TRƯỜNG ĐẠI HỌC KINH TẾ

**KHOA THỐNG KÊ – TIN HỌC**

–––––––––––––––––––––––––––––––



**BÁO CÁO CHUYÊN ĐỀ TỐT NGHIỆP**

**NGÀNH HỆ THỐNG THÔNG TIN QUẢN LÝ**

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

**Thực hiện kiểm thử Hệ thống quản lý khách sạn**

**SMART HOTEL: Phân hệ PMS\_FO**

Sinh viên thực hiện : Trần Thị Thuỳ Trang

Lớp : <45K21.1>

Đơn vị thực tập : Tập Đoàn APEC

Cán bộ hướng dẫn : Mentor. Trần Quốc Toản

Giảng viên hướng dẫn : <Học vị>. <Cao Thị Nhâm>

**Đà Nẵng, …/2023**

# NHẬN XÉT CỦA ĐƠN VỊ THỰC TẬP

Họ và tên sinh viên: Trần Thị Thuỳ Trang

Lớp: 45K21.1 Khoa: Thống Kê Tin Học

Trường: Đại học Kinh Tế Đà Nẵng

Thực tập từ ngày: 03./03/2023 đến ngày: Hiện tại

Tại: Tập Đoàn APEC

Địa chỉ: 117 Trần Duy Hưng, Cầu Giấy, Hà Nội

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

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **STT** | **Mục đánh giá** | **Rất không tốt** | **Không tốt** | **Bình thường** | **Tốt** | **Rất tốt** |
| 1 | Về thái độ, ý thức, đạo đức, kỷ luật và văn hóa công ty |  |  |  |  |  |
| 2 | Kiến thức chuyên môn |  |  |  |  |  |
| 3 | Khả năng hòa nhập, thích nghi và tác phong nghề nghiệp |  |  |  |  |  |
| 4 | Trách nhiệm |  |  |  |  |  |
| 5 | Sáng tạo trong công việc |  |  |  |  |  |

**Các nhận xét khác**

**Điểm:**

……….., ngày .......tháng ......năm 2023

**Xác nhận của đơn vị thực tập**

# LỜI CẢM ƠN

Lời đầu tiên em xin chân thành cảm ơn các thầy cô khoa Thống kê – Tin học, đặc biệt là cô Cao Thị Nhâm, giáo viên hướng dẫn đã quan tâm và giúp đỡ em, định hướng và cho em những đóng góp quý báu trong quá trình thực hiện đề tài này.

Em cũng xin bày tỏ lòng biết ơn sâu sắc đến người hướng dẫn trực tiếp là anh Trần Quốc Toản. Cảm ơn anh đã hướng dẫn, giúp đỡ để em bổ sung kiến thức và hoàn thành kỳ thực tập này. Cảm ơn những kinh nghiệm mà anh đã truyền đạt, những kiến thức mà anh đã chia sẻ để em hoàn thiện kỹ năng của bản thân hơn.

Trong thời gian thực tập cũng như làm báo cáo khó tránh khỏi những thiếu sót cũng như kinh nghiệm còn hạn chế, em kính mong sự đóng góp ý kiến từ cô và quý công ty để hoàn thiện báo cáo được tốt hơn.

Một lần nữa em xin chân thành cảm ơn!

# LỜI CAM ĐOAN

Em xin cam đoan đây là đề tài nghiên cứu về Thực hiện kiểm thử Hệ thống quản lý khách sạn SMART HOTEL: Phân hệ PMS được tiến hành dựa trên sự cố gắng, nỗ lực của em cũng như sự giúp đỡ tận tình từ phía công ty.

Nội dung báo cáo thực tập tốt nghiệp là sản phẩm mà em đã nỗ lực nghiên cứu trong quá trình học tập tại nhà cũng như học hỏi từ các dự án, quy trình, nghiệp vụ của doanh nghiệp thực tập. Các dữ liệu, kết quả trình bày trong báo cáo là hoàn toàn trung thực, nếu có gì sai sót em xin chịu hoàn toàn trách nhiệm.

# MỤC LỤC

[NHẬN XÉT CỦA ĐƠN VỊ THỰC TẬP i](#_Toc112859205)

[LỜI CẢM ƠN ii](#_Toc112859206)

[LỜI CAM ĐOAN iii](#_Toc112859207)

[MỤC LỤC v](#_Toc112859208)

[DANH MỤC HÌNH ẢNH vii](#_Toc112859209)

[DANH MỤC BẢNG BIỂU viii](#_Toc112859210)

[DANH MỤC CÁC TỪ VIẾT TẮT ix](#_Toc112859211)

[LỜI MỞ ĐẦU 1](#_Toc112859212)

1.Lý do nghiên cứu đề tài

2.Mục tiêu và ý nghĩa của đề tài

3.Đối tượng, phạm vi và phương pháp nghiên cứu

3.1.Đối tượng và phạm vi nghiên cứu

3.2. Phương pháp nghiên cứu

4.Kết cấu của đề tài

[CHƯƠNG 1. TỔNG QUAN VỀ ĐƠN VỊ THỰC TẬP 2](#_Toc112859213)

[1.1. Giới thiệu về VNPT 2](#_Toc112859214)

[1.1.1. Giới thiệu chung 2](#_Toc112859215)

[1.1.2. Quá trình phát triển 2](#_Toc112859216)

1.1.3. Cơ cấu tổ chức

1.1.4 Tầm nhìn, sứ mệnh

[1.2. Mục 1.2 2](#_Toc112859217)

[CHƯƠNG 2. TỔNG QUAN VỀ KIỂM THỬ PHẦN MỀM 3](#_Toc112859218)

[2.1. Kiểm thử thủ công 3](#_Toc112859219)

[2.1.1. Mục 2.1.1 3](#_Toc112859220)

[2.1.2. Mục 2.1.2 3](#_Toc112859221)

[2.2. Kiểm thử tự động 3](#_Toc112859222)

[CHƯƠNG 3. CÔNG CỤ KIỂM THỬ TỰ ĐỘNG KATALON 4](#_Toc112859223)

[3.1. Mục 3.1 4](#_Toc112859224)

[3.1.1. Mục 3.1.1 4](#_Toc112859225)

[3.1.2. Mục 3.1.2 4](#_Toc112859226)

[3.2. Mục 3.2 4](#_Toc112859227)

[CHƯƠNG 4. ỨNG DỤNG KATALON VÀO KIỂM THỬ WEB 5](#_Toc112859228)

[4.1. Mục 4.1… 5](#_Toc112859229)

[4.2. Mục 4.2… 5](#_Toc112859230)

[CHƯƠNG 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 6](#_Toc112859231)

[CHƯƠNG 6. TÀI LIỆU THAM KHẢO 7](#_Toc112859232)

[CHƯƠNG 7. PHỤ LỤC 8](#_Toc112859233)

# DANH MỤC HÌNH ẢNH

[Hình 1.1 Ngôn ngữ lập trình Python 2](#_Toc107673180)

[Hình 2.1 Kiến trúc phần mềm và phần cứng trong deep learning 3](#_Toc107673181)

# DANH MỤC BẢNG BIỂU

[Bảng 4.1 Kiến trúc 5](#_Toc107673187)

# DANH MỤC CÁC TỪ VIẾT TẮT

**AI** : Artificial Intelligence

**…**

# LỜI MỞ ĐẦU

1. **Lý do chọn đề tài**

Hệ thống quản lý khách sạn PMS (Property Management System) là một trong những phần mềm quan trọng nhất trong hoạt động của khách sạn, giúp quản lý các hoạt động vận hành, đặt phòng, thanh toán, quản lý kho, quản lý khách hàng. Tính chính xác và đáng tin cậy của PMS là rất quan trọng để đảm bảo hoạt động của khách sạn diễn ra suôn sẻ và hiệu quả.

Với sự phát triển của công nghệ thông tin, các phần mềm quản lý khách sạn PMS cũng được cải tiến và phát triển liên tục để đáp ứng nhu cầu của khách hàng. Việc nghiên cứu và kiểm thử PMS giúp đảm bảo rằng phần mềm hoạt động đúng cách và tốt nhất để đáp ứng nhu cầu của khách hàng.

Việc phát hiện và khắc phục các lỗi trong quá trình kiểm thử sẽ giúp tiết kiệm chi phí cho khách sạn. Việc sử dụng phần mềm không đáng tin cậy có thể dẫn đến các sai sót trong đặt phòng, thanh toán, quản lý kho, quản lý khách hàng và gây ra chi phí không đáng có cho khách sạn.

PMS là công cụ quản lý và cung cấp thông tin về khách hàng và dịch vụ của khách sạn. Việc đảm bảo chính xác và đáng tin cậy của thông tin này có thể giúp khách sạn nâng cao chất lượng dịch vụ và cải thiện trải nghiệm của khách hàng.

1. **Mục tiêu của đề tài**

* Xác định và phân tích các yêu cầu chức năng và phi chức năng của phần mềm PMS.
* Thiết kế và thực hiện các ca kiểm thử để đảm bảo tính chính xác, đáng tin cậy và hiệu quả của phần mềm PMS.
* Phát hiện và khắc phục các lỗi, sai sót trong phần mềm PMS.
* Đánh giá hiệu suất của phần mềm PMS trong các tình huống khác nhau và đề xuất các cải tiến cho phần mềm PMS.
* Tăng cường độ tin cậy của phần mềm PMS để đảm bảo hoạt động của khách sạn suôn sẻ và hiệu quả.

1. **Phương pháp thực hiện đề tài**

Phân hệ PMS (Property Management System) trong hệ thống quản lý khách sạn SMART HOTEL là một phần mềm quản lý tài nguyên và hoạt động của khách sạn. Phân hệ này sẽ giúp quản lý khách sạn quản lý thông tin về các phòng trống, đặt phòng, thanh toán, quản lý dịch vụ, quản lý đặt phòng trực tuyến và quản lý tài khoản khách hàng.

Để thực hiện kiểm thử phân hệ PMS trong hệ thống quản lý khách sạn SMART HOTEL, áp dụng các phương pháp sau:

* Xác định các chức năng và yêu cầu của phân hệ PMS: Từ tài liệu yêu cầu và thiết kế của phân hệ, xác định các chức năng và yêu cầu của phân hệ PMS.
* Lập kế hoạch kiểm thử: Xác định các phương pháp và kỹ thuật kiểm thử, xây dựng kế hoạch kiểm thử bao gồm các hoạt động kiểm thử, lịch trình kiểm thử và nguồn lực cần thiết cho kiểm thử.
* Tiến hành kiểm thử hộp trắng (white-box testing): Kiểm thử phân hệ PMS từ bên trong, tập trung vào việc kiểm tra mã nguồn, các phương pháp lập trình và các thuật toán để đảm bảo tính chính xác và độ tin cậy của phần mềm.
* Tiến hành kiểm thử hộp đen (black-box testing): Kiểm thử phân hệ PMS từ bên ngoài, tập trung vào việc kiểm tra chức năng và yêu cầu của phần mềm, đảm bảo tính đầy đủ và đúng đắn của chức năng.
* Tiến hành kiểm thử tích hợp (integration testing): Kiểm thử tích hợp giữa phân hệ PMS và các phân hệ khác của hệ thống, đảm bảo tính tương thích và tương tác giữa các phân hệ.
* Tiến hành kiểm thử hệ thống (system testing): Kiểm thử toàn bộ hệ thống để đảm bảo tính hoạt động, độ tin cậy và hiệu suất của hệ thống.
* Tiến hành kiểm thử chấp nhận (acceptance testing): Kiểm thử phân hệ PMS với người sử dụng cuối để đảm bảo tính tương thích với người dùng và đáp ứng các yêu cầu của khách hàng.

1. **Đối tượng và phạm vi của đề tài**

**Phạm vi**

Đối tượng kiểm thử là phần mềm Hệ thống quản lý khách sạn SMART HOTEL: Phân hệ PMS.

Phạm vi kiểm thử bao gồm các chức năng quản lý phòng, quản lý đặt phòng, quản lý khoản thanh toán, quản lý khách hàng và các hoạt động quản lý khác liên quan đến hoạt động kinh doanh của khách sạn.

**Đối tượng**

Nhân viên của khách sạn được đào tạo sử dụng Hệ thống quản lý khách sạn SMART HOTEL: Phân hệ PMS.

Khách hàng của khách sạn được tạo ra và quản lý thông tin của mình trên Hệ thống quản lý khách sạn SMART HOTEL: Phân hệ PMS.

1. **Nội dung của đề tài**

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

* Mở đầu
* **Chương 1**: TỔNG QUAN VỀ ĐƠN VỊ THỰC TẬP
* **Chương 2**: CƠ SỞ LÝ THUYẾT
* **Chương 3**: GIỚI THIỆU DỰ ÁN – THIẾT KẾ TEST CASE VÀ THỰC HIỆN TEST
* **Chương 4**: ỨNG DỤNG KIỂM THỬ TRONG HỆ THỐNG
* Kết luận và hướng phát triển

# TỔNG QUAN VỀ ĐƠN VỊ THỰC TẬP

## GIỚI THIỆU VỀ TẬP ĐOÀN APEC

### Giới thiệu chung

### Tập đoàn APEC (Asia-Pacific Economic Corporation) là một tập đoàn kinh tế đa quốc gia có trụ sở chính tại Hà Nội, Việt Nam. Tập đoàn APEC được thành lập vào năm 2005 và hiện tại đã trở thành một trong những tập đoàn kinh tế lớn nhất tại Việt Nam.Tập đoàn APEC hoạt động trong nhiều lĩnh vực, bao gồm bất động sản, xây dựng, tài chính, bán lẻ và khách sạn. Tập đoàn APEC đã phát triển nhiều dự án lớn tại Việt Nam và các quốc gia khác như Nhật Bản, Hàn Quốc, Singapore, Lào, Campuchia và Myanmar.

### Với phương châm hoạt động "Tạo giá trị bền vững cho cộng đồng và xã hội", Tập đoàn APEC không chỉ tập trung vào mục tiêu tăng trưởng kinh tế mà còn đảm bảo sự phát triển bền vững và đóng góp tích cực cho cộng đồng. Tập đoàn APEC đã thực hiện nhiều hoạt động đóng góp cho cộng đồng như xây dựng trường học, hỗ trợ người nghèo, quỹ học bổng cho sinh viên và các hoạt động xã hội khác.

### Tại địa chỉ 117 Trần Duy Hưng, Tập đoàn APEC cũng đã thực hiện nhiều dự án lớn như tòa nhà APEC Palace, APEC Mandala Wyndham Mũi Né Resort, Trung tâm thương mại Trần Duy Hưng, v.v. Tất cả các dự án của tập đoàn APEC đều được thực hiện với chất lượng cao và tính thẩm mỹ độc đáo, đem lại giá trị cho khách hàng và cộng đồng

### Quá trình phát triển

Tập đoàn APEC 117 Trần Duy Hưng là một thành viên của Tập đoàn APEC - một tập đoàn kinh tế đa quốc gia có trụ sở chính tại Hà Nội, Việt Nam. Tập đoàn APEC được thành lập vào năm 2005, và trong suốt hơn 15 năm hoạt động, tập đoàn đã phát triển mạnh mẽ và trở thành một trong những tập đoàn kinh tế lớn nhất tại Việt Nam.

Quá trình phát triển của Tập đoàn APEC 117 Trần Duy Hưng bắt đầu từ năm 2010 khi tập đoàn quyết định đầu tư vào dự án APEC Mandala Wyndham Mũi Né Resort tại tỉnh Bình Thuận. Đây là dự án đầu tiên của tập đoàn trong lĩnh vực du lịch và khách sạn. Với thiết kế độc đáo, đẳng cấp và đầy đủ tiện nghi, APEC Mandala Wyndham Mũi Né Resort đã trở thành một trong những điểm đến du lịch hàng đầu ở Việt Nam.

Sau đó, tập đoàn tiếp tục đầu tư vào các lĩnh vực bất động sản, bán lẻ và tài chính. Các dự án lớn của tập đoàn bao gồm tòa nhà văn phòng APEC Palace, Trung tâm thương mại Trần Duy Hưng, tòa nhà văn phòng Hanoi Tower và nhiều dự án bất động sản khác ở Hà Nội và các tỉnh thành khác.

Tập đoàn APEC cũng đang phát triển mạnh mẽ trong lĩnh vực bán lẻ với các thương hiệu nổi tiếng như Fivimart, Intimex, và V+ Mart. Tập đoàn cũng đang đầu tư vào lĩnh vực tài chính với việc sở hữu Ngân hàng TMCP Tiên Phong (TPBank), một trong những ngân hàng có tốc độ tăng trưởng nhanh nhất tại Việt Nam.

Tập đoàn APEC luôn đặt lợi ích của khách hàng lên hàng đầu và cam kết cung cấp các sản phẩm và dịch vụ tốt nhất để đáp ứng nhu cầu của khách hàng. Tập đoàn cũng đóng góp tích cực cho cộng đồng và xã hội bằng các hoạt động đóng góp, tài trợ và xây dựng cơ sở hạ tầng cho các trường

### Cơ cấu tổ chức

Cơ cấu tổ chức của Tập đoàn APEC bao gồm các cấp độ sau:

Cấp độ cao nhất: Hội nghị Cấp cao APEC (APEC Summit) là cấp độ cao nhất của tổ chức, được tổ chức hàng năm với sự tham gia của các nhà lãnh đạo các quốc gia thành viên.

Cấp bộ trưởng: Hội nghị Bộ trưởng Kinh tế APEC (APEC Ministerial Meeting) là cấp độ tiếp theo, cũng được tổ chức hàng năm và tập trung vào các vấn đề kinh tế và thương mại.

Các nhóm làm việc (Working Groups): Tập đoàn APEC có nhiều nhóm làm việc khác nhau, chịu trách nhiệm cho các lĩnh vực như thương mại, kinh tế, năng lượng, an ninh thực phẩm, tài chính, khoa học và công nghệ. Những nhóm làm việc này thường họp thường niên hoặc định kỳ để thảo luận và đưa ra các giải pháp cho các vấn đề trong lĩnh vực của mình.

Tổ chức kinh tế APEC (APEC Secretariat): Tổ chức này được thành lập để hỗ trợ các hoạt động của Tập đoàn APEC và giúp đỡ các thành viên trong việc triển khai các cam kết và chính sách của tổ chức. Tổ chức kinh tế APEC có trụ sở tại Singapore.

### Tầm nhìn, sứ mệnh

**Tầm nhìn**

Đứng trong top 100 công ty có môi trường làm việc lý tưởng và đáng mơ ước nhất Việt Nam. Đứng trong top 10 công ty phát triển bất động sản tại Việt Nam trong các phân khúc BĐS dân cư và BĐS du lịch, thông qua chiến lược cung cấp các sản phẩm, dịch vụ khác biệt và chuyên nghiệp. Với tư duy của một công ty công nghệ - nothing is impossible - chúng tôi kỳ vọng trong 10 năm tới trên 51% doanh thu và lợi nhuận của tập đoàn sẽ tới từ các nền tảng ứng dụng công nghệ mới với triết lý đầu tư hạnh phúc và nâng tầm khát vọng thế hệ trẻ Việt Nam thông qua quỹ đầu tư khởi nghiệp APEC cho các doanh nghiệp xã hội, phục vụ cộng đồng, phát triển y tế, giáo dục, sinh học, in 3D, trí tuệ nhân tạo AI,.

**Sứ mệnh**

Chúng tôi lựa chọn con đường “Phụng sự” do đó sứ mệnh của chúng tôi là trao “Hạnh phúc” cho những người đồng hành. Thực hiện sứ mệnh đó, chúng tôi đóng góp những điều tốt đẹp nhất cho cộng đồng, xã hội và đất nước, tối đa hóa lợi ích của nhân viên, cổ đông, khách hàng và đối tác, với tất cả niềm đam mê và niềm tin mãnh liệt rằng tất cả mọi người đều xứng đáng một cuộc sống tốt đẹp.

### Văn hoá doanh nghiệp

Văn hoá doanh nghiệp là những giá trị, tôn chỉ và quy tắc được đặt ra để hướng dẫn và tạo nên một môi trường làm việc chuyên nghiệp, hiệu quả và tích cực trong doanh nghiệp. Tập đoàn APEC có những giá trị và văn hoá doanh nghiệp rất đa dạng và đặc trưng riêng, bao gồm:

Tôn trọng đa dạng văn hóa: Tập đoàn APEC đề cao tính đa dạng văn hóa trong các hoạt động của mình và khuyến khích sự tôn trọng và hiểu biết đối với các giá trị văn hóa khác nhau.

Tập trung vào khách hàng: Tập đoàn APEC luôn đặt khách hàng lên hàng đầu, tập trung vào việc cung cấp sản phẩm và dịch vụ chất lượng cao, đáp ứng nhu cầu của khách hàng.

Đạo đức kinh doanh: Tập đoàn APEC tuân thủ các nguyên tắc đạo đức kinh doanh, đảm bảo hoạt động kinh doanh luôn tuân thủ đúng pháp luật và đạo đức đối với khách hàng, đối tác và cộng đồng.

Tính trách nhiệm xã hội: Tập đoàn APEC luôn hướng tới một môi trường làm việc tích cực và đóng góp cho sự phát triển của cộng đồng, bảo vệ môi trường và hỗ trợ các hoạt động từ thiện.

Tinh thần hợp tác: Tập đoàn APEC tạo ra một môi trường làm việc hợp tác và đồng đội, khuyến khích sự chia sẻ ý tưởng và kiến thức để phát triển cùng nhau

Khả năng thích nghi: Tập đoàn APEC có khả năng thích nghi với môi trường kinh doanh thay đổi, tìm kiếm cơ hội mới và cải tiến liên tục để đáp ứng nhu cầu của khách hàng.

Những giá trị và văn hoá doanh nghiệp này đã giúp Tập đoàn APEC phát triển và trở thành một trong những tập đoàn kinh tế lớn nhất trong khu vực Châu Á - Thái Bình Dương.

### Lĩnh vực hoạt động

Tập đoàn APEC hoạt động trong nhiều lĩnh vực kinh doanh khác nhau, bao gồm:

Bất động sản: Tập đoàn APEC là một trong những nhà phát triển bất động sản hàng đầu tại Việt Nam, với nhiều dự án như khu đô thị Vinhomes, khu đô thị Ecopark, khu nghỉ dưỡng Vinpearl, và nhiều dự án khác.

Thương mại điện tử: Tập đoàn APEC sở hữu trang thương mại điện tử Lazada.vn tại Việt Nam, đây là một trong những trang thương mại điện tử lớn nhất tại Việt Nam.

Xây dựng: Tập đoàn APEC thực hiện các dự án xây dựng lớn, bao gồm các công trình hạ tầng như cầu đường, sân bay và các công trình thương mại.

Du lịch và giải trí: Tập đoàn APEC sở hữu chuỗi khách sạn Vinpearl và thực hiện các dự án giải trí, bao gồm các khu vui chơi, rạp chiếu phim và các trung tâm thể thao.

Năng lượng: Tập đoàn APEC đầu tư vào các dự án năng lượng, bao gồm điện gió, điện mặt trời và năng lượng thủy điện.

Giáo dục: Tập đoàn APEC có các trường đại học, trung học và tiểu học, đào tạo các chương trình giáo dục từ mầm non đến đại học.

Thực phẩm và nước uống: Tập đoàn APEC thực hiện các hoạt động kinh doanh trong lĩnh vực sản xuất, chế biến và phân phối các sản phẩm thực phẩm và nước uống.

Tập đoàn APEC có sự đa dạng trong các lĩnh vực hoạt động kinh doanh, đồng thời tập trung vào những lĩnh vực có tiềm năng phát triển và mang lại giá trị cao cho khách hàng.

## Cơ hội làm việc của ngành kiểm thử

### Mô tả công việc Testing

* Tester là người sẽ thực hiện các công việc như: kiểm tra lỗi, đảm bảo chất lượng phần mềm nhằm đáp ứng yêu cầu khách hàng.
* Thiết kế test case cho các chức năng của phần mềm.
* Tìm lỗi, ngăn ngừa lỗi, đánh giá chất lượng sản phẩm, cải thiện sản phẩm, hiểu và làm rõ yêu cầu của khách hàng.
* Tham khảo với khách hàng để hiểu các yêu cầu sản phẩm.

### Các kĩ năng cần có của một Tester

* Kỹ năng phân tích
* Kỹ năng giao tiếp
* Kỹ năng tổ chức và quản lý thời gian
* Thái độ tốt
* Đam mê

### Mức lương khởi điểm cho Fresher:

Với vị trí của Tester, các ITers có thể nhận được số tiền dao động trong khoảng từ 7.160.000 đến 22.800.000 đồng cho một tháng làm việc. Tuy nhiên, đối với những sinh viên chuyên ngành CNTT mới tốt nghiệp, thực tập sinh hoặc nhân viên thử việc là vị trí đầu tiên mà họ chọn sau khi ra trường. Kinh nghiệm còn thiếu hụt cho phép giới hạn mức lương trong khoảng 4 – 8 triệu/tháng.

# CƠ SỞ LÝ THUYẾT

## Phần mềm

### Khái niệm phần mềm

Phần mềm máy tính (Computer Software) hay gọi tắt là Phần mềm (Software) là một tập hợp những câu lệnh hoặc chỉ thị (Instruction) được viết bằng một hoặc nhiều ngôn ngữ lập trình theo một trật tự xác định, và các dữ liệu hay tài liệu liên quan nhằm tự động thực hiện một số nhiệm vụ hay chức năng hoặc giải quyết một vấn đề cụ thể nào đó

### Phân loại phần mềm

a. Theo phương thức hoạt động

• Phần mềm hệ thống: Dùng để quản lý hành vi phần cứng máy tính, cung cấp các chức năng cơ bản được người dùng yêu cầu hoặc phần mềm khác chạy đúng (nếu có). Được thiết kế để cung cấp một nền tảng để chạy phần mềm ứng dụng như hệ điều hành Windows, iOS, Android, trình điều khiển thiết bị Driver…

• Phần mềm ứng dụng: Sử dụng hệ thống máy tính để thực hiện các chức năng đặc biệt hoặc cung cấp các chức năng giải trí ngoài hoạt động cơ bản của chính máy tính.

Phổ biến trong các phần mềm văn phòng như Microsoft Office, phần mềm game, các công cụ và tiện ích khác…

• Phần mềm dịch mã (trình biên dịch và trình thông dịch): gồm trình biên dịch và trình thông dịch, loại chương trình này sẽ đọc các câu lệnh từ mã nguồn được viết bởi các lập trình viên theo một ngôn ngữ lập trình và dịch nó sang dạng ngôn ngữ máy mà máy tính có thể hiểu được, hay dịch nó sang một dạng khác như là tập tin đối tượng (object file) và các tập tin thư viện (library file) mà các phần mềm khác (như hệ điều hành chẳng hạn) có thể hiểu để vận hành máy tính thực thi các lệnh.

• Nền tảng ứng dụng (ASP.net, PHP…): nền tảng ứng dụng web của Microsoft, cái này hỗ trợ việc tạo ra các ứng dụng web, dịch vụ web (web service).

b. Theo khả năng hay quyền hạn can thiệp vào mã nguồn

• Phần mềm mã nguồn đóng (closed source software): Là phần mềm mà mã nguồn của nó không được công bố. Để sử dụng phần mềm nguồn đóng phải được cấp bản quyền (mua, tặng là tùy).

• Phần mềm mã nguồn mở (open source software): Là phần mềm mà mã nguồn của nó được công bố rộng rãi, công khai và cho phép mọi người tiếp tục phát triển phần mềm đó. Thường thì loại phần mềm này miễn phí.

### Quy trình phát triển phần mềm

#### Định nghĩa

Quy trìnhphát triển phần mềm là một tập hợp các hoạt động tổ chức mà mục đích của chúng là xây dựng và phát triển phần mềm.

* Những câu hỏi được đặt ra là:
* Nhân sự: Ai sẽ làm? Ai làm gì?
* Thời gian: Khi nào làm? Làm mất bao nhiêu thời gian?
* Phương pháp: Làm như thế nào?
* Công cụ: Dùng công cụ gì để làm công việc này?
* Chi phí: Chi phí bỏ ra bao nhiêu? Thu về bao nhiêu? (ước tính)
* Mục tiêu: Mục tiêu hướng đến là gì?
* Mỗi loại hệ thống khác nhau thì cần những quy trình phát triển khác nhau.
* Có 4 thao tác là nền tảng của hầu hết các quy trình phát triển phần mềm:
* Đặc tả yêu cầu (Requirement Specification): Chỉ ra các yêu cầu chức năng, phi chức năng và điều kiện hoạt động của phần mềm.
* Phát triển phần mềm (Development): Là quá trình xây dựng các đặc tả, tạo ra phần mềm thỏa mãn các yêu cầu chỉ ra trong đặc tả.
* Kiểm thử phần mềm (Validation/Testing): Phầm mềm phải được đánh giá để chắc chắn rằng ít nhất có thể thực hiện những gì mà tài liệu đặc tả yêu cầu.
* Thay đổi phần mềm (Evolution): Đáp ứng các yêu cầu thay đổi từ phía khách hàng.

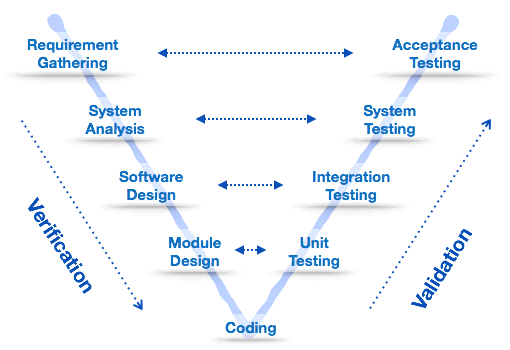
#### Các mô hình phát triển phần mềm

Mô hình phát triển phần mềm hay quy trình phát triển phần mềm xác định các pha/ giai đoạn trong xây dựng phần mềm. Có nhiều loại mô hình phát triển phần mềm khác nhau ví dụ như:

* Mô hình thác nước ( Waterfall model)
* Mô hình xoắn ốc ( Spiral model)
* Mô hình hình chữ V ( V model)
* Mô hình Scrum
* Mô hình tăng trưởng ( Incremental model)
* Mô hình chữ agile
* Mô hình tiếp cận lặp ( Iterative model)
* RAD model ( Rapid Application Development)

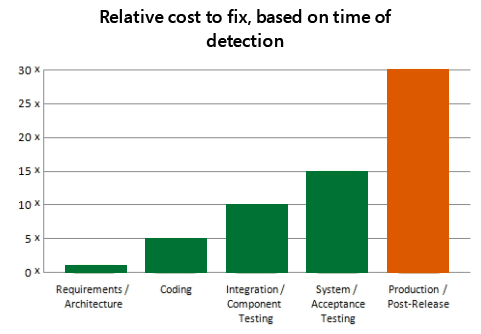
#### Mối quan hệ giữa quy trình phát triển phần mềm và kiểm thử phần mềm

Mô hình chữ V



Hình 2: Mô hình chữ V

Phát triển phần mềm và kiểm thử phần mềm có mối quan hệ khăng khít với nhau. Phát triển phần mềm ngay từ những phau đầu tiên như phân tích yêu cầu, phân tích thiết kế hệ thống,…phải được tiến hành kiểm thử một cách độc lập bởi một đội ngũ có kinh nghiệm để nếu có phát hiện ra sai sót thì phải tiến hành sửa chửa kịp thời, neesus càng để về sau mới phát hiện ra lỗi thì chi phí để sửa chữa là vô cùng lớn. Biểu đồ dưới đây minh họa cho điều này.



Hình 3: Bản đồ chi phí để sửa chữa

Nhìn vào biểu đồ ta cũng có thể dễ dàng nhận thấy là những pha đầu tiên của quy trình phất triển phần mềm thì chi phí là nhỏ, không đáng kể nhưng càng để về sau thì chi phí tăng lên rất nhiều lần. Vì vậy ta không thể chủ quan mà lơ là việc kiểm thử ngay từ giai đoạn đầu của phát triển phần mềm. Điều này là cần thiết nhất đối với các dự án phần ềm lớn của các doanh nghiệp ngày nay.

## Tổng quan về kiểm thử phần mềm

### Khái niệm

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.

- 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.

2.2.2.Mục tiêu của kiểm thử phần mềm

### Mục tiêu kiểm thử phần mềm

* Tìm các bug phát sinh do dev tạo ra khi code.
* Đạt được sự tự tin và cung cấp thông tin về mức độ chất lượng.
* Ngăn ngừa lỗi.
* Đảm bảo rằng kết quả cuối cùng đáp ứng các yêu cầu kinh doanh và người sử dụng.
* Để đạt được sự tín nhiệm của khách hàng bằng cách cung cấp cho họ một sản phẩm chất lượng.

- Kiểm thử phần mềm sẽ giúp hoàn thiện các ứng dụng phần mềm hoặc sản phẩm so với yêu cầu kinh doanh và người sử dụng. Nó rất quan trọng để đảm bảo kiểm thử tốt để kiểm thử các ứng dụng phần mềm hoàn toàn và chắc chắn rằng nó hoạt động tốt và theo các thông số kỹ thuật.

- Kiểm tra phần mềm để chắc chắn kiểm thử đang thực hiện đúng cách và hệ thống đã sẵn sàng để sử dụng. Kiểm thử bao phủ các lĩnh vực khác nhau như: chức năng của các ứng dụng, khả năng tương thích của các ứng dụng với các hệ điều hành, phần cứng và các loại khác nhau của các trình duyệt, thực hiện kiểm thử để kiểm tra hiệu năng của các ứng dụng để đảm bảo rằng hệ thống đáng tin cậy và không có trục trặc hay không nên có bất kỳ vấn đề cản trở.

- Kiểm tra xác nhận rằng hệ thống đáp ứng các yêu cầu khác nhau bao gồm: chức năng, hiệu suất, độ tin cậy, an toàn, khả năng sử dụng.

### Các thuật ngữ chính trong kiểm thử phần mềm

**-Error:** Là một lỗi trong phần mềm do con người gây ra. Một từ đồng nghĩa là mistake, là một sự nhầm lẫn hay một sự hiểu sai trong quá trình phát triển phần mềm.

**- Bug:** Hay còn được gọi là defect, là những lỗi phần mềm trong chương trình tạo ra một kết quả sai hoặc hoạt động không như mong muốn.

**- Fault:** Một bước (step), quy trình (process) hoặc định nghĩa dữ liệu (data definition) không chính xác.

- **Fail:** Thất bại – Một thử nghiệm được coi là thất bại nếu kết quả thực tế không khớp với kết quả mong đợi.

- **Failure**: Sự thất bại – Chính 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.

- **Feature:** Tính năng - Một thuộc tính về một đặc tả thành phần hoặc hệ thống hoặc tài liệu yêu cầu.

- **Pass:** Một ca kiểm thử được coi là vượt qua nếu kết quả thực tế của nó phù hợp với kết quả mong đợi.

- **Precondition:** Các điều kiện môi trường và trạng thái phải được đáp ứng trước khi thành phần hoặc hệ thống có thể được thực thi với một quy trình thử nghiệm hoặc thử nghiệm cụ thể.

- **Testcase(ca kiểm thử):** Một tập hợp các điều kiện mà người tester sẽ xác định xem liệu hệ thống có thỏa mãn yêu cầu kiểm tra và đáp ứng yêu cầu làm việc một cách chính xác.

**- Enviroment:** Môi trường chứa phần cứng, thiết bị, mô phỏng, công cụ phần mềm và các yếu tố hỗ trợ khác cần thiết để tiến hành kiểm tra.

- **Test report**: Tài liệu được tạo ra ở cuối quá trình thử nghiệm tóm tắt tất cả các hoạt động và kết quả thử nghiệm. Nó cũng chứa một đánh giá về quá trình kiểm tra và bài học kinh nghiệm.

- **Test plan:** Tài liệu mô tả phạm vi, cách tiếp cận, tài nguyên và lịch trình của các hoạt động kiểm tra dự định. Nó xác định trong số các mục kiểm tra khác, các tính năng sẽ được kiểm tra, các nhiệm vụ kiểm tra, ai sẽ thực hiện từng nhiệm vụ, mức độ độc lập của người kiểm tra, môi trường kiểm tra, kỹ thuật thiết kế kiểm tra và kỹ thuật đo kiểm được sử dụng và lý do lựa chọn và bất kỳ rủi ro nào cần lập kế hoạch dự phòng. Nó là một bản ghi của quá trình lập kế hoạch kiểm tra.

- **Capability** **Maturity Model Integration (CMMI) – Mô hình khả năng tăng trưởng tích hợp:** Một khung mô tả yếu tố chính của quy trình phát triển và bảo trì sản phẩm hiệu quả. Mô hình khả năng tăng trưởng tích hợp (CMMMI) bao gồm các phương thức tốt nhất cho việc lập kế hoạch, kỹ thuật và quản lý phát triển và bảo trì sản phẩm. CMMI là sự kế thừa từ CMM.

## Các nguyên tắc trong kiểm thử

Phần cần thiết của một ca kiểm thử là định nghĩa về kết quả hay đầu ra mong đợi.

- Lập trình viên nên tránh cố gắng kiểm thử chương trình do chính mình tạo ra.

- Một tổ chức lập trình nên tránh kiểm thử chương trình của chính mình.

- Kiểm tra kỹ lưỡng các kết quả của mỗi ca kiểm thử.

- Các ca kiểm thử phải được viết cho các điều kiện vào không có hiệu lực và không được mong đợi, cũng như điều kiện vào có hiệu lực và được mong đợi.

- Kiểm tra chương trình để xem là chương trình không chỉ không làm những điều gì được đề xuất mà còn có làm những điều gì không được đề xuất.

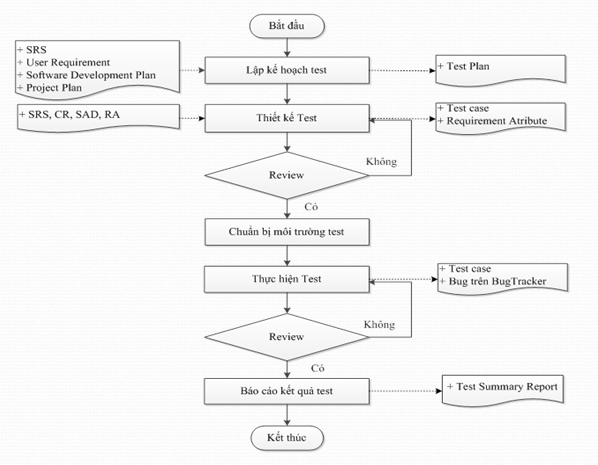
- Tránh vứt bỏ những ca kiểm thử trừ khi chương trình thực sự bị loại bỏ.

- Không lên kế hoạch cho một nỗ lực kiểm thử dưới một giả định ngầm rằng không có lỗi nào sẽ được tìm thấy.

- Xác suất tồn tại thêm lỗi trong một phần của chương trình tỷ lệ thuận với số lỗi đã được tìm thấy trong cùng phần của chương trình đó.

- Kiểm thử là một nhiệm vụ thách thức trí thông minh và khả năng sáng tạo.

## Quy trình kiểm thử phần mềm theo chuẩn CMMI



Hình 4: Quy trình kiểm thử phần mềm CMMI

Phát biểu quy trình:

- Sau khi dự án có kế hoạch tổng thể, Test Leader sẽ lập Test plan.

- Sau khi có Test plan, Tester sẽ tiến hành viết Test Case dựa vào các yếu tố đầu vào(Input) như: UR, SRS, UCD (Usecase Diagram) và GUI (Graphic User Interface).

- Sau khi viết Test Case xong, Tester gửi mail cho Test Leader yêu cầu review

+ Nếu chưa OK -> quay lại cập nhật Test Case

+ Nếu OK -> Chuẩn bị môi trường test

- Sau khi chuẩn bị môi trường test, tiến hnahf thực hiện test

- Trong quá trình thực hiện test cần thực hiện:

+ Thiếu Test Case -> quay lại update Test Case

+ Nếu có bug -> Post bug lên hệ thống

+ Đồng thời cập nhật trạng thái của Test Case thành các trạng thái: Pass/ Fail/ Not run

- Sau khi thực hiện test, Tester gửi mail yêu cầu Test Leader hoặc PM verify các bug trên hệ thống

+ Nếu có bug chưa OK -> Cancel bug

+ Nếu OK -> Tiến hành viết Test Report thông qua file Test Summary Report và kết thúc vòng test.

## Các phương pháp và kĩ thuật kiểm thử phần mềm

### Phương pháp kiểm thử hộp đen

#### Định nghĩa

- Kiểm thử hộp đen (Black Box Testing) là loại kiểm thử hướng dữ liệu hay là kiểm thử hướng in/out.

- Là phương pháp test dựa vào đầu vào và đầu ra của chương trình mà không quan tâm đến code bên trong được viết ra sao. Tester xem phần mềm như là một hộp đen.

- Người kiểm thử nên xây dựng các nhóm giá trị đầu vào mà sẽ thực thi đầy đủ tất cả các yêu cầu chức năng của chương trình.

- Black Box Testing chủ yếu là được thực hiện trong Function test và System test.

- Phương pháp này được đặt tên như vậy bởi vì các chương trình phần mềm, trong con mắt của các tester, giống như một hộp đen; bên trong mà người ta không thể nhìn thấy. Phương pháp này cố gắng tìm ra các lỗi trong các loại sau:

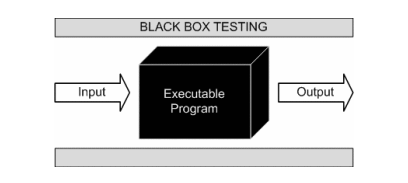
+ Chức năng không chính xác hoặc thiếu.

+ Lỗi giao diện

+ Lỗi trong cấu trúc dữ liệu hoặc truy cập cơ sở dữ liệu bên ngoài.

+ Hành vi hoặc hiệu suất lỗi.

+ Khởi tạo và chấm dứt các lỗi.



Hình 5:Black Box Testing

#### Các kĩ thuật trong kiểm thử hộp đen

Các kĩ thuật thường dùng:

**Phân vùng tương đương**

- Phân vùng tương đương là chia đầu vào thành những nhóm tương đương nhau. Nếu một giá trị trong nhóm hoạt động đúng thì tất cả các giá trị trong nhóm cũng hoạt động đúng và ngược lại.

- Mục đích: Giảm đáng kể số lượng testcase cần phải thiết kế vì vơí mỗi lớp tương đương ta chỉ cần test trên các phần tử đại diện.

- Thiết kế test case bằng phân lớp tương đương tiến hành theo 2 bước:

• Xác định các lớp tương đương và

• Xác định các ca kiểm thử.

**Phân tích giá trị biên**

- Phân tích giá trị biên là test tất cả các giá trị vùng biên của dữ liệu vào và dữ liệu ra. Chúng ta sẽ tập trung vào test các giá trị biên chứ không cần test toàn bộ dữ liệu.

- Kinh nghiệm cho thấy các ca kiểm thử mà khảo sát tỉ mĩ các điều kiện biên có tỉ lệ phàn trăm cao hơn các ca kiểm thử khác.

- Điều kiện biên là những điều kiện mà các tình huống ngay tại, trên và dưới các cạnh của các lớp tương đương đầu vào và các lớp tương đương đầu ra.

**Đồ thị nguyên nhân – kết quả**

- Tạo ra các ca kiểm thử có hệ thống mô tả sự kết hợp của các điều kiện. Sự thay đổi sẽ là một sự lựa chọn kết hpwj không thể dự tính trước, nhưng khi thực hiện như vậy, có vẻ như sẽ bỏ sót nhiều ca kiểm thử “thú vị” được xác định bằng đồ thị nguyên nhân – kết quả.

• Tất cả các nguyên nhân (các đầu vào) và các kết quả (các đầu ra) được liệt kê dựa trên đặc tả và được định danh cho mỗi nhân - quả.

• Các quan hệ giữa nguyên nhân (các đầu vào) và kết quả (các đầu ra) được biểu diễn trong đồ thị làm rõ ràng các mối quan hệ logic.

- Từ đồ thị tạo ra bảng quyết định biểu diễn các quan hệ giữa nguyên nhân và kết quả. Dữ liệu kiểm thử được sinh ra dựa trên các quy tắc trong các bảng này.

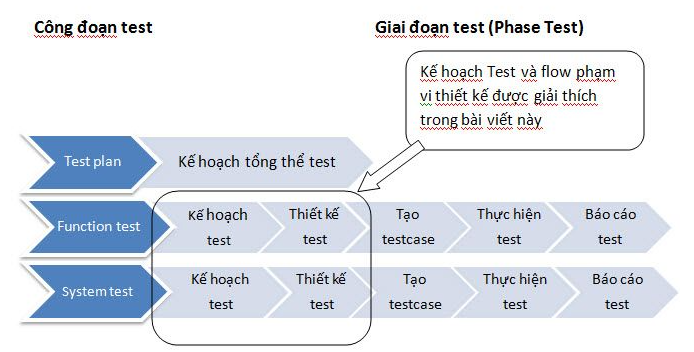
**Đoán lỗi**

- Tester sử dụng trực giác của đoán ra lỗi mà chưa cần đọc yêu cầu của khách hàng.

- Tester được đưa cho 1 chương trình đặc biệt, họ phỏng đoán, cả bằng trực giác lẫn kinh nghiệm, các loại lỗi có thể và sau đó viết các ca kiểm thử để đưa ra các lỗi đó.

- Thật khó để đưa ra một quy trình cho kỹ thuật đoán lỗi vì nó là một quy trình có tính trực giác cao và không thể dự đoán trước. Ý tưởng cơ bản là liệt kê một danh sách các lỗi có thể hay các trường hợp dễ xảy ra lỗi và sau đó viết các cakieemr thử dựa trên danh sách đó. Nói cách khác, bạn liệt kê những trường hợp đăc biệt đó mà có thể đã bị bỏ sót khi chương trình được thiết kế.

### Flow của Black Box Testing



Hình 6: Flow của Black Box Testing

Công đoạn test có lần lượt các phase sau: Kế hoạch test, thiết kế test, tạo testcase, thực hiện test, báo cáo test.

Trong đó: "kế hoạch test" và "thiết kế test" là phase quan trọng để phát hiện lỗi.Nội dung :

- **Test plan:** Chỉ ra rõ ràng mục đích và phạm vi của công đoạn test để kiểm tra xem là test bằng approach như thế nào. Điều chỉnh resource thành viên và quyết định cả schedule.

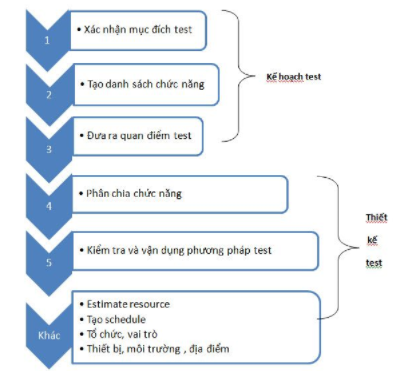
- **Thiết kế test:** Quyết định xem là sẽ sử dụng cái gì cho mục đích và loại test cần được thực hiện trong công đoạn test đó, chức năng đối tượng test, phương pháp test, import và export test. Ngoài ra cũng quyết định cụ thể hơn nguyên liệu cần thiết để thực hiện test hay tiêu chuẩn quyết định thành công/ không thành công.

- **Tạo testcase:** Tạo document ghi trạng thái trước khi bắt đầu test và kết quả mong đợi (kết quả chạy đối tượng test theo điều kiện và trình tự thao tác khi thực hiện test sẽ như thế nào) và cột trạng thái (cột ghi lại kết quả thao tác của đối tượng test).

**- Thực hiện test**: Vừa xem testcase vừa cho chạy phần mềm thực tế để tiến hành test, sau đó đánh dấu kết quả bằng dấu pass hoặc fail vào cột trạng thái testcase. Trường hợp có testcase khác với kết quả mong đợi thì ghi dấu fail vào cột trạng thái, rồi tạo bản báo cáo lỗi. Trong bản báo cáo lỗi: trình bày nội dung mô tả hiện tượng khác với kết quả mong đợi và hiện tượng đó phát sinh trong trường hợp như thế nào (thao tác, giả nhập, điều kiện,...)

- **Báo cáo test:** Tóm tắt kết quả để báo cáo. Căn cứ vào các loại dữ liệu (mục thực hiện, hiệu quả của việc test, công số thực hiện,...) và dữ liệu lỗi (số lỗi được tìm ra, số lỗi theo mức độ quan trọng,...) để đánh giá xem có thỏa mãn tiêu chuẩn pass/ fail của test không? Ngoài ra cũng đề xuất thêm risk có thể sinh ra sau khi release và mục cần bổ sung trong dự án cho giai đoạn tiếp theo.

**\* Flow của test plan và thiết kế test**



Hình 7: Flow của test plan và thiết kế test

- Xác nhận mục đích test

• Xem bản kế hoạch tổng thể test để xác nhận mục đích của test(đặc tính chất lượng, những điểm quan trọng ...).

• Quyết định phạm vi của test, nội dung của test, phương pháp test.

- Tạo danh sách chức năng

• Đưa ra toàn bộ chức năng làm đối tượng test.

• Cần hiểu trước về phần lớn các hoạt động của chức năng.

• Không phán đoán đối tượng hoặc phi đối tượng của test.

- Đưa ra quan điểm test

• Quan điểm test là " cánh cửa của Test ".

• Quan điểm test đã đáp ứng được đúng mục đích test?

- Phân chia chức năng cho từng quan điểm test

• Áp dụng quan điểm test cho từng chức năng để tránh bị quên.

• Có thể hình dung việc test một cách cụ thể.

• Có thể nắm bắt được quy mô test và mức độ quan trọng của các quan điểm test.

- Kiểm tra vận dụng phương pháp và kỹ thuật test

• Tiến hành thiết kế test dối với lần lượt từng kết hợp.

• Lựa chọn và quyết định phương pháp test có thể phát hiện ra lỗi một cách hiệu quả nhất từ một trong số các phương pháp test.

- Các mục kiểm tra khác

• Resource cần thiết

• Schedule

• Cơ cấu & tổ chức, vai trò khi thực hiện test.

• Thiết bị, môi trường, địa điểm làm việc cần để thực hiện test.

### Ưu điểm, nhược điểm của kiểm thử hộp đen

a. Ưu điểm

- Các tester được thực hiện từ quan điểm của người dùng và sẽ giúp đỡ trong việc sáng tỏ sự chênh lệch về thông số kỹ thuật.

- Các tester theo phương pháp black box không có “mối ràng buộc” nào với code, và nhận thức của một tester rất đơn giản: một source code có nhiều lỗi. Sử dụng nguyên tắc, "Hỏi và bạn sẽ nhận" các tester black box tìm được nhiều bug ở nơi mà các DEV không tìm thấy.

b.Nhược điểm

- Dữ liệu đầu vào yêu cầu một khối lượng mẫu (sample) khá lớn

- Nhiều dự án không có thông số rõ ràng thì việc thiết kế test case rất khó và do đó khó viết kịch bản kiểm thử do cần xác định tất cả các yếu tố đầu vào, và thiếu cả thời gian cho việc tập hợp này.

- Khả năng để bản thân kỹ sư lạc lối trong khi kiểm thử là khá cao.

- Chỉ có một số nhỏ các đầu vào có thể được kiểm tra và nhiều đường dẫn chương trình sẽ được để lại chưa được kiểm tra.

- Kiểm thử black box được xem như "là bước đi trong mê cung tối đen mà không mang đèn pin” bởi vì tester không biết phần mềm đang test đã được xây dựng như thế nào. Có nhiều trường hợp khi một tester viết rất nhiều trường hợp test để kiểm tra một số thứ có thể chỉ được test bằng một trường hợp test và/hoặc một vài phần cuối cùng không được test hết.

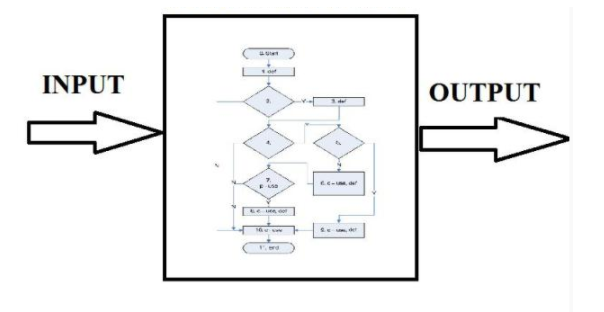
## Phương pháp kiểm thử hộp trắng

### Định nghĩa

- Kiểm thử hộp trắng (White Box Testing) là loại kiểm thử hướng cấu trúc logic, dựa vào cấu trúc code bên trong của phần mềm với mục đích đảm bảo rằng tất cả các câu lệnh và điều kiện sẽ được thực hiện ít nhất 1 lần.

- Người kiểm tra chọn đầu vào để thực hiện các đường dẫn thông qua mã và xác định đầu ra thích hợp. Kiến thức lập trình và kiến thức thực hiện là rất cần thiết trong kiểm thử hộp trắng.

- Người kiểm thử truy nhập vào mã nguồn chương trình và có thể kiểm tra nó, lấy nó làm cơ sở để hỗ trợ việc kiểm thử.



Hình8: White Box Testing

- Việc kiểm tra tập trung chủ yếu vào:

• Cấu trúc chương trình: Những câu lệnh và các nhánh, các loại đường dẫn chương trình

• Logic bên trong chương trình và cấu trúc dự liệu

• Những hành động và trạng thái bên trong chương trình

- Yêu cầu đặt ra cho các ca kiểm thử hộp trắng ít nhất phải đảm bảo:

• Mọi con đường độc lập trong một module cần được thực hiện ít nhất 1 lần

• Mọi ràng buộc logic cần được thực hiện hai phía đúng và sai

• Tất cả các vòng lặp ở biên của nó và biên vận hành phải được thực hiện

• Mọi cấu trúc dữ liệu nội tại được dùng để đảm bảo tính hiệu lực của nó

### Các kĩ thuật trong kiểm thử hộp trắng

- Để thực hiện kiểm thử hộp trắng, sử dụng các kĩ thuật sau:

+ Bao phủ mã lệnh (Code coverage)

+ Gán looux (Fault injection methods)

+ Kiểm thử hoán chuyển (Mutation testing methods)

+ Kiểm thử tĩnh (Fuzz testing)

+ Kiểm thử giao diện lập trình ứng dụng (API testing – Application programming interface)

### Đối tượng, mức độ áp dụng

a. Đối tượng áp dụng

- Đối tượng được kiểm thử là 1 thành phần phần mềm (TPPM).TPPM có thể là 1 hàm chức năng, 1 module chức năng, 1 phân hệ chức năng…

b. Mức độ áp dụng

- Phương pháp Kiểm tra Hộp trắng áp dụng cho các mức độ kiểm tra phần mềm sau đây:

• Unit Testing(Kiểm thử đơn vị): Để kiểm tra đường dẫn trong một đơn vị.

• Integration Testing(Test tích hợp): Để kiểm tra đường dẫn giữa các đơn vị.

• System Testing(Test hệ thống): Để kiểm tra các đường dẫn giữa các hệ thống con.

Tuy nhiên, nó là chủ yếu áp dụng cho các kiểm thử đơn vị.

### Ưu điểm, nhược điểm của kiểm thử hộp trắng

**a. Ưu điểm**

- Test có thể bắt đầu ở giai đoạn sớm hơn, không cần phải chờ đợi cho GUI để có thể test

- Test kỹ càng hơn, có thể bao phủ hầu hết các đường dẫn

- Thích hợp trong việc tìm kiếm lỗi và các vấn đề trong mã lệnh

- Cho phép tìm kiếm các lỗi ẩn bên trong

- Các lập trình viên có thể tự kiểm tra

- Giúp tối ưu việc mã hoá

- Do yêu cầu kiến thức cấu trúc bên trong của phần mềm, nên việc kiểm soát lỗi tối đa nhất.

**b. Nhược điểm**

- Việc kiểm thử bằng kĩ thuật hộp trắng không thể đảm bảo rằng chương trình đã tuân theo đặc tả.

- Một chương trình sai do thiếu đường dẫn. Việc kiểm thử hộp trắng không thể biết được sự thiếu sót này.

- Việc kiểm thử bằng ký thuật hộp trắng không thể phát hiện được lỗi do dữ liệu. Như vậy, việc kiểm thử chỉ dùng kỹ thuật hộp trắng là không đủ để phát hiện

- Đòi hỏi phải có các nguồn lực có tay nghề cao, với kiến thức sâu rộng về lập trình và thực hiện.

- Maintenance test script có thể là một gánh nặng nếu thể hiện thay đổi quá thường xuyên.

- Vì phương pháp thử nghiệm này liên quan chặt chẽ với ứng dụng đang được test, nên các công cụ để phục vụ cho mọi loại triển khai / nền tảng có thể không sẵn có.

## Phương pháp kiểm thử hộp xám

### Định nghĩa

- Là phương pháp đòi hỏi trình độ cao.

- Là kiểu trung gian giữa kiểm thử hộp đen và kiểm thử hộp trắng, trong đó tester phải vận dụng các kiến thức về thuật toán, cấu trúc bên trong chương trình,… như của hộp trắng nhưng để thiết kế testcase theo hướng người sử dụng hoặc có testcase như của hộp đen.

### Các kĩ thuật trong kiểm thử hộp xám

- Kiểm thử ma trận (Matrix Testing)

- Kiểm thử hồi quy (Regression Testing)

- Kiểm thử mảng trực giao (Orthogonal Array Testing or OAT)

- Kiểm thử mẫu (Pattern Testing)

- Ngoài ra, phương pháp kiểm thử hộp xám sử dụng các công cụ kiểm thử phần mềm tự động để tiến hành kiểm tra. Và còn có các stubs và module drivers được tạo giúp giải tỏa 1 phần quá trình kiểm thử bằng cách viết code thủ công.

### Ưu điểm, nhược điểm trong kiểm thử hộp xám

a. Ưu điểm

- Quan điểm kiểm thử của kiểm thử hộp xám là từ quan điểm của người dùng.

- Cung cấp các lợi ích của cả thử nghiệm hộp đen và hộp trắng cùng nhau.

- Sẽ dựa trên các đặc tả chức năng, mô tả của người dùng và sơ đồ kiến trúc hệ thống, từ đó xác nhận các yêu cầu ngay từ ban đầu.

- Việc kiểm tra sẽ tường minh vì sẽ có nhiều sự quan tâm giữa người kiểm thử phần mềm và người thiết kế hoặc kỹ sư.

b. Nhược điểm

- Kiểm tra hộp xám cũng có thể mất nhiều thời gian để kiểm tra từng đường dẫn và đôi khi điều này là không thực tế.

- Rất khó để liên kết lỗi khi thực hiện kiểm tra hộp xám cho một ứng dụng có hệ thống phân tán.

- Thông thường sẽ dẫn đến phạm vi kiểm tra thấp hơn so với thực hiện kiểm tra hộp trắng và đen riêng biệt.

- Có thể không phù hợp để thử nghiệm một số loại chức năng.

## Bug

### Khái niệm

- Bug là lỗi phát sinh trong phần mềm, phát sinh từ sự không khớp giữa chương trình và đặc tả.

- Phần mềm có quá nhiều bug thì sẽ không sử dụng được , gây mất uy tín của công ty trên thị trường và có thể khiến cả công ty phá sản.

- Bug là những lỗi xảy ra khi thực thi test phần mềm.

### Những nguyên nhân gây ra lỗi phần mềm

- Dev hiểu sai SRS

- Dev không chịu unit testing sau khi code

- Dev không cẩn thận dẫn đến code sai

- Môi trường chạy phần mềm không tương thích

- Lỗi trong quá trình build chương trình

### Mức độ nghiêm trọng

- Mức độ nghiêm trọng được định nghĩa là mức độ ảnh hưởng của bug đối với sự phát triển hoặc hoạt động của thành phần của ứng dụng đang được kiểm thử.

- Mức độ ảnh hưởng của bug với chức năng hệ thống càng cao thì mức độ nghiêm trọng của nó càng cao.Tester thường là người xác định mức độ nghiêm trọng của bug.

Trong kiểm thử, mức độ nghiêm trọng của bug được chia làm 4 mức độ:

Mức Critical: Lỗi làm chết hệ thống, không thể thực hiện bất kỳ thao tác gì với hệ thống.

Mức High: Lỗi về chức năng.

Mức Medium: Lỗi validate

Mức Low: Lỗi về giao diện

### Độ ưu tiên của Bug

- Độ ưu tiên quyết định thứ tự các bug được giải quyết. Độ ưu tiên càng cao thì bug cần phải được sửa càng sớm càng tốt.

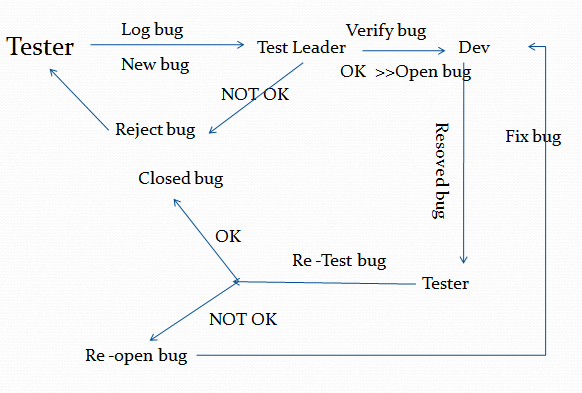
- Lỗi khiến toàn hệ thống không hoạt động thì sẽ có độ ưu tiên cao hơn những bug xảy ra tại các chức năng nhỏ.

- Priority có thể được đánh từ 1 đến 3.

- Khi phát hiện 1 bug, tester sẽ xác định mức độ nghiêm trọng của bug đó (gồm các mức: nghiêm trọng - Critical, cao - high, trung bình - medium, thấp-low), bug nào càng có độ nghiêm trọng cao thì developer nên ưu tiên fix trước. Thông thường thì bug về function - chức năng thì có độ nghiêm trọng cao hơn bug về giao diện và bug phi chức năng (hiệu năng, tính khả dụng, thân thiện,...). Nếu nhiều bug cùng là bug chức năng thì tùy yêu cầu dự án, và thứ tự ưu tiên release (phát hành - giao cho khách hàng) mà gán độ ưu tiên.

- Nói chung: Nếu bug nào xảy ra mà không thể test được các chức năng khác nữa thì thường là ưu tiên cao hơn.

### Vòng đời của Bug



Hình 9:Vòng đời của BUG

Trạng thái lỗi:

a. New (Mới tạo)

- Đây là trạng thái lỗi đầu tiên trong vòng đời của bug.

- Khi tester tìm thấy bug thì nó sẽ có trạng thái là NEW.

b. Assigned (Được chỉ định)

- Khi bug tìm thấy là hợp lệ, duy nhất và thuộc bản release hiện tại, thì team lead sẽ gán bug đó cho developer.

c. Open (Mở)

- Lỗi được log lên bởi tester. Team lead cần xác minh lại bug đó xem có đúng là bug hay không, thì bug có trạng thái OPEN.

d. Rejected (từ chối)

- Một bug được đánh dấu là Rejected khi bug đó không hợp lệ. Nghĩa là thỉnh thoảng tester có thể hiểu sai chức năng và có thể đánh dấu chức năng là bug. Trong trường hợp này, bug sẽ bị reject sau khi team lead kiểm tra lại.

- Khi tester báo cáo một bug nhưng nó lại là chức năng của ứng dụng thì team lead sẽ đánh dấu nó là REJECTED (H1).

e. Fixed/ Resoved (Đã sửa lỗi)

- Khi nhận được bug từ team lead, developer sẽ thực hiện thay đổi để fix bug cho đúng với yêu cầu, và đẩy lại cho tester kiểm tra lại lỗi đó.

f. Re-test (Kiểm tra lại)

- Sau khi fix xong bug, và chức năng/tính năng đã sẵn sàng để kiểm thử, thì tester sẽ thực hiện lại những test case lỗi và xác minh lại xem nó đã chạy đúng hay chưa. Việc này gọi là RE-TESTING.

g. Re-open (Mở lại)

Có 2 tình huống mà chúng ta cần phải re-open lại bug:

• Khi developer fix bug và tester thực hiện test lại nó, nhưng sau khi re-test, bug đó vẫn xảy ra thì tester sẽ RE-OPEN lại bug và assign cho developer.

• Có trường hợp lỗi đã fix và được close xuất hiện lại. Trong trường hợp này, tester cần RE-OPEN lại bug đã close và gán nó cho developer.

h. Closed (Đã đóng)

- Khi bug đã được fix, đã được kiểm thử lại và nó chạy đúng như yêu cầu thì tester sẽ đánh dấu nó là closed.

Phát biểu vòng đời bug:

- Khi tester phát hiện ra bug, tester sẽ log bug lên hệ thống Redmine, trạng thái bug lúc này là New bug và tester gán bug đó cho team leader của Dev hoặc Test lead để verify,Nếu bug đó không phải là bug, team leader sẽ reject bug đó và trả lại cho Tester, nếu Test lead verify bug đó đúng là bug, gán cho dev fix và trạng thái của bug là này là Open bug.

- Sau khi Dev fix xong, Dev sẽ bật trạng thái là resolved bug và gán bug đó cho tester test lại.

- Nếu bug đó được fix ok, Tester sẽ close bug. Nếu bug đó dược fix không ok, Tester sẽ Re-Open bug đó và gán lại cho Dev fix.

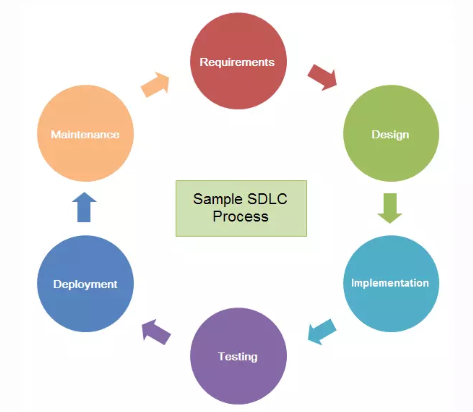
- Vòng đời của bug được lặp lại cho đến khi bug được Close.

## Vòng đời phát triển phần mềm (SDLC)

### Khái niệm

SDLC là một cách tiếp cận có hệ thống và trật tự để giải quyết các vấn đề liên quan đến hệ thống phần mềm hay một cấu trúc đối với sự phát triển của một sản phẩm phần mềm.

Gồm có 6 giai đoạn cơ bản trong vòng đời phát triển phần mềm



Hình. Vòng đời phát triển của sản phẩm

1. Requirment Analysis (Thu thập, phân tích yêu cầu): đây là bước khá quan trọng trong một phần mềm. Thông thường khi bắt đầu làm sản phẩm, khách hàng sẽ chưa xác định rõ sản phẩm mình cần là gì nên yêu cầu đưa ra thường không đầy đủ và rõ ràng. Lúc này, đội phát triển sẽ xác nhận và góp ý để có thể đưa ra một bản yêu cầu ban đầu.

2. Design (Thiết kế): 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, đảm bảo rằng các yêu cầu trong tương lai có thể giải quyết. Ngoài ra, nó cũng liên quan đến việc giao tiếp giữa các hệ thống phần mềm và các phần mềm khác, cũng như các phần cứng cơ bản hoặc các hệ điều hành chủ.

3. Implementation (Thực hiện/xây dựng): Thiết kế trước đó phải được lập trình viên dịch sang ngôn ngữ mà máy tính có thể thiểu. Nếu thiết kế được thực hiện một cách chi tiết và đầy đủ thì ở giai đoạn này việc code sẽ trở nên dễ dàng hơn. Ngược lại, nếu thiết kế hệ thống quá sơ sài thì đến giai đoạn này sẽ gặp nhiều khó khăn và tốn thời gian để giải quyết vấn đề.

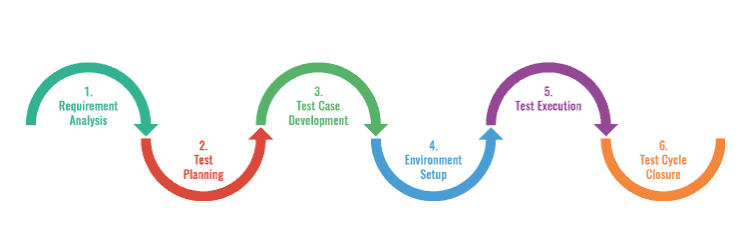
4. Testing (Kiểm thử): Sau khi các lập trình viên hoàn tất việc lập trình thì công việc kiểm thử được bắt đầu. Đội ngũ kiểm thử sẽ thực hiện các phương pháp kiểm thử khác nhau để có thể phát hiện lỗi trong hệ thống, trong giai đoạn này các công cụ kiểm thử sẽ được sử dụng nhằm phát hiện ra lỗi để lập trình viên có thể kịp thời sửa chữa trước khi chuyển đến người dung cuối. Ngoài ra, hiện nay cũng có nhiều công ty tự xây dựng và phát triển các công cụ kiểm thử để phục vụ cho mục đích của họ.

5. Deployment (Triển khai): Sau khi đội kiểm thử kết thúc công việc của mình, sản phẩm đảm bảo có thể đưa vào sử dụng thì nó sẽ được đưa vào sử dụng trong thực tế.

6. Maintenance (Bảo trì): Bảo trì và nâng cấp phần mềm để đối phó với các vấn đề được phát hiện hoặc yêu cầu mới có thể tốn nhiều thời gian hơn so với việc phát triển ban đầu của phần mềm.

### Quy trình kiểm thử phần mềm

Quy trình kiểm thử phần mềm xác định các giai đoạn hay còn gọi là pha trong kiểm thử phần mềm



Hình. Quy trình kiểm thử phần mềm

1. Requirenment analysis - Phân tích yêu cầu

2. Test planning - Lập kế hoạch kiểm thử

3. Test case development - Thiết kế kịch bản cho quy trình kiểm thử

4. Test environment set up - Thiết lập môi trường kiểm thử

5. Test execution - Thực hiện kiểm thử

6. 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.

Mô tả 6 giai đoạn cơ bản trong quy trình kiểm thử phần mềm:

1. Requirement Analysis – Phân tích yêu cầu

Phân tích yêu cầu đề cập đến bước bắt đầu STLC. Tại đây, nhóm tester sẽ đánh giá các yêu cầu của kiểm thử và phác thảo những yêu cầu phức tạp trong số các yêu cầu nhất định mà họ có thể kiểm tra thông qua những tài liệu bao gồm: tài liệu yêu cầu của khách hàng (User stories), tài liệu thiết kế phần mềm, tài liệu đặc tả yêu cầu của phần mềm (SRS)…

2. Test Planning – Lập kế hoạch kiểm thử

Giai đoạn này của vòng đời kiểm thử đưa ra các tài liệu như: Ước tính nỗ lực và kế hoạch kiểm thử. Mục tiêu chính của giai đoạn này là phác thảo nỗ lực và ước tính chi phí cho dự án của bạn.

Người quản lý có thể chuẩn bị kế hoạch kiểm thử cho các loại kiểm thử phần mềm khác nhau, chọn công cụ kiểm thử nào là tối ưu và đánh giá ước tính nỗ lực (chức năng, thời gian, nhân lực...). Đồng thời, họ cần phân công trách nhiệm và vai trò cho đội của mình.

3. Test case development – Thiết kế kịch bản cho quy trình kiểm thử

Giai đoạn Phát triển trường hợp kiểm thử để đáp ứng yêu cầu của sản phẩm. Nhóm tester cần tạo, xác minh và làm lại các trường hợp kiểm thử cụ thể, dựa trên các chức năng và yêu cầu khách hàng. Ngoài ra, nhóm tester cũng cần đưa ra dữ liệu thử nghiệm (test data) để có thể sử dụng cho các trường hợp kiểm thử.

4. Environment Setup – Thiết lập môi trường kiểm thử

Môi trường kiểm thử bao gồm các điều kiện kiểm thử như: thông số kỹ thuật phần cứng và phần mềm được sử dụng trong quy trình kiểm thử. Nhóm kiểm thử được yêu cầu thiết lập hoàn toàn môi trường kiểm thử và kiểm tra mức độ sẵn sàng của môi trường kiểm thử (kiểm thử khói - Smoke testing).

5. Test Execution – Thực thi kiểm thử

Trong giai đoạn Thực thi Kiểm thử, người kiểm thử thực hiện kiểm thử theo các kế hoạch kiểm thử và các trường hợp kiểm thử do nhóm tạo ra. Nếu phát hiện lỗi, tester sẽ báo cáo các lỗi thử nghiệm cho các developers đang làm việc trong dự án.

Nhóm tester cần ghi lại tất cả các kết quả kiểm tra và ghi lại bất kỳ trường hợp nào không thành công. Sau đó, các chuyên viên kiểm thử cần ghi lại test results (kết quả kiểm thử) và bug report (báo cáo lỗi).

6. Test cycle closure – Đóng chu trình kiểm thử

Để đóng chu trình kiểm thử, nhóm kiểm thử cần phân tích lại những tài liệu đã được tổng hợp và hoàn thiện từ những giai đoạn trước: tài liệu phân tích đặc tả yêu cầu, test plan, bug reports… Từ đó, nhóm kiểm thử sẽ tổng kết, báo cáo kết quả về quá trình kiểm thử, các bug đã được fix...

# GIỚI THIỆU DỰ ÁN

## Tổng quan về phần mềm

### Giới thiệu về phần mềm

#### Mục tiêu

Tài liệu nhằm mô tả các chức năng, nghiệp vụ để xây dựng phân hệ PMS của hệ thống quản lý chuỗi khách sạn Mandala (SmartHotel) của Apec Group

## Tổng quan về hệ thống

### Chức năng của hệ thống

- PMS gồm các chức năng:



### Chức năng người dùng

Hệ thống có những người dùng chính sau:

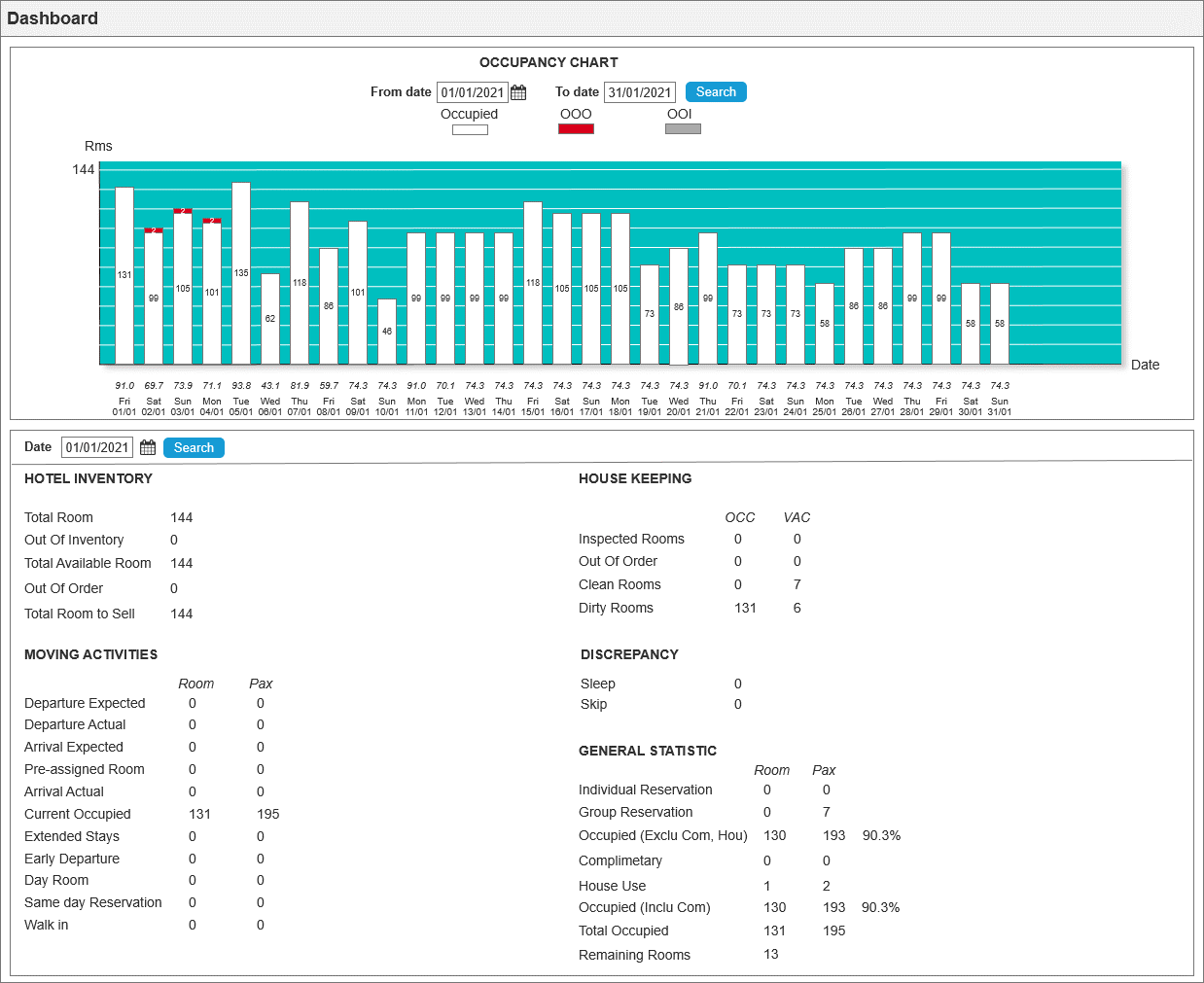
1.FO :Nhân viên lễ tân

2.FOM : Quản lý lễ tân

3.HK: Nhân viên buồng phòng

### Mô tả chi tiết hệ thống

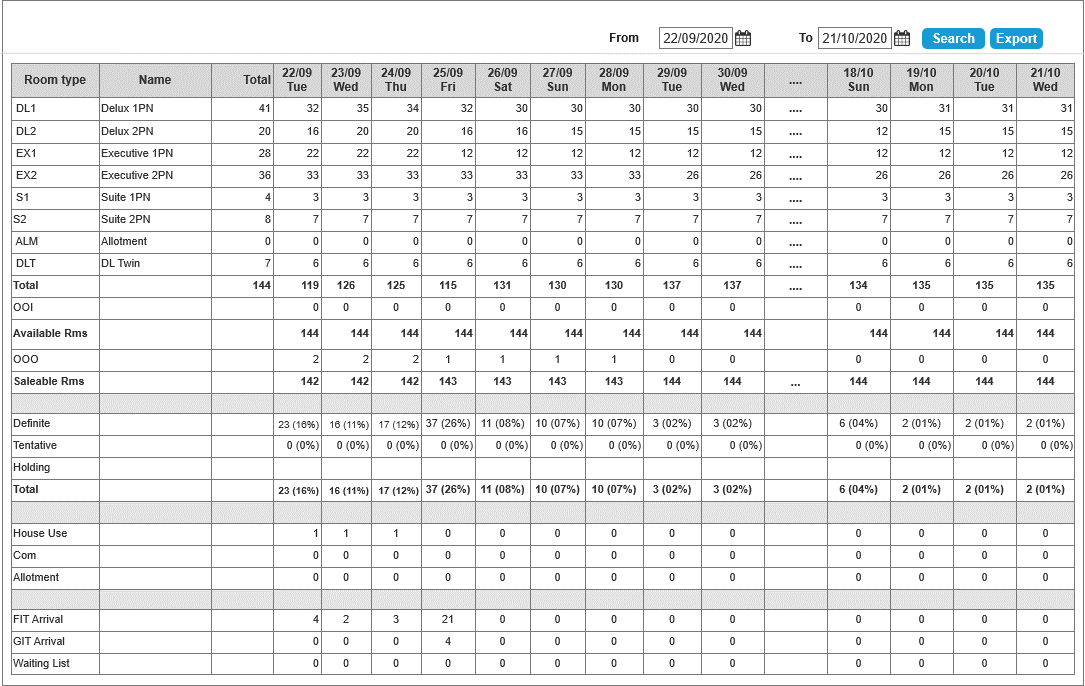
#### Màn hình Dashboard



Hình. Màn hình Dashboard

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field Name | Format/size | M/C/O | Default value | Rules Description |
| **OCCUPANCY CHART** | | | | |
| From date | Date | M | Ngày hiện tại trên hệ thống | Ngày xem dữ liệu từ |
| To date | Date | M | Ngày hiện tại trên hệ thống + 30 ngày | Ngày xem dữ liệu đến  To Date >= from |
| Search | Button |  |  | Nhấn để hiển thị biểu đồ theo khoảng thời gian tương ứng |
| Chỉ dẫn màu |  |  |  | - Màu đỏ: Phòng OOO  - Màu ghi xám: Phòng OOI  - Màu trắng: Phòng có khách lưu trú |
| Trục tung (Rms) |  |  |  | Biểu thị tổng số phòng active tại khách sạn (không tính phòng ảo Dummy) |
| Trục hoành (Date) |  |  |  | - Hiển thị lần lượt các ngày và thứ tương ứng trong khoảng thời gian từ From date đến To date  - Hiển thị % Occupancy theo công thức = (số lượng phòng (không tính phòng ảo) đã check in mà chưa check out tại ngày tương ứng)\*100/(Tổng số phòng active, không tính phòng ảo – Số lượng phòng OOO và OOI) |
| Thanh biểu đồ hình cột |  |  |  | - Biểu thị theo thứ tự từ trên xuống dưới của thanh biểu đồ như sau:  + Hiển thị lượng phòng đang khoá sửa với trạng thái OOI  + Hiển thị lượng phòng đang khoá sửa với trạng thái OOO  + Hiển thị lượng phòng (không tính phòng ảo) đã check in mà chưa check out tại ngày tương ứng |
| **PHẦN THÔNG TIN THỐNG KÊ** | | | | |
| Date | Date | M | Ngày hiện tại trên hệ thống | Ngày muốn xem dữ liệu thống kê |
| Search | Button |  |  | Nhấn để hiển thị thông tin theo Date đã chọn |
| **Hotel inventory** | | | | |
| Total Room | Numberic | C |  | Hiển thị Tổng số phòng có trạng thái active trong danh mục phòng đã thiết lập |
| Out Of Inventory | Numberic | C |  | Hiển thị Số lượng phòng bị block với trạng thái OOI |
| Total Available Room | Numberic | C |  | = Total room - Out of Inventory |
| Out Of Order | Numberic | C |  | Hiển thị số lượng phòng bị block với trạng thái OOO |
| Total Room to sell | Numberic | C |  | = Total Available Room-Out Of Order |
| **Moving Activities** | | | | |
| Departure Expected [Room] |  |  |  | Số lượng đặt phòng có Departure Date = Ngày tại Dashboard mà đã check in nhưng chưa check out |
| Departure Expected [Pax] |  |  |  | Số lượng khách có Departure Date = Ngày tại Dashboard mà đã check in nhưng chưa check out |
| Departure Actual [Room] |  |  |  | Số lượng đặt phòng có Departure Date = Ngày tại Dashboard mà đã check in và đã check out |
| Departure Actual [Pax] |  |  |  | Số lượng khách có Departure Date = Ngày tại Dashboard mà đã check in và đã check out |
| Arrival Expected [Room] |  |  |  | Số lượng đặt phòng Active có Arrival Date = Ngày tại Dashoard mà chưa check in |
| Arrival Expected [Pax] |  |  |  | Số lượng khách Active (Deleted=N) có Arrival Date = Ngày tại Dashoard mà chưa check in |
| Pre-assigned Room [Room] |  |  |  | Số lượng đặt phòng Active có Arrival Date = Ngày tại Dashoard mà chưa check in nhưng đã được gán phòng |
| Pre-assigned Room [Pax] |  |  |  | Số lượng khách Active thuộc đặt phòng Active có Arrival Date = Ngày tại Dashoard mà chưa check in nhưng đã được gán phòng |
| Arrival Actual [Room] |  |  |  | Số lượng đặt phòng có Arrival Date = Ngày tại Dashoard mà đã check in |
| Arrival Actual [Pax] |  |  |  | Số lượng khách Active có Arrival Date = Ngày tại Dashoard mà đã check in |
| Current Occupied [Room] |  |  |  | Số lượng phòng đã check in mà chưa check out tại ngày lựa chọn trên Dashboard |
| Current Occupied [Pax] |  |  |  | Số lượng khách đã check in mà chưa check out tại ngày lựa chọn trên Dashboard |
| Extended Stays [Room] |  |  |  | Số lượng phòng đã check in và có: Departure mới > Departure cũ và Departure cũ = Ngày tại ô Date |
| Extended Stays [Pax] |  |  |  | Số lượng khách đã check in và có: Departure mới > Departure cũ và Departure cũ = Ngày tại ô Date |
| Early Departure [Room] |  |  |  | Số lượng phòng đã check in và có Departure mới < Departure cũ và Departure cũ = Ngày tại ô Date |
| Early Departure [Pax] |  |  |  | Số lượng khách đã check in và có Departure mới < Departure cũ và Departure cũ = Ngày tại ô Date |
| Day Room [Room] |  |  |  | Số lượng đặt phòng Active có Arrival=Departure=Ngày tại ô Date |
| Day Room [Pax] |  |  |  | Số lượng khách Active có Arrival=Departure=Ngày tại ô Date |
| Same day Reservation [Room] |  |  |  | Số lượng đặt phòng có ngày đặt phòng = Ngày Check in và Ngày check in = Ngày tại ô Date |
| Same day Reservation [Pax] |  |  |  | Số lượng khách có ngày đặt phòng = Ngày Check in và Ngày check in = Ngày tại ô Date |
| Walk in [Room] |  |  |  | Số lượng đặt phòng có Account =Walk In và Arrival <=Ngày tại ô Date và Departure >Ngày tại ô Date (chỉ tính các đặt phòng Active) |
| Walk in [Pax] |  |  |  | Số lượng khách thuộc các đặt phòng có Account =Walk In và Arrival của khách <=Ngày tại ô Date và Departure của khách >Ngày tại ô Date (chỉ tính các đặt phòng Active) |

#### Màn hình Room Availability



Hình. Màn hình Room Availability

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field Name | Format/size | M/C/O | Default value | Rules Description |
| From | Date | M | Ngày hiện tại trên hệ thống | Ngày xem dữ liệu từ |
| To | Date | M | Ngày hiện tại trên hệ thống + 30 ngày | Ngày xem dữ liệu đến  To phải >= from  ~~To- from không vượt quá 60 ngày. Nếu vượt quá thì hệ thống cảnh báo “Only view Room ability in 60 days”~~ |
| Search | Button |  |  | Nhấn để hiển thị dữ liệu theo ngày From – To đã chọn |
| Export | Button |  |  | Nhấn để xuất bảng dữ liệu ra file excel |
| **Thống kê phòng theo Room type** | | | | |
| Room type | Text | C |  | Hiển thị danh sách các mã Room type  Room type sắp xếp theo số thứ tự (Sort) đã thiết lập trong Danh mục Room type |
| Name | Text | C |  | Hiển thị danh sách tên các room type tương ứng |
| Total | Numeric | C |  | Hiển thị số lượng phòng có trạng thái active tương ứng theo Room type |
| [Ngày – Thứ] | Text | C |  | - Tên cột hiển thị theo từng ngày trong khoảng thời gian đã chọn. Chỉ hiển thị ngày tháng (không hiển thị năm) và thứ tương ứng  - Giá trị trong Cột hiển thị số lượng phòng còn trống (Vacant) của từng loại phòng theo từng ngày tương ứng, đã loại trừ các phòng bị OOO/OOI. Được tính theo công thức:  = Tổng số phòng của loại phòng – Số phòng bị OOO/OOI – Số đặt phòng có trạng thái Active của loại phòng đó mà có Arrival date <= ngày tương ứng và Departure Date > ngày tương ứng trên cột |
| Dòng **Total** | Numeric | C |  | Tính tổng tất cả các cột tại phần Thống kê phòng theo Room type |
| Dòng **OOI/OOO** | Numeric | C |  | Hiển thị giá trị tại các cột [Ngày – thứ]  Giá trị hiển thị là số lượng các phòng có trạng thái block là OOI hoặc OOO tại các ngày tương ứng |
| Dòng **Available Rms** | Numeric |  |  | Hiển thị giá trị tại các cột [Ngày – thứ]  Giá trị hiển thị = Giá trị tại (dòng **Total** và cột **Tota**l) **-** Dòng **OOI/OOO** |
| Dòng **OOS** | Numeric |  |  | Hiển thị giá trị tại các cột [Ngày – thứ]  Giá trị hiển thị là số lượng các phòng có trạng thái block là OOS tại các ngày tương ứng |
| Dòng **Salesable Rms** | Numeric |  |  | Hiển thị giá trị tại các cột [Ngày – thứ]  Giá trị hiển thị = Dòng **Available Rms -** Dòng **OOS** |
| Dòng **Definite** | Numeric |  |  | Hiển thị giá trị tại các cột [Ngày – thứ]  Giá trị hiển thị là số lượng đặt phòng có trạng thái (reservation type)= Definite và có Arrival date <= ngày tương ứng và Departure Date > ngày tương ứng trên cột  Hiển thị giá trị phần trăm được tính bằng số lượng tại Definite/ **Salesable Rms** |
| Dòng **Tentative** | Numeric |  |  | Hiển thị giá trị tại các cột [Ngày – thứ]  Giá trị hiển thị là số lượng đặt phòng có trạng thái (reservation type)= Tentative và có Arrival date <= ngày tương ứng và Departure Date > ngày tương ứng trên cột  Hiển thị giá trị phần trăm được tính bằng số lượng tại Tentative/ **Salesable Rms** |
| Dòng **Holding** | Numeric |  |  | Hiển thị giá trị tại các cột [Ngày – thứ]  Giá trị hiển thị là số lượng đặt phòng có trạng thái (reservation type)= Holding và có Arrival date <= ngày tương ứng và Departure Date > ngày tương ứng trên cột  Hiển thị giá trị phần trăm được tính bằng số lượng tại Holding/ **Salesable Rms** |
| Dòng **Total** | Numeric |  |  | Hiển thị giá trị tại các cột [Ngày – thứ]  Giá trị hiển thị X là tổng số lượng tại Definite, Tentative, Holding và phần trăm được tính bằng số lượng X/ **Salesable Rms** |
| Dòng House use | Numeric |  |  | Hiển thị giá trị tại các cột [Ngày – thứ]  Giá trị hiển thị là số lượng đặt phòng là House use và có Arrival date <= ngày tương ứng và Departure Date > ngày tương ứng trên cột |
| Dòng Com | Numeric |  |  | Hiển thị giá trị tại các cột [Ngày – thứ]  Giá trị hiển thị là số lượng phòng có đặt phòng là Complimentary và có Arrival date <= ngày tương ứng và Departure Date > ngày tương ứng trên cột |
| Dòng Allotment | Numeric |  |  | ???? |
| FIT Arrival | Numeric |  |  | Hiển thị số lượng đặt phòng có khách lẻ/khách vãng lai đến vào ngày tương ứng (tính cả khách arrival đã check in) |
| GIT Arrival | Numeric |  |  | Hiển thị số lượng đặt phòng có khách đoàn đến vào ngày tương ứng (tính cả khách arrival đã check in) |
| Waiting list | Numeric |  |  | Hiển thị số lượng Reservation có trạng thái là Waiting |

# THIẾT KẾ TEST CASE VÀ THỰC HIỆN TEST – LOG BUG

## Công cụ ứng dụng

## Thực hiện thử nghiệm trên một ứng dụng Web

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 |
|  |  |  |
|  |  |  |
|  | | |

# 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, số, trang

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

# PHỤ LỤC