



TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI
HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

NHẬP MÔN CNPM

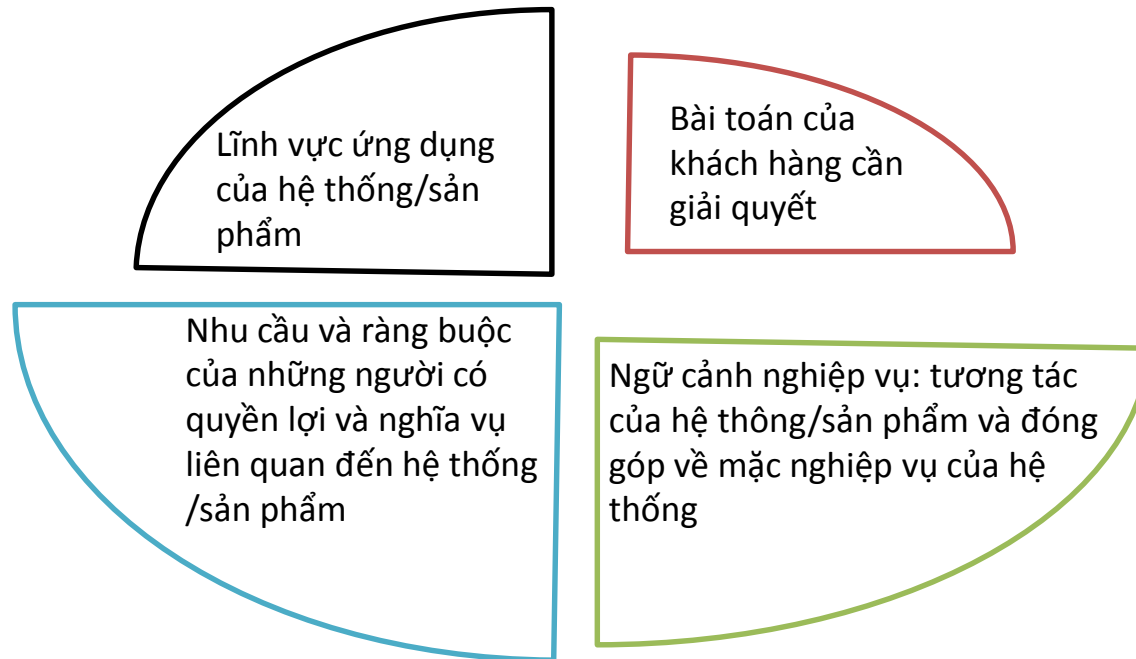
Nội dung / Chương 5: Kỹ nghệ yêu cầu phần mềm

Thông tin GV

Khái niệm

- **Yêu cầu:**

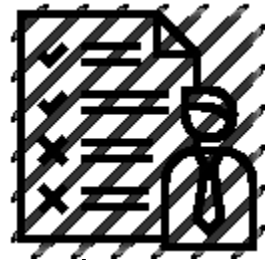
- Các đặc tính của hệ thống hay sản phẩm do khách hàng - người sử dụng phần mềm - nêu ra
- → Xác định được phần mềm đáp ứng được các yêu cầu và mong muốn của khách hàng - người sử dụng phần mềm



Khái niệm

- Tại sao cần phải đặt ra yêu cầu phần mềm ?

- Khách hàng chỉ có những ý tưởng còn mơ hồ về phần mềm cần phải xây dựng để phục vụ công việc của họ, chúng ta phải sẵn sàng, kiên trì theo đuổi để đi từ các ý tưởng mơ hồ đó đến “Phần mềm có đầy đủ các tính năng cần thiết”

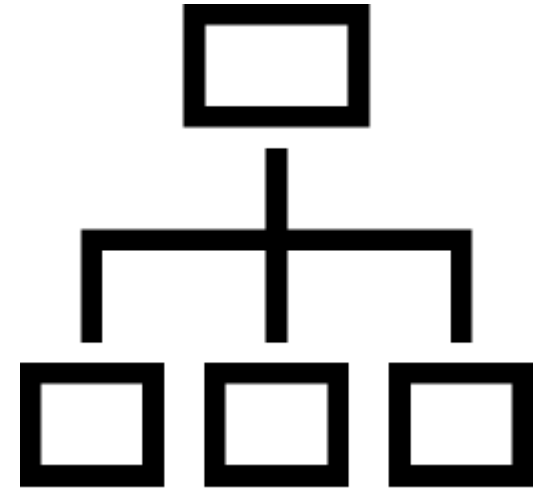


- Khách hàng rất hay thay đổi các đòi hỏi của mình, chúng ta nắm bắt được các thay đổi đó và sửa đổi các mô tả một cách hợp lý



Phân loại

- Theo 4 thành phần của phần mềm:
 - Các yêu cầu về phần mềm (Software)
 - Các yêu cầu về phần cứng (Hardware)
 - Các yêu cầu về dữ liệu (Data)
 - Các yêu cầu về con người (People, Users)
- Theo cách đặc tả phần mềm
 - Các yêu cầu chức năng
 - Các yêu cầu ngoài chức năng
 - Các ràng buộc khác

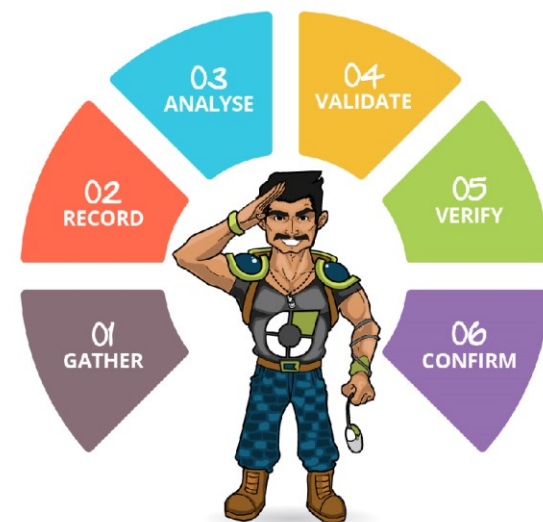


Kĩ thuật khảo sát Yêu cầu - I

- **Khởi đầu (Inception):** Hỏi một loạt các câu hỏi để xác định:
 - Hiểu biết căn bản về vấn đề cần giải quyết.
 - Người đang cần giải pháp
 - Loại giải pháp mong muốn
 - Mức độ hiệu quả ban đầu của việc trao đổi thông tin giữa khách hàng và nhà phát triển
- **Khám phá (Elicitation):** tìm ra yêu cầu của tất cả khách hàng.
- **Xây dựng (Elaboration):** tạo ra mô hình phân tích xác định dữ liệu, chức năng và hành vi được yêu cầu.
- **Đàm phán (Negotiation):** đồng ý với một hệ thống có thể bàn giao một cách thực tế đối với cả 2 bên.

Kĩ thuật khảo sát Yêu cầu - II

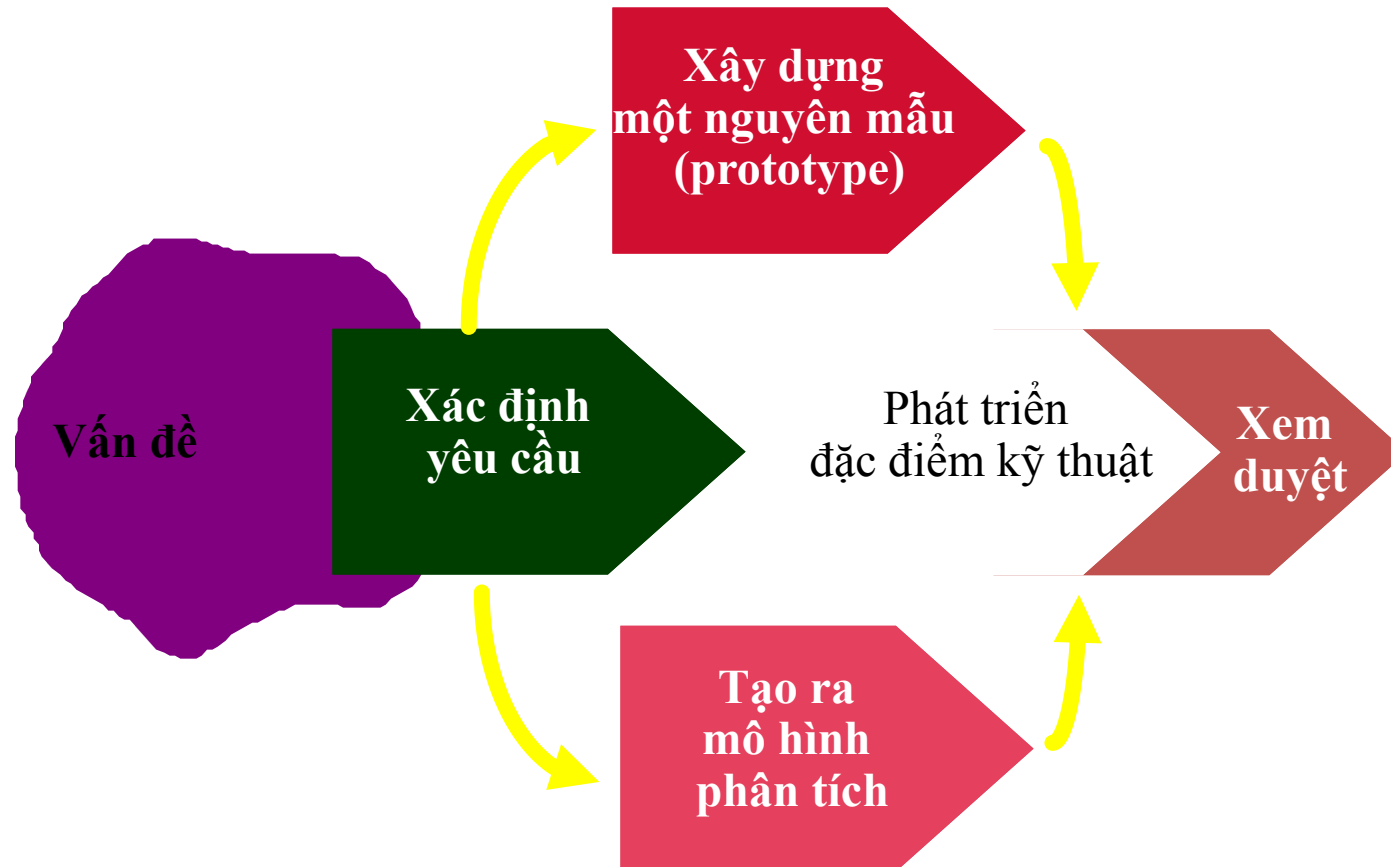
- **Đặc tả (Specification):** có thể là một/ nhiều những thứ sau:
 - Một Tài liệu được viết
 - Một Tập hợp các mô hình
 - Một hình thức biểu diễn toán học
 - Một tập các kịch bản người dùng (use-case)
 - Một Nguyên mẫu
- **Đánh giá (Validation):** tạo cơ chế xem xét các vấn đề:
 - Sai sót trong nội dung hoặc giải thích.
 - Phần được yêu cầu làm rõ.
 - Thông tin bị thiếu
 - Mâu thuẫn
 - Yêu cầu không thực tế, không thể đạt được.
- **Quản lý các Yêu cầu (Requirements management)**



Quy trình xác định yêu cầu PM

- Phát hiện các yêu cầu phần mềm (Requirements elicitation)
- Phân tích các yêu cầu phần mềm và thương lượng với khách hàng (Requirements analysis and negotiation)
- Đặc tả các yêu cầu phần mềm (Requirements specification)
- Mô hình hóa hệ thống (System modeling)
- Kiểm tra tính hợp lý của các yêu cầu phần mềm (Requirements validation)
- Quản trị các yêu cầu phần mềm (Requirements management)

Quy trình xác định yêu cầu PM



Khởi đầu:

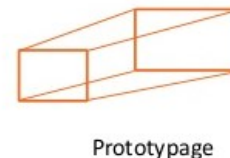
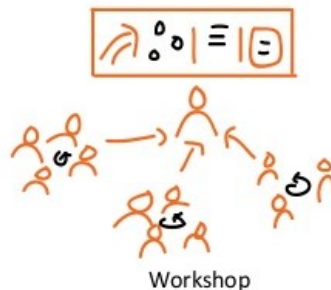
- Xác định các bên liên quan
 - “Ai là người Ngài nghĩ tôi cần phải gặp?”
- Tìm ra các quan điểm khác nhau
- Làm việc hướng tới sự hợp tác
- Câu hỏi đầu tiên:
 - Ai đứng sau những yêu cầu này?
 - Ai sẽ sử dụng các giải pháp?
 - Lợi ích kinh tế của một giải pháp
 - Có giải pháp nào khác hay không?

Phát hiện yêu cầu phần mềm

- Đánh giá tính khả thi về kỹ thuật và nghiệp vụ của phần mềm định phát triển
- Tìm kiếm các nhân sự (chuyên gia, người sử dụng) có những hiểu biết sâu sắc nhất, chi tiết nhất về hệ thống giúp chúng ta xác định yêu cầu phần mềm
- Xác định môi trường kỹ thuật trong đó sẽ triển khai phần mềm
- Xác định các ràng buộc về lĩnh vực ứng dụng của phần mềm (giới hạn về chức năng/hiệu năng phần mềm)

Phát hiện yêu cầu phần mềm

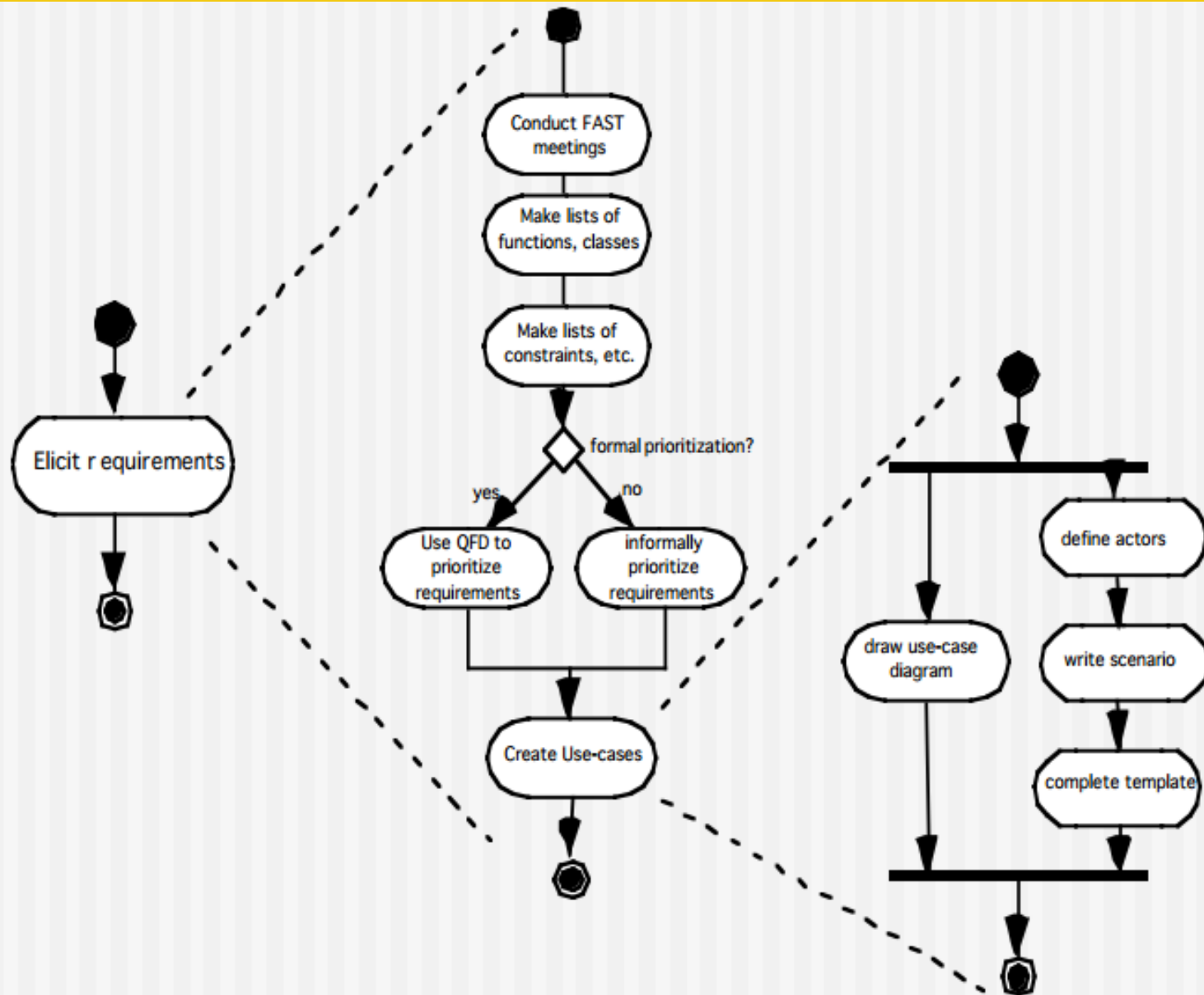
- Xác định các phương pháp sử dụng để phát hiện các yêu cầu phần mềm: phỏng vấn, làm việc nhóm, các buổi họp, gặp gỡ đối tác, v.v.
- Thu hút sự tham gia của nhiều chuyên gia, khách hàng để chúng ta có được các quan điểm xem xét phần mềm khác nhau từ phía khách hàng
- Xác định các yêu cầu còn nhập nhằng để làm mẫu thử
- Thiết kế các kịch bản sử dụng của phần mềm để giúp khách hàng định rõ các yêu cầu chính.



Khám phá các yêu cầu:

- Cuộc họp được tiến hành với sự tham gia của cả kỹ sư phần mềm và khách hàng.
- Quy tắc chuẩn bị và tham gia được xác lập
- Các vấn đề cần trao đổi được gợi ý
- Một Facilitator (khách hàng, nhà phát triển, hoặc người trung gian) sẽ điều khiển cuộc họp
- Một “Cơ chế xác định” (vd work sheet, biểu đồ, bảng điện tử, chat room) được sử dụng.
- Mục đích là:
 - Xác định vấn đề
 - Đề xuất các thành phần của giải pháp
 - Thảo luận giữa các cách tiếp cận khác nhau
 - Xác định một tập sơ bộ các yêu cầu giải pháp

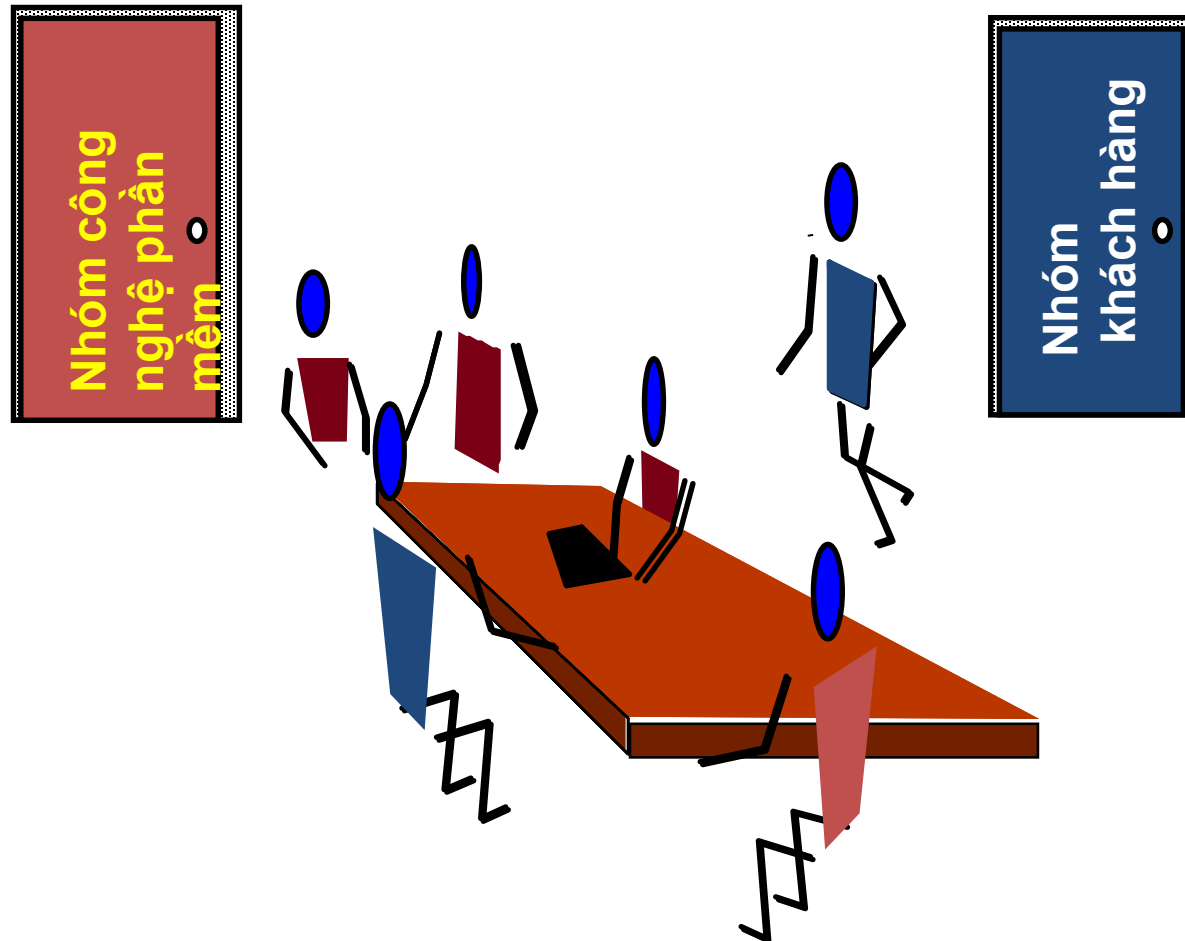
Khám phá yêu cầu:



Khám phá các sản phẩm

- Đầu ra của bước phát hiện yêu cầu phần mềm
 - Bảng kê các Yêu cầu và mức độ khả thi
 - Bảng kê Giới hạn phạm vi của hệ thống
 - Danh sách khách hàng, người sử dụng, các bên liên quan trong việc đưa ra yêu cầu
 - Mô tả về môi trường kỹ thuật của hệ thống
 - Danh sách các yêu cầu (sắp xếp theo chức năng)
 - Tập các kịch bản sử dụng để cung cấp một cách nhìn về vận hành hệ thống/ sản phẩm trong những điều kiện khác nhau
 - Nguyên mẫu được phát triển để xác định yêu cầu tốt hơn.

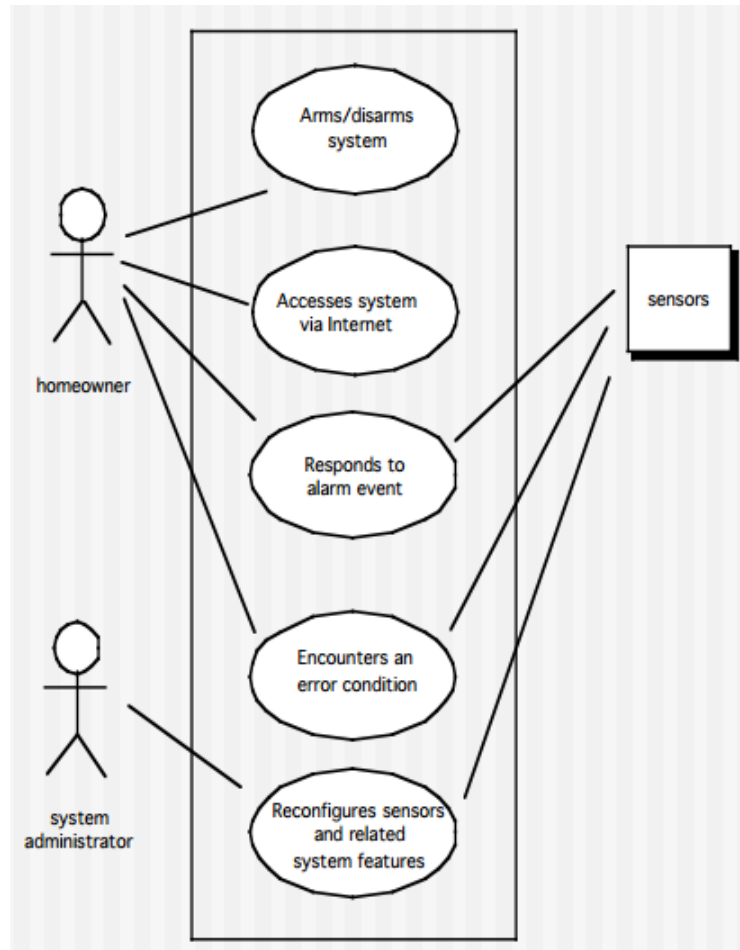
Phân tích các yêu cầu phần mềm và thương lượng với khách hàng



Xây dựng mô hình phân tích

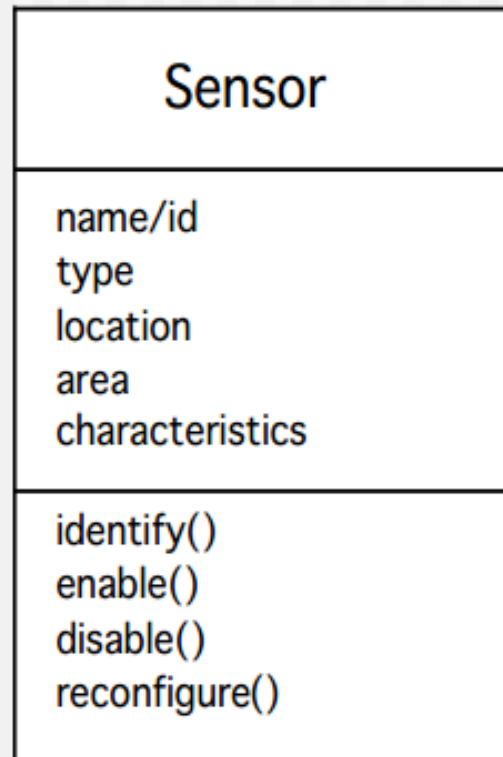
- Thành phần của mô hình phân tích
 - Các yếu tố dựa trên kịch bản sử dụng
 - Tường thuật quy trình – chức năng cho các chức năng phần mềm
 - Use-case mô tả sự tương tác giữa người dùng và hệ thống
 - Các lớp cơ sở
 - Dựa trên kịch bản sử dụng
 - Các yếu tố hành vi
 - Sơ đồ trạng thái
 - Thành phần luồng định hướng
 - Biểu đồ luồng dữ liệu

Biểu đồ use-case

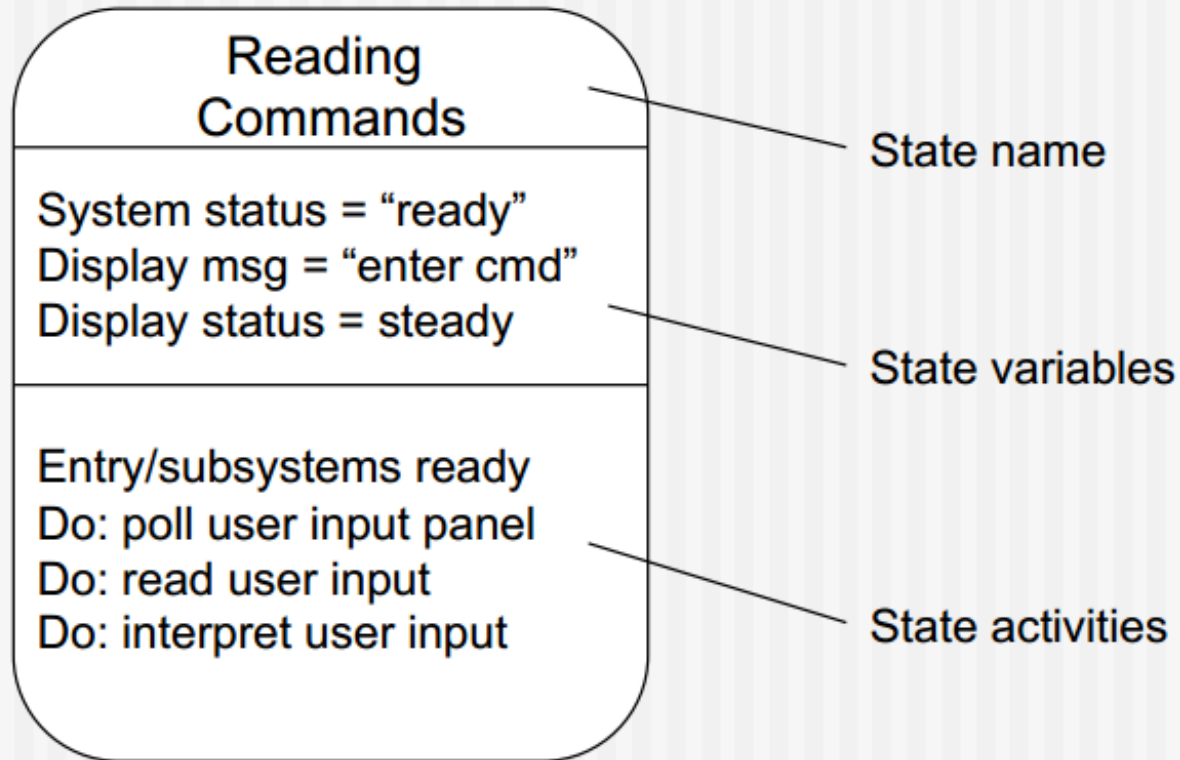


Biểu đồ lớp

From the *SafeHome* system ...



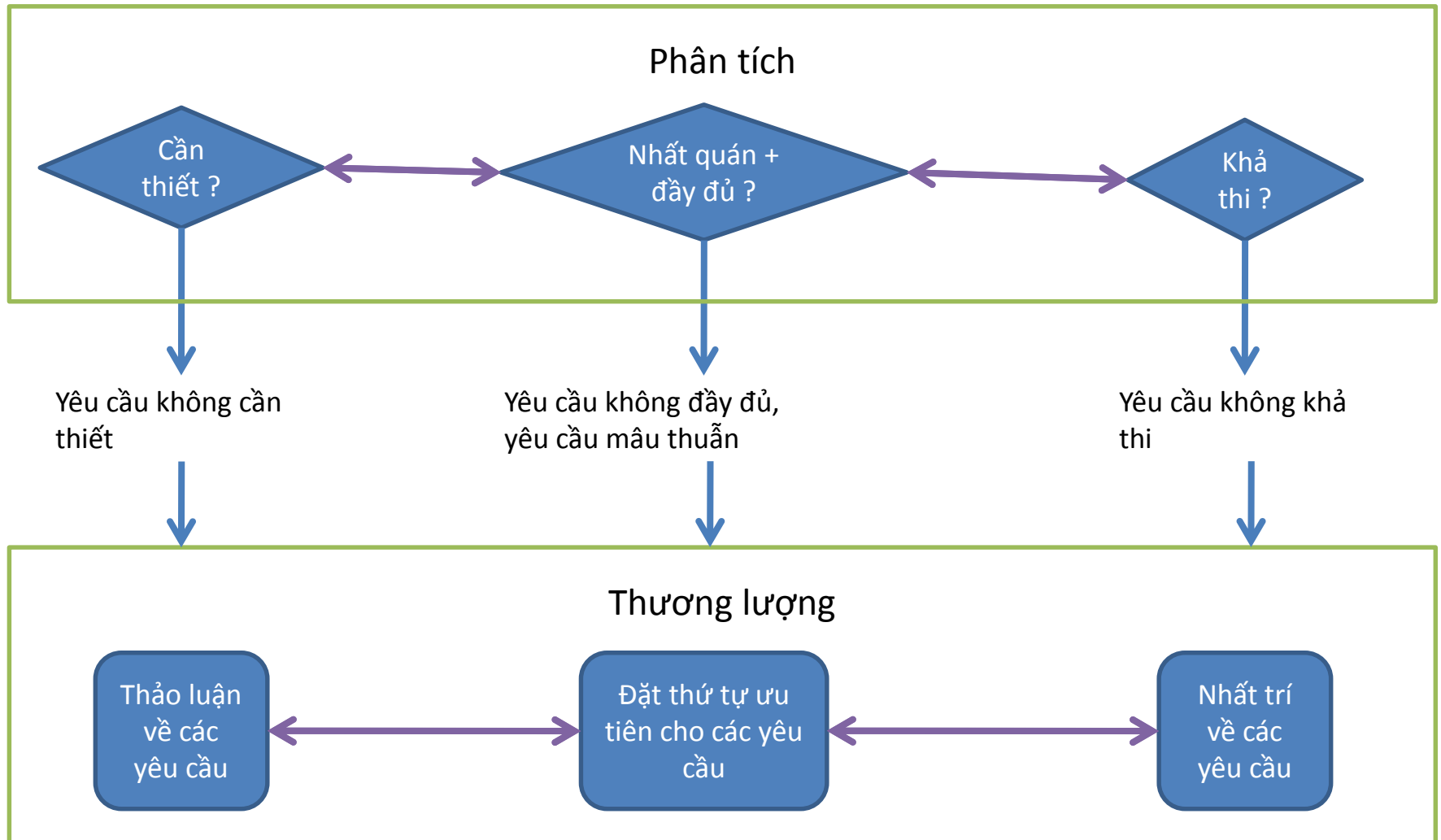
Biểu đồ trạng thái



Mẫu phân tích

- **Tên mẫu:** tên mô tả bản chất của mẫu.
- **Intent:** mô tả những gì mẫu hoàn thành hoặc đại diện
- **Bối cảnh:** mô tả những điều kiện xảy ra có thể ảnh hưởng đến việc sử dụng mẫu và những vấn đề có thể được giải quyết khi sử dụng mẫu.
- **Giải pháp:** mô tả về cách sử dụng mẫu để giải quyết vấn đề.
- **Hậu quả:** kết quả khi áp dụng mô hình và sự thỏa hiệp tồn tại trong ứng dụng.
- **Thiết kế:** thảo luận về kết quả các mẫu thiết kế có thể đạt được thông qua sử dụng các mẫu thiết kế đã biết
- **Công dụng:** ví dụ về sử dụng hệ thống trong thực tế.

Phân tích các yêu cầu phần mềm và thương lượng với khách hàng



Thương lượng yêu cầu

- **Xác định đâu là bên liên quan chính:** đây là những người sẽ tham gia thương lượng.
- **Xác định “win condition” của mỗi bên liên quan:** “win condition” của các bên liên quan thường không rõ ràng.
- **Thương lượng:** làm việc dẫn tới thành công cho các bên tham gia.



Phân tích các yêu cầu phần mềm và thương lượng với khách hàng

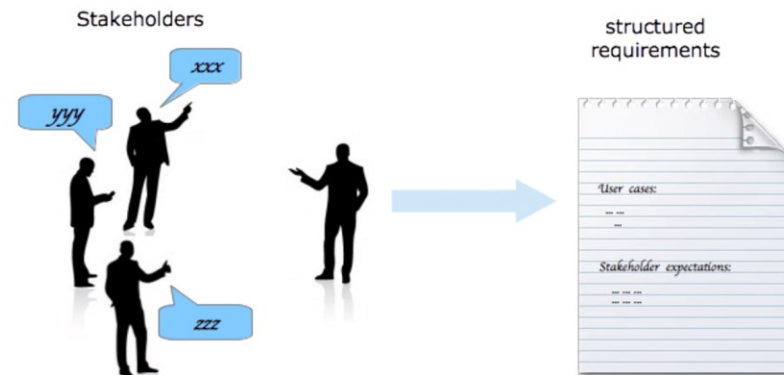
- Phân loại các yêu cầu phần mềm và sắp xếp chúng theo các nhóm liên quan
- Khảo sát tỉ mỉ từng yêu cầu phần mềm trong mối quan hệ của nó với các yêu cầu phần mềm khác
- Thẩm định từng yêu cầu phần mềm theo các tính chất: phù hợp, đầy đủ, rõ ràng, không trùng lặp
- Phân cấp các yêu cầu phần mềm theo dựa trên nhu cầu và đòi hỏi khách hàng / người sử dụng
- Thẩm định từng yêu cầu phần mềm để xác định:
 - Các yêu cầu PM có khả năng thực hiện được trong môi trường kỹ thuật hay không
 - Có khả năng kiểm định các yêu cầu phần mềm hay không

Phân tích các yêu cầu phần mềm và thương lượng với khách hàng

- Thẩm định các rủi ro có thể xảy ra với từng yêu cầu phần mềm
- Đánh giá thô (tương đối) về giá thành và thời gian thực hiện của từng yêu cầu phần mềm trong giá thành sản phẩm phần mềm và thời gian thực hiện phần mềm
- Giải quyết tất cả các bất đồng về yêu cầu phần mềm với khách hàng / người sử dụng trên cơ sở thảo luận và thương lượng các yêu cầu đề ra

Đặc tả yêu cầu phần mềm

- Đặc tả các yêu cầu phần mềm: xây dựng các tài liệu đặc tả, trong đó có thể sử dụng tới các công cụ như: mô hình hóa, mô hình toán học hình thức (a formal mathematical model), tập hợp các kịch bản sử dụng, các nguyên mẫu hoặc bất kỳ một tổ hợp các công cụ nói trên
- Phương pháp đặc tả:
 - Đặc tả phi hình thức (Informal specifications): viết bằng ngôn ngữ tự nhiên
 - Đặc tả hình thức (Formal specifications): viết bằng tập các ký pháp có các quy định về cú pháp (syntax) và ngữ nghĩa (semantic) rất chặt chẽ, thí dụ ký pháp đồ họa dùng các lưu đồ.
- Tiêu chí đánh giá chất lượng của hồ sơ đặc tả:
 - Tính rõ ràng, chính xác
 - Tính phù hợp
 - Tính đầy đủ, hoàn thiện



Các thành phần của hồ sơ đặc tả

- Đặc tả vận hành hay đặc tả chức năng (Operational specifications): mô tả các hoạt động của hệ thống phần mềm sẽ xây dựng:
 - Các dịch vụ mà hệ thống phải cung cấp
 - Hệ thống sẽ phản ứng với đầu vào cụ thể ra sao
 - Hành vi của hệ thống trong các tình huống đặc biệt.
- Đặc tả mô tả hay đặc tả phi chức năng (Descriptive specifications): đặc tả các đặc tính, đặc trưng của phần mềm:
 - Các ràng buộc về các dịch vụ hay các chức năng hệ thống cung cấp như thời gian, ràng buộc về các quá trình phát triển, các chuẩn,...
- Ngoài ra còn có yêu cầu về lĩnh vực, bắt nguồn từ lĩnh vực của ứng dụng hệ thống và các đặc trưng của lĩnh vực này.

Đặc tả chức năng

- Miêu tả các chức năng của hệ thống, phụ thuộc vào kiểu phần mềm và mong đợi của người dùng
 - Tương tác giữa phần mềm và môi trường, độc lập với việc cài đặt
 - Ví dụ: Hệ thống đồng hồ phải hiển thị thời gian dựa trên vị trí của nó
- Các công cụ đặc tả tiêu biểu:
 - Biểu đồ luồng dữ liệu (Data Flow Diagrams)
 - Máy trạng thái hữu hạn (Finite State Machines)
 - Mạng Petri (Petri nets),...
 - Tuy nhiên không bắt buộc và có thể dùng ngôn ngữ tự nhiên.

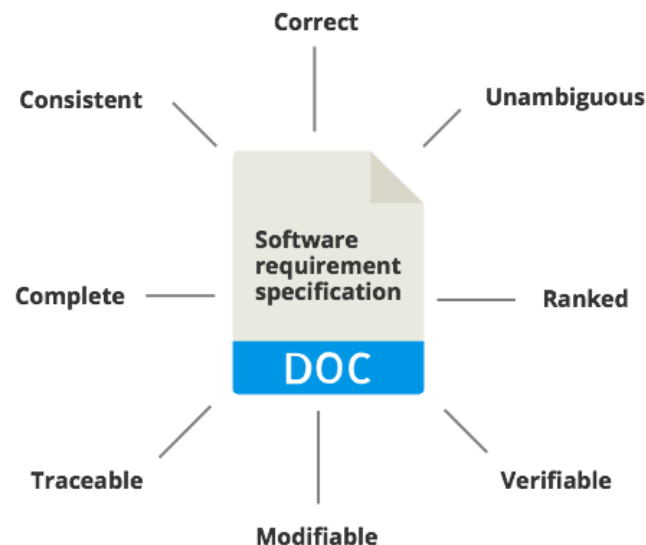
Đặc tả phi chức năng và ràng buộc

- Yêu cầu phi chức năng; Định nghĩa các khía cạnh sử dụng phần mềm, không liên quan trực tiếp tới các hành vi chức năng:
 - Các tính chất của hệ thống như độ tin cậy, thời gian trả lời, dung lượng bộ nhớ, ...
 - Thời gian trả lời phải nhỏ hơn 1 giây
- Ràng buộc: do khách hàng hay môi trường thực thi phần mềm đặt ra
 - Các yêu cầu do tổ chức qui định như qui định chuẩn về quá trình tiến hành, chuẩn tài liệu, ...
 - Ngôn ngữ cài đặt phải là COBOL
 - Các yêu cầu từ bên ngoài
 - Phải giao tiếp với hệ thống điều phối được viết vào năm 1956.
- Thường sử dụng các công cụ
 - Biểu đồ thực thể liên kết (Entity-Relationship Diagrams)
 - Đặc tả Logic (Logic Specifications)
 - Đặc tả đại số (Algebraic Specifications)

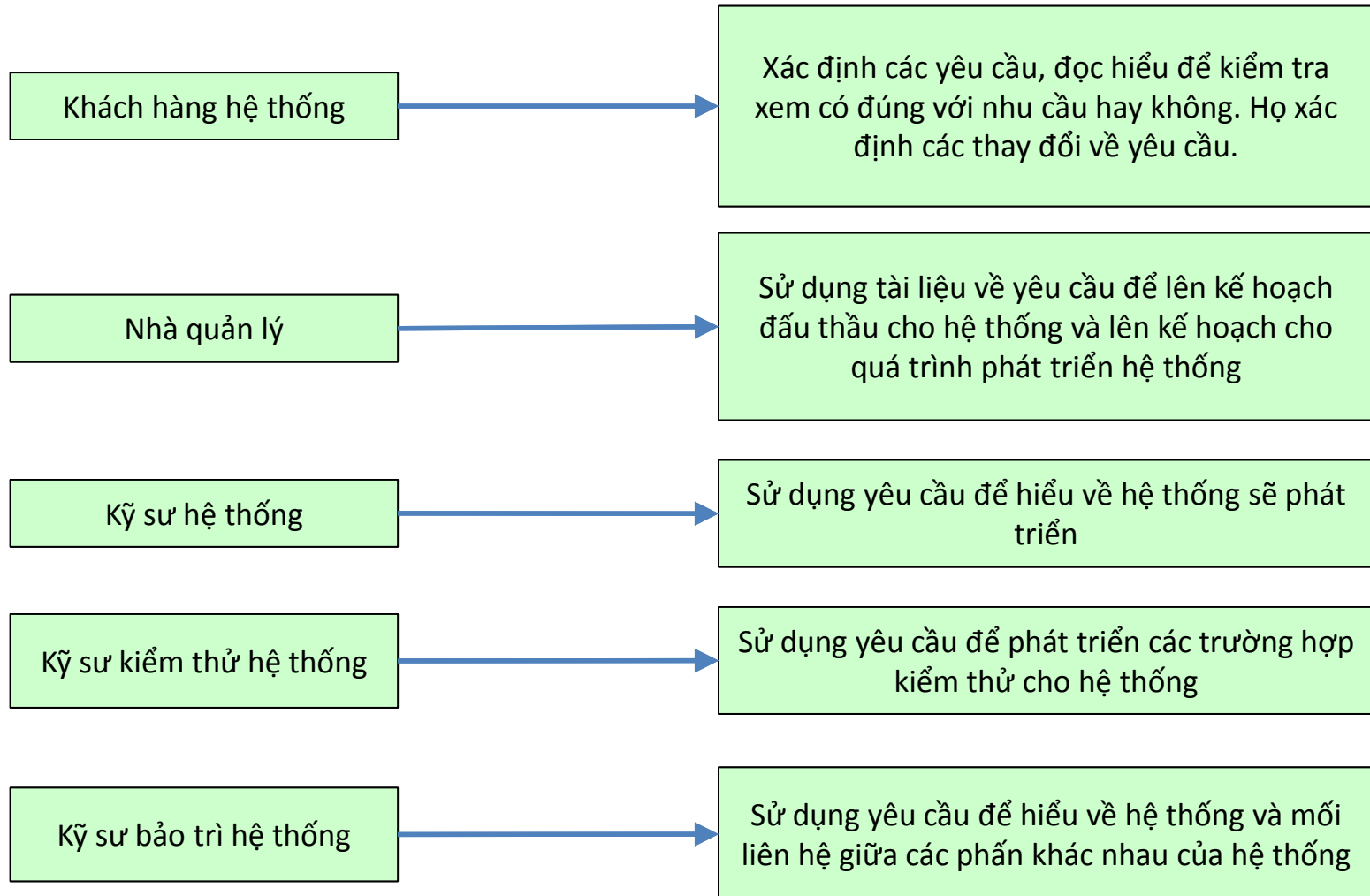
→ Khó phát biểu chính xác, Rất khó kiểm tra

Tài liệu yêu cầu

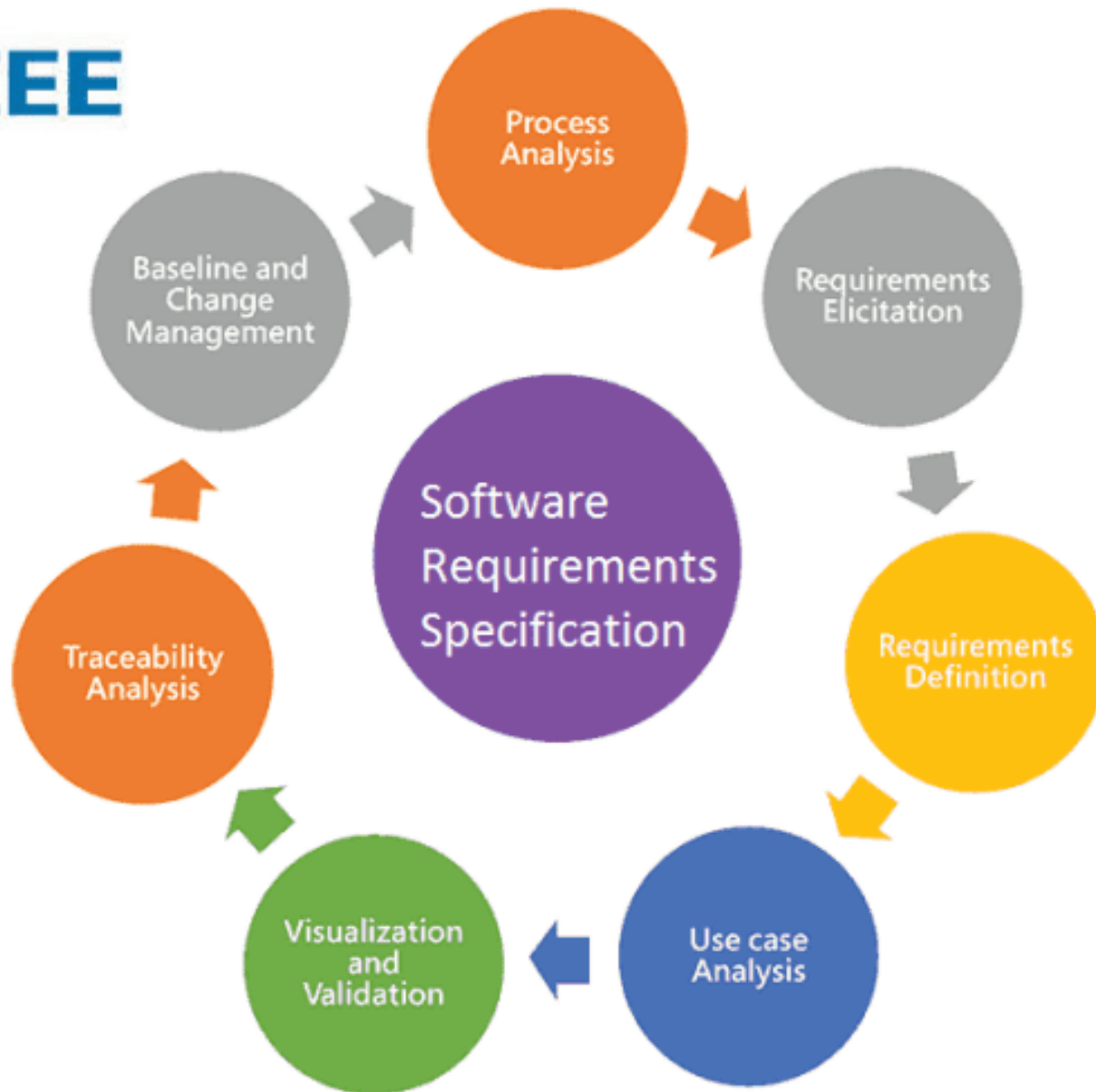
- Tài liệu về yêu cầu là các phát biểu chính thức về cái được yêu cầu bởi các nhà phát triển HT
- Nó bao gồm cả 2 phần: định nghĩa và đặc tả yêu cầu
- Nó không phải là tài liệu thiết kế. Tốt hơn có thể nó chỉ là 1 tập các cái mà HT phải làm hơn là HT phải làm thế nào (PT chứ không phải là TK)



Nội dung cần có của tài liệu yêu cầu



Tài liệu yêu cầu



Use-cases

- Tập hợp các kịch bản mô tả việc sử dụng hệ thống của người dùng.
- Mỗi kịch bản được mô tả dưới tầm nhìn của tác nhân- là người hoặc máy móc, tương tác với phần mềm theo một cách nào đó
- Mỗi kịch bản trả lời các câu hỏi sau:
 - Đây là tác nhân chính, tác nhân phụ
 - Mục tiêu của tác nhân là gì?
 - Điều kiện khi kịch bản bắt đầu?
 - Chức năng chính thực hiện bởi tác nhân?
 - Các biến thể trong tương tác của tác nhân?
 - Thông tin mà tác nhân mong muốn từ hệ thống?
 - Tác nhân thông báo cho hệ thống biết về những thay đổi bên ngoài

Tài liệu yêu cầu

1 Giới thiệu

1.1 Mục đích

Mục này giới thiệu mục đích của tài liệu yêu cầu. Thường chỉ đơn giản là định nghĩa “đây là tài liệu yêu cầu về phần mềm XYZ”.

1.2 Phạm vi

Nêu phạm vi đề cập của tài liệu yêu cầu.

1.3 Định nghĩa

Định nghĩa các khái niệm, các từ viết tắt, các chuẩn được sử dụng trong tài liệu yêu cầu.

1.4 Tài liệu tham khảo

Nêu danh mục các tài liệu tham khảo dùng để tạo ra bản đặc tả yêu cầu.

1.5 Mô tả chung về tài liệu

Mô tả khái quát cấu trúc tài liệu, gồm có các chương, mục, phụ lục chính nào.

2 Mô tả chung

2.1 Tổng quan về sản phẩm

Mô tả khái quát về sản phẩm.

2.2 Chức năng sản phẩm

Khái quát về chức năng sản phẩm.

2.3 Đối tượng người dùng

Mô tả về đối tượng người dùng.

2.4 Ràng buộc tổng thể

Khái quát về các ràng buộc của phần mềm: ràng buộc phần cứng, môi trường sử dụng, yêu cầu kết nối với các hệ thống khác...

2.5 Giả thiết và sự lệ thuộc

Mô tả các giả thiết khi áp dụng tài liệu: ví dụ như tên phần cứng, phần mềm, hệ điều hành cụ thể.

3 Yêu cầu chi tiết

Mô tả các yêu cầu

3.1 Yêu cầu chức năng

Mô tả chi tiết về các yêu cầu chức năng.

3.1.1 Yêu cầu chức năng 1

3.1.1.1 Giới thiệu

3.1.1.2 Dữ liệu vào

3.1.1.3 Xử lý

3.1.1.4. Kết quả

3.1.2 Yêu cầu chức năng 2

...

3.1.n Yêu cầu chức năng n

3.2 Yêu cầu giao diện ngoài

Mô tả các giao diện của phần mềm với môi trường bên ngoài.

3.2.1 Giao diện người dùng

3.2.2 Giao diện phần cứng

3.2.3 Giao diện phần mềm

3.2.4 Giao diện truyền thông

3.3 Yêu cầu hiệu suất

Mô tả về hiệu suất, ví dụ như thời gian phản hồi với sự kiện, số giao dịch được thực hiện/giây,....

3.4 Ràng buộc thiết kế

Mô tả các ràng buộc thiết kế, ví dụ các ràng buộc về ngôn ngữ, về công nghệ, về cơ sở dữ liệu và về chuẩn giao tiếp.

3.5 Thuộc tính

Mô tả các thuộc tính của phần mềm.

3.5.1 Tính bảo mật, an toàn và khả năng phục hồi

Mức độ bảo mật dữ liệu, cách thức truy cập vào hệ thống. Độ an toàn của hệ thống đối với các trường hợp bất thường như mất điện... Khả năng phục hồi của hệ thống sau khi gặp sự cố.

3.5.2 Tính bảo trì

Các chức năng, giao diện đòi hỏi phải dễ sửa đổi (dễ bảo trì).

3.6 Các yêu cầu khác

Các yêu cầu khác liên quan đến sản phẩm.

Đánh giá yêu cầu – I

- Xem xét mỗi yêu cầu có phù hợp với mục tiêu tổng thể của hệ thống hay không?
- Yêu cầu có thực sự cần thiết hay không? Hay nó là đại diện cho một add-on không cần thiết của hệ thống?
- Yêu cầu có cụ thể và rõ ràng hay không?
- Mỗi yêu cầu có xung đột, đụng độ với yêu cầu khác hay không?
- Mỗi yêu cầu đó có đóng góp gì vào hoạt động của hệ thống hay không?
- Mỗi yêu cầu có phù hợp với môi trường mà hệ thống sẽ được triển khai hay không?
- Từng yêu cầu đã được kiểm chứng hay chưa?



Đánh giá yêu cầu - II

- Các mô hình đã phản ánh đúng thông tin, chức năng, hành vi của hệ thống hay chưa?
- Các mô hình yêu cầu được phân chia theo cách giúp làm rõ hơn về hệ thống hay chưa?
- Liệu các mô hình yêu cầu đúng phản ánh thông tin, chức năng và hành vi của hệ thống được xây dựng?
- Có các yêu cầu mô hình được "phân chia" theo một cách mà cho thấy nhiều thông tin dần dần chi tiết hơn về hệ thống?
- Có yêu cầu mô hình được sử dụng để đơn giản hóa các mô hình yêu cầu. Đã tất cả các mẫu được xác nhận đúng? Có phải tất cả các mô hình phù hợp với yêu cầu của khách hàng?

SCM cho kỹ nghệ Web-II

- **Khả năng mở rộng.**

- Bởi vì kích thước và độ phức tạp phát triển, thay đổi nhỏ có thể có ảnh hưởng sâu rộng và không lường trước được ảnh hưởng có thể có vấn đề. Do đó, sự chặt chẽ của các cơ chế kiểm soát cấu hình nên được tỷ lệ thuận với quy mô ứng dụng.

- **Về mặt pháp lý/chính trị.**

- Ai là người sở hữu WebApp?
- Ai chịu trách nhiệm trực tiếp đến thông tin của trang Web ?
- Ai đảm bảo rằng kiểm soát chất lượng quy trình đã được tuân theo trước khi thông tin được công bố để đưa các trang web vào hoạt động?
- Ai chịu trách nhiệm thực hiện thay đổi?
- Ai giả định chi phí thay đổi?

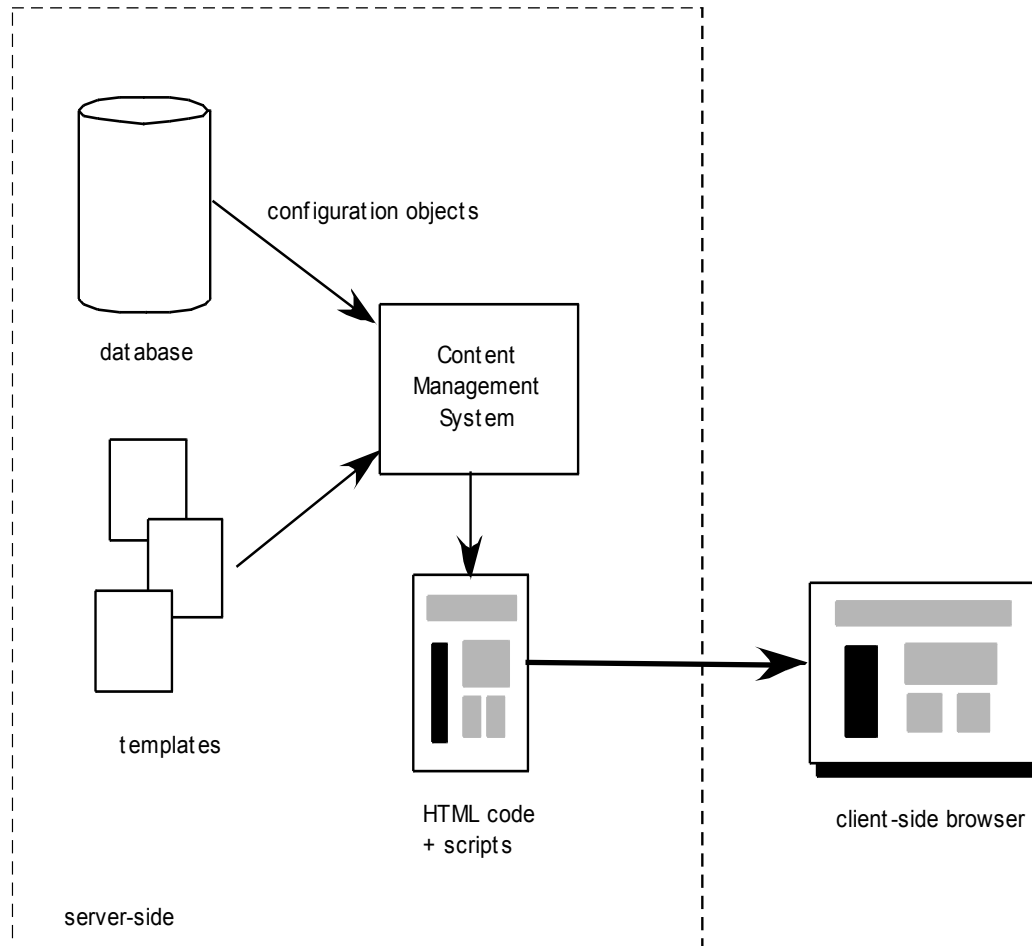
Quản lý nội dung-I

- **Tập hợp những hệ thống con** bao gồm tất cả các hành động yêu cầu để tạo và/hoặc lấy nội dung, và những chức năng kỹ thuật mà cần thiết để:
 - Chuyển nội dung bên trong một form mà được trình bày bởi ngôn ngữ HTML, XML.
 - Tập hợp nội dung bên trong các gói được hiển thị tốt phía máy cá nhân.
- **Quản lý hệ thống con** thực thi một kho chứa bao gồm những yếu tố sau đây:
 - **Nội dung cơ sở dữ liệu**—là cấu trúc thông tin mà được thiết lập lưu trữ tất cả nội dung của các đối tượng.
 - **Những khả năng cơ sở dữ liệu**—những chức năng mà cho phép CMS tìm kiếm các đối tượng cụ thể(hoặc giới hàng của đối tượng), lưu trữ và truy xuất các đối tượng, và quản lý the cấu trúc file mà được thiết lập sẵn nội dung.
 - Những chức năng quản lý cấu hình —là các yếu tố chức năng và công việc liên quan mà hỗ trợ xác định các đối tượng nội dung, điều khiển các phiên bản, thay đổi sự quản lý, thay đổi tác giả, và những báo cáo.

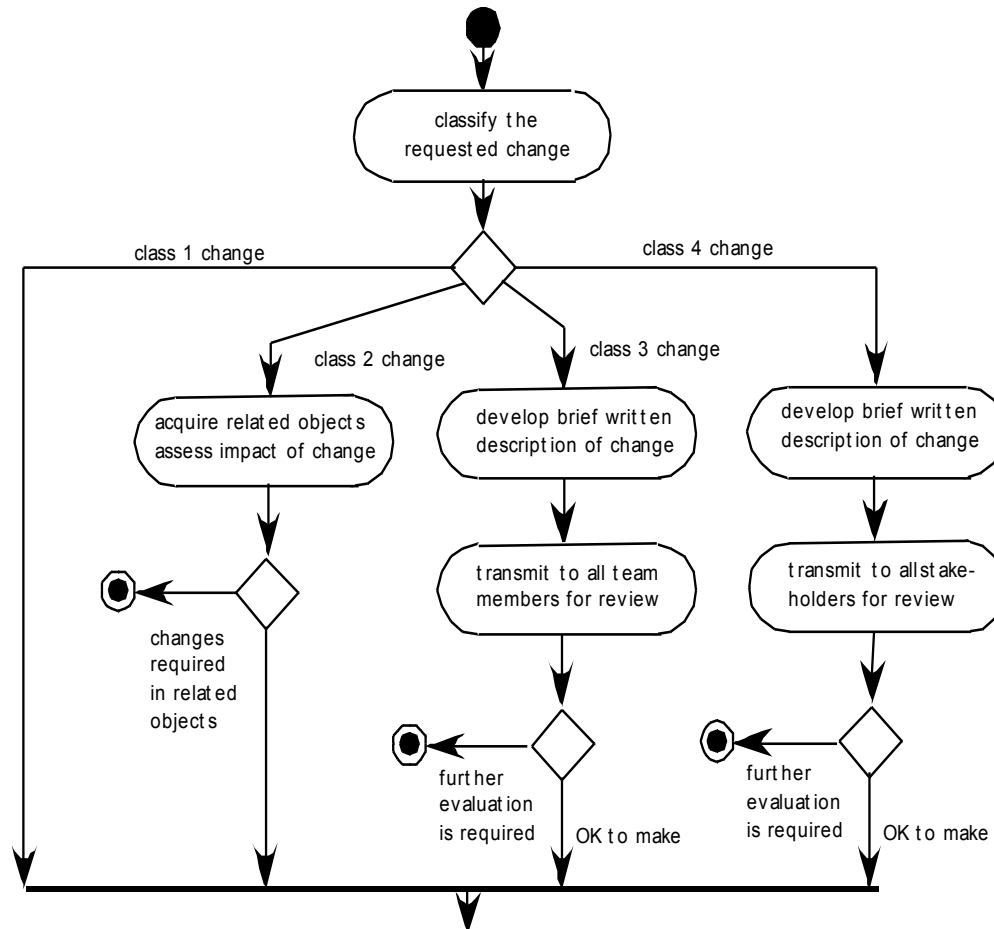
Quản lý nội dung-II

- Các hệ thống phụ xuất bản chất chiết xuất từ kho lưu trữ, chuyển nó đến một hình thức phụ tùng để xuất bản và định dạng nó để nó có thể được truyền đến các trình duyệt phía khách hàng. Các hệ thống xuất bản hoàn thành các tác vụ này bằng cách sử dụng một loạt các mẫu.
- Mỗi mẫu là một chức năng mà xây dựng xuất bản sử dụng một trong ba thành phần khác nhau [BOI02]:
 - **Những yếu tố tĩnh** - chữ , đồ hoạ, truyền thông và các nguyên bản mà không cần tiếp tục xử lý được chuyển đổi trực tiếp đến máy cá nhân.
 - **Dịch vụ xuất bản** - gọi các chức năng để thu hồi và định dạng các dịch vụ mà cá nhân hoá nội dung (sử dụng các quy tắc được xác định trước), chuyển đổi dữ liệu, xây dựng những liên kết điều hướng thích hợp.
 - **Dịch vụ bên ngoài** - cung cấp truy cập đến cơ sở thông tin như các dữ liệu doanh nghiệp hoặc những ứng dụng “back-rooms”.

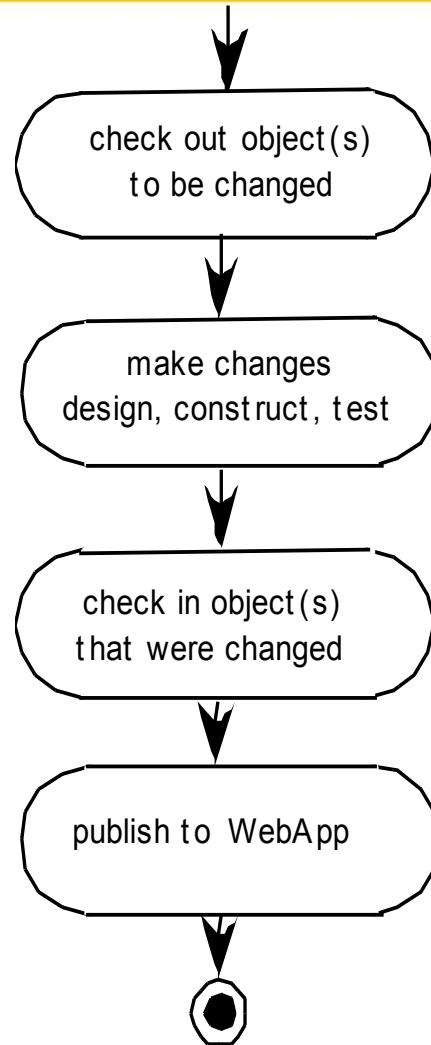
Quản lý nội dung



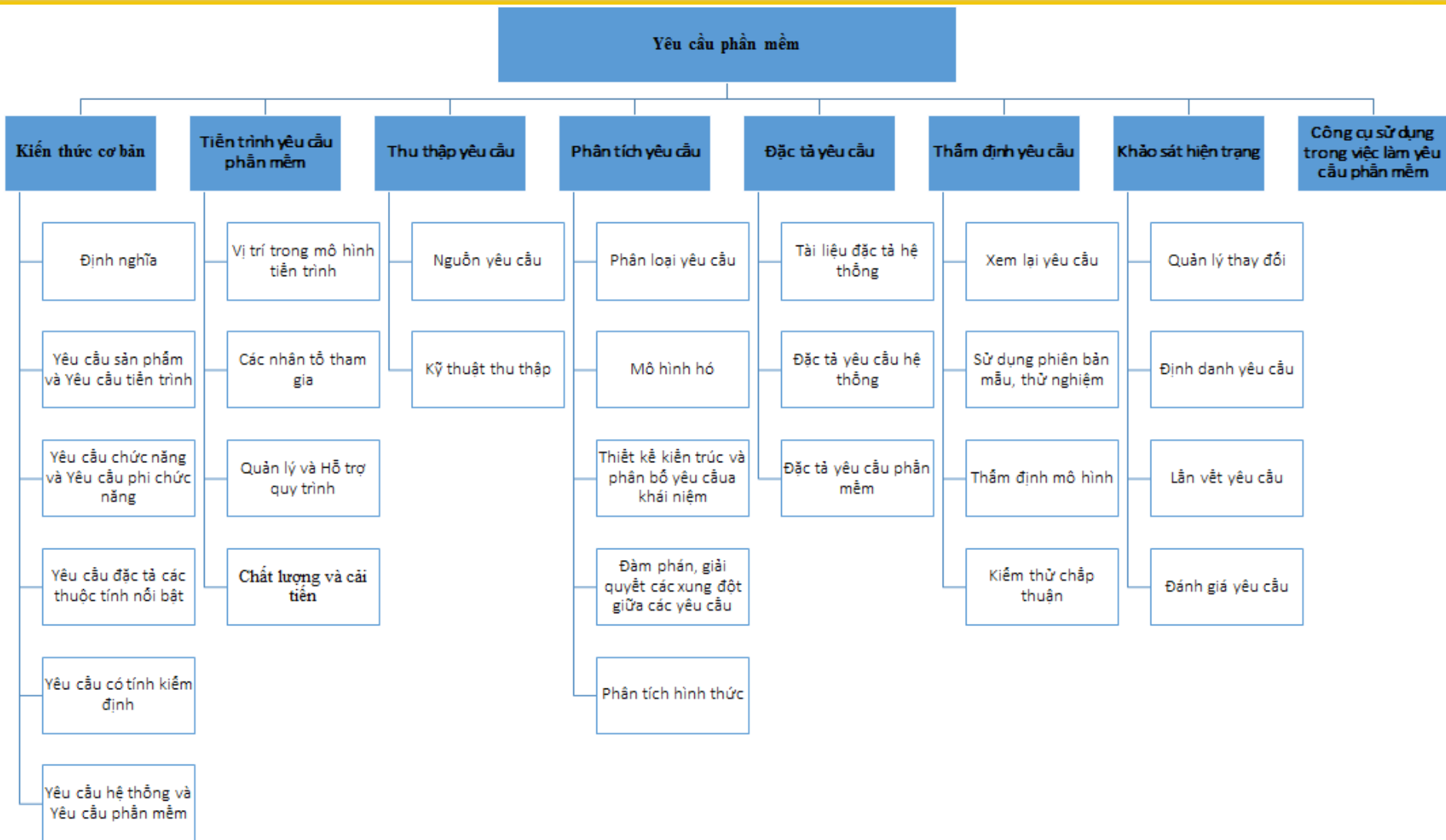
Quản lý Thay đổi cho WebApps-I



Quản lý Thay đổi cho WebApps-II



Tổng kết



Tài liệu tham khảo

- Slide đi kèm với Software Engineering: A Practitioner's Approach, 7/e by Roger S. Pressman
- Slides copyright © 1996, 2001, 2005, 2009 by Roger S. Pressman
- Chỉ dùng cho mục đích giáo dục phi lợi nhuận.
- Có thể sửa đổi slide chỉ nhằm mục đích phục vụ sinh viên đại học trong những môn học liên quan tới sách Software Engineering: A Practitioner's Approach, 7/e. Nghiêm cấm mọi hoạt động sửa đổi khác hoặc sử dụng không được sự cho phép của tác giả.
- Mọi thông tin bản quyền phải được đi kèm nếu những slide này được đăng lên mạng để phục vụ sinh viên.