trên môi trường đã được cài đặt.

So sánh với kết quả mong đợi sau báo cáo các bug xảy ra lên tool quản lý lỗi và theo dõi trạng thái của lỗi đến khi được sửa thành công.

Thực hiện re-test để verify các bug đã được fix và regression test khi có sự thay đổi liên quan.

Trong quá trình thực hiện kiểm thử, kiểm thử viên cũng có thể hỗ trợ, đề xuất cho cả đội dự án để có giải pháp hợp lý và kết hợp công việc hiệu quả.

Đo lường và phân tích tiến độ: kiểm thử viên cần kiểm soát chặt chẽ tiến độ công việc của mình bằng cách so sánh tiến độ thực tế với kế hoạch, nếu chậm cần phải điều chỉnh sao cho kịp tiến độ dự án, nếu nhanh cũng cần điều chỉnh vì có thể test lead lên kế hoạch chưa sát với thực tế dự án. Từ đó có thể sửa chữa test plan cần điều chỉnh để phù hợp với tiến độ dự án đưa ra.

Report thường xuyên cho PM và khách hàng về tình hình thực hiện dự án: Cung cấp thông tin trong quá trình kiểm thử đã làm được những chức năng nào, còn chức năng nào, hoàn thành được bao nhiều phần trăm công việc, báo cáo các trường hợp phát sinh sớm, tránh ảnh hưởng tiến độ công việc của cả ngày.

* Đầu ra

Đầu ra của giai đoạn này là test results (kết quả kiểm thử), defect reports (danh sách các lỗi tìm được).

* + - 1. **Test cycle closure - Đóng chu trình kiểm thử**
* Đầu vào

Đầu vào của giai đoạn đóng chu trình kiểm thử là bao gồm tất cả những tài liệu liên quan đã được tổng hợp, ghi chép và hoàn thiện đầy đủ trong suốt quy trình kiểm thử của dự án: tài liệu phân tích đặc tả yêu cầu, test plan, test results, defect reports, tài liệu Q&A, ...

* Hoạt động

Đây là giai đoạn cuối cùng trong quy trình kiểm thử phần mềm.

Ở giai đoạn này, QA team thực hiện tổng kết, báo cáo kết quả về việc thực thi test case, bao nhiêu case pass/ fail, bao nhiêu case đã được fix, mức độ nghiêm trọng của lỗi, bao nhiêu lỗi cao/ thấp, lỗi còn nhiều ở chức năng nào, dev nào nhiều lỗi. Chức năng nào đã hoàn thành test/ chưa hoàn thành test/ trễ tiến độ bàn giao.

Đánh giá các tiêu chí hoàn thành như phạm vi kiểm tra, chất lượng, chi phí, thời gian, mục tiêu kinh doanh quan trọng.

Ngoài ra, giai đoạn này cũng thảo luận tất cả những điểm tốt, điểm chưa tốt và rút ra bài học kinh nghiệm cho những dự án sau, giúp cải thiện quy trình kiểm thử.

* Đầu ra

Đầu ra của giai đoạn này bao gồm các tài liệu: Test report, Test results (final)

### 7 nguyên tắc kiểm thử phần mềm

* + - 1. **Kiểm thử đưa ra lỗi**

Kiểm thử có thể cho thấy rằng phần mềm đang có lỗi, nhưng không thể chứng minh rằng phần mềm không có lỗi. Kiểm thử được thực hiện bằng những kĩ thuật khác nhau. Kiểm thử làm giảm xác suất lỗi chưa tìm thấy vẫn còn trong phần mềm, ngay cả khi đã kiểm thử nghiêm ngặt phần mềm vẫn có thể còn lỗi. Vì vậy chúng ta phải tìm được càng nhiều lỗi càng tốt.

* + - 1. **Kiểm thử toàn bộ là không thể**

Nguyên tắc này nói rằng kiểm tra mọi thứ trong phần mềm một cách trọn vẹn là không thể. Kiểm thử với tất cả các kết hợp đầu vào và đầu ra, với tất cả các kịch bản là không thể trừ phi nó chỉ bao gồm ít trường hợp thì có thể kiểm thử toàn bộ. Thay vì kiểm thử toàn bộ, việc phân tích rủi ro và dựa trên sự mức độ ưu tiên chúng ta có thể tập trung việc kiểm thử vào một số điểm cần thiết, có nguy cơ lỗi cao hơn.

* + - 1. **Kiểm thử càng sớm càng tốt**

Nguyên tắc này yêu cầu bắt đầu thử nghiệm phần mềm trong giai đoạn đầu của vòng đời phát triển phần mềm. Các hoạt động kiểm thử phần mềm từ giai đoạn đầu sẽ giúp phát hiện bug sớm hơn. Nó cho phép chuyển giao phần mềm theo yêu cầu đúng thời gian với chất lượng dự kiến.

* + - 1. **Sự tập trung của lỗi**

Thông thường, lỗi tập trung vào những module, thành phần chức năng chính của hệ thống. Nếu xác định được điều này bạn sẽ tập trung vào tìm kiếm lỗi quanh khu vực được xác định. Nó được coi là một trong những cách hiệu quả nhất để thực hiện kiểm tra hiệu quả.

* + - 1. **Nghịch lí thuốc trừ sâu**

Nếu bạn sử dụng cùng một tập hợp các trường hợp kiểm thử liên tục, sau đó một thời gian các trường hợp kiểm thử không tìm thấy lỗi nào mới. Hiệu quả của các trường hợp kiểm thử bắt đầu giảm xuống sau một số lần thực hiện, vì vậy luôn luôn chúng ta phải luôn xem xét và sửa đổi các trường hợp kiểm thử trên một khoảng thời gian thường xuyên.

* + - 1. **Kiểm thử phụ thuộc vào ngữ cảnh**

Theo nguyên tắc này thì việc kiểm thử phụ thuộc vào ngữ cảnh và chúng ta phải tiếp cận kiểm thử theo nhiều ngữ cảnh khác nhau Nếu bạn đang kiểm thử ứng dụng web và ứng dụng di động bằng cách sử dụng chiến lược kiểm thử giống nhau, thì đó là sai.

* + - 1. **Không có lỗi - Sai lầm**

Việc không tìm thấy lỗi trên sản phẩm không đồng nghĩa với việc sản phẩm đã sẵn sàng để tung ra thị trường. Việc không tìm thấy lỗi cũng có thể là do bộ trường hợp kiểm thử được tạo ra chỉ nhằm kiểm tra những tính năng được làm đúng theo yêu cầu thay vì nhằm tìm kiếm lỗi mới.

### Giai đoạn kiểm thử phần mềm

Kiểm thử phần mềm là một quá trình, được thực thi để đánh giá giao diện và chức năng của phần mềm đó nhằm mục đích tìm kiếm các lỗi đảm bảo rằng hệ thống không có khuyến khuyết và hệ thống phần mềm được tạo ra hoạt động theo đúng yêu cầu đặc tả.

Các giai đoạn của kiểm thử phần mềm:

* Unit testing
* Intergration testing
* System testing
* Acceptance testing
  + - 1. **Unit testing (kiểm thử mức đơn vị)**

Kiểm thử đơn vị là triển khai việc kiểm thử trên từng đơn vị nhỏ nhất của phần mềm. Kiểm thử đơn vị được thực hiện riêng lẻ cho từng module, kiểm tra từng đơn vị của ứng dụng một cách riêng biệt.

Phát hiện lỗi của hệ thống càng sớm thì chi phí để khắc phục lỗi càng thấp. Vậy nên nếu phát hiện lỗi ở giai đoạn kiểm thử đơn vị thì việc tìm ra nguyên nhân và khắc phục lỗi tương đối dễ, có thể khoanh vùng đơn vị mà tại đó lỗi xảy ra.

Kiểm thử đơn vị là kiểu kiểm thử hộp trắng. Người thực hiện việc kiểm thử đơn vị thường là lập trình viên.

* + - 1. **Intergration testing (kiểm thử tích hợp)**

Kiểm thử tích hợp là mức độ kiểm thử mà tại đó các module riêng lẻ được kết hợp lại với nhau và kiểm tra sự tương thích của chúng. Tích hợp: các hàm lại với nhau, tích hợp các màn hình dựa theo chức năng)

Kiểm tra tích hợp có hai mục tiêu chính đó là:

* Tìm ra lỗi khi tích hợp các module riêng lẻ.
* Tích hợp các module đơn lẻ thành một hệ thống nhỏ (hệ thống này đã có thể hoạt động) và làm tiền đề cho việc kiểm thử hệ thống.

Người thực hiện kiểm thử tích hợp thường là lập trình viên.

Các phương thức tiếp cận trong kiểm thử tích hợp:

* Top Down: Kiểm thử tích hợp Top Down (Kiểm thử từ trên xuống dưới) khi đó các module được tích hợp và kiểm tra từ các module cấp cao đến các module cấp thấp hơn, tức là: Kiểm tra các module chức các chức năng chính đến các module chứa các chức năng phụ.
* Button Up: Kiểm thử tích hợp Button Up (Kiểm thử từ dưới lên trên) khi đó các module được tích hợp và kiểm tra từ các module cấp thấp đến các module cấp cao hơn, tức là: kiểm tra các module phụ đến module chính.
* Sandwich / Hybrid là cách kiểm thử tích hợp, khi đó kết hợp phương pháp Top Down và Bottom Up.
  + - 1. **System testing (Kiểm thử hệ thống)**

Kiểm thử hệ thống là mức độ kiểm thử phần mềm, lúc này một phần mềm đã hoàn chỉnh và được tích hợp kiểm tra. Thực hiện kiểm tra về thiết kế và hệ thống thỏa mãn các yêu cầu đặt tả. Cần phải thực hiện kiểm thử đơn vị và kiểm thử tích hợp trước khi kiểm thử hệ thống để đảm bảo sự tương thích, kết hợp giữa các module với nhau.

Tại mức độ kiểm thử này chú trọng vào việc kiểm thử chức năng và phi chức năng của toàn hệ thống: hoạt động, thao tác, sự tin cậy. mức độ sự lý và các yêu cầu khác liên quan đến chất lượng đã được đề ra cho dự án.

Người thực hiện kiểm thử hệ thống thường là kiểm thử viên. Có thể là một hoặc một nhóm kiểm thử viên cùng thực hiện.

* + - 1. **Acceptance testing (Kiểm thử nghiệm thu/ sự chấp nhận sản phẩm)**

Kiểm tra sự chấp nhận có ý nghĩa rất quan trọng. Kiểm tra sự chấp nhận được xây dựng có phù hợp với yêu cầu của khách hàng, mục đích của hệ thống được xây dựng, thân thiện với người dùng. Ở giai đoạn kiểm thử này tìm ra lỗi không phải là mục đích chính.

Mục tiêu của kiểm thử chấp nhận là xác nhận sự chấp nhận, sự tin tưởng, các được tính về chức năng và phi chức năng và đánh giá sự sẵn sàng của hệ thống để triển khai và sử dụng.

Có ba loại kiểm thử chấp nhận:

* Alpha testing: là một dạng cả kiểm thử chấp nhận, đối với loại kiểm thử này nhóm người thực hiện kiểm thử là người của nơi sản xuất phần mềm (nhưng không phải là nhóm người đã thực hiện kiểm thử trước đó). Là một dạng kiểm tra mức độ chấp nhận nội bộ.
* Beta testing: Là hiền thức kiểm thử có tiền đề là Alpha testing. Được thực hiện bởi những thành viên từ phía khách hàng sắp xếp.

### Các loại kiểm thử

Loại kiểm thử là sự định nghĩa rõ ràng về các đối tượng kiểm thử. Tập trung vào kiểm tra những đối tượng cụ thể sau đó chọn một kiểm thử thích hợp giúp việc kiểm thử đối tượng trở nên đơn giãn và dễ dàng hơn. Một loại kiểm thử chỉ tập trung vào một mục đích cụ thể.

Có bốn loại kiểm thử đó là:

* Kiểm thử chức năng
* Kiểm thử đặc tính hệ thống
* Kiểm thử cấu trúc
* Kiểm thử liên quan đến các thay đổi
  + - 1. **Kiểm thử chức năng (function testing)**

Chức năng là một hệ thống hay một thành phần phải thực hiện được mô tả trong yêu cầu đặc tả. Kiểm thử chức năng được thực hiện dựa trên các chức năng, các tài liệu đặc tả hoặc sự hiểu biết của chuyên viên kiểm thử phần mềm. Kiểm thử chức năng được thực hiện bởi tất cả của giai đoạn kiểm thử.

Kiểm thử chức năng thực hiện trên cả hai cách:

* Kiểm thử dựa trên yêu cầu: Sử dụng các đặc tả chức năng của hệ thống như là cơ sở để thiết kế hệ thống,
* Kiểm thử dựa trên quy trình: Sử dụng những kiến thức về quy trình. Quy trình mô tả kịch bản hoạt động hằng ngày của hệ thống.

Các loại kiểm thử chức năng

* Kiểm thử đơn vị (Unit testing)
* Smoke testing
* Sanity testing
* Kiểm thử giao diện
* Kiểm thử tích hợp (Intergration testing)
* Kiểm thử hệ thống (System testing)
* Kiểm thử hồi quy (Regression Testing)
* Kiểm thử chấp nhận (Acceptance testing)
  + - 1. **Kiểm thử đặc tính sản phẩm phần mềm (non - function)**

Khi sử dụng một sản phẩm mềm, chúng ta quan tâm đến việc mọi thứ hoạt động tốt như thế nào, tốc độ xử lý trong bao lâu, thời gian phản hồi của hệ thống với người dùng, … Kiểm thử đặc tính sản phẩm cũng như kiểm thử chức năng, nó được thực hiện ở tất cả các giai đoạn kiểm thử.

Kiểm thử phi chức năng bao gồm:

* Kiểm thử hiệu năng
* Kiểm thử khả năng chịu tải
* Kiểm thử áp lực
* Kiểm thử tính khả dụng
* Kiểm thử bảo trì
* Kiểm thử độ tin cậy
* Kiểm thử tính tương thích

Các đặc điểm và các đặc điểm phụ tương ứng:

* Chức năng (Functionality) gồm 5 đặc điểm phụ: sự phù hợp, chính xác, bảo mật, khả năng tương tác và tuân thủ.
* Độ tin cậy (Reliability) gồm 4 đặc điểm phụ: độ bền, khả năng chịu lỗi, khả năng phục hồi và tuân thủ.
* Khả năng sử dụng (Usability) gồm 5 đặc điểm phụ: dễ hiểu, khả năng học hỏi, khả năng hoạt động, sự thu hút và tính tuân thủ.
* Tính hiệu quả (Efficiency) gồm 3 đặc điểm phụ: thời gia (hiệu suất), sử dụng tài nguyên và tuân thủ.
* Khả năng bảo trì (Maintainability) gồm 5 đặc điểm phụ: khả năng phân tích, khả năng thay đổi, tính ổn định, khả năng kiểm tra và tuân thủ.
* Tính tương thích (Portability) gồm 5 đặc điểm phụ: khả năng thích ứng, khả năng cài đặt, cùng tồn tại, khả thăng thay thế và tuân thủ.

# phân tích và thiết kế ứng dụng

## Mô tả bài toán

Diễn đạt lại nội dung mô tả bài toán ở đây

## Thiết kế các chức năng

Trình bày các chức năng của ứng dụng ở đây

## Thiết kế cơ sở dữ liệu

Trình bày các bảng cơ sở dữ liệu ở đây

# PHÁT TRIỂN ỨNG DỤNG

## Sơ đồ chức năng

Trong này vẽ sơ đồ các form có trong ứng dụng

## Thông tin chi tiết các form làm việc

### Form X

Trong này giới thiệu chức năng, các mục có trong màn hình

### Form Y

Trong này giới thiệu chức năng, các mục có trong màn hình

# KẾT QUẢ

Phần này trình bày kết quả đạt được

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Đề tài đã thực hiện được …

# TÀI LIỆU THAM KHẢO

1. Phan, D.-V., et al., *Liver cancer prediction in a viral hepatitis cohort: A deep learning approach.* International Journal of Cancer, 2020. **147**(10): p. 2871-2878.

2. Phan, D.V., et al., *Deep learning approaches for sleep disorder prediction in an asthma cohort.* J Asthma, 2020: p. 1-9.

Tên tác giả, Tên tác giả… *Tên công trình*. Nơi xuất bản, năm, trang

# PHỤ LỤC (nếu có)