**ĐẠI HỌC QUỐC GIA TP.HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

🙤🙧🟍🙥🙦



**Kế hoạch, qui trình, công cụ**

**Đề tài quản lý xổ tiết kiệm**

|  |  |  |
| --- | --- | --- |
| Sinh viên thực hiện: | | |
| STT | Họ tên | MSSV |
| 1 | Nguyễn Lộc Linh | 19521754 |
| 2 | Trần Duy Quang | 19522102 |
| 3 | Nguyễn Trí Tài | 19522572 |

# Kế hoạch phát triển (Release Roadmap)

• GIAI ĐOẠN 1: Phân tích, thiết kế phần mềm theo yêu cầu ban đầu của khách hàng.

• GIAI ĐOẠN 2: Kiểm thử, thu phập ý kiến của khách hang về chương trình (lần 1).

• GIAI ĐOẠN 3: Cập nhật theo mong muốn của khách hang ở giai đoạn 2

• GIAI ĐOẠN 4: Kiểm thử, thu thập ý kiến của khách hàng về chương trình (lần 2). Từ đó hoàn thiện sản phẩm đầu ra cuối cùng.

# Qui trình phát triển đồ án

Sơ đồ mô hình Thác nước cải tiến (Căn bản):

Diagram

Description automatically generated

- Từ mô hình trên, ta có thể chia quy trình thành 6 giai đoạn:

**+ *Giai đoạn 1*: Analysis and Definition - Phân tích và xác định:**

Bước 1: Phân tích và đặc tả yêu cầu:

\* Gồm có 6 yêu cầu:

1. Mở sổ tiết kiệm

2. Lập phiếu gởi tiền

3. Lập phiếu rút tiền

4. Tra cứu sổ

5. Lập báo cáo tháng

6. Thay đổi qui định

Dựa vào bản mẫu "Phân tích xác định yêu cầu Đề tài quản lý sổ tiết kiêm" để phân tích 6 yêu cầu trên

Bước 2: Xác định quy trình phát triển:

Xác định sử dụng mô hình Thác nước cải tiến. Xem xét các bước và giai đoạn tiếp theo sau khi đã phân tích các yêu cầu (Gồm 6 yêu cầu)

[ Cụ thể xác định các giai đoạn tổng quát của quy trình như sau:

A screenshot of a computer

Description automatically generated]

Bước 3: Phân bổ nguồn lực

Có rất nhiều cách đê phân bổ nguồn lực. Ta có thể làm như sau:

Do ta đã xác định sử dụng Mô hình thác nước cải tiến nên ta có thể phân bố nguồn lực đảm nhiệm 5 vai trò:

1. Analysis and Definition: Bộ phận này nhằm phân tích xác định các yêu cầu

2. Design and Implementation: Bộ phận này nhằm thiết kế phần mềm phù hợp với các yêu cầu đã được phân tích của bộ phận nắm chức năng Analysis ad Definition

3. Test: Bộ phận này kiểm tra code, kiểm tra lỗi code, sai sót, lập danh sách các lỗi, danh sách các yêu cầu không đáp ứng, ... Nếu có lỗi code thì đưa lại cho bộ phận Design khắc phục code

4. Usage and Maintenance: Thu thập feedback của user. Thu thập, cập nhật các yêu cầu của user, danh sách các yêu cầu thay đổi, các yêu cầu không được thỏa. Nếu tất cả đã thỏa thì thực hiện nhiệm vụ bảo trì phần mềm

5. Update: Nếu ở bộ phận Usage and Maintenance thu thập được feedback của user, bộ phận này có chức năng phân loại các feedback của user, sau đó đưa về lại cho từng bộ phận đảm nhiệm phần liên quan đến các feedback đó cho đến khi thỏa hết các yêu cầu user

\* Như ta biết nhược điểm của Mô hình thác nước là sau khi đã hoàn tất ở bộ phận nào và chuyển giao nhiệm vụ cho bộ phận kế tiếp thì không thể thay đổi hay sửa chữa lỗi của bộ phận trước đó (Nếu có lỗi). Vậy ta nên phân bố nguồn lực chủ yếu ở các bộ phận đầu tiên (Bộ phận Analysis and Definition và Design and Implementation), trung bình ở bộ phận Test và các bộ phận như Usage and Maintenance và Update thì không cần quá nhiều nguồn nhân lực

**+ *Giai đoạn 2:* Design**

Từ những yêu cầu đã được phân tích ở giai đoạn 1, ta thiết kế ra mô hình tổng quát xây dựng phần mềm

Phân chia nguồn lực trong bộ phận có nhiệm vụ Desgin thành nhiều bộ phận nhỏ hơn, mỗi bộ phận nhỏ đảm nhiệm 1 yêu cầu nhỏ nào đó

\* Trong quá trình thực hiện giai đoạn 2, ta bắt buộc phải thiết kế mô hình đáp ứng các yêu cầu đã được phân tích như trên, sau đó ta mới thiết kế và tự cải tiến phần mềm (Vd: Tốc độ phần mềm, giao diện, độ khả dụng, khả năng tương tác giữa phần mềm và user,...)

**+ *Giai đoạn 3:* Implement**

Chạy thử phần mềm. Sau khi các bộ phận đảm nhiệm 1 yêu cầu nhỏ trong bộ phận Design, ta tổng hợp lại 'code' của các bộ phận đó, điều chỉnh và chỉnh sửa để các 'code' đó có thể thực thi với nhau thành 1 phần mềm hoàn chỉnh. Sau khi phần mềm đã thực thi được thì chuyển sang giai đoạn 4

**+ *Giai đoạn 4:* Test**

Sau khi đã có 1 phần mềm hoàn chỉnh, ta đưa phần mềm đó cho bộ phận Test để kiểm tra phần mềm.

Kiểm tra xem phần mềm đã thỏa các yêu cầu đã được phân tích xác định từ trước hay chưa. Kiểm tra các trường hợp đặc biệt, kiểm tra các trường hợp gây lỗi, tự gây lỗi cho phần mềm, Kiêm tra thời gian, chất lượng phần mềm,....

Nếu không thỏa thì yêu cầu bộ phận Design and Implementation khắc phục. Sau khi khắc phục thì đưa lại cho bộ phận Test cho tới khi phần mềm được chấp nhận

**+ *Giai đoạn 5:* Usage and Maintenance**

Do bộ phận Usage and Maintenance và bộ phận Update đảm nhiệm. Bộ phận Usage and Maintenance chuyển giao phần mềm 'thử' cho user sử dung (Phần mềm sau khi đã qua bộ phận Test). Thu thập các feedback của user. Hệ thống các yêu cầu mới của user và cập nhật các yêu cầu của user

Sau khi đã hệ thống được các feedback của user, bộ phận Update chuyển về cho từng bộ phận các 'lỗi' hay 'không hài lòng' trong feedback của user. Yêu cầu các bộ phận đó chỉnh sửa khắc phục các 'lỗi' đó.

Quá trình này lặp đi lặp lại cho đến phần mềm thỏa các yêu cầu của user. Sau đó ta đưa ra bả hoàn chỉnh của phần mềm

# Công cụ (CASE Tools) hỗ trợ trong suốt quá trình quản lý công việc

Các công cụ hỗ trợ bao gồm:

* Roadmap: làm timeline chi tiết về kế hoạch phát triển đồ án
* Trello: công cụ quản lý công việc và thời gian

Graphical user interface, application, Teams

Description automatically generated

* Android studio, Flutter: Công cụ code chương chình phần mềm

Graphical user interface, text

Description automatically generated

Text

Description automatically generated