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**

**NGHIÊN CỨU KIỂM THỬ TỰ ĐỘNG**

**BẰNG ROBOT FRAMEWORK**

Sinh viên thực hiện : Nguyễn Thị Ái

Lớp : 45K21.1

Đơn vị thực tập : TMA Solutions Bình Định

Cán bộ hướng dẫn : Nguyễn Thành Đời

Giảng viên hướng dẫn : Ths Cao Thị Nhâm

**Đà Nẵng, 8/2022**

**NHẬN XÉT CỦA ĐƠN VỊ THỰC TẬP**

Họ và tên sinh viên: Nguyễn Thị Ái

Lớp: 45K21.1

Khoa: Thống Kê – Tin Học

Trường: Đại học Kinh Tế - Đại Học Đà Nẵng

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

Tại: TMA Solutions Bình Định

Địa chỉ: 12 Đại lộ Khoa học, Thung lũng Sáng tạo, Ghềnh Ráng, Tp. Qui Nhơn, Bình Định

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

[NHẬN XÉT CỦA DOANH NGHIỆP THỰC TẬP ii](#_Toc98339732)

[LỜI CẢM ƠN iii](#_Toc98339733)

[LỜI CAM ĐOAN iv](#_Toc98339734)

[MỤC LỤC vi](#_Toc98339735)

[DANH MỤC HÌNH ẢNH viii](#_Toc98339736)

[DANH MỤC BẢNG BIỂU ix](#_Toc98339737)

[DANH MỤC CÁC TỪ VIẾT TẮT x](#_Toc98339738)

[LỜI MỞ ĐẦU 1](#_Toc98339739)

[CHƯƠNG 1. TỔNG QUAN … 2](#_Toc98339740)

[1.1. Mục 1.1 2](#_Toc98339741)

[1.1.1. Mục 1.1.1 2](#_Toc98339742)

[1.1.2. Mục 1.1.2 2](#_Toc98339743)

[1.2. Mục 1.2 2](#_Toc98339744)

[CHƯƠNG 2. LÝ THUYẾT 3](#_Toc98339745)

[2.1. Mục 2.1 3](#_Toc98339746)

[2.1.1. Mục 2.1.1 3](#_Toc98339747)

[2.1.2. Mục 2.1.2 3](#_Toc98339748)

[2.2. Mục 2.2 3](#_Toc98339749)

[CHƯƠNG 3. TRIỂN KHAI … 4](#_Toc98339750)

[3.1. Mục 3.1 4](#_Toc98339751)

[3.1.1. Mục 3.1.1 4](#_Toc98339752)

[3.1.2. Mục 3.1.2 4](#_Toc98339753)

[3.2. Mục 3.2 4](#_Toc98339754)

[CHƯƠNG 4. KẾT QUẢ 5](#_Toc98339755)

[4.1. Mục 4.1… 5](#_Toc98339756)

[4.2. Mục 4.2… 5](#_Toc98339757)

[KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 6](#_Toc98339758)

[TÀI LIỆU THAM KHẢO 7](#_Toc98339759)

[PHỤ LỤC 8](#_Toc98339760)

*(Mục lục này chỉ là ví dụ)*

# DANH MỤC HÌNH ẢNH

[Hình 1.1 Ngôn ngữ lập trình Python 2](#_Toc98336120)

[Hình 4.1 Kiến trúc của mô hình 5](#_Toc98336121)

# DANH MỤC BẢNG BIỂU

[Bảng 4.1 Kiến trúc 5](#_Toc74235471)

# DANH MỤC CÁC TỪ VIẾT TẮT

|  |  |  |
| --- | --- | --- |
| Từ viết tắt |  |  |
| STLC | Software testing life cycle | Vòng đời kiểm thử phần mềm |
| QA | Quality Assurance | Đảm bảo chất lượng |
| QC | Quality Control | Kiểm soát chất lượng |
| SDLC | Software Development Life Cycle | Chu trình phát triển phần mềm |

# 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 là gì?

Kiểm thử phần mềm là quá trình thực thi 1 chương trình với mục đích tìm ra lỗi nhằm 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.

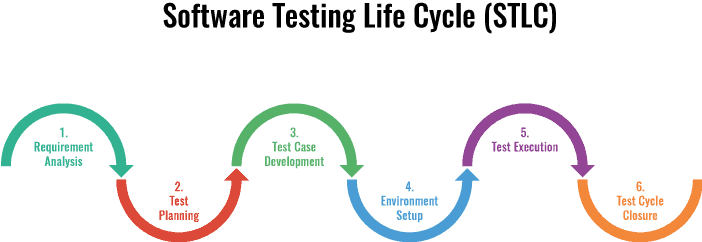
Kiểm thử phần mềm cũng cung cấp mục tiêu, cái nhìn độc lập về phần mềm, điều này cho phép việc đánh giá và hiểu rõ các rủi ro khi thực thi phần mềm.

### Mục tiêu kiểm thử phần mềm là gì?

* Đảm bảo chất lượng của sản phẩm
* Phòng ngừa và phát hiện khiếm khuyết khi khách hàng phát hiện ra
* Sẵn sàng tích hợp và sửa đổi thành phần
* Cung cấp thông tin để đưa ra quyết định cho giai đoạn tiếp theo
* Xác minh và xác thực yêu cầu của người dùng

### Vòng đời kiểm thử phần mềm (STLC) là gì?

Vòng đời kiểm thử phần mềm (Software testing life cycle - STLC) là quy trình kiểm thử được thực hiện theo hệ thống và có kế hoạch rõ ràng. Trong quá trình kiểm thử, rất nhiều giai đoạn khác nhau được thực hiện một cách tuần tự. Mỗi giai đoạn đều có đầu vào và đầu ra khác nhau nhưng đều hướng tới mục tiêu cuối đảm bảo chất lượng sản phẩm.



Hình 01. Mô hình vòng đời kiểm thử phần mềm

### Error, Bug, Fault & Failure

**Error:** Là hành động của con người dẫn đến kết quả sai.

**Bug:** Sự hiện diện của lỗi tại thời điểm thực thi phần mềm.

**Fault:** Lỗi khi làm sai các bước quy trình hoặc chuẩn bị dữ liệu

**Failure:** Lỗi khi có 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 phần, hệ thống hoặc service nào đó.

### Phân biệt QA và QC

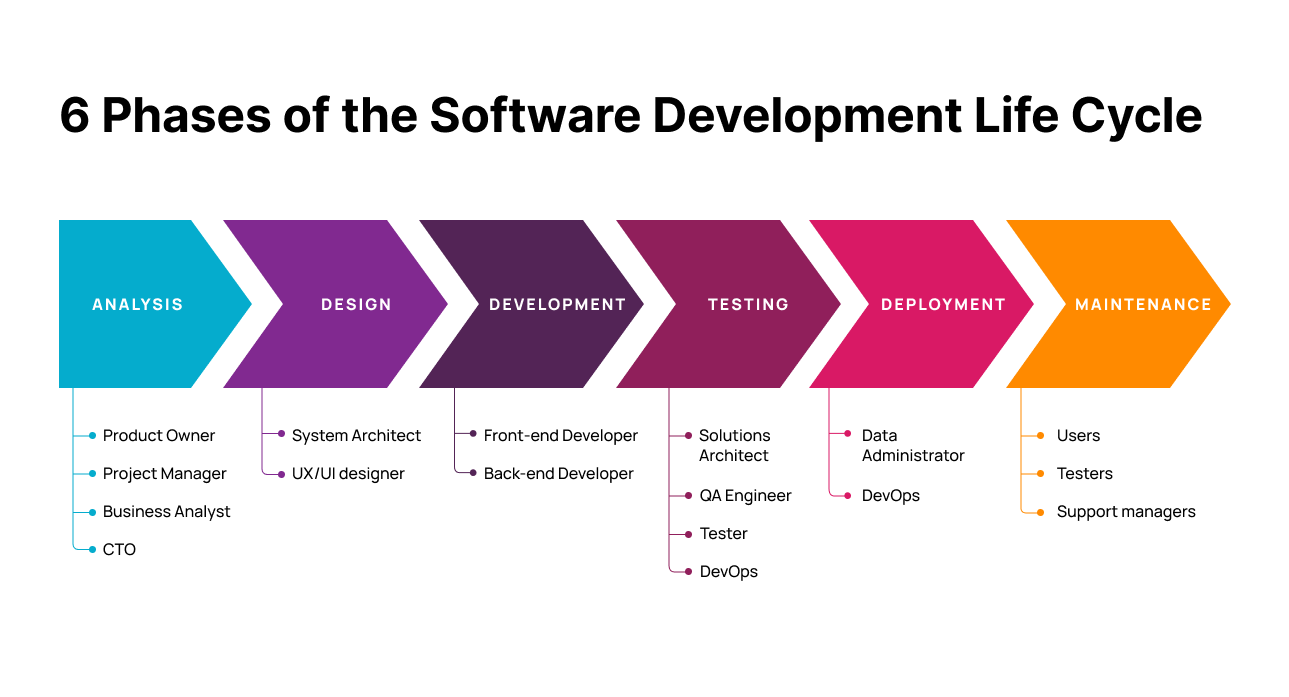
|  |  |
| --- | --- |
| **QA (Đảm bảo chất lượng)** | **QC (Kiểm soát chất lượng)** |
| Là người chịu trách nhiệm đảm bảo chất lượng sản phẩm thông qua việc đưa ra qui trình làm việc cho các bên liên quan | Là những người trực tiếp làm kiểm tra cho các sản phẩm thực tế từng công đoạn của sản xuất |
| Ngăn ngừa khiếm khuyết | Xác định và cải thiện khiếm khuyết |
| Tất cả các thành viên trong nhóm có trách nhiệm đảm bảo chất lượng | Testing team chịu trách nhiệm cho QC |
| Đảm bảo rằng bạn đang làm đúng điều phải làm | Đảm bảo kết quả cua những gì bạn đã làm là những gì bạn mong đợi |
| Verification: xác minh | Validation: xác nhận |
| Toàn bộ vòng đời | Dành cho phần kiểm tra trong SDLC |

Bảng 01. Phân biệt QA và QC

## Vòng đời phát triển phần mềm (SDLC)

### Vòng đời phát triển phần mềm là gì?

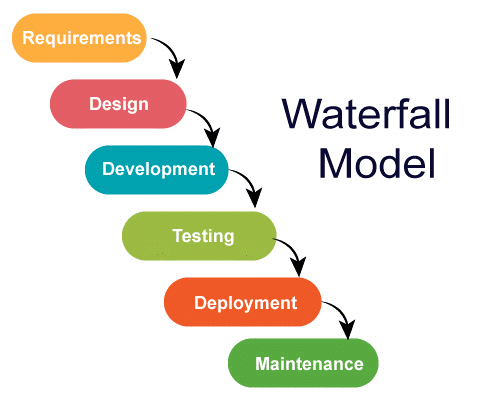
Vòng đời phát triển phần mềm (SDLC – Software Development Life Cycle) là một quá trình theo sau cho một dự án phần mềm, trong một tổ chức phần mềm. Nó bao gồm một kế hoạch chi tiết mô tả làm thế nào để phát triển, duy trì, thay đổi hoặc nâng cấp phần mềm cụ thể.



Hình 02. 6 giai đoạn của chu trình phát triển phần mềm

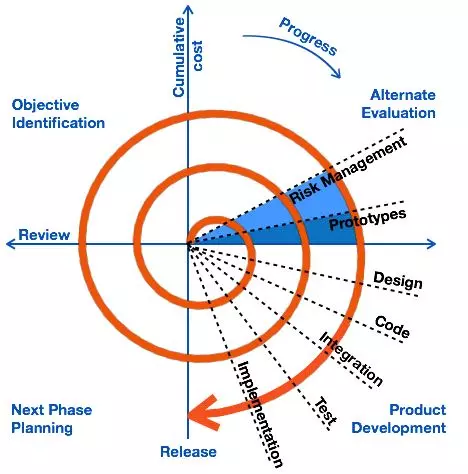
### Các mô hình phát triển phần mềm

* **Mô hình thác nước (Waterfall Model):**
* Là mô hình phát triển phần mềm đầu tiên được sử dụng.
* Mô hình này áp dụng tuần tự các giai đoạn của phát triển phần mềm.
* Đầu ra của giai đoạn trước là đầu vào của giai đoạn sau. Giai đoạn sau chỉ được thực hiện khi giai đoạn trước đã kết thúc. Đặc biệt không được quay lại giai đoạn trước để xử lý các yêu cầu khi muốn thay đổi.



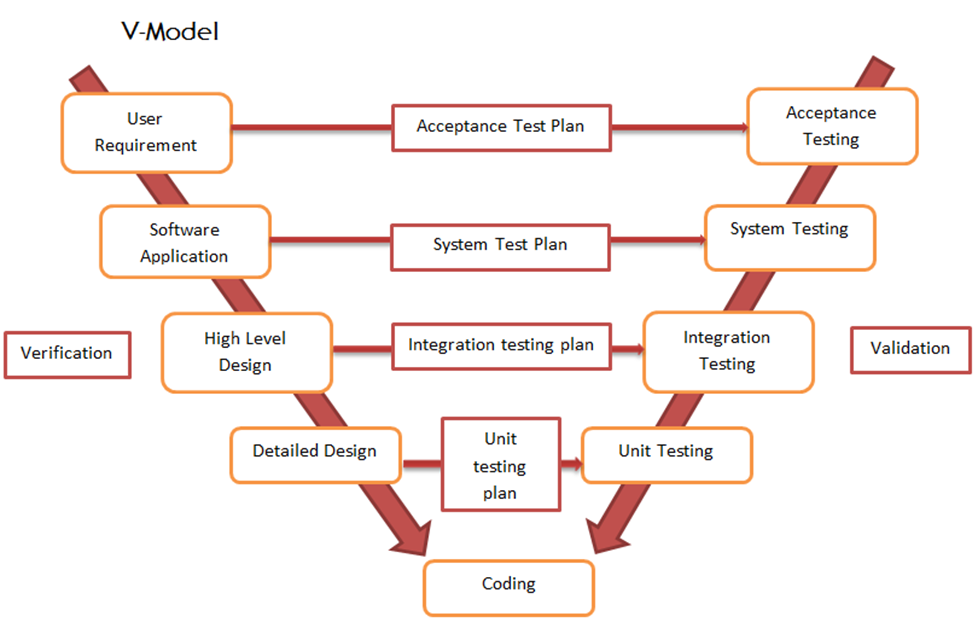
Hình 03. Mô hình thác nước (Waterfall Model)

* **Mô hình xoắn ốc (Spiral model)**
* Quy trình phát triển phần mềm được thực hiện như một vòng xoáy ốc. Mỗi vòng xoắn ốc biểu diễn một hoạt động trong tiến trình phát triển phần mềm
* Mô hình xoắn ốc dựa trên ý tưởng là tối thiểu hóa rủi ro, bằng viêc phân tích yếu tố rủi ro ở mỗi bước lặp và sử dụng phương pháp làm bản mẫu



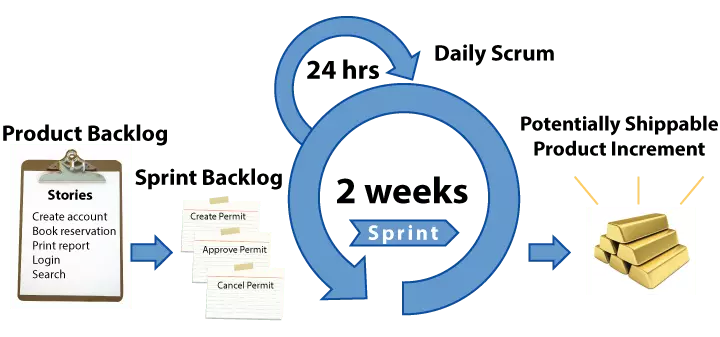
Hình 04. Mô hình xoắn ốc (Spiral model)

* **Mô hình chữ V (V Model)**
* Mô hình chữ V là một phần mở rộng của mô hình thác nước và được dựa trên sự kết hợp của một giai đoạn thử nghiệm cho từng giai đoạn phát triển tương ứng. Đây là một mô hình có tính kỷ luật cao và giai đoạn tiếp theo chỉ bắt đầu sau khi hoàn thành giai đoạn trước.
* Với V model thì công việc test được tham gia ngay từ đầu.



Hình 05. Mô hình chữ V (V Model)

* **Mô hình Agile:** Phương thức phát triển phần mềm Agile là một tập hợp các phương thức phát triển lặp và tăng dần trong đó các yêu cầu và giải pháp được phát triển thông qua sự liên kết cộng tác giữa các nhóm tự quản và liên chức năng.



Hình 06. Mô hình Agile

## Loại và phương pháp kiểm thử phần mềm

### Các loại kiểm thử phần mềm

* **Kiểm thử thủ công (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 kỳ hành vi hoặc lỗi không mong đợi.
* **Kiểm thử tự động (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 thử. Automation testing giúp chạy lại các kịch bản kiểm thử đã thực hiện một cách thủ công, nhanh chóng và lặp đi lặp lại.
* **Kiểm thử bảo mật (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.
* **Kiểm thử API (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

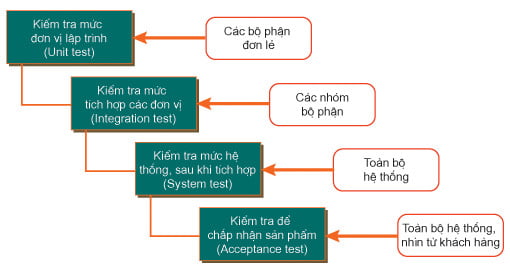
* **Kiểm thử hộp trắng (White box testing):** Trong kiểm thử hộp trắng, cấu trúc mã hoặc thuật toán của chương trình được đưa vào xem xét. Các trường hợp kiểm thử được thiết kế dựa vào cấu trúc mã hoặc cách thức làm việc của chương trình. Người kiểm thử truy cập vào mã nguồn của chương trình và có thể kiểm tra nó, lấy đó làm cơ sở để hỗ trợ việc kiểm thử.
* **Kiểm thử hộp đen (Black box testing):** Không yêu cầu người kiểm thử cần phải có bất kỳ kiến thức về mã hoặc thuật toán của chương trình. Kiểm tra các chức năng của hệ thống tức là những gì hệ thống được cho là cần phải làm dựa trên các " Đặc tả yêu cầu".

Hay nói cách khác, kiểm thử hộp đen là phương pháp test dựa trên đầu vào và đầu ra của chương trình để test mà không quan tâm tới code bên trong được viết ra sao. Tester xem phần mềm như là một hộp đen.

* **Kiểm thử hộp xám (Grey box testing):** Là một phương pháp kiểm thử phần mềm được kết hợp giữa phương pháp kiểm thử hộp đen và phương pháp kiểm thử hộp trắng.

Trong kiểm thử hộp xám, cấu trúc bên trong sản phẩm chỉ được biết một phần, tester có thể truy cập vào cấu trúc dữ liệu bên trong và thuật toán của chương trình với mục đích là để thiết kế test case, nhưng khi test thì test như là người dùng cuối hoặc là ở mức hộp đen.

### Mức độ kiểm thử phần mềm



Hình 07. Mức độ kiểm thử phần mềm

* **Unit testing (Kiểm thử đơn vị):** Kiểm thử đơn vị là cấp độ kiểm thử cơ bản, thực hiện test từng module nhỏ trong hệ thống và thường do lập trình viên thực hiện. Mục đích: để xác nhận mỗi thành phần của phần mềm thực hiện đúng với thiết kế
* **Integration testing (Kiểm thử tích hợp):** Tích hợp kiểm tra việc truyền dữ liệu giữa các module, tích hợp kiểm tra các hàm lại với nhau, các màn hình với nhau theo từng module hoặc theo chức năng. Mục đích: để đảm bảo rằng hệ thống tích hợp đã sẵn sàng để thử nghiệm hệ thống

Một số phương pháp kiểm thử tích hợp: Phương pháp kiểm thử Topdown, Phương pháp kiểm thử Bottom up, Phương pháp kiểm thử Sandwich

* **System Testing (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 nằm trong phạm vi kiểm thử hộp đen và do đó, không yêu cầu kiến thức về thiết kế bên trong của mã hoặc logic.
* **Acceptance Testing (Kiểm thử chấp nhận):** 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.

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

* Kiểm thử alpha: được thực hiện tại nơi phát triển phần mềm bởi những người trong tổ chức nhưng không tham gia phát triển phần mềm.
* Kiểm thử beta: được thực hiện tại bởi khách hàng/ người dùng cuối tại địa điểm của người dùng cuối.

## Testcase

### Khái niệm về testcase

Test case là tập hợp các trường hợp điều kiện mà Tester dựa vào đó để xác định ứng dụng, hệ thống phần mềm hoặc là 1 trong các chức năng của nó có hoạt động như mong muốn hay không.

Quá trình phát triển test case có thể giúp tìm ra lỗi trong các yêu cầu hoặc thiết kế của ứng dụng, vì nó đòi hỏi phải tư duy hoàn toàn thông qua các hoạt động của ứng dụng.

### Cấu trúc của testcase

* Test Case ID: Giá trị cần để xác định số lượng trường hợp cần để kiểm thử.
* Test Items: Dựa theo chức năng của hệ thống có thể chia nhỏ các functions ra để tạo TCs rõ ràng hơn.
* Pre-condition: Điều kiện tiền đề nếu có
* Test Data: Những dữ liệu cần chuẩn bị để test
* Test Steps: Mô tả các bước thực hiện test
* Expected results: Kết quả mong đợi từ các bước thực hiện trên
* A result: Thông thường sẽ là pass, fail
* Comments: Cột này dùng để note lại thông tin liên quan khi thực hiện test case.

## Giới thiệu về Robot Framework

### Robot Framework là gì?

Robot Framework là test framework có thể mở rộng để dùng cho giai đoạn acceptance testing (AT) và giai đoạn acceptance test driven development (ATDD) và có thể dùng để kiểm thử tự động. Robot Framework có thể được sử dụng trong các môi trường phân tán, không đồng nhất, nơi yêu cầu sử dụng các công nghệ và giao diện khác nhau. Robot Framework là phần mềm mã nguồn mở. Robot Framework rất dễ sử dụng cũng như viết test script và có thể chạy được trên mọi nền tảng khác nhau mà không cần chỉnh sửa test script.

### Ưu điểm của robot Framework

* Cú pháp rõ ràng dễ hiểu, gần giống với ngôn ngữ tiếng Anh, các từ khóa và đối số được phân cách bởi 4 kí tự trống, các từ khóa được định nghĩa khá giống với ngôn ngữ lập trình Python khi sử dụng 4 dấu cách để phân chia.
* Có nhiều thư viện hỗ trợ cho nhiều phương thức test khác nhau như Selenium2Library, Android library, OperatingSystem, Swing, Windows GUIs,... (Libraries)
* Linh hoạt, có thể mở rộng các chức năng thông qua các module Python hoặc Java
* Report rõ ràng, chi tiết
* Được hỗ trợ bởi đa số trình editor và có RIDE hỗ trợ (Robot Framework IDE)

# 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

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

Hình 4.1 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