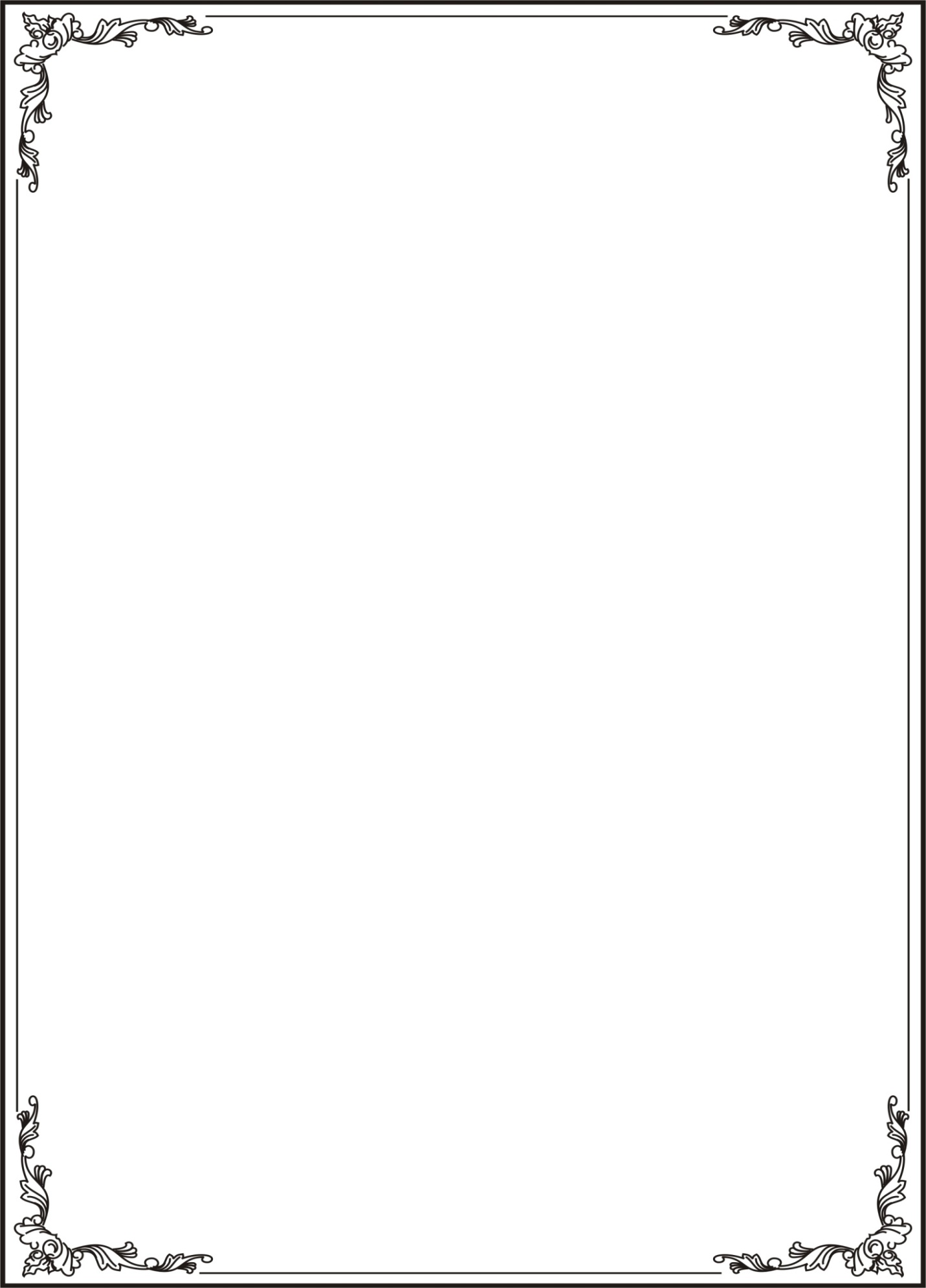
**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔN**G

**KHOA CÔNG NGHỆ THÔNG TIN**

-----🙞🙜🕮🙞🙜-----

****

**BÁO CÁO NHÓM**

**NHẬP MÔN CÔNG NGHỆ PHẦN MỀM**

***CHAPTER 11: REQUIREMENTS***

**Giảng viên : ĐẶNG NGỌC HÙNG**

**Nhóm lớp : 08 - Nhóm bài tập: 08**

**Sinh viên : PHẠM CÔNG BÌNH – MSV: B18DCAT020**

**LÊ TRUNG ĐÔNG – MSV: B18DCCN160**

**LÊ ĐẮC DUY – MSV: B18DCCN108**

**ĐỖ QUANG HUY – MSV: B18DCAT106**

**LẠI VĂN XUÂN – MSV: B18DCCN701**

BẢNG PHÂN CÔNG CÔNG VIỆC

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Họ và tên | Công việc | Đánh giá |
| 1 | Đỗ Quang Huy(Trưởng nhóm) |  |  |
| 2 | Phạm Công Bình |  |  |
| 3 | Lê Trung Đông |  |  |
| 4 | Lê Đắc Duy |  |  |
| 5 | Lại Văn Xuân |  |  |

MỤC LỤC

BẢNG THUẬT NGỮ.…………………………………………………………………………….1

NỘI DUNG ...……………………………………………………………………………………..2

PHẦN I. XÁC ĐỊNH YÊU CẦU KHÁCH HÀNG ……………………………………………….2

PHẦN II. TỔNG QUAN VỀ LUỒNG CÔNG VIỆC XÁC ĐỊNH YÊU CẦU…………………...2

PHẦN III. HIỂU VỀ MIỀN………………………………………………………………………..2

PHẦN IV. MÔ HÌNH NGHIỆP VỤ……………………………………………………………….2

4.1. Phỏng vấn……………………………………………………………………………..3

4.2. Các kỹ thuật khác……………………………………………………………………...3

4.3. Các Use Case………………………………………………………………………….3

PHẦN V. CÁC YÊU CẦU BAN ĐẦU…………………………………………………………...5

PHẦN VI. HIỂU VỀ DOMAIN: CASE STUDY MSG FOUNDATION…………………………5

PHẦN VII. MÔ HÌNH NGHIỆP VỤ BAN ĐẦU: CASE STUDY MSG FOUNDATION………6

PHẦN VIII. CÁC YÊU CẦU BAN ĐẦU: CASE STUDY MSG FOUNDATION………………9

PHẦN IX. TIẾP TỤC VỚI LUỒNG CÔNG VIỆC XÁC ĐỊNH YÊU CẦU: CASE STUDY MSG

FOUNDATION ………………………………………………………………………………….10

PHẦN X. SỬA ĐỔI CÁC YÊU CẦU: CASE STYDY MSG FOUNDATION……………….…11

PHẦN XI. LUỒNG CÔNG VIỆC KIỂM THỬ: CASE STUDY MSG FOUNDATION………..16

PHẦN XII. PHA XÁC ĐỊNH YÊU CẦU CỔ ĐIỂN……………….…………………………….21

PHẦN XIII. BẢN MẪU NHANH………………………………………………………………..21

PHẦN XIV. NHÂN TỐ CON NGƯỜI…………………………………………………………..21

PHẦN XV. SỬ DỤNG LẠI BẢN MẪU NHANH………………………………………………22

PHẦN XVI. CÁC CÔNG CỤ CASE CHO XÁC ĐỊNH YÊU CẦU…………………………….22

PHẦN XVII. CÁC THƯỚC ĐO CHO XÁC ĐỊNH YÊU CẦU…………………………………22

PHẦN XVIII. NHỮNG THỬ THÁCH CHO PHA XÁC ĐỊNH YÊU CẦU……………………23

PHẦN XIX. TỔNG KẾT ...………………………………………………………………………23

PHẦN XX. TÀI LIỆU THAM KHẢO ...………………………………………………………...23

BẢNG THUẬT NGỮ

|  |  |
| --- | --- |
| Thuật Ngữ | Ý Nghĩa |
| Workflow | Luồng công việc |
| Application Domain | Miền ứng dụng |
| Business Model | Mô hình nghiệp vụ |
| Structured Interview | Phỏng vấn có cấu trúc |
| Unstructured Interview | Phỏng vấn không có cấu trúc |
| Questionnaire | Bảng câu hỏi |
| Direct Observation | Quan sát trực tiếp |
| Use Case | Chức năng nhỏ nhất của ứng dụng hoặc là nghiệp vụ của một hệ thống nào đó và được sử dụng bởi một tác nhân(actor) hoặc một nhóm tác nhân. |
| Functional Requirement | Yêu cầu chức năng |
| Nonfunctional Requirement | Yêu cầu phi chức năng |
| Case Study | Một phương pháp nghiên cứu tình huống hay sự việc thực tế, giúp việc tiếp thu kiến thức trở lên hiệu quả hơn. |
| Rapid Prototyping | Bản mẫu nhanh |
| User Interface | Giao diện người dùng |
| Human-computer Interface (HCI) | Giao diện người-máy tính |
| System Architect  Software through Pictures | Các môi trường CASE đồ họa được mở rộng để trợ giúp UML |
| IBM Rational Rose  Together  ArgoUML(open source) | Các môi trường CASE hướng đối tượng |

NỘI DUNG

I. XÁC ĐỊNH YÊU CẦU KHÁCH HÀNG

* Một quan niệm sai lầm khá phổ biến trong *luồng công việc xác định yêu cầu* là người phát triển phải xác định phần mềm mà khách hàng mong muốn,mà thực tế họ phải xác định phần mềm mà khách hàng cần.
* Khách hàng thường gặp khó khăn về các vấn đề kỹ thuật cũng như trong việc truyền đạt ý tưởng của họ với nhóm phát triển.
* Nhóm phát triển phải làm rõ được những thông tin cần thiết cho khách hàng hiểu.
* Giải pháp cho việc này:
  + Thu nhập thông tin ban đầu từ khách hàng và người dùng.
  + Sử dụng thông tin ban đầu làm đầu vào cho các luồng yêu cầu của quy trình hợp nhất.

II. TỔNG QUAN VỀ LUỒNG CÔNG VIỆC XÁC ĐỊNH YÊU CẦU

* Mục đích chung của *luồng công việc xác định yêu cầu* là để cho tổ chức phát triển xác định nhu cầu khách hàng.
* Bước đầu để hướng tới mục tiêu này là hiểu về *miền ứng dụng*(miền) – môi trường cụ thể mà sản phẩm hoạt động. Từ đó, nhóm phát triển có thể xây dựng *mô hình nghiệp vụ*, sử dụng biểu đồ UML để mô tả quá trình công việc của khách hàng.

III. HIỂU VỀ DOMAIN

* Để hiểu được nhu cầu của khách hàng, các thành viên của đội phân tích yêu cầu phải làm quen với *miền ứng dụng*.
* Việc sử dụng chính xác các thuật ngữ khi giao tiếp với khách hàng là vô cùng quan trọng.
* Việc sử dụng từ ngữ không phù hợp có thể gây hiểu lầm dẫn đến bàn giao sản phẩm bị lỗi.
* Để giải quyết vấn đề các thuật ngữ trong *miền*, ta có thể xây dựng các bảng chú giải thuật ngữ với danh sách các từ ngữ chuyên ngành kỹ thuật và ý nghĩa của chúng. Bảng thuật ngữ không chỉ làm giảm sự nhầm lẫn giữa khách hàng và nhà phát triển mà nó còn có ích trong việc giảm bớt hiểu lầm giữa các thành viên trong nhóm phát triển.

IV. MÔ HÌNH NGHIỆP VỤ

* *Mô hình nghiệp vụ* là sự miêu tả các quá trình nghiệp vụ của một tổ chức.
* *Mô hình nghiệp vụ* cung cấp thông tin về toàn bộ nghiệp vụ của khách hàng, giúp nhà phát triển đưa ra thông tin cho khách hàng về mặt tính toán.
* Các nhà phân tích hệ thống cần có hiểu biết chi tiết về các loại quá trình nghiệp vụ khác nhau. Các kỹ thuật khác nhau được sử dụng để có được thông tin cần thiết để xây dựng *mô hình nghiệp vụ*.

4.1. Phỏng vấn

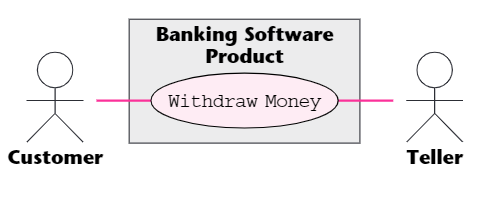
* Các thành viên của nhóm xác định yêu cầu cần gặp gỡ khách hàng và người dùng để thu thập những thông tin liên quan.
* Có hai loại câu hỏi cơ bản:
  + Câu hỏi kết thúc đóng: yêu cầu một câu trả lời cụ thể.
  + Câu hỏi kết thúc mở: khuyến khích người được phỏng vấn nói ra ý kiến của mình.
* Có hai kiểu phỏng vấn:
  + Trong *cuộc phỏng vấn có cấu trúc*, các câu hỏi được lập kế hoạch cụ thể từ trước thường là những câu hỏi kết thúc đóng.
  + Trong *cuộc phỏng vấn không có cấu trúc*, các câu hỏi được đưa ra để phản ứng lại những câu trả lời nhận được thường là câu hỏi có kết thúc mở.
* Cuộc phỏng vấn phải có cấu trúc mới có khả năng mang lại nhiều thông tin liên quan.
* Một cuộc phỏng vấn tốt không phải lúc nào cũng dễ dàng:
  + Đầu tiên, người phỏng vấn phải thực sự quen thuộc với *miền ứng dụng*.
  + Thứ hai, người phỏng vấn phải lắng nghe, tiếp thu cẩn thận thông tin từ người được phóng vẫn.
* Sau cuộc phỏng vấn, người phỏng vấn phải chuẩn bị một báo cáo bằng văn bản đưa ra kết quả của cuộc phỏng vấn cũng như một bản sao cho người được phỏng vấn.

4.2. Các kỹ thuật khác

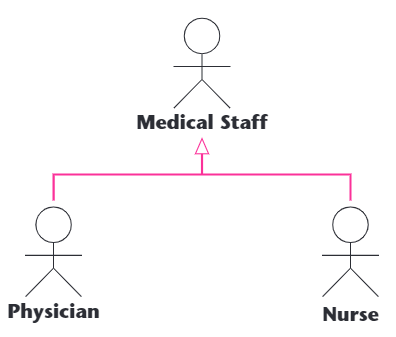
* Bên cạnh phỏng vấn có một cách khác để có được thông tin về các hoạt động của tổ chức khách hàng là gửi một *bảng câu hỏi* cho các thành viên của tổ chức đó. Một bản thăm dò ý kiến rất hữu ích khi lấy ra ý kiến của hàng trăm người.
* Một cách khác là kiểm tra các định dạng nghiệp vụ. Một tập hợp toàn diện các loại dữ liệu khác nhau về cách khách hàng thực hiện những công việc nghiệp vụ có thể rất hữu ích trong việc xác định nhu cầu khách hàng.
* *Việc quan sát trực tiếp* những người công nhân thực hiện nhiệm vụ của họ cũng có thể là một cách hữu ích. Trong kỹ thuật này, các máy quay được thiết lập tại nơi làm việc để ghi lại những gì đang được thực hiện. Tuy nhiên, kỹ thuật này gặp phải một vài khó khăn khi việc phân tích các băng video tốn rất nhiều thời gian và có thể là hành vi xâm phạm quyền riêng tư.

4.3. Các Use Case

* Một Use Case mô hình hóa sự tương tác giữa sản phẩm phần mềm và người dùng sản phẩm phần mềm đó.
* Ví dụ mô tả một Use Case từ một sản phẩm phần mềm ngân hàng:



* + Có hai tác nhân trong ví dụ là khách hàng và giao dịch viên.
  + Hoạt động nghiệp vụ trong Use Case này là rút tiền.
* Một cách khác để xem xét một Use Case là sự tương tác giữa sản phẩm phần mềm và môi trường mà nó hoạt động. Nghĩa là một tác nhân là một thành viên của thế giới bên ngoài phần mềm.
* Thường dễ dàng nhận dạng ra tác nhân:
  + Một tác nhân thường là người dùng sản phẩm phần mềm.
  + Một tác nhân đóng vai trò nào đó đối với sản phẩm phần mềm (không nhất thiết phải là người sử dụng phần mềm).
* Một người dùng hệ thống có thể giữ nhiều vai trò. Ví dụ một người khách của ngân hàng có thể là một người vay tiền hoặc một người gửi tiền.
* Một tác nhân có thể tham gia vào nhiều Use Case. Ví dụ một người vay tiền có thể là tác nhân trong Use Case *tiền đi vay*, trong Use Case *khoản vay phải trả*, trong Use Case *khoản vay trả nợ gốc*.
* Một tác nhân không nhất thiết phải là một con người. Ví dụ hệ thống thông tin thương mại điện tử phải tương tác với hệ thống thông tin công ty thẻ tín dụng:
  + Hệ thống thông tin công ty thẻ tín dụng là một tác nhân theo quan điểm của hệ thống thương mại điện tử.
  + Hệ thống thương mại điện tử là một tác nhân của hệ thống thông tin theo quan điểm của công ty thẻ tín dụng.
* Vấn đề trong việc xác định các tác nhân là sự nạp chồng tác nhân. Ví dụ trong sản phẩm phần mềm bệnh viện:



* Một Use Case có tác nhân Y tá.
* Một Use Case khác có tác nhân là Nhân viên y khoa.
* Vì tất cả Y tá đều thuộc Nhân viên y khoa nhưng một số nhân viên Y khoa(Bác sĩ) không phải là Y tá. Sẽ tốt hơn khi có cả Bác sĩ và Y tá.
* Giải pháp: Nhân viên y khoa có thể định nghĩa với hai chuyên môn là Bác sĩ và Y tá.

V. CÁC YÊU CẦU BAN ĐẦU

* Các *yêu cầu ban đầu* được đưa ra dựa trên *mô hình nghiệp vụ* đầu tiên. Sau đó, *miền* và *mô hình nghiệp vụ* được tinh chỉnh trên cơ sở thảo luận thêm với khách hàng.
* Các yêu cầu có sự thay đổi thường xuyên không chỉ đối với bản thân nó mà còn về quan điểm của nhóm phát triển, khách hàng và người dùng đối với từng yêu cầu. Một cách tốt để xử lý những thay đổi thường xuyên này là duy trì một danh sách những yêu cầu quan trọng, cùng với các Use Case của những yêu cầu đã được phê chuẩn bởi khách hàng.
* Các yêu cầu được chia thành hai loại:
  + *Yêu cầu chức năng* chỉ rõ hành động mà sản phẩm phần mềm phải có khả năng thực hiện. Thường được biểu diễn dưới dạng đầu vào và đầu ra.
  + *Yêu cầu phi chức năng* chỉ rõ những đặc trưng của sản phẩm phần mềm như các ràng buộc về nền tảng, thời gian phản hồi, hoặc độ tin cậy.
* Những yêu cầu chức năng được xử lý như là một phần của *luồng công việc xác định yêu cầu và phân tích*.
* *Các yêu cầu phi chức năng* phải đợi đến *luồng công việc thiết kế* vì cần những thông tin chi tiết cho *yêu cầu phi chức năng* không có sẵn cho đến khi *luồng công việc phân tích và xác định yêu cầu* hoàn thành.

VI. HIỂU VỀ DOMAIN: CASE STUDY MSG FOUNDATION

* Sau khi Marth Stockton Greengage qua đời ở tuổi 87, bà để lại toàn bộ tài sản 2,3 tỷ USD của mình để làm từ thiện. Cụ thể, bà thành lập một quỹ mang tên mình(quỹ MSG). Mục đích của quỹ này để giúp các cặp vợ chồng trẻ mua nhà riêng bằng cách cho vay lãi suất thấp.
* Quỹ MSG tổ chức phát triển phần mềm thực hiện các tính toán cần thiết để xác định xem sẽ có bao nhiêu tiền có sẵn mỗi tuần để cho người vay mua nhà.
* Không có nhiều người có đủ khả năng chi trả tiền mua nhà. Thay vào đó, họ trả một phần nhỏ và vay phần còn lại.
* Ví dụ: Có một cặp đôi muốn mua một ngôi nhà giá $100.000. Họ trả khoản tiền đặt cọc $10.000 và đi vay $90.000 còn lại.
  + Cách thức tính tiền cho vay:

Số tiền vay: $100.000

Ký gửi trước: $10.000 (10%)

Vốn cần phải trả: $90.000 (trong 30 năm)

Lãi suất: 7,5% một năm (0,625% một tháng)

* Vì trả trong vòng 30 năm (360 tháng) nên số tiền mỗi tháng người vay phải trả được áp cứng là: $629,3.
* Số thiền người vay trả mỗi tháng = lãi suất + tiền gốc. Nghĩa là trong $629,3 bao gồm cả tiền lãi trong tháng đó + số tiền gốc của tháng đó.
* Cụ thể trong cách tính toán:
  + Tháng đầu tiên:
    - Người vay trả trước 10%: $10.000
    - Tiền gốc còn lại là: $90.000
  + Cuối tháng đầu tiên:
    - Người vay trả: $629,3
    - Lãi suất: $90.000 \* 0,625% = $562,5
    - Tiền gốc đã trả: $629,3 - $562,5 = $66,8
    - Tiền gốc còn lại: $90.000 – $66,8 = $89.933,2
  + Cuối tháng thứ 2:
    - Người vay trả: $629,3
    - Lãi suất: $89.933,2 \* 0,625% = $562,08
    - Tiền gốc đã trả: $629,3 – $562,08 = $67,22
    - Tiền gốc còn lại: $89.933,2 – $67,22 = $89.865,98

………………….

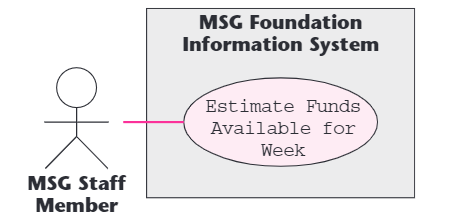
* + 15 năm sau (180 tháng):
    - Người vay trả: $629,3
    - Lãi suất: $67.881,61 \* 0,625% = $424,26
    - Tiền gốc đã trả: $629,3 – $424,26 = $205,04
    - Tiền gốc còn lại: $67.881,61 – $205,04 = $67.676, 57

………………….

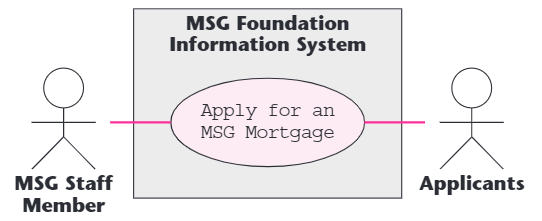
* + Sau 30 năm(360 tháng) người vay sẽ trả hết nợ.
* Công ty tài chính muốn chắc chắn rằng họ sẽ nhận được số tiền $90.000 cùng với lãi suất thì phải yêu cầu người vay thỏa thuận theo các cách sau:
  + Đầu tiên, người vay phải ký một văn bản pháp lý quy định rằng: Nếu người vay không thực hiện các khoản thanh toán hàng tháng thì công ty tài chính có thể bán căn nhà đó và sử dụng số tiền thu được để trả hết số dư nợ của người vay.
  + Thứ hai, công ty tài chính yêu cầu người vay mua bảo hiểm cho căn nhà của mình, nếu căn nhà xuống cấp thì công ty bảo hiểm sẽ chịu trách nhiệm chi trả phần mất mát cho công ty tài chính. Phí bảo hiểm thường được trả mỗi năm một lần bởi công ty tài chính.
  + Thứ ba, điều kiện để công ty tài chính không thực hiện hành động thế chấp nếu tổng số tiền trả hàng tháng(tiền lãi + tiền vốn + tiền bảo hiểm + thuế bất động sản) vượt quá 28% tổng thu nhập của người vay.

VII. MÔ HÌNH NGHIỆP VỤ BAN ĐẦU: CASE STUDY MSG FOUNDATION

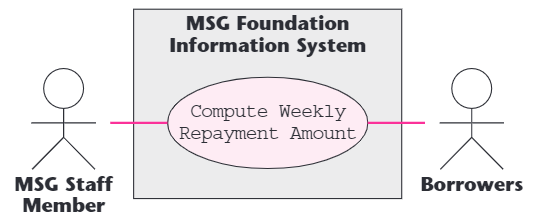
* Vào đầu mỗi tuần, MSG ước tính số tiền sẽ được dùng trong thế chấp.
* Một nhân viên của MSG Foundation xác định:
  + Người vay có đủ tiêu chuẩn để được thế chấp hay không.
  + MSG Foundattion có đủ tiền để mua nhà hay không.
* Nếu khoản thế chấp được chấp nhận thì việc trả nợ thế chấp hàng tuần được tính theo quy định của MSG Foundation.
* Có ba Use Case:
  + *Ước tính số tiền có sẵn cho tuần đó*.
  + *Đăng ký thế chấp MSG*.
  + *Tính toán số tiền phải trả hàng tuần*.
* Hình ảnh minh họa các Use Case:



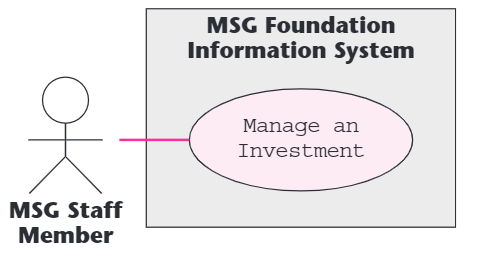
* Mô tả: Use Case này cho phép nhân viên của MSG Foundation ước tính số tiền mà tổ chức có sẵn trong tuần đó để tài trợ cho các khoản thế chấp.



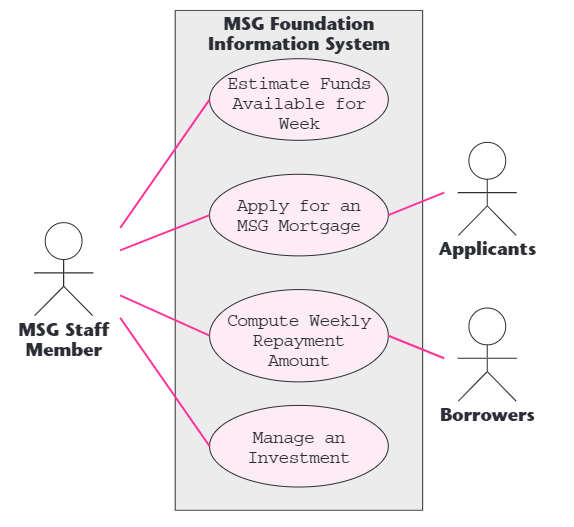
* Mô tả: Use Case này cho phép nhân viên MSG Foundation xác định xem người vay có đủ điều kiện để được được thế chấp hay không, và nếu có thế, liệu hiện tại có đủ tiền để thế chấp hay không.
* Người nộp đơn là một tác nhân của Use Case *đăng ký thế chấp MSG* mặc dù không tương tác với sản phẩm phần mềm. Vì:
  + Người nộp đơn mở đầu Use Case.
  + Người nộp đơn cung cấp thông tin cho nhân viên MSG nhập.



* Mô tả: Use Case này cho phép nhân viên MSG Foundation tính toán số tiền người vay phải trả mỗi tuần.
* Người vay là một tác nhân của Use Case: *tính toán số tiền phải trả hàng tuần* mặc dù cũng không tương tác với sản phẩm phần mềm. Vì:
  + Người vay mở đầu Use Case.
  + Người vay cung cấp thông tin cho nhân viên MSG nhập.
* Một khía cạnh khác của *mô hình nghiệp vụ* MSG Foundation là các khoản đầu tư của MSG Foundation. Ở phần này, không có chi tiết nào được biết vể việc mua bán các khoản đầu tư cũng như làm thế nào thu nhập đầu tư có sẫn cho các khoản thế chấp.
* Use Case *quản lý một khoản đầu tư* là một phần không thể thiếu của *mô hình nghiệp vụ ban đầu*.

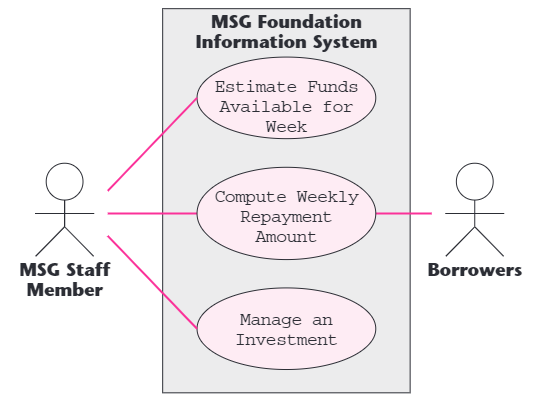


* Mô tả: Use Case này cho phép nhân viên MSG Foundation mua bán và quản lý danh mục đầu tư.
* Biểu đồ Use Case về *mô hình nghiệp vụ ban đầu* của Case Study MSG Foundation:



VIII. CÁC YÊU CẦU BAN ĐẦU: CASE STUDY MSG FOUNDATION

* Sẽ không rõ ràng khi trả lời câu hỏi: Liệu bốn Use Case phía trên có phải là tất cả yêu cầu của sản phẩm phần mềm được phát triển hay không? Cách tốt nhất để giải quyết vấn đề trong câu hỏi trên là đưa ra các *yêu cầu ban đầu* trên cơ sở những gì khách hàng muốn, và sau đó lặp đi lặp lại.
* Xem xét lần lượt từng Use Case:
  + *Ước tính số tiền có sẵn cho tuần đó* là phần rõ ràng của các *yêu cầu ban đầu*.
  + *Đăng ký thế chấp MSG* dường như không liên quan đến dự án thí điểm, vì vậy nó bị loại khỏi các *yêu cầu ban đầu*.
  + *Tính toán số tiền phải trả hàng tuần* và *Quản lý một khoản đầu tư* có vẻ như không liên quan đến dự án thí điểm. Tuy nhiên, dự án thí điểm giải quyết vấn đề “số tiền có sẵn mỗi tuần để mua nhà” bằng nguồn tiền đến từ việc trả nợ hàng tuần và từ các khoản đầu tư nên chúng cũng được coi là một phần của *yêu cầu ban đầu*.
* Biểu đồ Use Case về các *yêu cầu ban đầu* của Case Study MSG Foundation:



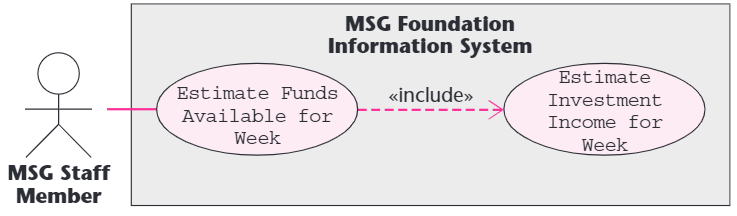
IX. TIẾP TỤC VỚI LUỒNG CÔNG VIỆC XÁC ĐỊNH YÊU CẦU: CASE STUDY MSG FOUNDATION

* Các thành viên của nhóm phát triển biết được rằng MSG Foundation cấp khoản thế chấp 100% để mua nhà với các điều kiện sau:
  + Các cặp đôi mua nhà đã kết hôn ít nhất một năm nhưng không quá mười năm.
  + Cả vợ và chồng đều có việc làm ổn định.
  + Giá của ngôi nhà phải thấp hơn giá trung bình của những ngôi nhà khác trong khu vực đó trong 12 qua.
  + Thu nhập hoặc tiền tiết kiệm của họ không đủ để trả khoản thế chấp 90% lãi xuất cố định tiêu chuẩn trong 30 năm.
  + Tổ chức có đủ tiền để mua nhà.
* Nếu đơn đăng ký được chấp thuận, thì mỗi tuần trong 30 năm tới, hai vợ chồng phải trả cho MSG Foundation:
  + Tổng số tiền cả gốc và lãi không thay đổi trong suốt thời hạn của khoản thế chấp.
  + Các khoản tiền giao kèo, bằng 1/52 tổng thuế bất động sản và phí bảo hiểm hàng năm đối với chủ sở hữu ngôi nhà.
* Nếu số tiền đó vượt quá 28% tổng thu nhập hàng tuần của cặp vợ chồng, MSG Foundation sẽ trả khoản chênh lệch dưới dạng trợ cấp. Khi đó, cặp vợ chồng phải cung cấp bằng chứng về thu nhập hiện tại của họ.
* MSG Foundation sử dụng thuật toán dưới đây để xác định xem nó có đủ tiền để chấp thuận một đơn đăng ký thế chấp hay không:
  1. Đầu mỗi tuần, *thu nhập ước tính hàng năm từ các khoản đầu tư* của MSG Foundation được tính và chia cho 52.
  2. *Chi phí hoạt động ước tính hàng năm* của MSG Foundation được chia cho 52.
  3. *Tổng số tiền thanh toán thế chấp ước tính cho tuần đó* được tính.
  4. *Tổng số tiền trợ cấp ước tính cho tuần đó* được tính.
  5. *Tổng số tiền có sẵn vào đầu tuần sau đó* là: (1) - (2) + (3) - (4).
  6. Nếu chi phí của ngôi nhà không quá số tiền có sẵn để thế chấp, số tiền đó sẽ được giảm bớt.
  7. Vào cuối mỗi tuần, các khoản tiền chưa được sử dụng sẽ được đem đi đầu tư.
* Để giữ cho chi phí của dự án thí điểm càng thấp càng tốt, các nhà phát triển cho rằng chỉ những hạng mục dữ liệu cần thiết cho việc tính tiền hàng tuần mới nên được đưa vào sản phẩm phần mềm.
* Đối với các khoản đầu tư, cần có các dữ liệu:
  + Số mục.
  + Tên mục.
  + Lợi tức ước tính hàng năm.
  + Ngày lợi tức hàng ước tính hàng năm được cập nhật lần cuối.
* Đối với chi phí hoạt động, các dữ liệu sau đây là bắt buộc:
  + Chi phí hoạt động ước tính hàng năm.
  + Ngày chi phí hoạt động ước tính hàng năm được cập nhật lần cuối.
* Đối với mỗi khoản thế chấp, các dữ liệu sau đây là bắt buộc:
  + Số lượng tài khoản.
  + Tên họ của người thế chấp.
  + Giá gốc mua nhà.
  + Ngày thế chấp có hiệu lực.
  + Trả gốc và lãi tuần.
  + Tổng thu nhập hàng tuần hiện tại.
  + Ngày tổng thu nhập hàng tuần được cập nhật lần cuối.
  + Thuế bất động sản hàng năm.
  + Ngày thuế bất động sản hàng năm được cập nhật lần cuối.
  + Phí bảo hiểm hàng năm của chủ sở hữu ngôi nhà.
  + Ngày phí bảo hiểm hàng năm của chủ sở hữu ngôi nhà được cập nhật lần cuối.
* Trong quá trình thảo luận thêm với các nhà quản lý của MSG Foundation, các nhà phát triển nhận thấy rằng cần có ba loại báo cáo:
  + Kết quả của tính toán tiền trong tuần.
  + Một danh sách tất cả các khoản đầu tư.
  + Một danh sách tất cả các khoản thế chấp.

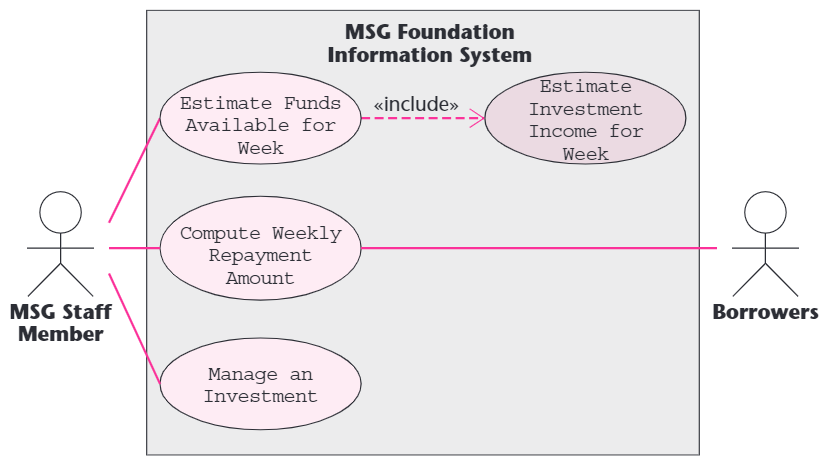
X. SỬA ĐỔI CÁC YÊU CẦU: CASE STYDY MSG FOUNDATION

* Với thông tin bổ sung đã nhận được, các *yêu cầu ban đầu* có thể được sửa đổi.
* Xem xét từng thành phần của công thức xác định số tiền có sẵn đầu tuần:

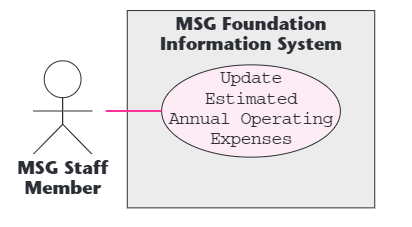
1. *Thu nhập ước tính hàng năm từ các khoản đầu tư*. Đối với mỗi khoản đầu tư, tính tổng lợi tức ước tính hàng năm trên mỗi khoản đầu tư và lấy kết quả chia cho 52. Để thực hiện điều này, cần bổ sung một Use Case *ước tính thu nhập từ các khoản đầu tư cho tuần đó*.
   * Mũi tên nét đứt “include” biểu thị Use Case đó là một phần của Use Case *uớc tính số tiền có sẵn cho tuần đó*.



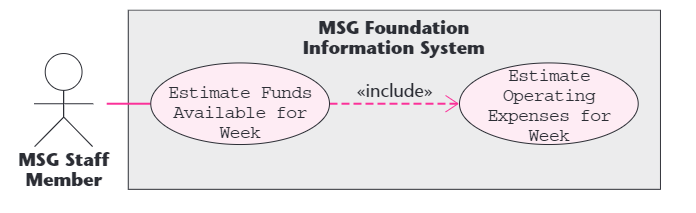
* Lần lặp đầu tiên của biểu đồ Use Case về *các yêu cầu sửa đổi* của Case Study MSG Foundation. Use Case mới được tô đậm.



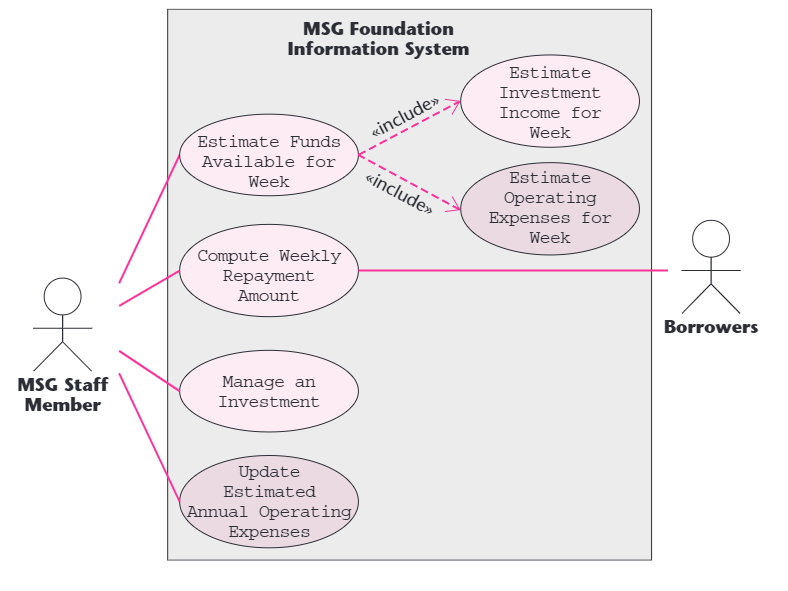
1. *Chi phí hoạt động ước tính hàng năm*. Để xác định chi phí này cần bổ sung hai Use Case:
   * Use Case *cập nhật chi phí hoạt động ước tính hàng năm* mô hình hóa sự điều chỉnh giá trị chi phí hoạt động ước tính hàng năm.



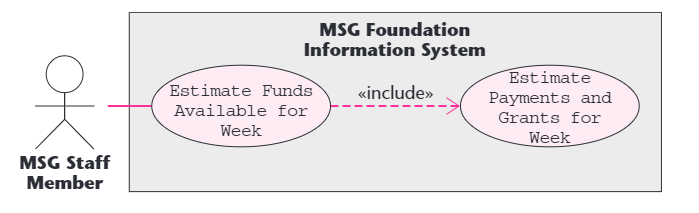
* + Use Case *ước tính chi phí hoạt động cho tuần đó* cung cấp chi phí hoạt động ước tính được yêu cầu.



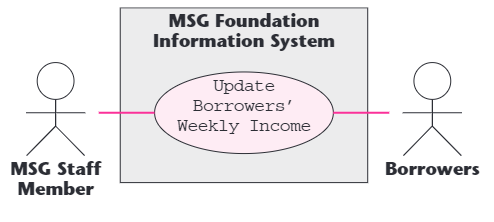
* + Lần lặp thứ hai của biểu đồ Use Case về *các yêu cầu sửa đổi* của Case Study MSG Foundation. Hai Use Case mới được tô đậm.



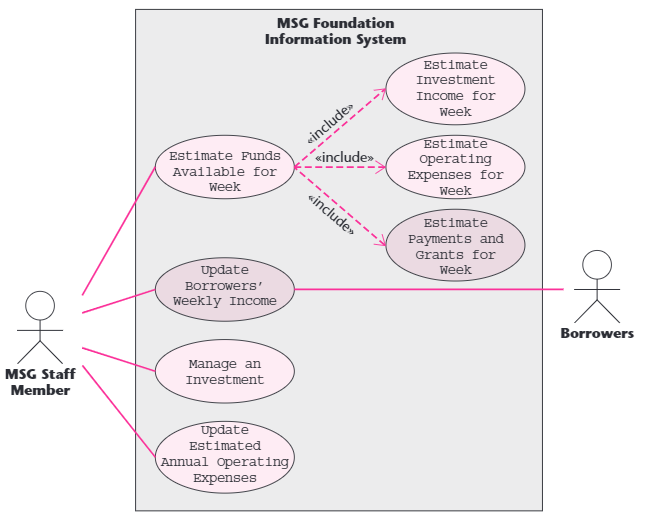
1. *Tổng số tiền thanh toán thế chấp ước tính cho tuần đó*.
2. *Tổng số tiền trợ cấp ước tính cho tuần đó*.
   * Use Case *tính tổng số tiền phải trả cho tuần đó* mô hình hóa sự tính toán của *số tiền thanh toán thế chấp ước tính* và *số tiền trợ cấp ước tính cho mỗi khoản thế chấp*.
   * Tổng hợp các số lượng riêng biệt này sẽ tạo ra *tổng số tiền thanh toán thế chấp ước tính* cũng như *tổng số tiền trợ cấp ước tính cho tuần đó*.
   * *Tính tổng số tiền phải trả hàng tuần* cũng mô hình hóa việc cập nhật thu nhập hàng tuần của người vay.
   * *Tính tổng số tiền phải trả hàng tuần* nên được chia thành hai Use Case:
     + *Uớc tính số tiền phải trả và trợ cấp cho tuần đó*.
     + *Cập nhật thu nhập người vay hàng tuần*.
   * Use Case *uớc tính số tiền phải trả và trợ cấp cho tuần đó* về *các yêu cầu sửa đổi* của Case Study MSG Foundation:



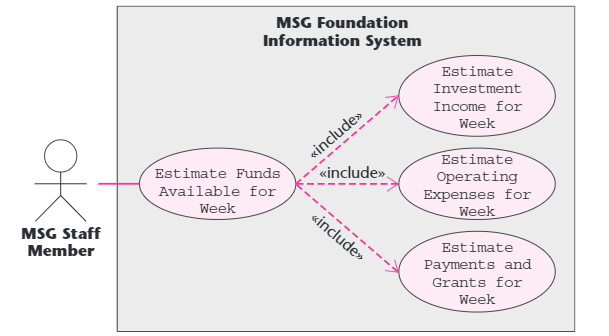
* Use Case *cập nhật thu nhập người vay hàng tuần* về *các yêu cầu sửa đổi* của Case Study MSG Foundation:



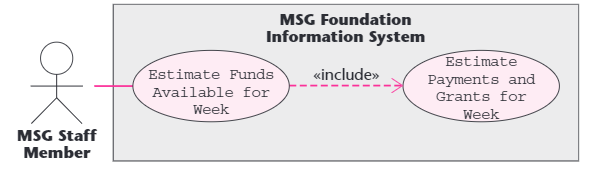
* Lần lặp thứ ba của biểu đồ Use Case về *các yêu cầu sửa đổi* của Case Study MSG Foundation. Hai Use Case mới xuất phát từ Use Case *tính tổng số tiền phải trả mỗi tuần* được tô đậm:



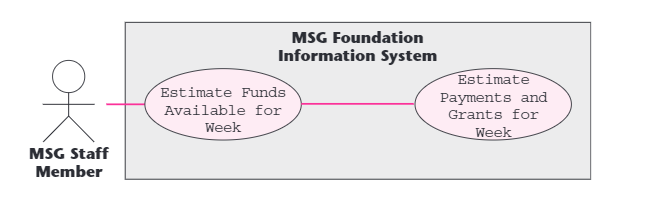
* Lần lặp thứ hai của Use Case *Ước tính số tiền có sẵn cho tuần đó* về *các yêu cầu sửa đổi* của Case Study MSG Foundation.



* Đối với mối quan hệ “include” trong biểu đồ UML
  + Use Case đúng

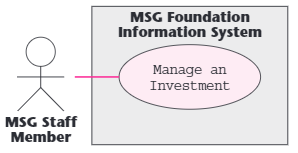
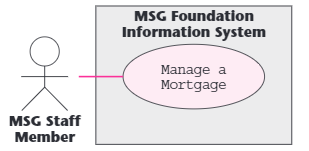


* + Use Case sai

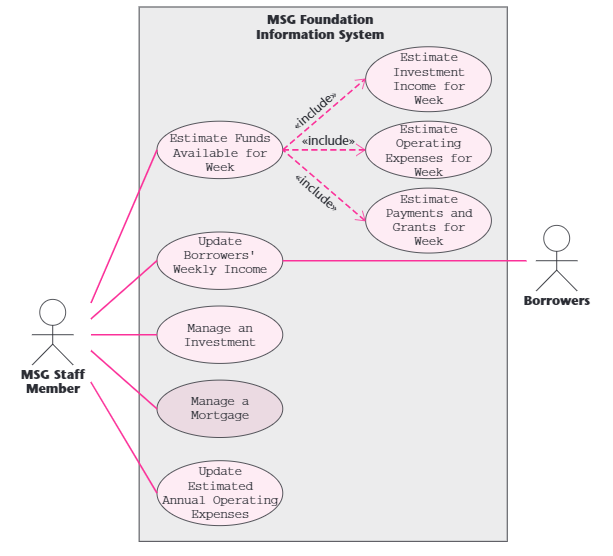


XI. LUỒNG CÔNG VIỆC KIỂM THỬ: CASE STUDY MSG FOUNDATION

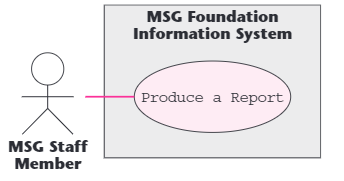
* Một kết quả phổ biến của mô hình vòng đời lặp đi lại và tăng dần là các chi tiết bị hoãn lại bằng cách nào đó bị bỏ quên. Vì vậy việc kiểm thử là cần thiết.
* Use Case *quản lý một khoản đầu tư* và Use Case *quản lý một khoản thế chấp* đã bị bỏ qua.

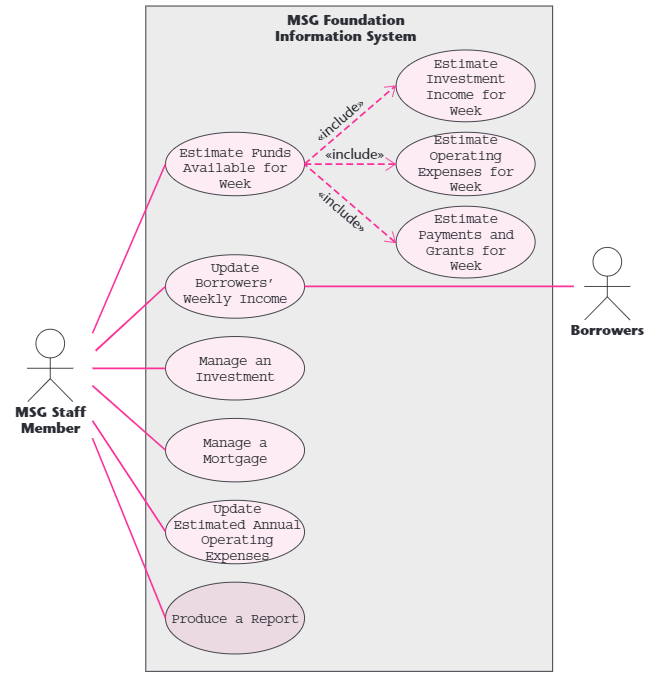
* Lần lặp thứ tư của biểu đồ Use Case về *các yêu cầu sửa đổi* của Case Study MSG Foundation. Use Case mới, *quản lý một khoản thế chấp* được tô đậm.



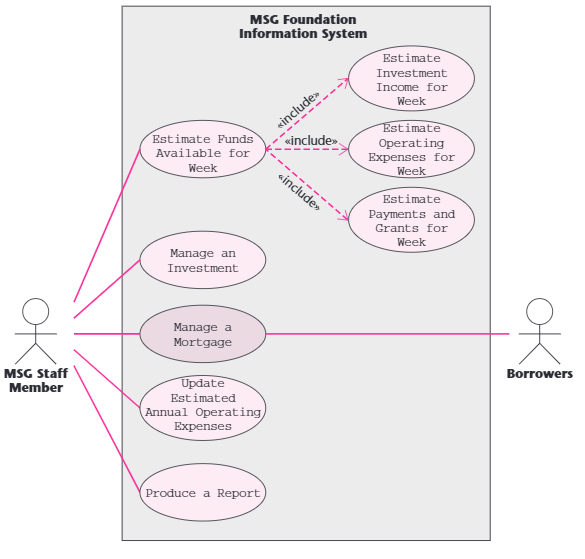
* Use Case *lập báo cáo* dùng để mô hình hóa việc in báo cáo cũng bị bỏ qua.



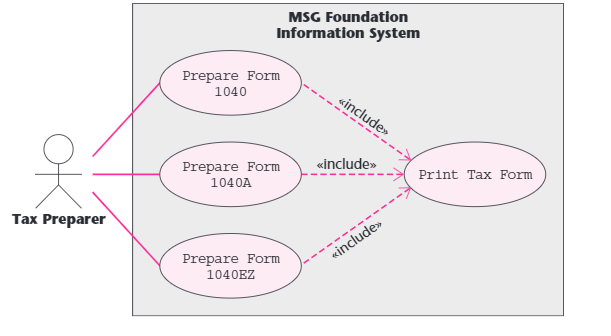
* + Lần lặp thứ năm của biểu đồ Use Case về *các yêu cầu sửa đổi* của Case Study MSG Foundation. Use Case mới, *lập báo cáo* chấp được tô đậm.



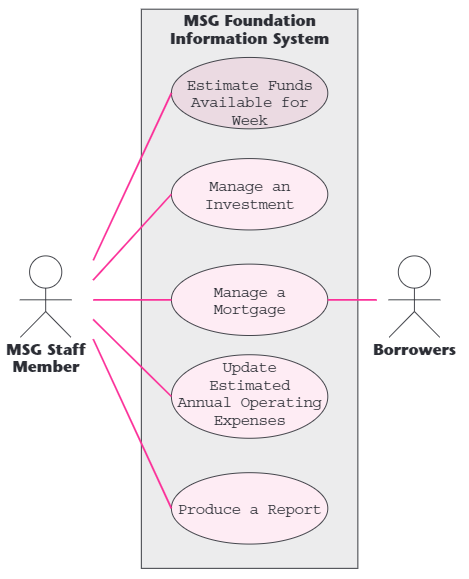
* *Các yêu cầu sửa đổi* được kiểm tra lại và hai vấn đề mới được phát hiện:
  + Đầu tiên, một Use Case đã được sao chép một phần.
  + Thứ hai, hai trong số các Use Case cần được tổ chức lại.
* Use Case *quản lý một khoản thế chấp*: một hành động là sửa đổi một khoản thế chấp.
* Use Case *cập nhật thu nhập của người vay hàng tuần*: chỉ có hành động là cập nhật thu nhập của người vay hàng tuần.
* *Thu nhập của người vay hàng tuần* là một thuộc tính của khoản thế chấp.
* Use Case *quản lý một khoản thế chấp* đã bao gồm Use Case *cập nhật thu nhập người vay hàng tuần*. Theo đó, Use Case *cập nhật thu nhập người vay hàng tuần* không cần thiết và bị xóa.
* Lần lặp thứ sáu của biểu đồ Use Case về *các yêu cầu sửa đổi* của Case Study MSG Foundation. Use Case được sửa đổi, *quản lý một khoản thế chấp* được tô đậm.



* Sự lặp lại này dẫn đến giảm hơn là tăng.
* Thực tế, việc xóa diễn ra thường xuyên, bất cứ khi nào có sai sót xảy ra.
* Điểm mấu chốt là khi phát hiện ra lỗi thì không phải bắt đầu toàn bộ *quá trình yêu cầu* lại từ đầu. Thay vào đó là sửa lỗi lặp hiện tại.
* Nếu sai lầm quá nghiêm trọng, phải quay lại lần lặp trước đó và tìm cách tốt hơn để tiếp tục từ đó.
* Xác định số tiền có sẵn cho tuần hiện tại:
  + Use Case *ước tính số tiền có sẵn cho tuần đó* mô hình hóa việc thực hiện tính toán.
  + Use Case *lập báo cáo* mô hình hóa việc in kết quả của việc tính toán.
* Không có ích gì khi ước tính số tiền hiện có trừ khi kết quả được in ra.
* Những mô tả các Use Case *ước tính số tiền có sẫn cho tuần đó* và *lập hóa đơn* phải được sửa đổi.
* Lý do thông thường cho mối quan hệ “include” là một Use Case là một phần của hai hay nhiều Use Case khác
  + Ví dụ: biểu mẫu thuế Hoa Kỳ tránh việc gấp ba:



* Đối với Case Study MSG Foundation, tất cả các Use Case liên quan chỉ là một phần của Use Case *ước tính số tiền có sẵn cho tuần đó*.
* Kết hợp ba Use Case vào Use Case *ước tính số tiền có sẵn cho tuần đó* sẽ được biểu đồ Use Case như sau:



XII. PHA XÁC ĐỊNH YÊU CẦU CỔ ĐIỂN

* Không có cái gọi là “xác định yêu cầu hướng đối tượng*”. Luồng công việc xác định yêu cầu* không liên quan đến cách sản phẩm sẽ được xây dựng.
* Cách tiếp cận trong chương này là hướng mô hình, và bản chất của nó là hướng đối tượng.
* Phương pháp cổ điển để xác định yêu cầu:
  + Làm rõ yêu cầu.
  + Phân tích yêu cầu.
  + Xây dựng bản mẫu nhanh.
  + Khách hàng và người dùng trong tương lai thử nghiệm với bản mẫu nhanh.

XIII. BẢN MẪU NHANH

* *Bản mẫu nhanh* là phần mềm được xây dựng nhanh thể hiện các chức năng chính của sản phẩm đích, giúp phản ánh chức năng mà khách hàng thấy như màn hình đầu vào hay in báo cáo, nhưng bỏ qua các khía cạnh “ẩn” như kiểm tra lỗi hay cập nhật tệp.
* Người dùng sẽ cho biết *bản mẫu nhanh* đã đáp ứng được nhu cầu của họ hay chưa và những phần cần được cải thiện. Từ đó, các nhà phát triển sẽ thay đổi đến khi mọi nhu cầu của khách hàng đều nằm trong *bản mẫu nhanh*.
* Mục đích của *bản mẫu nhanh* là cung cấp cho khách hàng sự hiểu biết về sản phẩm phần mềm càng sớm càng tốt.
* *Bản mẫu nhanh* được xây dựng để có thể thay đổi nhanh chóng nhằm đáp ứng tốt hơn nhu cầu của khách hàng. Ngôn ngữ cho *bản mẫu nhanh* bao gồm các ngôn ngữ thế hệ thứ tư và các ngôn ngữ được thông dịch.

XIV. NHÂN TỐ CON NGƯỜI

* Cả khách hàng và người dùng tương lai của sản phẩm phải tương tác với *bản mẫu nhanh* của *giao diện người dùng*.
* Việc người dùng thử nghiệm với *giao diện người – máy tính* làm giảm đáng kể nguy cơ sản phẩm hoàn thành bị thay đổi. Đặc biệt còn giúp tăng *sự thân thiện với người dùng*-mục tiêu quan trọng đối với các sản phẩm phần mềm.
* Giao diện người - máy tính
  + Dạng bảng chọn, không dòng lệnh.
  + Thao tác trỏ và nhấp chuột.
  + Giao diện cửa sổ, biểu tượng, bảng chọn sổ xuống.
* Ngay cả khi sản phẩm đích sử dụng công nghệ hiện đại, yếu tố con người vẫn phải được xem xét:
  + Tránh các bảng đơn dài dòng.
  + Cho phép mức thay đổi thành thạo của giao diện.
  + Tính đồng đều của hình thức là quan trọng.
  + Tâm lý tiến bộ so với cảm giác chung.
* Bản mẫu nhanh của giao diện người-máy của mỗi sản phẩm phần mềm là bắt buộc.

XV. SỬ DỤNG LẠI BẢN MẪU NHANH

* Việc sử dụng lại *bản mẫu nhanh* giúp việc phát triển phần mềm nhanh hơn, thay vì loại bỏ, *bản mẫu nhanh* được chuyển đổi thành sản phẩm cuối cùng. Tuy nhiên cũng gặp phải vấn đề:
  + Trong quá trình tinh chỉnh *bản mẫu nhanh*, các thay đổi được thực hiện đối với sản phẩm đang hoạt động nên rất tốn kém.
  + *Bản mẫu nhanh* được làm một cách vội vã. Trong trường hợp thiếu tài liệu đặc tả và thiết kế thì code sẽ khó bảo trì và tốn kém.
* Một lý do để loại bỏ *bản mẫu nhanh* là vấn đề hiệu suất, đặc biệt là các ràng buộc về hệ thống thời gian thực.
* Một cách để đảm bảo rằng *bản mẫu nhanh* được bỏ qua là cài đặt *bản mẫu nhanh* bằng một ngôn ngữ khác với ngôn ngữ đã lựa chọn cho sản phẩm đích.
* Mã được sinh ra có thể sử dụng lại.
* Chúng ta có thể giữ lại *bản mẫu nhanh* một cách an toàn:
  + Phải quyết định trước những phần trong *bản mẫu nhanh* có thể được sử dụng trong sản phẩm cuối cùng.
  + Các phần của *bản mẫu nhanh* đã qua kiểm tra kỹ lưỡng của nhóm SQA.
  + Các tài liệu thiết kế không phải là một phần của *bản mẫu nhanh* cổ điển.

XVI. CÁC CÔNG CỤ CASE CHO XÁC ĐỊNH YÊU CẦU

* Các công cụ đồ họa có tầm ảnh hưởng quan trọng trong việc hỗ trợ *luồng công việc xác định yêu cầu*:
  + Dễ dàng thay đổi các biểu đồ UML.
  + Tài liệu được lưu trong các công cụ và luôn có sẵn.
* Một số điểm yếu của các công cụ CASE:
  + Các công cụ không thực sự thân thiện với người dùng.
  + Biểu đồ UML vẽ bằng các công cụ không có tính thẩm mỹ cao như khi vẽ bằng tay.
  + Giá thành của các công cụ còn đắt.
* Nhìn chung, hai điểm mạnh của công cụ CASE ở trên vượt trội hơn những điểm yếu này.
* Các môi trường CASE đồ họa được mở rộng để trợ giúp UML:
  + System Architect
  + Software through Pictures
* Các môi trường CASE hướng đối tượng:
  + IBM Rational Rose
  + Together
  + ArgoUML(open source)

XVII. CÁC THƯỚC ĐO CHO XÁC ĐỊNH YÊU CẦU

* Sự biến động và tốc độ tập trung là những thước đo về mức độ nhanh chóng mà nhu cầu khách hàng được xác định.
* Số lượng thay đổi được đưa ra trong suốt chuỗi con các pha.
* Những thay đổi được đề xướng bởi những người phát triển:
  + Quá nhiều thay đổi có thể đồng nghĩa với việc quy trình không hoàn thiện.
  + Những thay đổi được đề xướng bởi khách hàng.
  + Thay đổi sản phẩm phần mềm cuối cùng.

XVIII. NHỮNG THỬ THÁCH CHO PHA XÁC ĐỊNH YÊU CẦU

* Nhân viên của tổ chức khách hàng thường cảm giác bị đe dọa bởi máy tính.
* Những thành viên đội xác định yêu cầu phải có khả năng thương lượng. Phải biết giảm quy mô những gì khách hàng muốn.
* Nhân viên chính của tổ chức khách hàng không thể có thời gian cho những cuộc thảo luận cốt yếu và sâu sắc.
* Tính linh hoạt và khách quan là điều cần thiết cho việc kích thích yêu cầu.

XIX. TỔNG KẾT

Sau khi tìm hiểu về chương Requirements chúng ta có thể:

* Thực hiện luồng công việc xác định yêu cầu.
* Tạo ra được mô hình nghiệp vụ ban đầu.
* Tạo ra được các yêu cầu.
* Tạo ra được bản mẫu nhanh.

XX. TÀI LIỆU THAM KHẢO

1. Stephen R. Schach, Object-Oriented and Classical Software Engineering, Eighth Edition, U.S: McGraw Hill, 2011.
2. Trần Đình Quế, Nguyễn Mạnh Hùng, Giáo trình Nhập Môn Công Nghệ Phần Mềm, Hà Nội: Học Viện Công Nhệ Bưu Chính Viễn Thông, 2016.