

PHÂN TÍCH THIẾT KẾ HỆ THỐNG THÔNG TIN

Bài 5. Requirements – Yêu cầu

Giáo viên: TS. Trần Mạnh Tuấn

Bộ môn: Hệ thống thông tin

Khoa: Công nghệ thông tin

Email: tmtuan@tlu.edu.vn

Điện thoại: 0983.668.841

Nội dung

- 1. Quy trình phát triển HĐT(OO) - RUP**
- 2. Mô hình hóa nghiệp vụ**
- 3. Requirement – Yêu cầu**

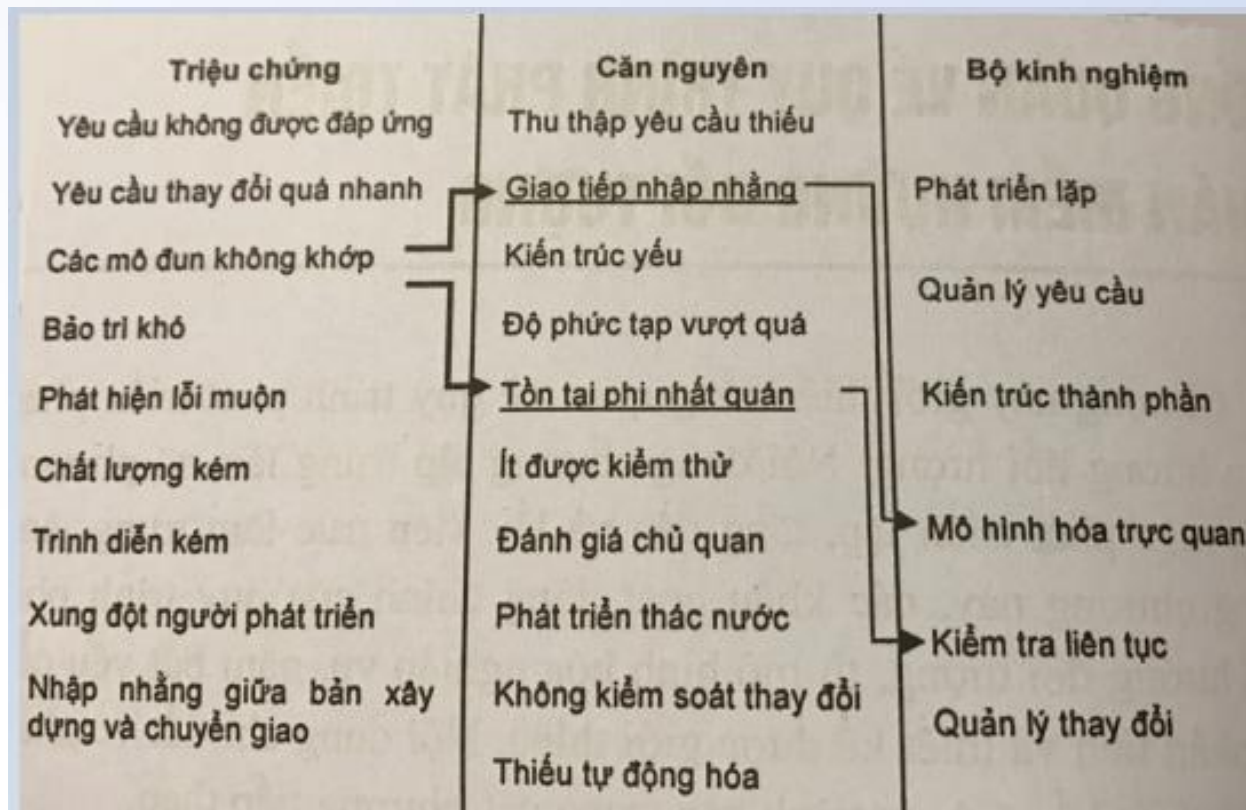
Quy trình phát triển HĐT

- ❖ Quy trình phát triển theo chức năng gặp nhiều hạn chế, thiếu các tiêu chí chất lượng và đặc biệt tính bảo trì được.
- ❖ Thúc đẩy phát sinh ra quy trình phát triển theo hướng đối tượng (OO).
- ❖ RUP – Rational Unified Process – Quy trình đồng nhất hợp nhất là một trong những quy trình như vậy

Quy trình phát triển HĐT

Các triệu chứng của vấn đề phát triển phần mềm

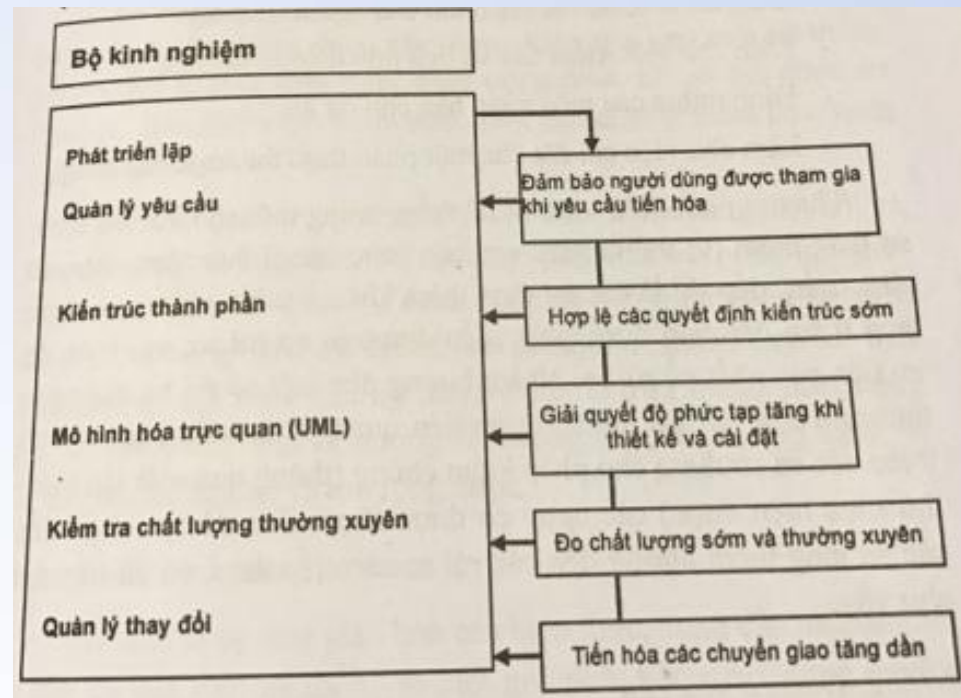
- Triệu chứng (Symptoms) – Các vấn đề xấu trong phần mềm. Việc xử lý các triệu chứng xấu sẽ làm chất lượng phần mềm cao theo định hướng lập và dự đoán được



Quy trình phát triển HĐT

Bộ kinh nghiệm thực tế (Best practises)

- Tập các phương pháp phát triển phần mềm đã được kiểm nghiệm bằng các phần mềm thương mại.
- Tính đúng đắn được khẳng định thông qua quá trình được sử dụng thường xuyên và thành công trong công nghiệp và các tổ chức.
- Bộ kinh nghiệm thu được:
 - Khách hàng
 - Dự án
 - Chuyên gia



Quy trình phát triển HĐT

Bộ kinh nghiệm thực tế

❖ Phát triển lặp

- Kỹ thuật được sử dụng để chuyển các chức năng của hệ thống vào một chuỗi liên tục các phiên bản hoàn thiện tăng dần.
- Mỗi phiên bản được phát triển trong thời gian cố định, gọi là vòng lặp. Mỗi vòng lặp tập trung vào: 1. định nghĩa – 2. phân tích – 3. thiết kế - 4. xây dựng – 5. kiểm thử một tập các **yêu cầu**

Quy trình phát triển HĐT

Bộ kinh nghiệm thực tế

- Vòng lặp giải quyết vấn đề:
 - Giải quyết các rủi ro lớn trước khi đầu tư
 - Sớm nhận được các phản hồi người dùng
 - Làm cho việc kiểm thử và tích hợp diễn ra liên tục
 - Định nghĩa các mốc ngắn hạn cho dự án
 - Làm cho việc cài đặt của một phần thực thi được sẵn sàng.

Quản lý yêu cầu:

- Tỷ lệ thành công của dự án phụ thuộc rất lớn (yếu tố quyết định) trong việc quản lý các yêu cầu dự án.
- Các khía cạnh quản lý yêu cầu:
 - Phân tích vấn đề
 - Hiểu sự mong đợi của người sử dụng
 - Định nghĩa hệ thống
 - Quản lý phạm vi
 - Làm mịn định nghĩa hệ thống
 - Quản lý thay đổi yêu cầu.

Quy trình phát triển HĐT

Bộ kinh nghiệm thực tế

❖ Sử dụng kiến trúc phần mềm

- Kiến trúc là một phần của thiết kế. Nó bao gồm các quyết định làm thế nào hệ thống được xây dựng.
- Kiến trúc phần mềm là khía cạnh quan trọng nhất, nó điều khiển quy trình phát triển lặp và tăng thêm của hệ thống trong suốt vòng đời phát triển.
- Tính chất của kiến trúc:
 - Khả năng đàn hồi và linh động
- Đề đạt được tính chất này cần dự đoán trong cả lĩnh vực phần mềm và công nghệ phát triển, để đưa ra một bản thiết kế tính đến sự thay đổi này.
- Kỹ thuật chính:
 - Trừu tượng hóa
 - Đóng gói
 - Phân tích thiết kế hướng đối tượng
- Kết quả: đưa ứng dụng về cơ bản có thể bảo trì và mở rộng.

Quy trình phát triển HĐT

Bộ kinh nghiệm thực tế

❖ Mô hình hóa trực quan

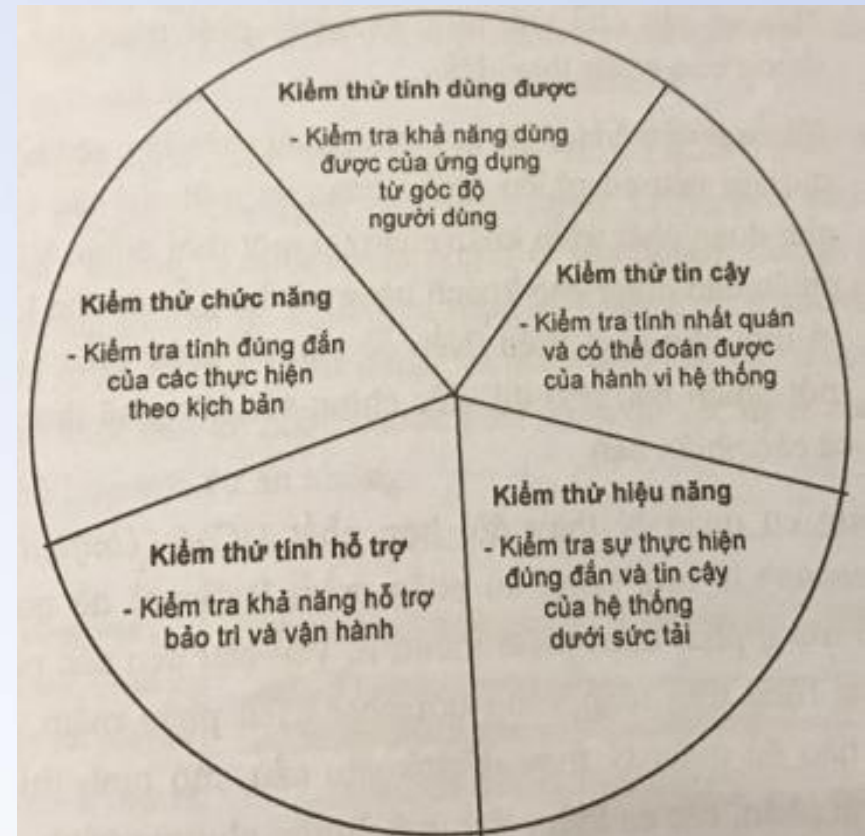
- Mô hình là sự đơn giản hóa của hiện thực, cung cấp một sự mô tả đầy đủ của một hệ thống từ một góc nhìn nào đó.
- Mô hình hóa rất quan trọng vì nó giúp việc phát triển hiện thị, đặc tả, xây dựng và tài liệu hóa cấu trúc và hành vi của kiến trúc của hệ thống. Sử dụng UML các thành viên trong nhóm phát triển có thể trao đổi các quyết định về hệ thống với nhau.
- Giúp đội phát triển quản lý sự phức tạp của hệ thống.

Quy trình phát triển HĐT

Bộ kinh nghiệm thực tế

❖ Kiểm tra thường xuyên

- Kiểm thử chức năng (Functional Testing)
- Kiểm thử tính dùng được (Usability Testing)
- Kiểm thử tin cậy (Reliability Testing)
- Kiểm thử hiệu năng (Performance Testing)
- Kiểm thử tính hỗ trợ (Supportability Testing)



Quy trình phát triển HĐT

Bộ kinh nghiệm thực tế

❖ Quản lý sự thay đổi

- Vấn đề thách thức của phát triển hệ thống phần mềm:
 - Nhiều người phát triển
 - Nhiều team phát triển
 - Các team ở các vị trí, địa điểm khác nhau

Nếu thiếu nguyên tắc điều khiển, quy trình phát triển phần mềm sẽ bị hỗn loạn.

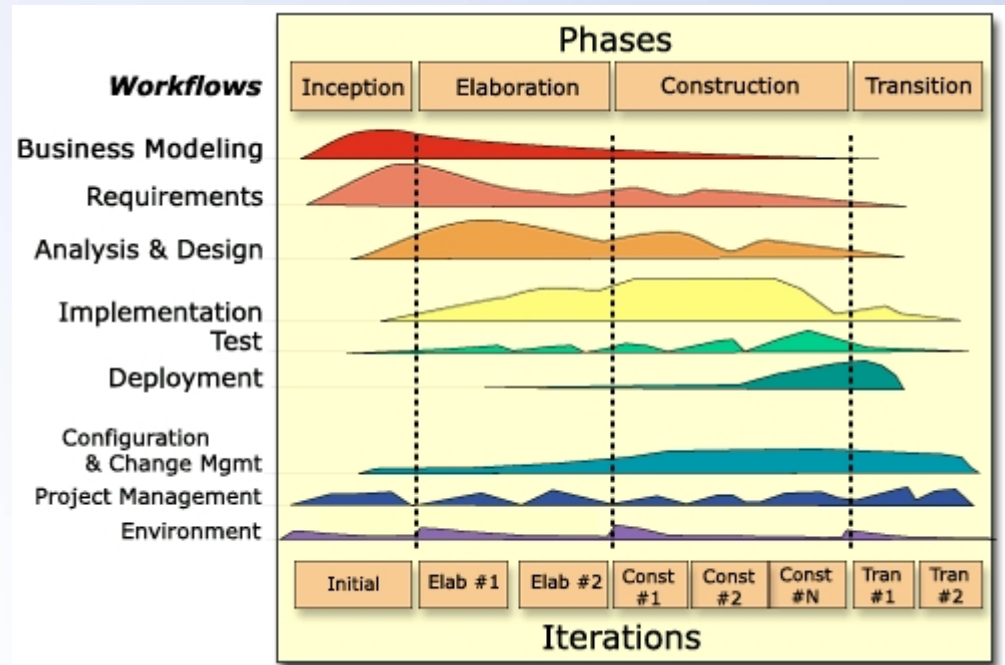
- Ba vấn đề thường gặp:
 - Cập nhật đồng thời
 - Thông báo hạn chế
 - Nhiều phiên bản
- UCM – Unified Change Mngt – Tạo ra sự thống nhất giữa các hoạt động sử dụng để lập kế hoạch và theo dõi sự tiến triển của dự án.

Quy trình phát triển HĐT

Bộ kinh nghiệm thực tế

❖ RUP – Rational Unified Process – Quy trình hợp nhất

- Quy trình nghiệp vụ cho kỹ thuật phần mềm hướng đối tượng, dùng mô tả một họ các tiến trình kỹ thuật phần mềm cùng chia sẻ cấu trúc và kiến trúc tiến trình chung.
- Có 4 pha (phases):
 - Khởi tạo
 - Chi tiết
 - Xây dựng
 - Chuyển giao



Quy trình phát triển HĐT

Bộ kinh nghiệm thực tế

❖ Các khâu hoạt động của RUP:

- Mô hình hóa nghiệp vụ: bao gồm tất cả kỹ thuật trực quan hóa mô hình nghiệp vụ.
- Yêu cầu: Định nghĩa hệ thống cần làm gì.
- Phân tích & Thiết kế: Chỉ ra làm thế nào các ca sử dụng (usecases) của hệ thống được cài đặt.
- Thực thi: Coding
- Kiểm thử: Testing
- Cài đặt: Cung cấp sản phẩm phần mềm đến người sử dụng.
- Quản lý thay đổi & Cấu hình: Điều khiển và theo dõi sự thay đổi.
- Quản lý dự án: Bảo đảm tất cả các công việc được lập lịch, cung cấp và hoàn chỉnh phù hợp với lịch, yêu cầu và kinh phí dự án.
- Môi trường: Định nghĩa và quản lý môi trường trên đó hệ thống được phát triển.

Quy trình phát triển HĐT

Mô hình hóa

❖ Mô hình

- là bức tranh hay mô tả vấn đề đang cố gắng giải quyết hay mô tả chính giải pháp vấn đề
- là ngôn ngữ của người thiết kế (trong nhiều lĩnh vực)
- là trình diễn hệ thống sẽ xây dựng
- là phương tiện giao tiếp giữa các stakeholders
- là kế hoạch chi tiết (**blueprints**)

❖ Mô hình cho khả năng suy diễn một số đặc tính của hệ thống thực

❖ Mô hình hóa trực quan

- Bảng các phần tử đồ họa

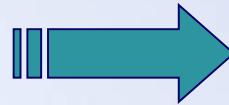
❖ **Ngôn ngữ mô hình hóa** là ngôn ngữ mô tả hệ thống hay tác nghiệp

Mô hình hóa nghiệp vụ

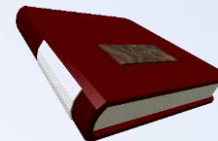
Mô hình hóa



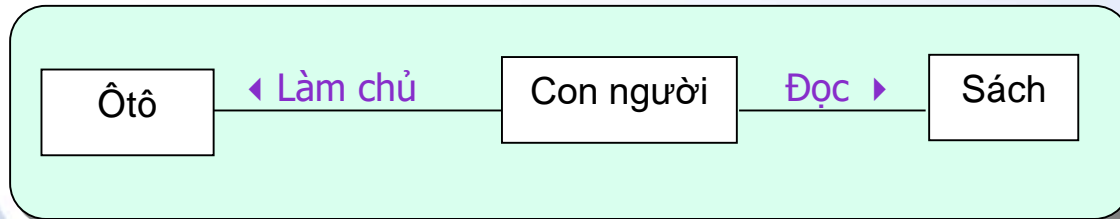
Thế giới thực



Mô hình: Quả địa cầu học sinh



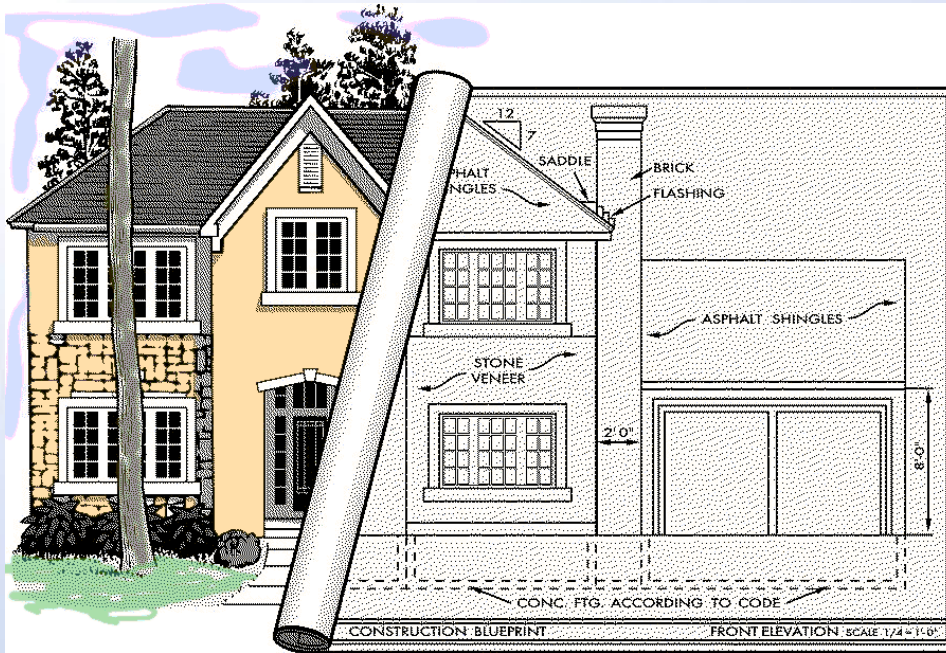
Thế giới thực



Mô hình

Mô hình hóa nghiệp vụ

Mô hình hóa



Mô hình hóa nghiệp vụ

- ❖ Cách tiếp cận chính thống và hiệu quả để nắm bắt các yêu cầu cho việc xây dựng các công cụ và hệ thống hỗ trợ nghiệp vụ.
- ❖ Nó giúp chúng ta hiểu chính xác về quy trình nghiệp vụ và là cơ sở cho tính chính đúng đắn của hệ thống xây dựng lên.

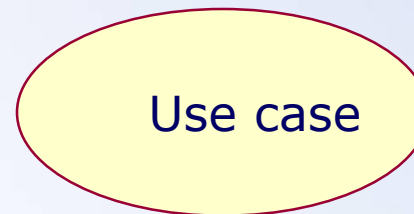
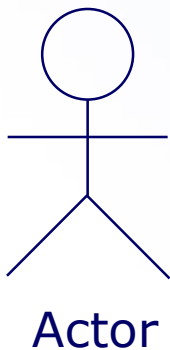
❖ Mục đích của mô hình hóa nghiệp vụ:

- Để hiểu được cấu trúc và khía cạnh động của tổ chức trong hệ thống được triển khai
- Để hiểu được vấn đề thực tại của tổ chức, xác định các cải tiến nhằm nâng cao hiệu quả của tổ chức
- Để đảm bảo cái hiểu thống nhất về tổ chức giữa khách hàng, người dùng cuối và người phát triển.
- Để nắm bắt các yêu cầu của hệ thống cần hỗ trợ cho tổ chức.

Mô hình hóa nghiệp vụ

❖ Hoạt động của mô hình hóa nghiệp vụ:

- Tìm tác nhân nghiệp vụ: là các cá nhân, các nhóm, các tổ chức, cơ quan, các hệ thống tương tác với nghiệp vụ đó. (Actors)
- Tìm các ca sử dụng nghiệp vụ (Use cases): Để tìm các Use cases, chúng ta xuất phát từ các tác nhân nghiệp vụ (Actors) và phân tích các giá trị thu được của chúng từ các hoạt động nghiệp vụ.



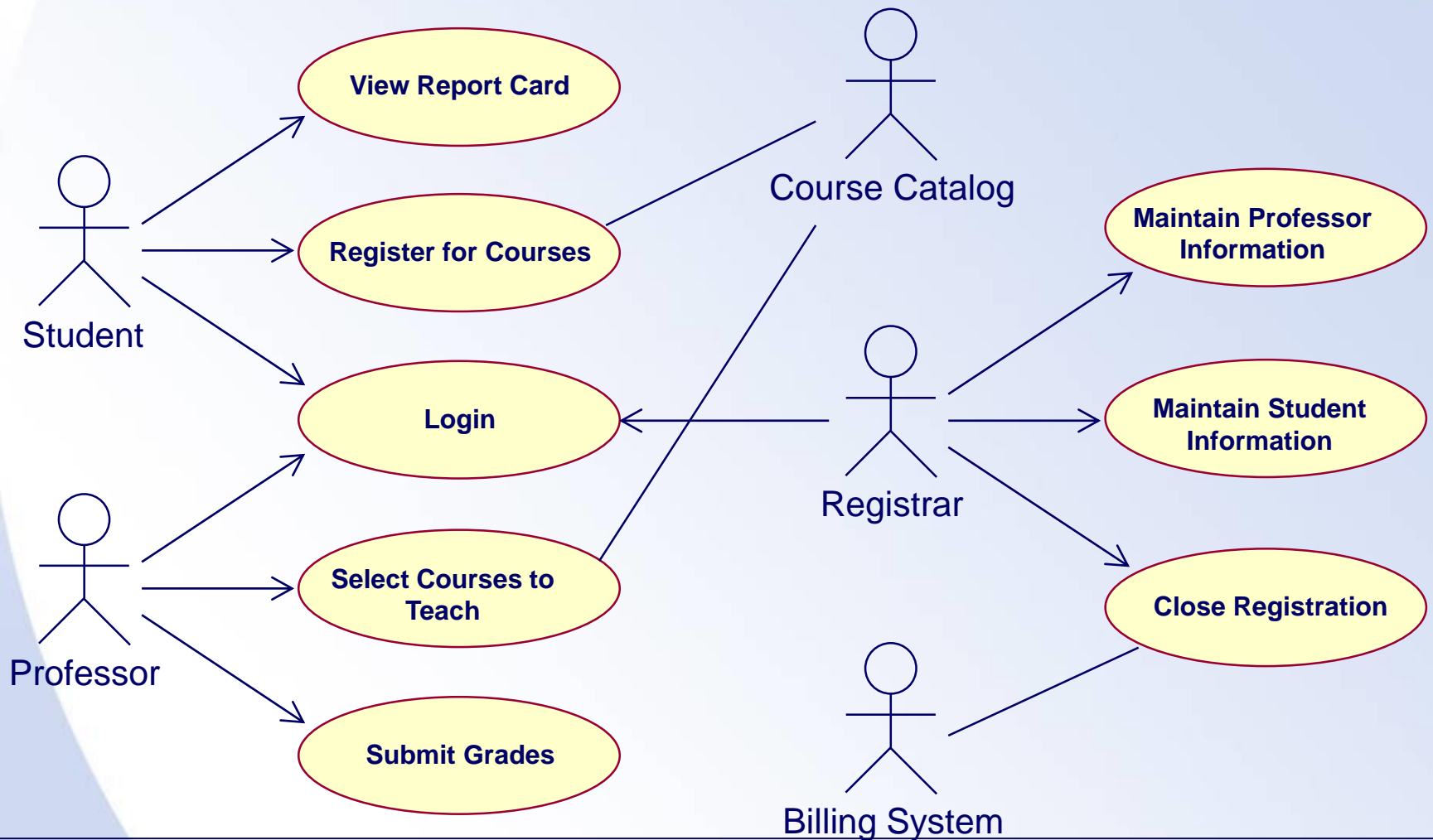
Requirements – Yêu cầu

- ❖ **Nắm bắt yêu cầu là khâu diễn ra ngay trước hoạt động phân tích và thiết kế trong quá trình phát triển phần mềm.**
- ❖ **Mục đích của nắm bắt yêu cầu phần mềm:**
 - Để tìm ra giải pháp đúng cho một hệ thống phần mềm, chúng ta phải hiểu được vấn đề, cũng như những yêu cầu đặt ra cho hệ thống.
 - Các câu hỏi đặt ra trong quá trình nắm bắt yêu cầu:
 - Để làm gì – What to do?
 - Làm như thế nào – How to do?

❖ Chế tác cho mô hình yêu cầu

- Phát biểu vấn đề
 - Một phần tài liệu hệ thống
 - Làm rõ thực trạng của hệ thống hiện thời, đặc biệt các vấn đề khó đặt ra cho hệ thống
- Mô hình ca sử dụng (Use cases)
 - Mô tả những gì mà hệ thống sẽ làm.
 - Cam kết giữa khách hàng, người dùng và người phát triển
 - Mỗi ca sử dụng trong mô hình ca sử dụng mô tả chi tiết những gì mà hệ thống làm trong quá trình tương tác và tác nhân. Nó phản ánh chuỗi các sự kiện diễn ra trong từng kịch bản ca sử dụng.
 - Những mô tả này làm thành tài liệu đặc tả ca sử dụng (use case specification).

Requirements – Yêu cầu



Requirements – Yêu cầu

- Từ điển thuật ngữ (Glossary)
 - Bao gồm các thuật ngữ trong miền của hệ thống, được mô tả ở dạng văn bản và được dùng chung cho tất cả các mô hình của hệ thống.
 - Định nghĩa các thuật ngữ quan trọng.
 - Mục đích
 - Thống nhất thuật ngữ
 - Thuật tiện giao tiếp
 - Được tạo ra trong giai đoạn đầu – Pha khởi đầu (Inception) và pha chi tiết (Elaboration).

Requirements – Yêu cầu

Mô hình ca sử dụng

- Đặc tả bổ sung (Supplementary Specification)
 - Chức năng
 - Tính khả dụng (Usability)
 - Độ tin cậy (Reliability)
 - Hiệu năng (Performance)
 - Khả năng hỗ trợ (Supportability)
 - Ràng buộc về thiết kế (Design constraints)

Requirements – Yêu cầu

Mô hình ca sử dụng

Tên

Mô tả ngắn gọn

Luồng sự kiện

Các quan hệ

Các lược đồ hoạt động
(activity diagram)

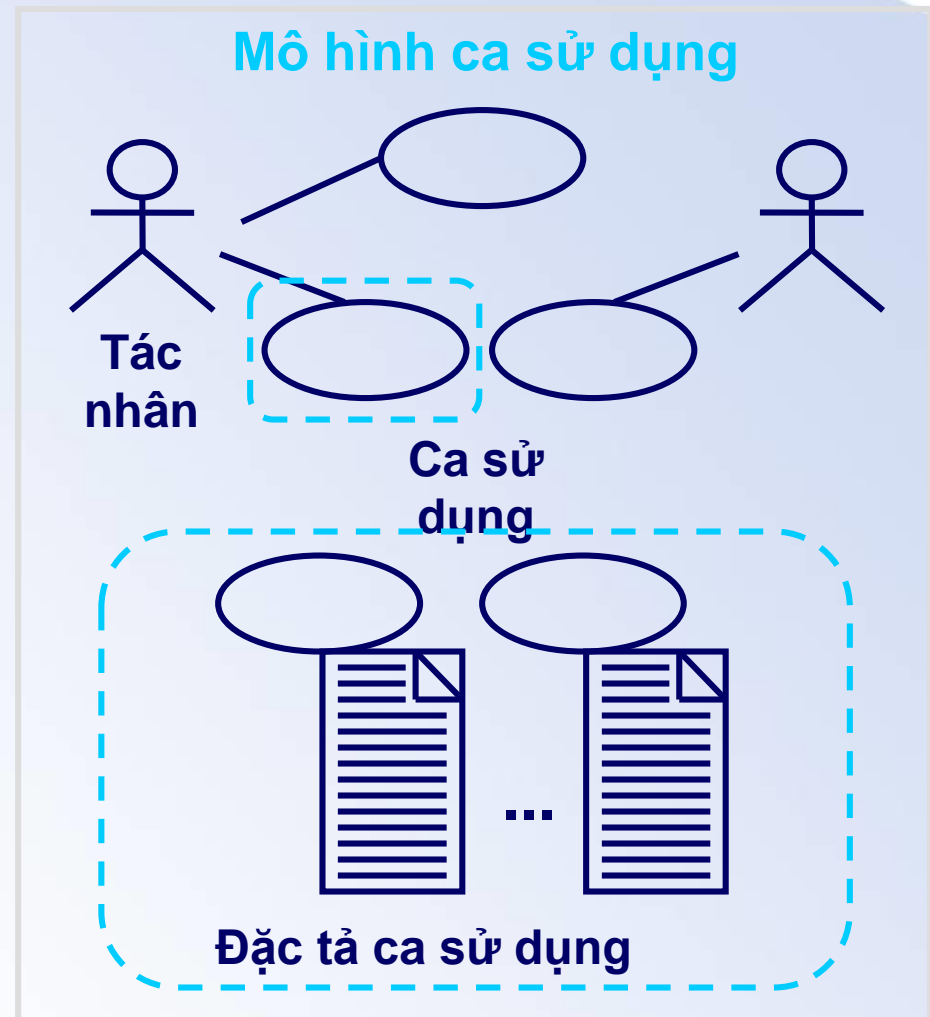
Các lược đồ ca sử dụng

Các yêu cầu cụ thể

- Tiền điều kiện

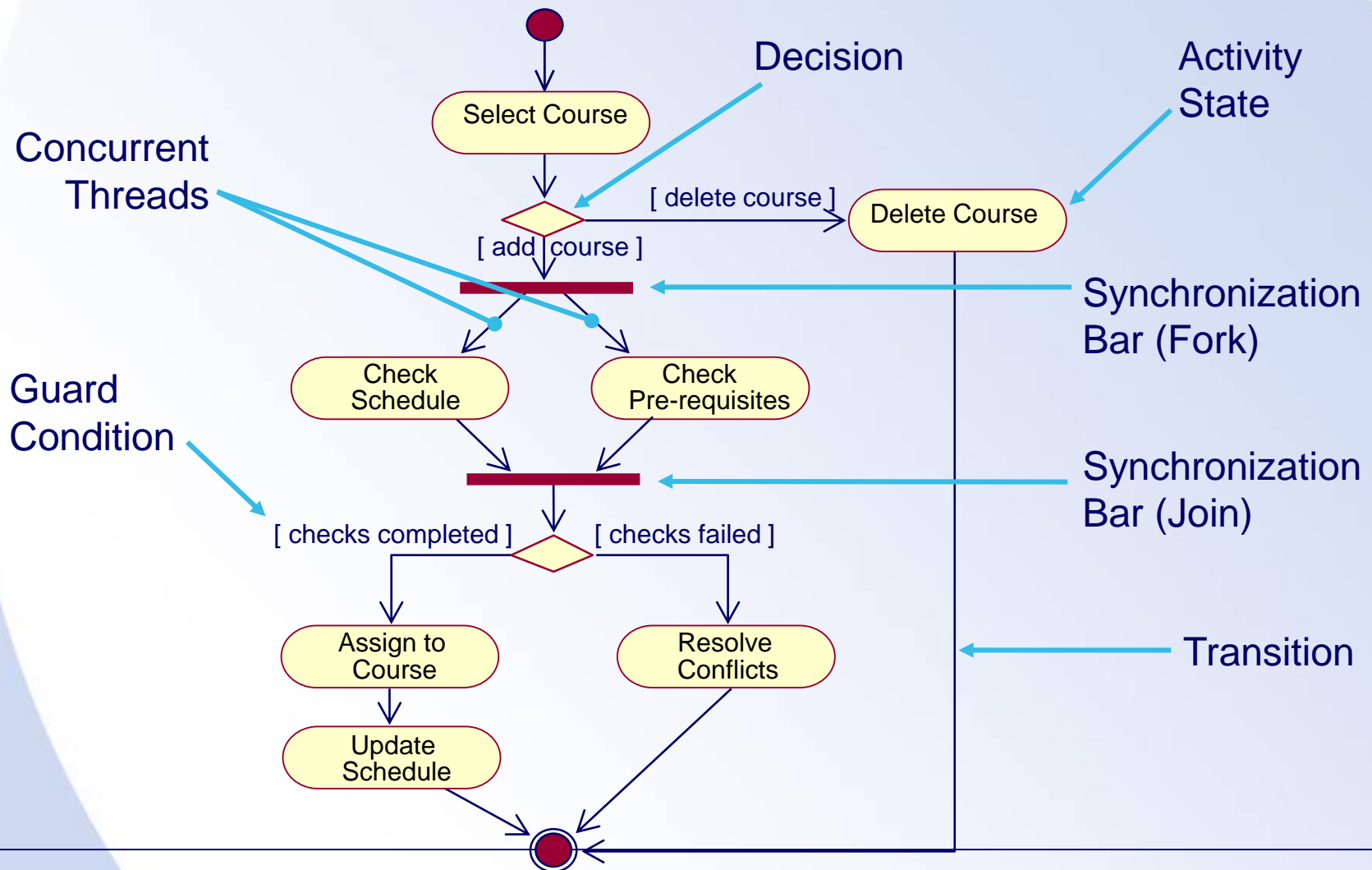
- Hậu điều kiện

- Các lược đồ khác



Requirements – Yêu cầu

Đặc tả ca sử dụng với biểu đồ hoạt động



Bài tập

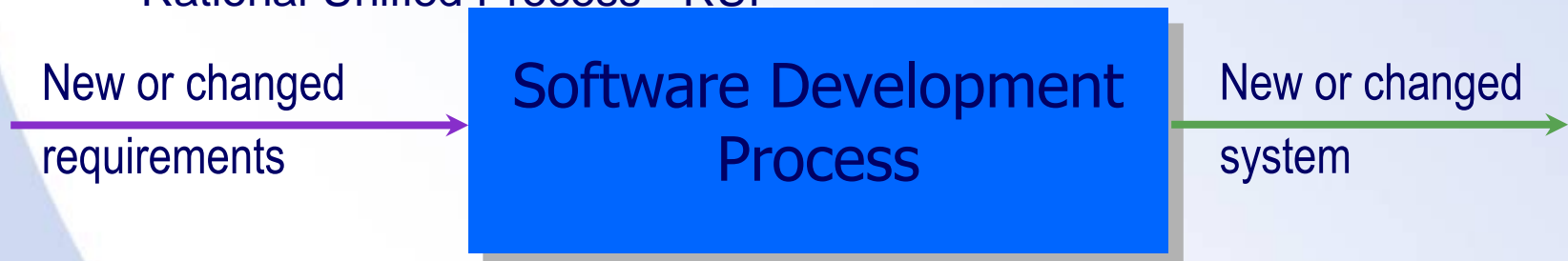
Bài toán: Một cửa hàng buôn bán đồ dùng thể thao muốn xây dựng một chương trình quản lý bán hàng qua môi trường Web và tại cửa hàng.

Hãy xác định yêu cầu người dùng

Trao đổi, câu hỏi?

Tiến trình phát triển phần mềm

- ❖ Mọi kỹ nghệ (**engineering**) đều đề cập đến sản xuất sản phẩm theo tiến trình
- ❖ Tổng quát thì tiến trình (**process**) xác định ai (**Who**) làm gì (**What**); làm khi nào (**When**) và làm như thế nào (**How**) để đạt tới mục đích mong muốn.
- ❖ Tiến trình phát triển phần mềm (**Software Development Process - SDP**) là tiến trình xây dựng sản phẩm phần mềm hay nâng cấp phần mềm đang có.
- ❖ **Thí dụ tiến trình phát triển phần mềm:**
 - Rational Unified Process - RUP



Tiến trình phát triển phần mềm

- ❖ **Tiến trình phát triển phần mềm mô tả tập các hoạt động cần thiết để chuyển đổi từ yêu cầu người sử dụng sang hệ thống phần mềm**
- ❖ **Yêu cầu người sử dụng xác định mục tiêu phát triển phần mềm**
 - Khách hàng và kỹ sư tin học xác định các dịch vụ mà hệ thống cần có (yêu cầu chức năng của hệ thống)
- ❖ **Yêu cầu chức năng mô tả cái mà hệ thống phải làm (**What**) không mô tả hệ thống làm như thế nào (**How**)**
 - Khách hàng cũng có các ràng buộc phi chức năng: thời gian đáp ứng, chuẩn ngôn ngữ...

Tiến trình phát triển phần mềm

- ❖ **Thu thập và phân tích yêu cầu là công việc rất khó khăn**
 - Các yêu cầu thường là không hoàn chỉnh
 - Yêu cầu của khách hàng thường được mô tả bằng khái niệm, đối tượng và các thuật ngữ khó hiểu với kỹ sư tin học
 - Các yêu cầu của khách hàng thường thiếu cấu trúc, thiếu chính xác, dư thừa, phỏng chừng, thiếu nhất quán
 - Các yêu cầu thiếu tính khả thi
- ❖ **Do vậy**
 - Bất kỳ tiến trình phát triển nào đều bắt đầu từ thu thập và phân tích yêu cầu
- ❖ **Các hoạt động trong SDP và các kết quả liên quan hình thành pha đầu tiên của tiến trình và gọi nó là Phân tích yêu cầu**

Thu thập và phân tích yêu cầu

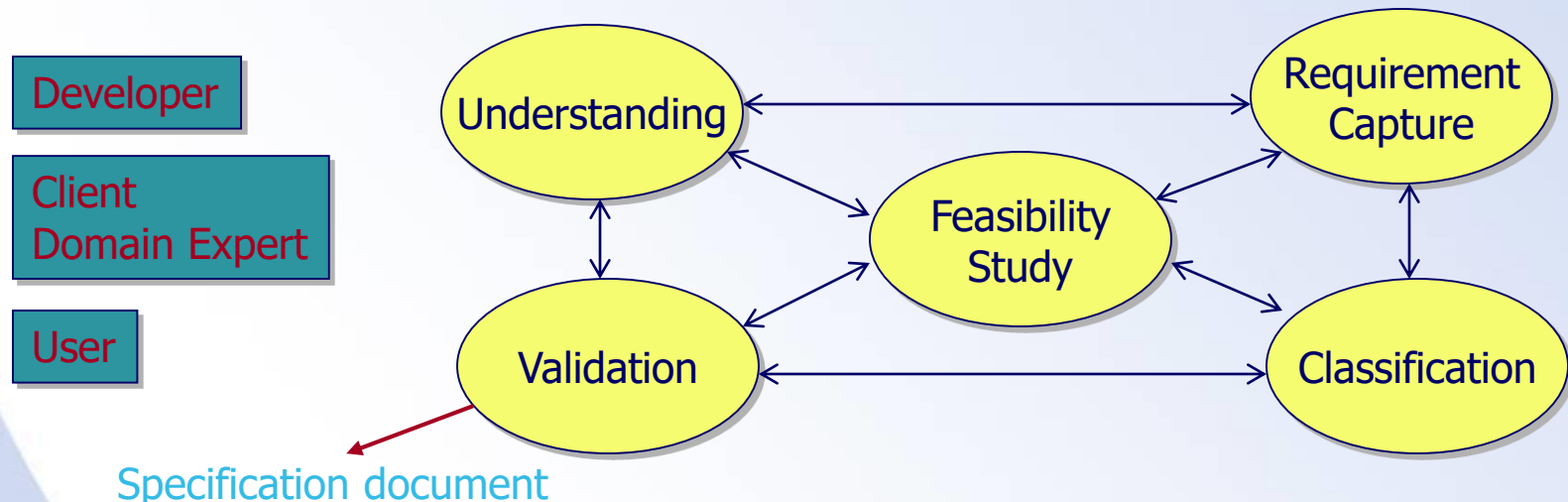
❖ Mục tiêu

- Hình thành tài liệu đặc tả yêu cầu (Requirement Specification)

❖ Tài liệu đặc tả yêu cầu được sử dụng như

- Cam kết giữa khách hàng và tổ chức phát triển hệ thống về cái mà hệ thống có thể làm (và cái mà hệ thống không thể làm)
- Cơ sở để đội ngũ phát triển phát triển hệ thống
- Mô hình tương đối đầy đủ về cái hệ thống đòi hỏi

❖ Tiến trình phân tích yêu cầu bao gồm các hoạt động lặp



Các hoạt động của phân tích yêu cầu

❖ **Hiểu lĩnh vực vấn đề (Understanding)**

- Phân tích viên trình bày hiểu biết về lĩnh vực vấn đề
- Khám phá các quan niệm
- Suy ra các yêu cầu khách hàng

❖ **Thu thập yêu cầu (Requirement Capture)**

- Phân tích viên cần có cách thu thập nhu cầu khách hàng sao cho họ có thể cùng tham gia vào dự án
- Phân tích viên, khách hàng, chuyên gia lĩnh vực ứng dụng và người sử dụng hệ thống cùng phát hiện và thu thập yêu cầu
- Kỹ năng **trừu tượng** là rất quan trọng để thu thập những cái chính, bỏ qua cái không cần thiết

❖ **Phân lớp**

❖ **Đánh giá**

❖ **Nghiên cứu khả thi**

Các hoạt động của phân tích yêu cầu

❖ Hiểu lĩnh vực vấn đề

❖ Thu thập yêu cầu

❖ Phân lớp (Classification)

- Đầu vào của hoạt động này là tập hợp phi cấu trúc của các yêu cầu thu thập được trong pha trước để tổ chức chúng thành các nhóm dính liền nhau
- Gắn mức ưu tiên cho các yêu cầu theo tầm quan trọng của chúng đối với khách hàng và người sử dụng

❖ Đánh giá (Validation)

- Kiểm tra xem các yêu cầu có nhất quán và đầy đủ
- Giải quyết các mâu thuẫn giữa các yêu cầu

❖ Nghiên cứu khả thi (Feasibility study)

- Dự báo khả năng thỏa mãn sử dụng phần cứng, phần mềm của các yêu cầu đã nhận ra
- Quyết định các bước tiếp theo nếu hệ thống đề xuất có hiệu quả

Phân tích yêu cầu

- ❖ **Khi nào kết thúc phân tích yêu cầu?**
 - Không có quy luật nhất định
- ❖ **Để tiến tới bước phát triển phần mềm tiếp theo hãy trả lời các câu hỏi sau:**
 - Khách hàng, người sử dụng cuối cùng và người phát triển đã hiểu trọn vẹn hệ thống?
 - Mô hình của hệ thống đòi hỏi xây dựng đã được hình thành đầy đủ?
 - có đầy đủ các chức năng (dịch vụ)
 - có đầy đủ đầu vào- đầu ra
 - cần loại dữ liệu nào
- ❖ **Chú ý: Chưa mô tả quyết định cài đặt nào ở mô hình này**
- ❖ **Đặc tả yêu cầu và mô hình của hệ thống tại mức này cần phải được hiệu chỉnh, bổ sung khi cần thiết trong các pha phát triển tiếp theo.**

Phân tích yêu cầu

❖ Đặc tả yêu cầu

- là thông báo chính thức cái đòi hỏi hệ thống phải được phát triển
- Nó không phải là tài liệu thiết kế

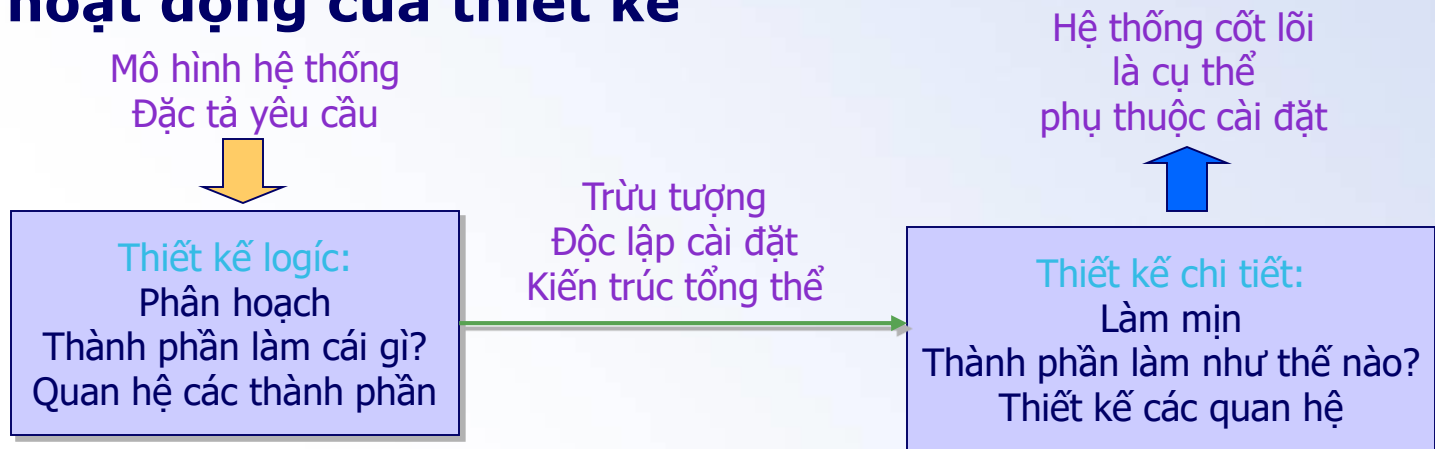
❖ Mô tả đặc tả yêu cầu

- Ngôn ngữ đặc tả
- Ký pháp đồ họa

Pha thu thập và phân tích yêu cầu rất quan trọng.
Nếu không phát hiện ra lỗi tại pha này thì rất khó
và tốn kém để phát hiện ra nó ở pha tiếp theo.

Thiết kế hệ thống

- ❖ **Sau khi có đặc tả yêu cầu, hai tiến trình thiết kế hệ thống tiếp theo**
 - Thiết kế kiến trúc (logic)
 - Phân hoạch các yêu cầu thành các thành phần
 - Tài liệu thiết kế kiến trúc mô tả mỗi thành phần cần làm gì và chúng tương tác với nhau như thế nào để hình thành các chức năng hệ thống
 - Thiết kế chi tiết (vật lý)
 - Thiết kế từng thành phần
 - Tài liệu thiết kế chi tiết mô tả **mỗi thành phần** và cả **hệ thống** phải làm cái nó cần làm như thế nào
- ❖ **Các hoạt động của thiết kế**



Thiết kế hệ thống

❖ Tài liệu của pha thiết kế kiến trúc là mô hình kiến trúc

- Đặc tả thành phần, mô tả cái mà thành phần phải làm bằng cách chỉ ra giao diện giữa các thành phần
- Mô hình hệ thống ở đây chủ yếu mô tả “what”, ít mô tả “how”

❖ Thiết kế chi tiết thực hiện nhiều bước làm mịn mô hình kiến trúc

❖ Mô hình thiết kế chi tiết mô tả:

- thiết kế chức năng của mỗi thành phần
- thiết kế giao diện của mỗi thành phần

❖ Mô hình hệ thống tại mức này được xem như hệ thống cốt lõi

- nó là cụ thể
- phụ thuộc cài đặt
- xác định “How”

- ❖ **Mỗi thành phần trong pha thiết kế được hiện thực thành một modun chương trình**
- ❖ **Kiểm chứng hay kiểm thử mỗi modun chương trình theo đặc tả có từ pha thiết kế**

Tích hợp và kiểm thử hệ thống

- ❖ **Tổ hợp các modul chương trình thành hệ thống**
- ❖ **Kiểm thử hệ thống chương trình để đảm bảo đáp ứng đầy đủ yêu cầu**
- ❖ **Khi người phát triển thỏa mãn với sản phẩm**
 - khách hàng kiểm thử hệ thống
- ❖ **Pha này kết thúc khi khách hàng chấp nhận sản phẩm**

Bảo trì hệ thống

- ❖ Pha này bắt đầu khi hệ thống được cài đặt sử dụng thực tế, sau khi đã cấp phát sản phẩm cho khách hàng
- ❖ Bảo trì bao gồm mọi thay đổi sản phẩm để khách hàng đồng ý rằng họ đã thỏa mãn với sản phẩm.
- ❖ **Bảo trì bao gồm**
 - sửa phần mềm
 - loại bỏ các lỗi mà không phát hiện trong các pha trước đó
 - nâng cấp phần mềm
 - Hiệu năng: Bổ sung chức năng, tăng tốc độ thực hiện chương trình
 - Thích nghi: Các thay đổi cho phù hợp với môi trường phần mềm hoạt động thay đổi, thí dụ yêu cầu mới của chính phủ
- ❖ **Thời gian trung bình:**
 - sửa lỗi 17,5%, hiệu năng 60%, thích nghi 18%.