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 ỨNG DỤNG KIỂM THỬ TỰ ĐỘNG CHO WEBSITE**

Sinh viên thực hiện : Nguyễn Thị Tường Vi

Lớp : 45K21.1

Đơn vị thực tập : TMA Solutions Bình Định

Cán bộ hướng dẫn : Huỳnh Thanh Nhã

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: ........./ .……./ 202 đến ngày: ........./ .……./ 202

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

# MỤC LỤC

[LỜI CẢM ƠN 3](#_Toc108709110)

[LỜI CAM ĐOAN 4](#_Toc108709111)

[MỤC LỤC 5](#_Toc108709112)

[DANH MỤC HÌNH ẢNH 8](#_Toc108709113)

[DANH MỤC BẢNG BIỂU 9](#_Toc108709114)

[DANH MỤC CÁC TỪ VIẾT TẮT 10](#_Toc108709115)

[LỜI MỞ ĐẦU 11](#_Toc108709116)

[CHƯƠNG 1. TỔNG QUAN VỀ CƠ SỞ LÝ THUYẾT 12](#_Toc108709117)

[1.1. Tổng quan về kiểm thử phần mềm 12](#_Toc108709118)

[1.1.1. Kiểm thử phần mềm là gì? 12](#_Toc108709119)

[1.1.2. Mục tiêu của kiểm thử phần mềm 12](#_Toc108709120)

[1.1.3. STLC là gì? 12](#_Toc108709121)

[1.1.4. Nguyên tắc kiểm thử 13](#_Toc108709122)

[1.1.5. Error/ Fault/ Failure/ Bug/ Defect 13](#_Toc108709123)

[1.1.6. Verification – Validation 14](#_Toc108709124)

[1.1.7. QA - QC 14](#_Toc108709125)

[1.2. Vòng đời phát triển phần mềm (SDLC) 14](#_Toc108709126)

[1.2.1. SDLC là gì? 14](#_Toc108709127)

[1.2.2. Các mô hình của SDLC 15](#_Toc108709128)

[1.3. Hình thức và phương pháp kiểm thử phần mềm 17](#_Toc108709129)

[1.3.1. Hình thức kiểm thử phần mềm 17](#_Toc108709130)

[1.3.2. Phương pháp kiểm thử phần mềm 17](#_Toc108709131)

[1.4. Cấp độ của kiểm thử phần mềm 18](#_Toc108709132)

[1.4.1. Unit testing 18](#_Toc108709133)

[1.4.2. Integration testing 18](#_Toc108709134)

[1.4.3. System testing 18](#_Toc108709135)

[1.4.4. Acceptance testing 18](#_Toc108709136)

[1.5. Test case 19](#_Toc108709137)

[1.5.1. Khái niệm 19](#_Toc108709138)

[1.5.2. Kỹ thuật thiết kế Test case 19](#_Toc108709139)

[CHƯƠNG 2. LÝ THUYẾT 21](#_Toc108709140)

[2.1. Mục 2.1 21](#_Toc108709141)

[2.1.1. Mục 2.1.1 21](#_Toc108709142)

[2.1.2. Mục 2.1.2 21](#_Toc108709143)

[2.2. Mục 2.2 21](#_Toc108709144)

[CHƯƠNG 3. TRIỂN KHAI … 22](#_Toc108709145)

[3.1. Mục 3.1 22](#_Toc108709146)

[3.1.1. Mục 3.1.1 22](#_Toc108709147)

[3.1.2. Mục 3.1.2 22](#_Toc108709148)

[3.2. Mục 3.2 22](#_Toc108709149)

[CHƯƠNG 4. KẾT QUẢ 23](#_Toc108709150)

[4.1. Mục 4.1… 23](#_Toc108709151)

[4.2. Mục 4.2… 23](#_Toc108709152)

[KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 24](#_Toc108709153)

[TÀI LIỆU THAM KHẢO 25](#_Toc108709154)

[PHỤ LỤC 26](#_Toc108709155)

# DANH MỤC HÌNH ẢNH

[*Hình 1. Mô hình Waterfall* 14](#_Toc108703805)

[*Hình 2. Mô hình chữ V* 14](#_Toc108703806)

[*Hình 3. Mô hình Aglie* 15](#_Toc108703807)

[*Hình 4. Quy trình Scrum* 15](#_Toc108703808)

# DANH MỤC BẢNG BIỂU

[Bảng 4.1 Kiến trúc 5](#_Toc74235471)

# DANH MỤC CÁC TỪ VIẾT TẮT

|  |  |  |
| --- | --- | --- |
| **STT** | **Kí hiệu viết tắt** | **Chữ viết đầy đủ** |
| 1 | STLC | Software testing life cyle |
| 2 | QA | Quality Assurance |
| 3 | QC | Quality Control |
| 4 | SDLC | Software Developer Life Cycle |
| 5 | API | Application Programming Interface Testing |
| 6 | EP | Equivalence partitioning |
| 7 | BVA | Boundary value analysis |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# 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

# TỔNG QUAN VỀ CƠ SỞ LÝ THUYẾT

## Tổng quan về kiểm thử phần mềm

### Kiểm thử phần mềm là gì?

Kiểm thử phần mềm là quá trình thao tác thực hiện một chương trình hoặc một ứng dụng nào đó với mục đích tìm ra các lỗi phần mềm và được sử dụng để đảm bảo phầm mềm đáp ứng chính xác, đầy đủ và đúng theo yêu cầu của khách hàng, yêu cầu của sản phẩm đề ra.

### Mục tiêu của kiểm thử phần mềm

* Đảm bảo chất lượng của sản phẩm.
* Ngăn ngừa và phát hiện các khiếm khuyết.
* Tìm các khuyết điểm có thể được tạo ra bởi các lập trình viên khi phát triển phần mềm.
* Giúp đạt được sự tín nhiệm của khách hàng bằng cách cung cấp cho họ một sản phẩm chất lượng.

### STLC là gì?

* Vòng đời kiểm thử phần mềm (STLC) là quá trình kiểm thử được thực hiện một cách có hệ thống và có kế hoạch. Trong quá trình STLC, các hoạt động khác nhau được thực hiện để nâng cao chất lượng sản phẩm.
* Các giai đoạn của STLC: Có 6 giai đoạn chính
  + - * Phân tích yêu cầu (Requirement Analysis)
      * Lập kế hoạch kiểm thử (Test Planning)
      * Phát triển trường hợp kiểm thử (Test Case Development)
      * Thiết lập môi trường kiểm thử (Enviroment Setup)
      * Thực hiện kiểm thử (Test Execution)
      * Kết thúc chu kì kiểm thử (Test Cycle Closure)

### Nguyên tắc kiểm thử

Có 7 nguyên tắc kiểm thử:

* Kiểm thử cho thấy sự hiện diện của các khiếm khuyết (Testing shows presence of defects)
* Kiểm thử toàn diện là không thể (Exhaustive testing is impossible)
* Kiểm thử càng sớm càng tốt (Early testing)
* Phân cụm khiếm khuyết (Defect clustering)
* Nguyên lý thuốc trừ sâu (Pesticide paradox)
* Kiểm thử phụ thuộc vào ngữ cảnh (Testing is context dependent)
* Quan niệm sai lầm về việc “hết lỗi” (Absence of errors fallacy)

### Error/ Fault/ Failure/ Bug/ Defect

* ***Error:*** Là hành động của con người dẫn đến kết quả sai.
* ***Fault:*** Trạng thái phần mềm được gây ra bởi một Error. Xảy ra khi làm sai các bước, quy trình hoặc chuẩn bị dữ liệu.
* ***Failure:*** Kết quả sai lệch so với yêu cầu đặc tả, là sự khác biệt giữa kết quả thực tế trên màn hình và kết quả mong đợi của một thành phẩm, hệ thống nào đó.
* ***Bug:*** Là một khiếm khuyết trong một thành phần hoặc hệ thống mà nó có thể làm cho thành phần hoặc hệ thống này không thực hiện đúng chức năng yêu cầu của nó.
* ***Defect:*** Lỗi trong quá trình phát triển hoặc lỗi logic làm cho chương trình hoạt động sai yêu cầu đặt ra.

### Verification – Validation

|  |  |
| --- | --- |
| **Verification** | **Validation** |
| * Xác minh là quá trình xác nhận rằng phần mềm đáp ứng được các đặc điểm kỹ thuật của nó. * Trả lời cho câu hỏi: “Sản phẩm có đúng hay không?” | * Xác thực là quá trình xác nhận rằng phần mềm đáp ứng được yêu cầu của người dùng. * Trả lời cho câu hỏi: “Đó có phải là sản phầm phù hợp hay không?” |

### QA - QC

|  |  |
| --- | --- |
| **Quality Assurance** | **Quality Control** |
| Đảm bảo chất lượng là tập hợp các hoạt động được lên kế hoạch và có hệ thống để cung cấp sự đảm bảo rằng phần mềm sẽ phù hợp với các yêu cầu được chỉ định và đáp ứng nhu cầu của người dùng. | Kiểm soát chất lượng là quy trình kiểm tra sự hoàn thành của các yêu cầu về chất lượng phần mềm. |
| Ngăn ngừa khiếm khuyết. | Xác định và cải thiện các khiếm khuyết. |
| Là quy trình để tạo phần mềm. | Là quá trình để xác minh phần mềm. |
| Đảm bảo những gì đang làm là đúng điều phải làm. | Đảm bảo kết quả những gì đang làm là những gì mong đợi. |

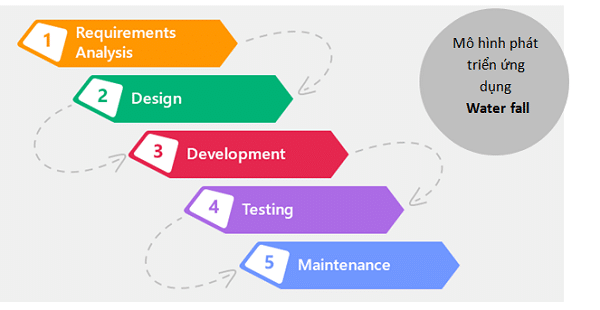
## Vòng đời phát triển phần mềm (SDLC)

### SDLC là gì?

* SDLC là một cách tiếp cận có hệ thống và có trật tự để giải quyết các vấn đề liên quan đến hệ thống phần mềm hay nói cách khác đó là một cấu trúc đối với sự phát triển của mộ sản phẩm phầm mềm.
* Có 6 giai đoạn:
  + - * Thu thập và phân tích yêu cầu (Requirement gathering and analysis)
      * Thiết kế (Design)
      * Thực hiện hoặc mã hóa (Implementation or coding)
      * Kiểm thử (Testing)
      * Triển khai (Deployment)
      * Duy trì (Maintenance)

### Các mô hình của SDLC

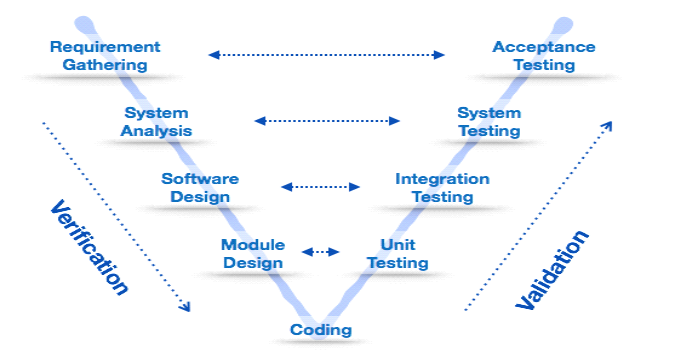
##### Mô hình Waterfall



*Hình 1. Mô hình Waterfall*

* Waterfall là một trong những phương pháp phát triển phần mềm có từ lâu đời. Các giai đoạn sẽ được thực hiện tuần tự nối tiếp nhau.
* Giai đoạn sau chỉ được thực hiện khi giai đoạn trước đã kết thúc và không được quay lại giai đoạn trước để xử lý các yêu cầu khi muốn thay đổi.

##### Mô hình chữ V



*Hình 2. Mô hình chữ V*

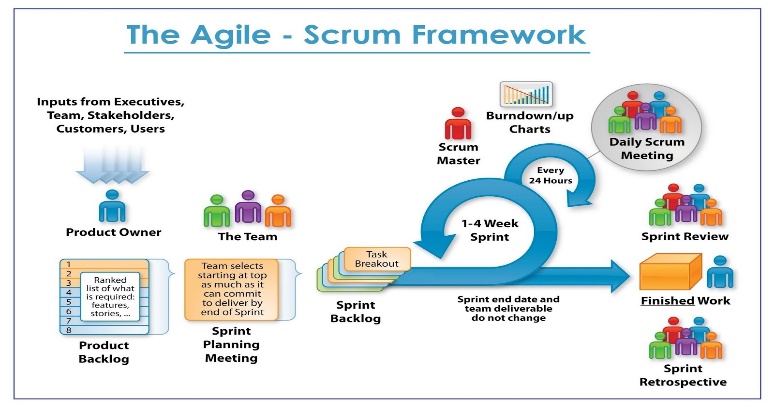
* Là quy trình phát triển phần mềm mở rộng của quy trình phát triển phần mềm Waterfall. Toàn bộ quy trình được chia thành hai nhánh: Phát triển và Kiểm thử.
* Mỗi giai đoạn phát triển sẽ tiến hành song song với một giai đoạn kiểm thử tương ứng, do vậy các lỗi sẽ được phát hiện sớm từ đầu.

##### Mô hình Agile – Quy trình Scrum



*Hình 3. Mô hình Aglie*

* Agile là một phương pháp phát triển phần mềm linh hoạt để làm sao đưa sản phẩm đến tay người dùng càng nhanh càng tốt và được xem như là sự cải tiến so với các mô hình cũ.
* Mô hình này được ứng dụng với bất kỳ loại dự án nào, nhưng cần sự tham gia và tương tác của khách hàng.



*Hình 4. Quy trình Scrum*

* Scrum là một dạng của mô hình Agile và là framework phổ biến nhất khi thực hiện mô hình Agile.
* Scrum là mô hình phát triển lặp đi lặp lại. Những khoảng lặp cố định thường kéo dài 1 đến 2 tuần được gọi là Sprint hay Iteration.

## Hình thức và phương pháp kiểm thử phần mềm

### Hình thức kiểm thử phần mềm

* ***Manual testing:*** Là việc kiểm thử một phần mềm mà không sử dụng bất kỳ công cụ tự động nào hoặc bất kỳ tập lệnh nào. Trong hình thức này, người kiểm thử đảm nhận vai trò là người dùng cuối cùng và kiểm tra phần mềm để xác định bất kỳhành vi hoặc lỗi không mong đợi.
* ***Automation testing:*** Là việc kiểm thử một phần mềm mà người thực hiện kiểm thử viết các kịch bản và sử dụng các công cụ để thực hiện các trường hợp kiểm. Automation testing giúp chạy lại các kịch bản kiểm thử đã thực hiện một cách thủ công, nhanh chóng và lặp đi lặp lại.
* ***Security testing:*** Là hình thức kiểm thử phần mềm nhằm khám phá các lỗ hổng, mối đe dọa và rủi ro trong một ứng dụng phần mềm và ngăn chặn các cuộc tấn công độc hại từ những kẻ xâm nhập. Mục đích là xác định tất cả các lỗ hổng và điểm yếu có thể có của hệ thống phần mềm có thể dẫn đến việc mất thông tin, doanh thu hoặc danh tiếng dưới tay của nhân viên hoặc người ngoài tổ chức.
* ***API testing:*** Là hình thức kiểm thử phần mềm liên quan đến việc kiểm thử các giao diện lập trình ứng dụng một cách trực tiếp và là một phần của kiểm thử tích hợp để xác định xem hệ thống có đáp ứng các yêu về tính năng, độ tin cậy, hiệu suất và bảo mật.

### Phương pháp kiểm thử phần mềm

* ***Black box testing:*** Là một phương pháp kiểm thử dựa trên đầu vào và đầu ra của phần mềm để kiểm tra mà không quan tâm đến code bên trong được viết như thế nào.
* ***White box testing:*** Còn đượcc gọi là Glass testing hay Open box testing và là phương pháp kiểm thử điều tra chi tiết về logic và cấu trúc bên trong của code. Phương pháp này đòi hỏi tester phải có kiến thức về ngôn ngữ lập trình.
* ***Grey box testing:*** Là phương pháp kiểm thử đòi hỏi tester phải có một lượng kiến thức về hoạt động bên trong của một ứng dụng. Đây là phương pháp kết hợp giữa Black box testing và White box testing.

## Cấp độ của kiểm thử phần mềm

### Unit testing

* Là cấp độ kiểm thử cơ bản, thực hiện kiểm thử từng module nhỏ trong hệ thống.
* Mục đích: Xác nhận mỗi thành phần của phần mềm thực hiện đúng với thiết kế.

### Integration testing

* Là kiểm tra các module riêng lẻ với nhau thành một nhóm bởi vì một dự án phần mềm được kết hợp bởi nhiều module riêng lẻ khác nhau và được code bởi nhiều lập trình viên khác nhau.
* Mục đích: Để đảm bảo rằng hệ thống tích hợp đã sẵn sàng để thử nghiệm hệ thống.

### System testing

* Là thực hiện kiểm thử một hệ thống đã được tích hợp hoàn chỉnh để xác minh rằng nó đúng với yêu cầu của phần mềm.
* Mục đích: Đánh giá sự tuân thủ của hệ thống với các yêu cầu được chỉ định.

### Acceptance testing

* Sau khi kiểm tra hệ thống đã sửa tất cả hoặc hầu hết các lỗi, hệ thống sẽ được gửi đến người dùng hoặc khách hàng để kiểm tra chấp nhận.
* Mục đích: Đảm bảo phần mềm đáp ứng đúng yêu cầu của khách hàng. Sản phẩm nhận được sự chấp nhận từ khách hàng – người dùng cuối.
* Được chia thành 2 mức: Alpha testing và Beta testing.

## Test case

### Khái niệm

* Test case là tập hợp các tình huống có thể xảy ra giúp tester xác định được một ứng dụng, hệ thống phần mềm hay một ứng dụng có hoạt động đúng hay không.
* Một bộ test case thường bao gồm: Mã test case, tên test case, mục đích thực hiện test, dữ liệu đầu vào, các bước thực hiện và các kết quả mong đợi.

### Kỹ thuật thiết kế Test case

##### Static testing

* Là việc kiểm tra từng phần của phần mềm, chủ yếu dựa trên các tài liệu của phần mềm hoặc tự phân tích các cú pháp của code để kiểm tra tính logic mà không cần phải chạy phần mềm một cách trực tiếp.
* Các loại Static testing:
  + - * Informal review
      * Walkthroughs
      * Technical review
      * Inspection

##### Dynamic testing

* Thực hiện khi code đang ở chế độ thực thi. Khi code được thực thi, sẽ truyền một giá trị vào đầu vào, sau đó xem kết quả hoặc đầu ra để so sánh với kết quả dự kiến ban đầu.
* Các loại Dynamic testing:
  + - * *Kỹ thuật dựa trên đặc điểm kỹ thuật (Specification – based techniques)*
* Phân vùng tương đương (EP)
* Phân tích giá trị biên (BVA)
* Bảng quyết định (Decision table testing)
* Chuyển đổi trạng thái (State transition testing)
  + - * *Kỹ thuật dựa trên kinh nghiệm (Experience – based techniques)*
* Kiểm thử thăm dò (Exploratory testing)
* Phỏng đoán lỗi (Error guessing)
  + - * *Kỹ thuật dựa trên cấu trúc (Structure – based techniques)*
* Kiểm thử câu lệnh (Statement testing)
* Kiểm thử quyết định (Decision testing)
* Kiểm thử điều kiện (Condition testing)
* Kiểm thử đa điều kiện (Multiple condition testing)

# 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 4.1 Kiến trúc

# 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