TRƯỜNG ĐẠI HỌC KINH TẾ

**KHOA THỐNG KÊ – TIN HỌC**

–––––––––––––––––––––––––––––––



**BÁO CÁO THỰC TẬP NGHỀ NGHIỆP**

**NGÀNH HỆ THỐNG THÔNG TIN QUẢN LÝ**

**CHUYÊN NGÀNH QUẢN TRỊ HỆ THỐNG THÔNG TIN**

**ĐỀ TÀI**

**NGHIÊN CỨU VÀ THỰC HIỆN KIỂM THỬ THỦ CÔNG CHO WEBSITE**

Sinh viên thực hiện : Nguyễn Thị Nữ

Lớp : 45K21.1

Đơn vị thực tập : Công ty TMA Bình Định

Cán bộ hướng dẫn : Phạm Minh Tuấn

Giảng viên hướng dẫn : Ths. Cao Thị Nhâm

**Đà Nẵng, 8/2022**

**NHẬN XÉT CỦA ĐƠN VỊ THỰC TẬP**

Họ và tên sinh viên:

Lớp: Khoa: Trường:

Thực tập từ ngày: …./……/ 2022 đến ngày: ........./ .……./ 2022

Tại:

Địa chỉ:

Sau quá trình thực tập tại đơn vị của sinh viên, chúng tôi có một số nhận xét, đánh giá như sau:

**1. Về thái độ, ý thức, đạo đức, kỷ luật**

**2. Kiến thức chuyên môn**

**3. Khả năng hòa nhập và thích nghi với công việc**

**4. Trách nhiệm, sáng tạo trong công việc**

**5. Các nhận xét khác**

**Đánh giá chung:**

**Điểm:**

……….., ngày .......tháng ......năm 2022

**Xác nhận của đơn vị thực tập**

# 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, hàng đâu tiên lùi vào 1.27 cm; 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

[LỜI CẢM ƠN i](#_Toc108814411)

[LỜI CAM ĐOAN ii](#_Toc108814412)

[MỤC LỤC iv](#_Toc108814413)

[DANH MỤC HÌNH ẢNH vi](#_Toc108814414)

[DANH MỤC BẢNG BIỂU vii](#_Toc108814415)

[DANH MỤC CÁC TỪ VIẾT TẮT viii](#_Toc108814416)

[LỜI MỞ ĐẦU 1](#_Toc108814417)

[CHƯƠNG 1. CƠ SỞ LÝ THUYẾT 2](#_Toc108814418)

[1.1. Tổng quan về kiểm thử phần mềm 2](#_Toc108814419)

[1.1.1. Kiểm thử phần mềm (Software Testing) là gì? 2](#_Toc108814420)

[1.1.2. Bảy nguyên tắc trong kiểm thử phần mềm: 2](#_Toc108814421)

[1.2. Quy trình kiểm thử phần mềm: 3](#_Toc108814422)

[1.3. Các quy trình phát triển phần mềm: 5](#_Toc108814423)

[1.3.1. Mô hình Waterfall 5](#_Toc108814424)

[1.3.2. Mô hình chữ V 7](#_Toc108814425)

[1.4. Các cấp độ kiểm thử (Test Level) 8](#_Toc108814426)

[1.4.1. Kiểm thử đơn vị ( Unit test) 8](#_Toc108814427)

[1.4.2. Kiểm thử tích hợp ( Integration test) 8](#_Toc108814428)

[1.4.3. Kiểm thử hệ thống (System test) 8](#_Toc108814429)

[1.4.4. Kiểm thử chấp nhận ( Acceptance test) 9](#_Toc108814430)

[1.5. Kỹ thuật kiểm thử tĩnh (Static Techniques) 9](#_Toc108814431)

[1.5.1. Kỹ thuật kiểm thử tĩnh là gì? 9](#_Toc108814432)

[1.5.2. Quy trình thực hiện Static Techniques 10](#_Toc108814433)

[1.6. Test case 10](#_Toc108814434)

[1.6.1. Khái niệm: 10](#_Toc108814435)

[1.6.2. Các thông số của Test case: 10](#_Toc108814436)

[1.7. Các kỹ thuật kiểm thử: 11](#_Toc108814437)

[CHƯƠNG 2. LÝ THUYẾT 13](#_Toc108814438)

[2.1. Mục 2.1 13](#_Toc108814439)

[2.1.1. Mục 2.1.1 13](#_Toc108814440)

[2.1.2. Mục 2.1.2 13](#_Toc108814441)

[2.2. Mục 2.2 13](#_Toc108814442)

[CHƯƠNG 3. TRIỂN KHAI … 14](#_Toc108814443)

[3.1. Mục 3.1 14](#_Toc108814444)

[3.1.1. Mục 3.1.1 14](#_Toc108814445)

[3.1.2. Mục 3.1.2 14](#_Toc108814446)

[3.2. Mục 3.2 14](#_Toc108814447)

[CHƯƠNG 4. KẾT QUẢ 15](#_Toc108814448)

[4.1. Mục 4.1… 15](#_Toc108814449)

[4.2. Mục 4.2… 15](#_Toc108814450)

[KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 16](#_Toc108814451)

[TÀI LIỆU THAM KHẢO 17](#_Toc108814452)

[PHỤ LỤC 18](#_Toc108814453)

*(Mục lục này chỉ là ví dụ)*

# DANH MỤC HÌNH ẢNH

[Hình 1.1.2‑1Hình ảnh quy trình kiểm thử phần mềm 3](#_Toc108814729)

[Hình 1.3.1‑1Hình ảnh Waterfall model 5](#_Toc108814730)

[Hình 1.3.2‑1Hình ảnh V-Model 7](#_Toc108814731)

[Hình 3.1.2‑1 Kiến trúc của mô hình 15](#_Toc108814732)

# 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 nghiên cứu của đề tài**

* Đề tài này nghiên cứu …

1. **Nhiệm vụ của đề tài**

* Nghiên cứu ...
* ...

1. **Phương pháp nghiên cứu**

* ...

1. **Đối tượng và phạm vi nghiên cứu**

* ...

1. **Kết cấu của đề tài**

Đề tài được tổ chức gồm phần mở đầu, x chương nội dung và phần kết luận...

* Mở đầu
* **Chương 1**:
* **Chương 2**:
* **Chương 3**:
* **Chương 4**:
* Kết luận và hướng phát triển

# CƠ SỞ LÝ THUYẾT

## Tổng quan về kiểm thử phần mềm

### Kiểm thử phần mềm (Software Testing) là gì?

Kiểm thử phần mềm là quá trình thực thi 1 chương trình với mục đích tìm ra lỗi. Kiểm thử phần mềm đảm bảo sản phẩm phần mềm đáp ứng chính xác, đầy đủ và đúng theo yêu cầu của khách hàng, yêu cầu của sản phẩm đề đã đặt ra.

Vai trò của kiểm thử phần mềm: Kiểm thử phần mềm đóng vai trò quan trọng trong việc đánh giá và thu được chất lượng cao của sản phẩm phần mềm trong quá trình phát triển. Thông qua chu trình “ kiểm thử – tìm lỗi – sửa lỗi”, để sản phẩm mang lại độ hiệu quả cao nhất mà không còn lỗi.

### Bảy nguyên tắc trong kiểm thử phần mềm:

* **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 kiểm thử phần mềm nghiêm ngặt vẫn còn lỗi. Vì vậy phải tìm được càng nhiều lỗi càng tốt.
* **Kiểm thử cạn kiệt là không thể:** Việc kiểm tra mọi thứ trong phần mềm là không thể. Kiểm thử tất cả các kết hợp đầu vào và đầu ra, với tất cả kịch bản là không thể, trừ khi nó chỉ bao gồm ít trường hợp.
* **Kiểm thử càng sớm càng tốt:** Yêu cầu bắt đầu thử nghiệm phần mềm ngay trong giai đoạn đầu của vòng đời sản phẩm. Hoạt động này sẽ giúp sớm tìm ra bug hơn.
* **Sự tập trung của lỗi:** Thông thường, lỗi tập trung vào các module, thành phần chức năng chính của hệ thống. Nếu xác định được lỗi sẽ tập trung tìm kiếm lỗi quanh khu vực được xác định.
* **Nghịch lý thuốc trừ sâu:** Nếu 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 một lỗi nào mới. Hiệu quả của các lần kiểm thử bắt đầu sẽ giảm xuống sau các lần kiểm thử, vậy nên luôn phải xem xét và sửa đổi các trường hợp kiểm thử một cách thường xuyên.
* **Kiểm thử phụ thuộc vào ngữ cảnh:** Việc kiểm thử phụ thuộc vào ngữ cảnh, và người kiểm thử phải tiếp cận theo nhiều ngữ cảnh khác nhau.
* **Không có lỗi – Sai lầm:** Việc không tìm thấy lỗi trên sản phẩm không đồng nghĩa với việc sản phẩm đã sẵn sàng tung ra thị trường. Việc không tìm thấy lỗi có thể do bộ trường hợp kiểm thử sinh ra chỉ nhằm kiểm tra các chức năng được làm theo đúng yêu cầu thay vì tìm kiếm lỗi.

## Quy trình kiểm thử phần mềm:



Hình ‑Hình ảnh quy trình kiểm thử phần mềm

**Giai đoạn 1:** Lập kế hoạch và kiểm soát kiểm thử

**Lập kế hoạch kiểm thử:**

* Xác định scope, risk và mục đích của hoạt động kiểm thử
* Xác định các tiếp cận kiểm thử
* Xác định quy định kiểm thử hoặc chiến lượng kiểm thử
* Xác định yêu cầu về nguồn lực ( con người, môi trường kiểm thử, thiết bị…)
* Lên lịch trình cho việc phân tích kiểm thử và thiết kế các trường hợp kiểm thử, thực thi kiểm thử và đánh giá kết quả kiểm thử
* Xác định các tiêu chí kết thúc việc kiểm thử

**Kiểm soát kiểm thử:**

* Đo lường và phân tích các kết quả của hoạt động kiểm thử
* Theo dõi và ghi lại tiến độ, độ bao phủ của kiểm thử và các tiêu chí kết thúc kiểm thử
* Cung cấp các thông tin, tài liệu cho việc kiểm thử
* Tiến hành các hành động khắc phục nếu cần thiết
* Đưa ra quyết định

**Giai đoạn 2:** Phân tích và thiết kế kiểm thử.

* Rà soát các yêu cầu cần thiết trước khi tiến hành kiểm thử như: tài liệu đặc tả, tài liệu đặc tả, tài liệu thiết kế.
* Xác định các điều kiện kiểm thử
* Thiết kế test case
* Đánh giá tính khả thi trong việc kiểm thử cũng như yêu cầu của hệ thống.
* Chuẩn bị môi trường test cũng như xác định các yêu cầu về cơ sở hạ tầng cũng như các công cụ kiểm thử tương ứng.

**Giai đoạn 3:** Thực thi và chạy kiểm thử.

* Thực hiện test dựa trên test case đã viết.
* Chạy lại các case bị lỗi trước đó để xác nhận là case đó đã được sửa.
* So sánh kết quả khi thực thi với kết quả mong đợi.
* Đánh giá kết quả kiểm thử cho các trường hợp kiểm thử.
* Biết báo cáo lỗi khi có bug

**Giai đoạn 4:** Đánh giá Exit criteria và báo cáo:

* Đối chiếu các kết quả thực thi các test case so với các tiêu chí kết thúc kiểm thử được định ra trong lúc lập kế hoạch kiểm thử.
* Đánh giá xe liệu có cần phải test thêm hay điều chỉnh các tiêu chí kết thúc kiểm thử trong bản kế hoạch.
* Viết báo cáo tóm tắt hoạt động kiểm thử cũng như kết quả kiểm thử gửi cho các bên liên quan.

**Giai đoạn 5:** Đóng hoạt động kiểm thử

* Kiểm tra khách hàng được nhận sản phẩm theo dự kiến từ đầu và đảm bảo rằng tất cả sự cố đã được giải quyết.
* Hoàn thiện và lưu trữ phần mềm testware như script, môi trường kiểm thử để sau này sử dụng lại.
* Bàn giao lại testware cho đội bảo trì.
* Đánh giá cách kiểm thử và đưa ra bài học cho lần phát hành và các dự án trong tương lai.

## Các quy trình phát triển phần mềm:

### Mô hình Waterfall



Hình ‑Hình ảnh Waterfall model

Mô hình này gồm các giai đoạn xử lý nối tiếp nhau như sau:

* Thu thập yêu cầu (Requirement gathering): Đây là giai đoạn xác định các yêu cầu chức năng và phi chức năng mà hệ thống phần mềm cần có. Kết quả của giai đoạn này là bản tài liệu đặc tả yêu cầu. Tài liệu này sẽ là nền tảng cho những giai đoạn tiếp theo cho đến cuối dự án.
* Phân tích hệ thống ( System Analysis): Là giai đoạn định ra làm thế nào để hệ thống phần mềm đáp ứng đúng yêu cầu của khách hàng. Giai đoạn này thực hiện phân tích, thiết kế hệ thống phần mềm.
* Coding: Là giai đoạn thực hiện sản phẩm dựa trên đặc tả yêu cầu và tài liệu thiết kế module.
* Testing: Tester sẽ nhận sản phẩm từ developer và thực hiện kiểm thử cho nhóm các thành phần và kiểm thử hệ thống. Khâu kiểm thử cuối cùng sẽ là Kiểm thử chấp nhận, giai đoạn này còn có sự tham gia của khách hàng.
* Implementation: Triển khai hệ thống ra môi trường của khách hàng.
* Operations & Maintenance: Đây là giai đoạn cài đặt, cấu hình và đào tạo cho khách hàng. Giai đoạn này sửa chữa những lỗi của sản phẩm (nếu có) và phát triển những thay đổi mới được khách hàng yêu cầu.

**Đặc điểm:**

Thường áp dụng cho các phần mềm có quy mô vừa và nhỏ.

Các dự án có yêu cầu rõ ràng, ít thay đổi.

Nguồn lực được đào tạo và sẵn sàng.

**Ưu điểm**: Vì có yêu cầu rõ ràng nên dễ hiểu, dễ áp dụng. Dễ phân công công việc, bố trí , giám sát

**Nhược điểm**: Thực tế cho thấy rằng đến những giai đoạn cuối cùng của dự án mới có khả năng nhận ra sai sót trong những giai đoạn trước để có thể quay lại sửa chữa.

### Mô hình chữ V



Hình ‑Hình ảnh V-Model

**Đặc điểm:**

Hoạt động tốt với các dự án có quy mô vừa và nhỏ.

Dễ dàng quản lý vì mỗi giai đoạn có các mục tiêu và mục tiêu được xác định rõ ràng.

Toàn bộ quy trình được chia thành 2 nhóm giai đoạn tương ứng nhau là phát triển và kiểm thử. Mỗi giai đoạn phát triển sẽ tiến hành song song với một giai đoạn kiểm thử tương ứng. Do đó, các lỗi được phát hiện sớm ngay từ đầu.

**Ưu điểm**

Ngay từ lúc nhận được tài liệu đặc tả yêu cầu, tester sẽ tham gia vào review tài liệu đặc tả yêu cầu sau đó lên kế hoạch và thực hiện viết test case. Lỗi được phát hiện từ giai đoạn này sẽ ít tốn thời gian và chi phí hơn các giai đoạn sau.

**Nhược điểm**

Các yêu cầu vẫn được đưa vào thực hiện cùng một lúc mà rủi ro về thay đổi yêu cầu từ phía khách hàng là rất lớn. Do đó, mô hình này vẫn có thể gặp rắc rối khi khách hàng thường xuyên thay đổi yêu cầu.

## Các cấp độ kiểm thử (Test Level)

### Kiểm thử đơn vị ( Unit test)

Kiểm thử đơn vị là loại kiểm thử phần mềm trong đó các đơn vị hay thành phần đơn lẻ của phần mềm được kiểm tra như: hàm (function), lớp (class), phương thức (method). Kiểm thử đơn vị được thực hiện ở giai đoạn sớm nhất của quá trình phát triển, thường do lập trình viên thực hiện và sử dụng phương pháp kiểm thử hộp trắng

### Kiểm thử tích hợp ( Integration test)

Kiểm thử tích hợp là loại kiểm thử trong đó các module phần mềm hay từng chức năng riêng lẻ được tích hợp logic và được kiểm tra theo nhóm nhằm mục đích đánh giá xem chúng có hoạt động chính xác với nhau hay không.

Kiểm thử tích hợp được thực hiện sau khi kiểm thử đơn vị và trước khi kiểm thử hệ thống, thường được thực hiện bởi một Tester cụ thể hoặc một nhóm kiểm thử.

Các phương pháp phổ biến của kiểm thử tích hợp được sử dụng là:

* Kiểm thử tích hợp Big Bang,
* Kiểm thử tích hợp Top-down,
* Kiểm thử tích hợp Bottom up
* Kiểm thử tích hợp Sandwich

### Kiểm thử hệ thống (System test)

Kiểm thử hệ thống là thực hiện kiểm thử một hệ thống đã được tích hợp hoàn chỉnh để xác minh rằng nó đúng yêu cầu của phần mềm. Kiểm thử hệ thống thường là thử nghiệm cuối cùng để xác minh rằng hệ thống được phân phối đáp ứng các đặc điểm kỹ thuật và mục đích của nó. Kiểm thử hệ thống sử dụng phương pháp Kiểm thử hộp đen và thường được thực hiện bởi Tester.

Một số loại kiểm thử thường được thực hiện trong System Test:

* Kiểm thử chức năng (Functional Test)
* Kiểm thử hiệu năng (Performance Test)
* Kiểm thử cơ sở dữ liệu (Database Test)
* Kiểm thử khả năng bảo mật (Security Test)
* Kiểm thử tính khả dụng (Usability Test)
* Kiểm tra tính tương thích (Compatibility Test)
* Kiểm tra khả năng phục hồi (Recovery Test)

### Kiểm thử chấp nhận ( Acceptance test)

Mục đích của việc kiểm thử chấp nhận là để đảm bảo phần mềm đáp ứng đúng yêu cầu của khách hàng. Sản phẩm nhận đươc sự chấp nhận của khách hàng, người dùng cuối.

Phạm vi kiểm tra chấp nhận bao gồm từ việc đơn giản là tìm lỗi chính tả và lỗi thẩm mỹ, đến việc phát hiện ra các lỗi có thể gây ra lỗi lớn trong ứng dụng. Bằng cách thực hiện các thử nghiệm chấp nhận, nhóm thử nghiệm có thể tìm hiểu sản phẩm sẽ hoạt động như thế nào khi được cài đặt trên hệ thống của người dùng.

Kiểm thử chấp nhận được chia thành 2 mức khác nhau:

* Kiểm thử alpha
* Kiểm thử beta

## Kỹ thuật kiểm thử tĩnh (Static Techniques)

### Kỹ thuật kiểm thử tĩnh là gì?

Static testing là một kỹ thuật kiểm thử phần mềm để tìm kiếm lỗi trong phần mềm/ ứng dụng mà không thực thi code. Kiểm thử tĩnh được thực hiện để tránh lỗi sớm trong giai đoạn đầu phát triển phần mềm, nó cũng dễ dàng xác định lỗi và giải quyết lỗi.

Có 2 loại chính của kỹ thuật kiểm thử tĩnh là:

* Kiểm thử thủ công hay còn gọi là các hoạt động review
* Kiểm thử tự động là sử dụng các tool để phân tích

### Quy trình thực hiện Static Techniques

Gồm 5 hoạt động chính:

* Planning
* Initiate review
* Individual review
* Issue communication and analysis
* Fixing and report

## Test case

### Khái niệm:

Test case ( Kịch bản kiểm thử ) là một tập hợp các hành động được thực thi để xác minh một function, một hệ thống phần mềm có hoạt động đúng hay không. Test case mô tả dữ liệu đầu vào (input) , hành động (action) hoặc sự kiện (event) và một kết quả mong đợi (expected result).

### Các thông số của Test case:

* Test Case ID (Mã và tên của Test case): Giá trị cần để xác định số lượng trường hợp cần kiểm thử.
* Test Items (Mục đích kiểm thử): Mô tả mục đích sử dụng test case. Giúp tester hiểu và thực hiện đúng khi kiểm thử phần mềm theo test case mô tả.
* Pre-Condition (Điều kiện tiên quyết): Mô tả điều kiện cần có để thực hiện test case này.
* Test Steps (Mô tả các bước): Mô tả các bước cần thực hiện để tái hiện nội dung test case khi tester thực hiện kiểm thử phần mềm.
* Test Data (Dữ liệu đầu vào): Là dữ liệu nhập vào các trường của phần mềm để thực hiện kiểm thử phần mềm.
* Expected results (Kết quả mong đợi): Một test case được viết tốt cần phải đề cập một cách rõ ràng kết quả mong đợi của ứng dụng hoặc hệ thống. Chỉ ra những gì mong đợi như là đầu ra của bước kiểm tra đó.

## Các kỹ thuật kiểm thử:

* **Equivalence Partitioning (Phân vùng tương đương):**

Phân vùng tương đương là phương pháp chia đều điều kiện đầu vào thành những vùng tương đương nhau.Tất cả các giá trị trong một vùng tương đương sẽ cho một kết quả đầu ra giống nhau. Vì vậy chúng ta có thể kiểm tra một giá trị đại diện trong vùng tương đương.

* **Boundary-value Analysis (Phân tích giá trị biên):**

Phân tích giá trị biên là trường hợp đặc biệt của phân vùng tương đương, dựa trên những phân vùng tương đương, tester sẽ xác định giá trị biên giữa những phân vùng này và lựa chọn test case phù hợp.

* **Error Guessing (Đoán lỗi):**

Phương pháp này không có quy trình cụ thể vì có tính trực giác cao và không thể dự đoán trước. Phương pháp chỉ phù hợp với những tester có kinh nghiệm. Họ phỏng lỗi phần mềm dựa vào trực giác, dựa vào kinh nghiệm, dữ liệu lịch sử về các lỗi đã từng xảy ra và sau đó viết các trường hợp kiểm thử để đưa ra các lỗi đó.

# LÝ THUYẾT

## Mục 2.1

Nội dung văn bản….

### Mục 2.1.1

### Mục 2.1.2

## Mục 2.2

# TRIỂN KHAI …

## Mục 3.1

### Mục 3.1.1

### Mục 3.1.2

## Mục 3.2

# KẾT QUẢ

## Mục 4.1…

## Mục 4.2…

Kết quả được xây dựng dựa trên ngôn ngữ lập trình thể hiện như (Bảng 4.1).

Bảng . Kiến trúc

|  |  |  |
| --- | --- | --- |
| Lớp | Ý nghĩa | Tham số |
|  |  |  |
|  |  |  |
|  | | |

Hình ‑ Kiến trúc của mô hình

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Đề tài đã thực hiện được …

# TÀI LIỆU THAM KHẢO

* + - 1. Tên tác giả, Tên tác giả… *Tên công trình*. Nơi xuất bản, năm, trang
      2. Tên tác giả, Tên tác giả… *Tên công trình*. Nơi xuất bản, năm, trang

# PHỤ LỤC