

# PHÂN TÍCH THIẾT KẾ HỆ THỐNG THÔNG TIN

## XÁC ĐỊNH YÊU CẦU NGƯỜI DÙNG

Giảng viên: TS. Trần Mạnh Tuấn

Email: [tmtuan@tlu.edu.vn](mailto:tmtuan@tlu.edu.vn)



# NỘI DUNG

- Tổng quan về yêu cầu phần mềm
- Vai trò của yêu cầu phần mềm
- Xác định yêu cầu phần mềm
- Các đặc điểm cần lưu ý



# QUY TRÌNH PHÁT TRIỂN DỰ ÁN PHẦN MỀM



# Quy trình phát triển dự án phần mềm (SDLC - Software Development Lifecycle )



# Các nhân tố tham gia vào dự án phần mềm (Các bên liên quan - Stakeholders)

- Khách hàng;
- Người dùng cuối;
- Những người phát triển;
- Những người kiểm thử;
- Những người bảo trì;



# Các nhân tố tham gia vào dự án phần mềm (Các bên liên quan - Stakeholders)

- Người đảm bảo chất lượng;
- Người quản trị Cơ sở dữ liệu;
- Người quản lý cấu hình;
- Những nhà cung cấp;



# Các nhân tố tham gia vào dự án phần mềm (Các bên liên quan - Stakeholders)

- Những nhà tiếp thị;
- Người quản lý;
- Các cơ quan quy định tính an toàn hệ thống (Safety regulation agencies).



# Ví dụ về yêu cầu

**A (chủ nhà):** Tôi muốn có 1 chiếc xích đu trong vườn để bọn trẻ có thể chơi đùa dưới gốc cây.

**B (Kiến trúc sư):** Tôi sẽ hoàn thành công việc này trong 2 tuần tới.





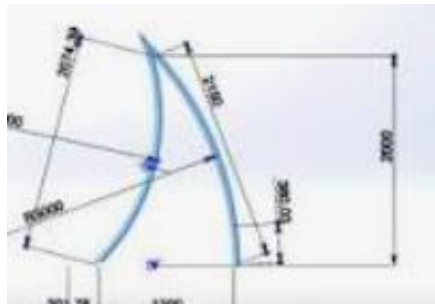
# Ví dụ về yêu cầu – Kiến trúc sư nghĩ



nhưng  
nào sẽ  
phù  
hợp?



# Ví dụ về yêu cầu – Kiến trúc sư làm việc



# Ví dụ về yêu cầu – Sản phẩm



# Ví dụ về yêu cầu – Khách hàng muốn





# Ví dụ về yêu cầu – So sánh



Sản phẩm



Suy nghĩ của khách hàng



# Trao đổi

- Kiến trúc sư đã **hiểu rõ yêu cầu** của khách hàng?
- Trong quá trình thực hiện, có **sự tham gia của khách hàng**?
- Sản phẩm cuối cùng so với mong muốn/suy nghĩ của khách hàng?



# Tổng quan về yêu cầu phần mềm

- Yêu cầu cho 1 phần mềm cụ thể là **tổng hợp những yêu cầu từ nhiều đối tượng khác nhau** về tổ chức, mức độ chuyên môn và mức độ tham gia, tương tác với phần mềm trong môi trường hoạt động của nó.
- **Có thể kiểm chứng** một cách riêng rẽ ở mức chức năng(các yêu cầu chức năng) hoặc mức hệ thống (các yêu cầu bổ sung)

# Tổng quan về yêu cầu phần mềm

- Cung cấp các chỉ số đánh giá độ ưu tiên về các mặt khi cân nhắc về nguồn tài nguyên.
- Cung cấp các giá trị trạng thái để theo dõi tiến độ của dự án.
- Cần xác định được đầu vào của hệ thống là gì



# Tổng quan về yêu cầu phần mềm

- Những **quá trình cần xử lý trong hệ thống**, hay hệ thống phần mềm sẽ phải **xử lý những gì**.
- Xác định được đầu ra (kết quả xử lý) của hệ thống là gì
- Những **ràng buộc** trong hệ thống, chủ yếu là mối quan hệ giữa đầu vào và đầu ra như thế nào

# ĐỊNH NGHĨA YÊU CẦU PHẦN MỀM (REQUIREMENTS ENGINEERING)

Không phải một thời kỳ hay một giai đoạn !

Truyền đạt rất quan trọng khi phân tích

Chất lượng nghĩa là đáp ứng mục tiêu. Không thể nói điều gì về chất lượng trừ khi bạn hiểu rõ mục tiêu

Cần nhận dạng tất cả các đối tác – không chỉ là người dùng và khách hàng !

**Requirements Engineering (RE)** là một

**tập các hoạt động** liên quan tới

**việc xác định và truyền đạt**

**mục tiêu** của một hệ thống phần mềm

chuyên nghiệp, trong **lĩnh vực** mà

chúng được sử dụng. Ở đây, các hoạt

động RE như là cầu nối giữa

**các nhu cầu trong thực tế** của

người dùng, khách hàng, và những

**ứng viên** khác có ảnh hưởng đến một

hệ thống phần mềm, và **những khả**

**năng và cơ hội** được tạo ra bởi những

kỹ thuật phần mềm chuyên nghiệp

Người thiết kế cần biết hệ thống sẽ được sử dụng ở đâu và như thế nào?

Yêu cầu là một phần của ... nhu cầu là gì ???

Và một phần của ... nó thực hiện được gì ???

# Mục đích của phân tích yêu cầu

- Tìm ra được những cái mà hệ thống phần mềm phải có; những giới hạn của phần mềm và cách phần mềm tương tác với tổ chức và môi trường hoạt động của nó.
- Phát hiện và giải quyết mâu thuẫn giữa các yêu cầu



# Phân loại yêu cầu

- ✓ Yêu cầu chức năng: đặc tả các chức năng mà phần mềm phải thực hiện.
- ✓ Yêu cầu phi chức năng: là các ràng buộc về giải pháp và chất lượng(hiệu năng, việc bảo trì, độ an toàn, bảo mật,...).
- ✓ Yêu cầu đặc tả các thuộc tính nổi bật: là các đặc tả cho các thuộc tính phụ thuộc vào sự vận hành,... đặc biệt là kiến trúc hệ thống. Các thuộc tính này không thể xác định được cho từng thành phần đơn lẻ.

# Phân loại yêu cầu

- ✓ **Yêu cầu chức năng:** đặc tả các chức năng mà phần mềm phải thực hiện.
- ✓ Yêu cầu phi chức năng: là các ràng buộc về giải pháp và chất lượng (hiệu năng, việc bảo trì, độ an toàn, bảo mật,...).
- ✓ Yêu cầu đặc tả các thuộc tính nổi bật: là các đặc tả cho các thuộc tính phụ thuộc vào sự vận hành,... đặc biệt là kiến trúc hệ thống. Các thuộc tính này không thể xác định được cho từng thành phần đơn lẻ.

# PHÂN BIỆT PHÂN TÍCH YÊU CẦU PM VÀ PHÂN TÍCH THIẾT KẾ HT

➤ **Yêu cầu:** Trả lời câu hỏi **What**

What: Cái mà hệ thống nên làm

➤ **Thiết kế:** Trả lời câu hỏi **How**

How: Hệ thống làm điều đó như thế nào.

# Các đặc tính của một yêu cầu tốt

- ✓ Có thể hiểu
- ✓ Khả thi (hiện thực, có thể thực hiện)
- ✓ Cần thiết
- ✓ Độc lập với cài đặt (trừu tượng)



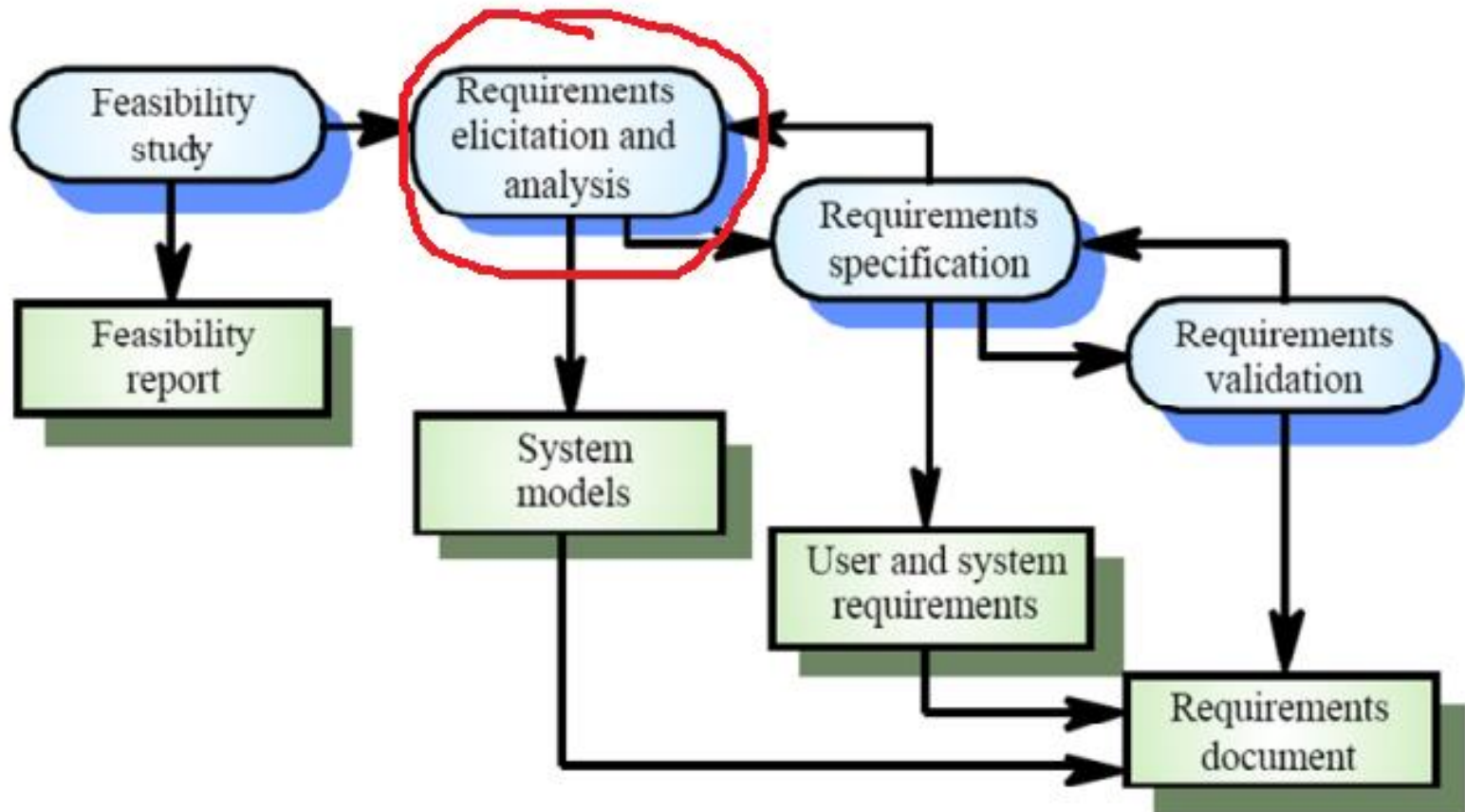
# Các đặc tính của một yêu cầu tốt

- ✓ Không mập mờ (Rõ ràng)
- ✓ Có thể kiểm thử (có thể thẩm định)
- ✓ Ngắn gọn, súc tích, đơn giản
- ✓ Đúng đắn






# Các bước trong quy trình



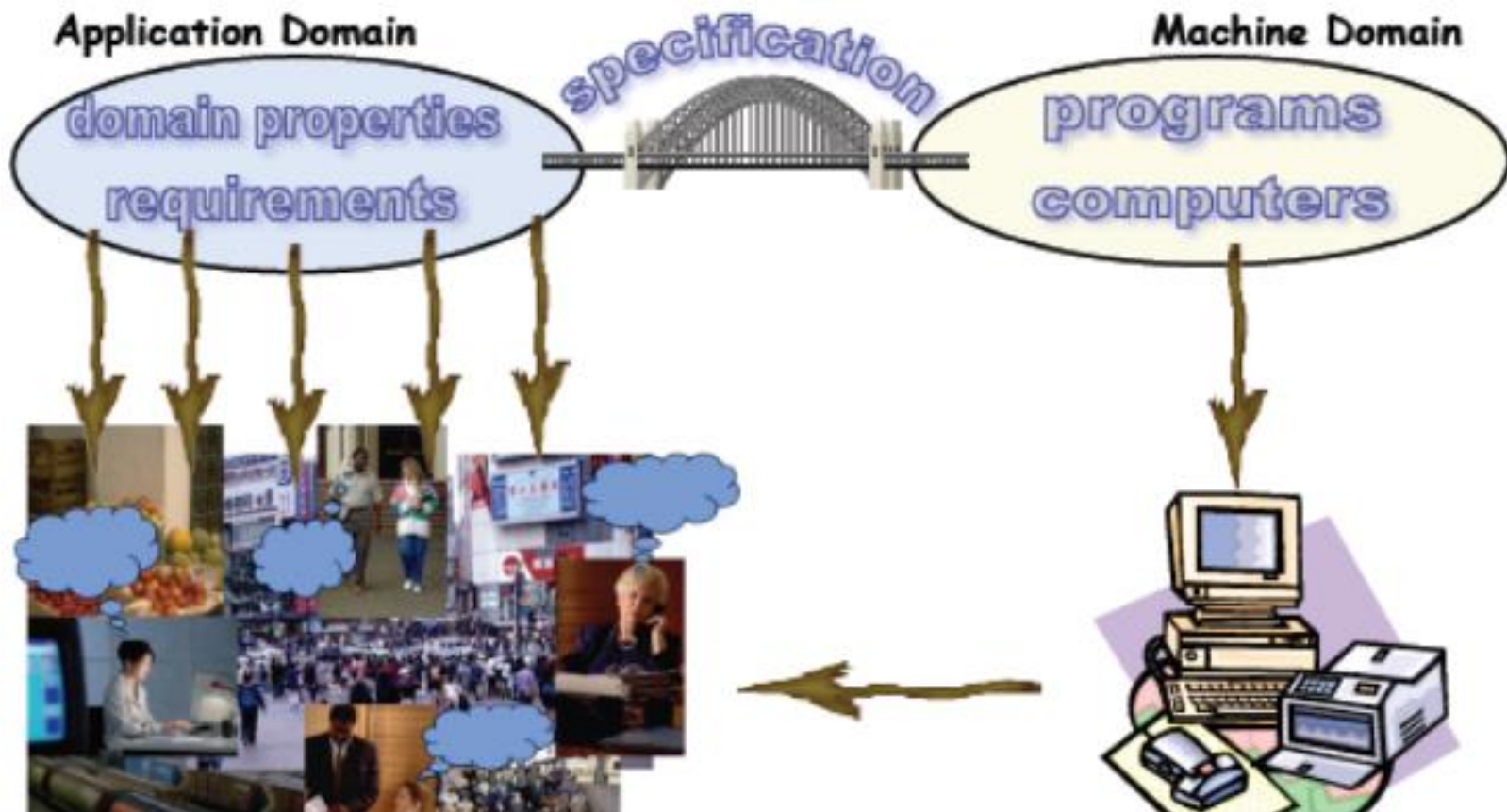
# NGHIÊN CỨU KHẢ THI

- ✓ **Khả thi về kinh tế:** chi phí và lợi nhuận
  - ✓ **Khả thi về kỹ thuật:** tài nguyên sẵn có, công nghệ, ...
  - ✓ **Khả thi về hợp pháp:** có sự xâm phạm, vi phạm hay khó khăn nào gây ra khi xây dựng hệ thống hay không
  - ✓ **Các phương án:** đánh giá về phương án tiếp cận đến việc xây dựng hệ thống.
- 

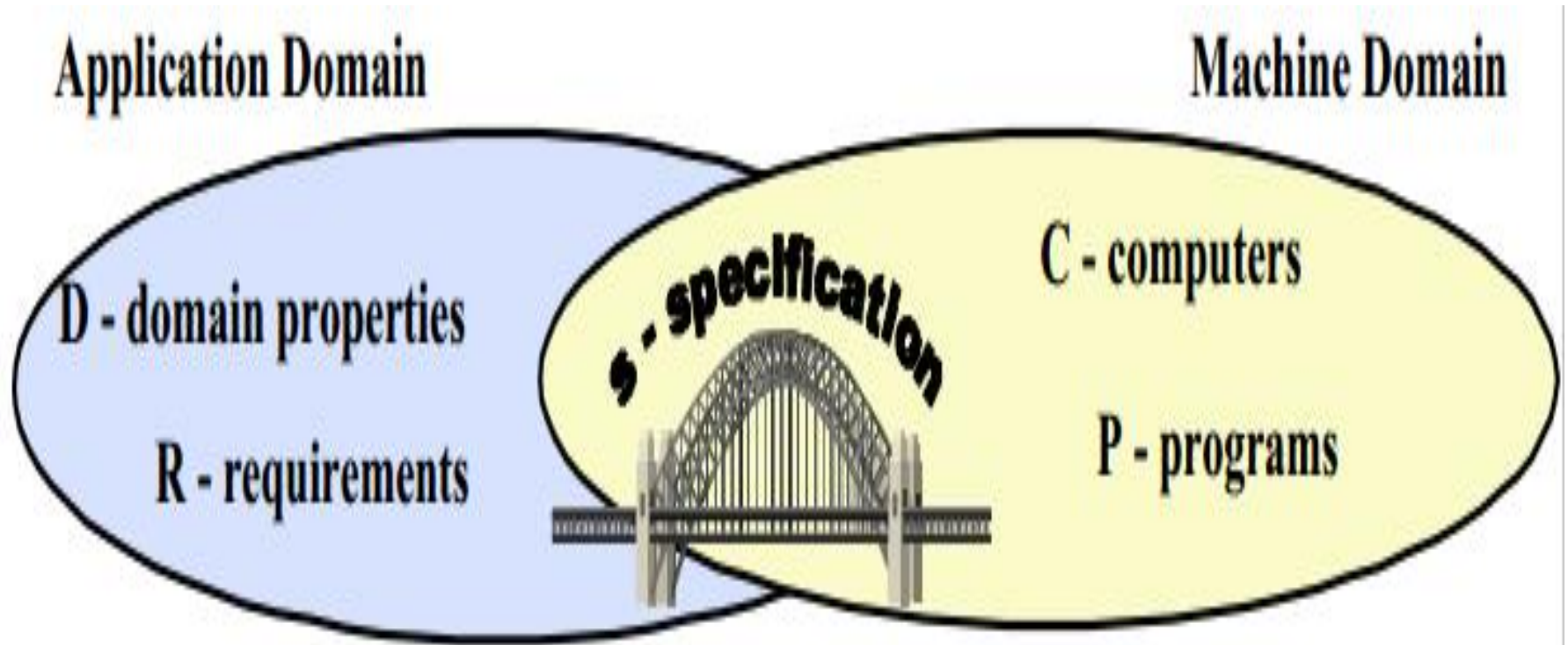
# PHÂN TÍCH YÊU CẦU

- **Đầu vào** của hệ thống là gì
- Những quá trình cần xử lý trong hệ thống, hay hệ thống phần mềm sẽ phải **xử lý** những cái gì.
- **Đầu ra:** kết quả xử lý của hệ thống là gì
- Những ràng buộc trong hệ thống, chủ yếu là **mối quan hệ giữa đầu vào và đầu ra** như thế nào

# ĐẶC TẢ YÊU CẦU PHẦN MỀM



# CÁC KHÁI NIỆM CƠ BẢN

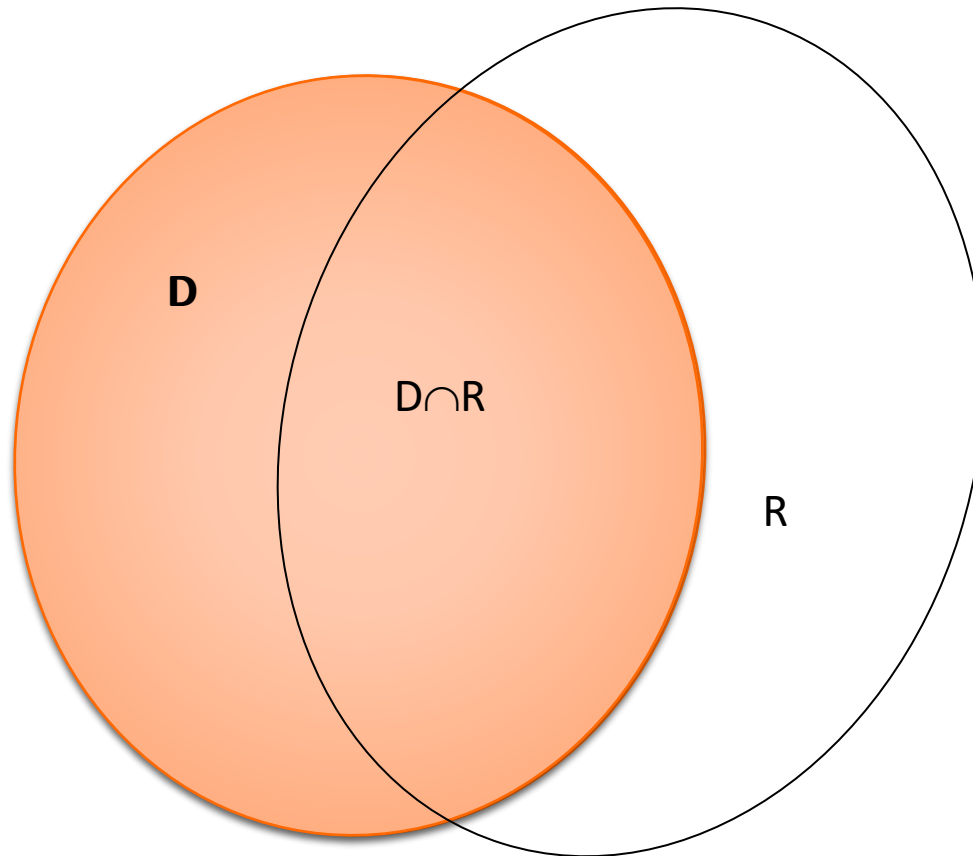


# CÁC KHÁI NIỆM CƠ BẢN

- **D – Đặc tính lĩnh vực:** những thứ có thật trong lĩnh vực ứng dụng dù chúng ta có thiết kế chúng trong hệ thống hay không.
- **R – Các yêu cầu:** những thứ trong lĩnh vực ứng dụng mà chúng ta mong muốn trở thành hiện thực bằng cách đưa vào hệ thống.



# MÔ TẢ MIỀN D VÀ MIỀN R



# CÁC KHÁI NIỆM CƠ BẢN

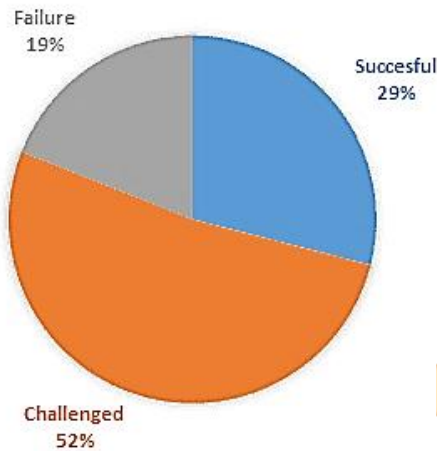
- **S** – Đặc tả: mô tả các hành vi chương trình cần thực hiện để đáp ứng với các yêu cầu
- **C**- Máy tính: Môi trường để hệ thống được vận hành.
- **P** – Chương trình: sản phẩm (hệ thống) mà dự án phải đáp ứng





# DỰ ÁN PHẦN MỀM

SUCCESS RATE



## Project Success Factors

There are obviously many reasons for the failure of software development projects. From the customer perspective, the following 3 reasons are most attributed to a successful software development project.

- User involvement from the customer
- Senior management support
- Clear statement of requirements from the outset



# HẬU QUẢ CỦA SAI SÓT

↳ Lỗi yêu cầu (requirements errors) có thể phải trả giá đắt nếu chúng không được phát hiện và sửa chữa sớm trong tiến trình phát triển.

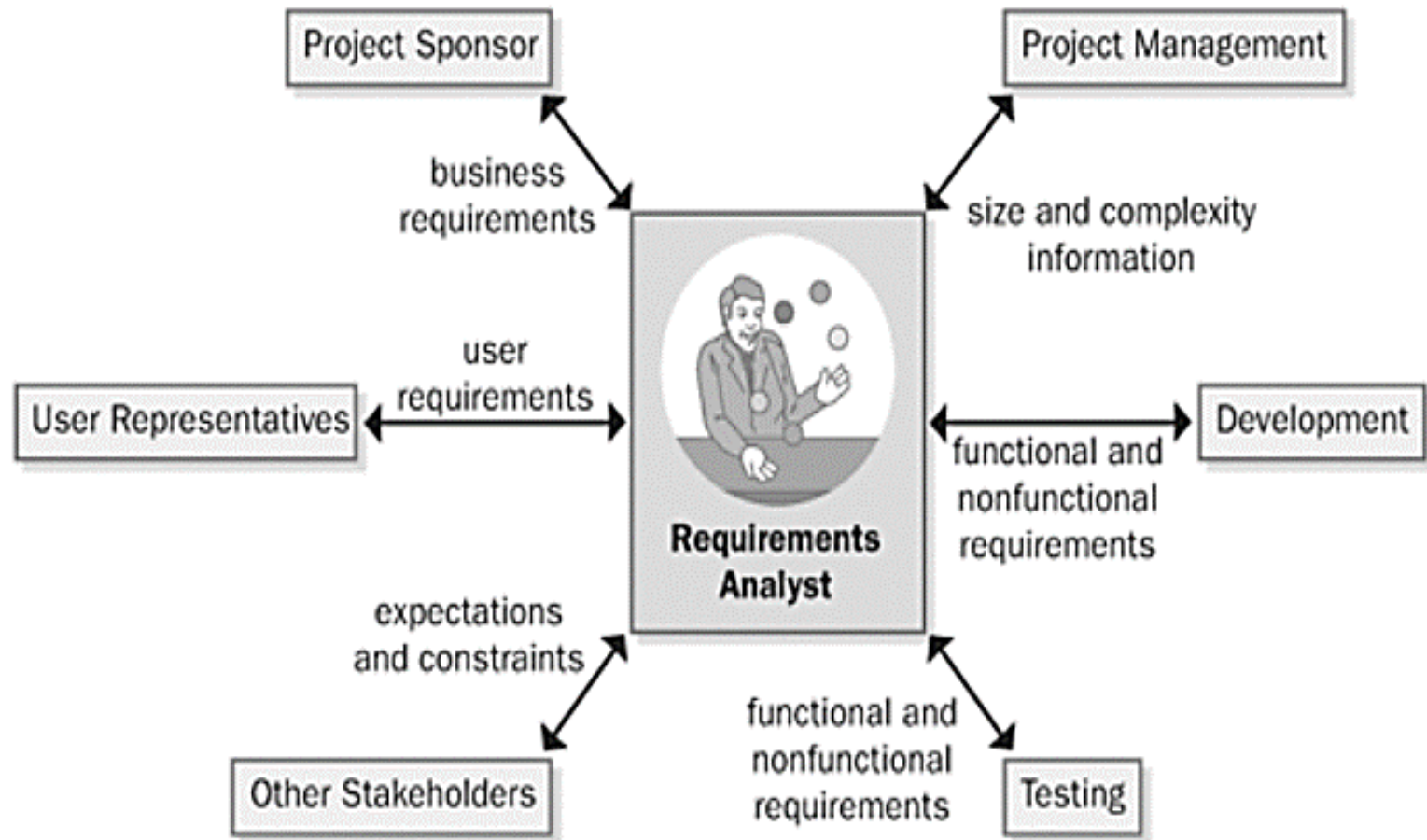
↳ Báo cáo của Boehm và Papaccio (1988) cho thấy ước lượng giá trị tiêu tốn cho việc phát hiện lỗi ở các giai đoạn của một tiến trình phát triển phần mềm như sau :

Phân tích yêu cầu (1\$)  $\Rightarrow$  Thiết kế (5\$)  $\Rightarrow$  Lập trình (10\$)  
 $\Rightarrow$  Kiểm thử (20\$)  $\Rightarrow$  Triển khai hệ thống (>200\$)

→ Cần dành thời gian để tìm hiểu kỹ vấn đề trong lĩnh vực của chúng và thu thập yêu cầu thật chính xác trong giai đoạn đầu tiên.



# VAI TRÒ CỦA NHÀ PHÂN TÍCH YÊU CẦU




# Các kỹ thuật thu thập, phân tích yêu cầu

- Phỏng vấn (đối thoại riêng lẻ với stakeholder).
- Bản câu hỏi thăm dò (một tập các câu hỏi gửi đến một tập hợp lớn hơn các stakeholder).
- Hội thảo (Workshop - các stakeholders tập hợp để tập trung thảo luận).
- Thẻ sự kiện (sử dụng công cụ đồ họa, trực quan để minh họa hành vi hệ thống).
- Phân vai (mỗi thành viên trong nhóm được gán một vai, thường là một trong các người dùng cuối).



# Các kỹ thuật thu thập, phân tích yêu cầu

- Các phiên làm việc tập trung (story boarding - trình bày các ý tưởng trong phiên tập trung ngắn).
  - Sử dụng các biểu đồ quan hệ.
  - Mẫu thử (phát triển một mẫu thử để có sự phản hồi).
  - Các ca sử dụng (sự tương tác giữa người dùng và hệ thống được trình bày như một chuỗi các bước).
- 

# Các kỹ thuật thu thập, phân tích yêu cầu

- Phân tích các tài liệu hiện có (trích rút thông tin từ các tài liệu word, các email, ...).
- Quan sát và minh họa nhiệm vụ (xem người dùng thực hiện nhiệm vụ cụ thể).
- Phân tích các hệ thống đang tồn tại (thu thập các yêu cầu từ hệ thống bị thay thế hoặc từ các hệ thống cạnh tranh đã được xây dựng).



# PHỎNG VẤN

## ➤ ***Ưu điểm:***

- Linh hoạt, hiệu quả cao, nhận được câu trả lời tức thời với độ tin cậy cao.
- Có thể mở rộng phạm vi câu hỏi để nhận được yêu cầu chi tiết, khai thác được các yêu cầu phi chức năng.

## ➤ ***Nhược điểm:***

- Khó thực hiện, Kết quả phụ thuộc nhiều vào người thực hiện phỏng vấn.

# BẢN CÂU HỎI THĂM DÒ (QUESTIONNAIRE)

## ➤ *Ưu điểm:*

- Dễ thực hiện,
- Tiếp cận đến một lượng lớn các stakeholders,

## ➤ *Nhược điểm:*

- Cấu trúc bộ câu hỏi không linh hoạt
- Thời gian chờ phản hồi lâu
- Độ tin cậy của các câu trả lời phụ thuộc nhiều vào sự hợp tác của stakeholder



# HỘI THẢO YÊU CẦU (WORKSHOP)

## ➤ **Ưu điểm:**

- Thu thập được nhiều yêu cầu trong thời gian diễn ra hội thảo
- Dễ dàng giải quyết các mâu thuẫn giữa các yêu cầu

## - **Nhược điểm:**

- Cần nhiều sự chuẩn bị
- Buổi hội thảo cần được kiểm soát tốt bởi người tổ chức.

# PHÂN VAI (ROLE PLAYING)

## ➤ ***Ưu điểm:***

- Dễ thực hiện,
- Có thể triển khai linh hoạt

## ➤ ***Nhược điểm:***

- Các yêu cầu đưa ra có thể không khách quan
- Người thực hiện không có chuyên môn về lĩnh vực trong miền ứng dụng

# CÁC PHIÊN LÀM VIỆC TẬP TRUNG

## ➤ ***Ưu điểm:***

- Dễ thực hiện,
- Giải quyết được các vấn đề phát sinh trong quá trình thực hiện

## ➤ ***Nhược điểm:***

- Thường chỉ được áp dụng trong các tình huống cần thiết



# PHÂN TÍCH CÁC TÀI LIỆU HIỆN CÓ

## ➤ ***Ưu điểm:***

- Dễ thực hiện,
- Nguồn tài liệu lớn

## ➤ ***Nhược điểm:***

- Độ tin cậy của các tài liệu cần được kiểm chứng
- Các tài liệu thu thập được đôi khi lạc hậu



# QUAN SÁT VÀ MÔ PHỎNG NHIỆM VỤ

## ➤ ***Ưu điểm:***

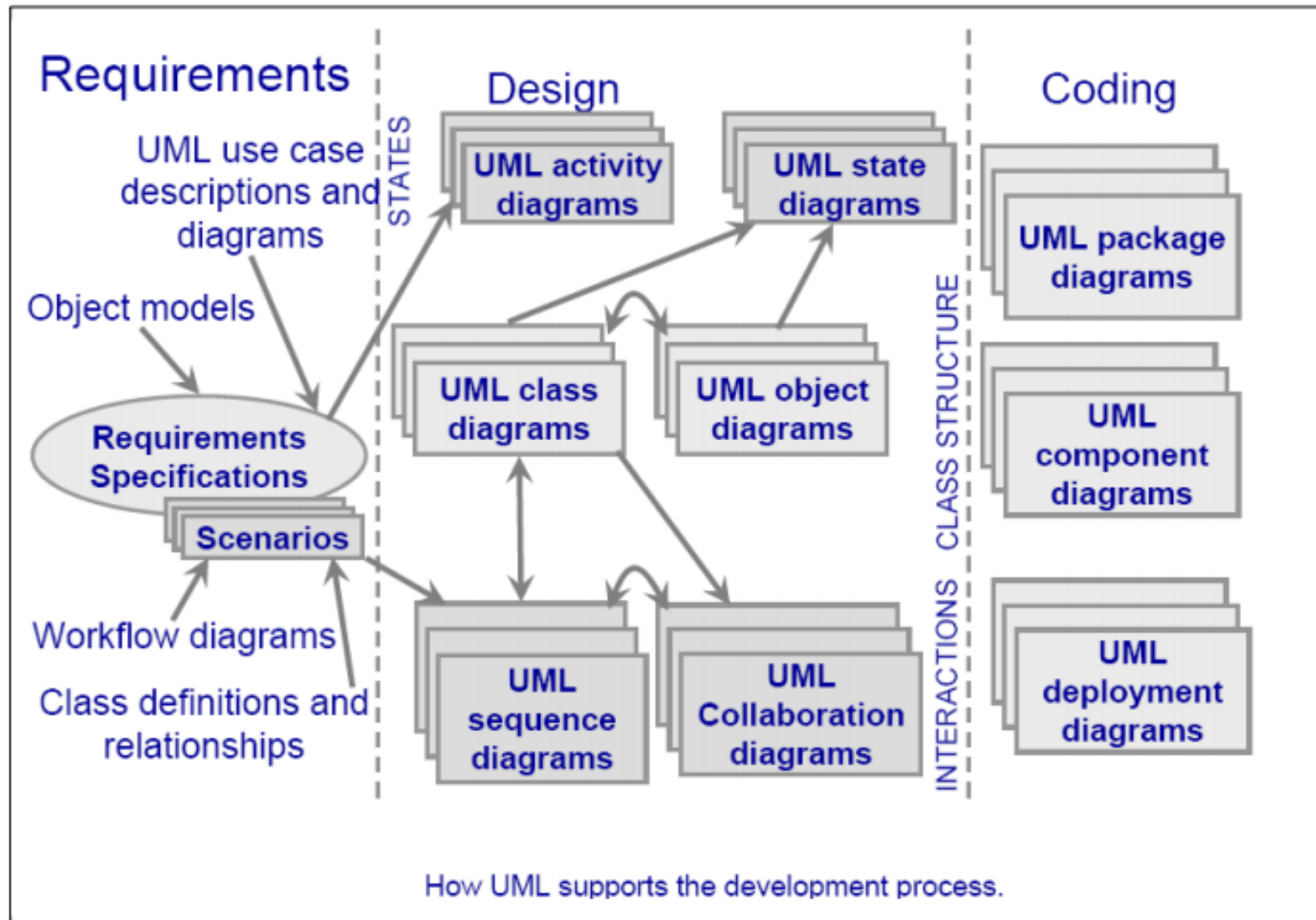
- Khắc phục được các khó khăn khi thu thập yêu cầu bằng các phương pháp khác trong các tình huống cụ thể (nói và làm).

## ➤ ***Nhược điểm:***

- Khó thực hiện trong 1 vài trường hợp,

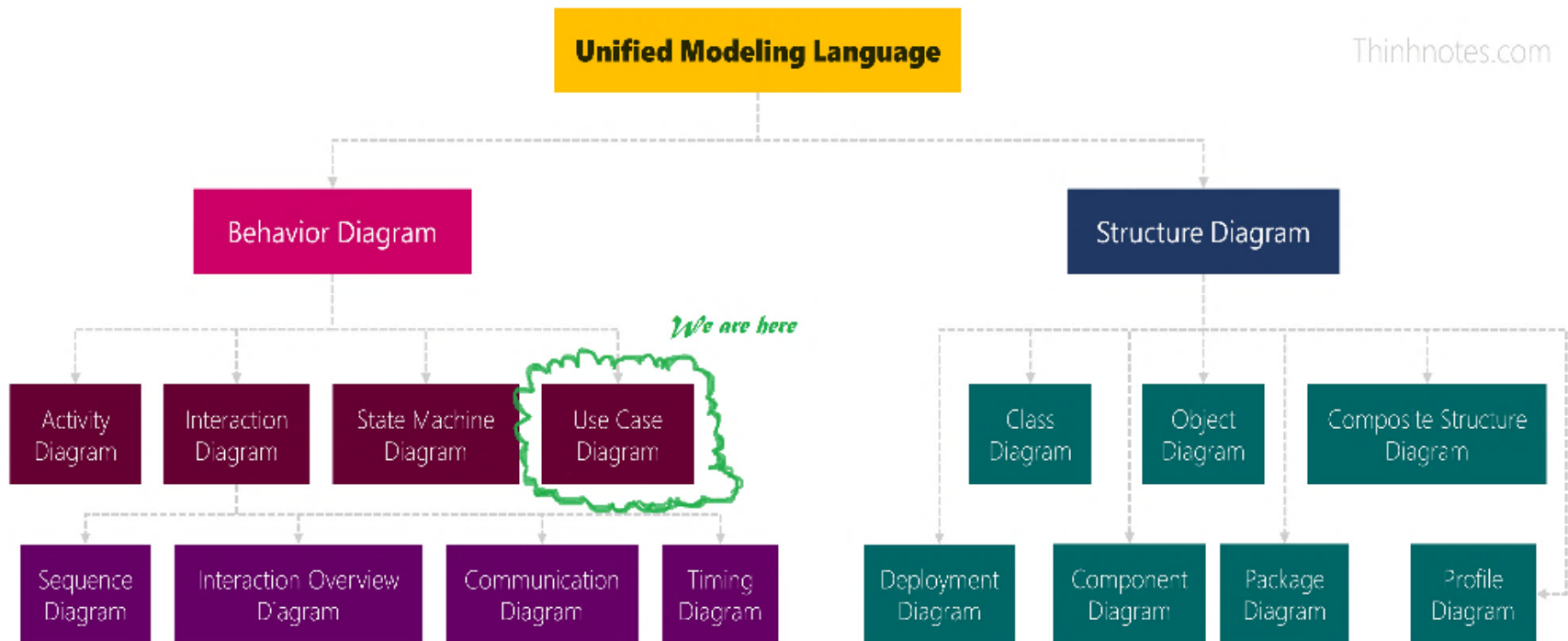


# VAI TRÒ CỦA UML TRONG DỰ ÁN PM



# Use cases (UCs)

Thinnotes.com



# CÁC THÀNH PHẦN TRONG SƠ ĐỒ UC

- Tác nhân (Actor)
- Ca sử dụng (Use Case)
- Các liên kết (Communication Link)
- Phạm vi (Boundary of System)
- Các mối quan hệ (Relationships)

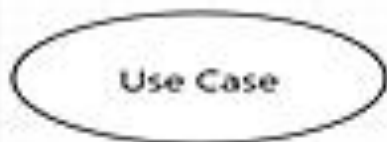




# CÁC THÀNH PHẦN TRONG SƠ ĐỒ UC



**Actor:** có thể là người dùng, hoặc một hệ thống nào khác bất kỳ



**Use Case:**

Communication Link

**Communication Link:** Kết nối giữa Actor và Use Case, cho biết Actor đó có những sự tương tác nào với hệ thống.

**Boundary of System**

**Boundary of System:** Phạm vi của các sự tương tác. Có thể là trong một hệ thống, một module, hoặc một tính năng bất kỳ

<<include>>

<<extend>>

Generalization

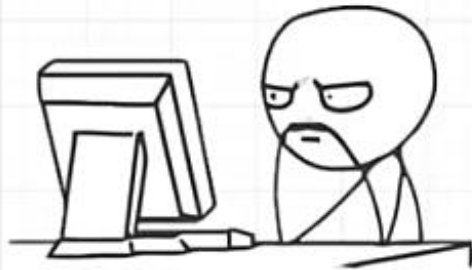
**Relationship:** Mối quan hệ giữa các Use Case với nhau.

# Quan hệ <<extend>> và <<include>>

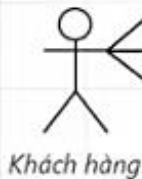
- ✓ Include (included) là mối quan hệ bắt buộc phải có giữa các Use Case với nhau.
- ✓ Extend (extended) là mối quan hệ mở rộng giữa các Use Case với nhau.



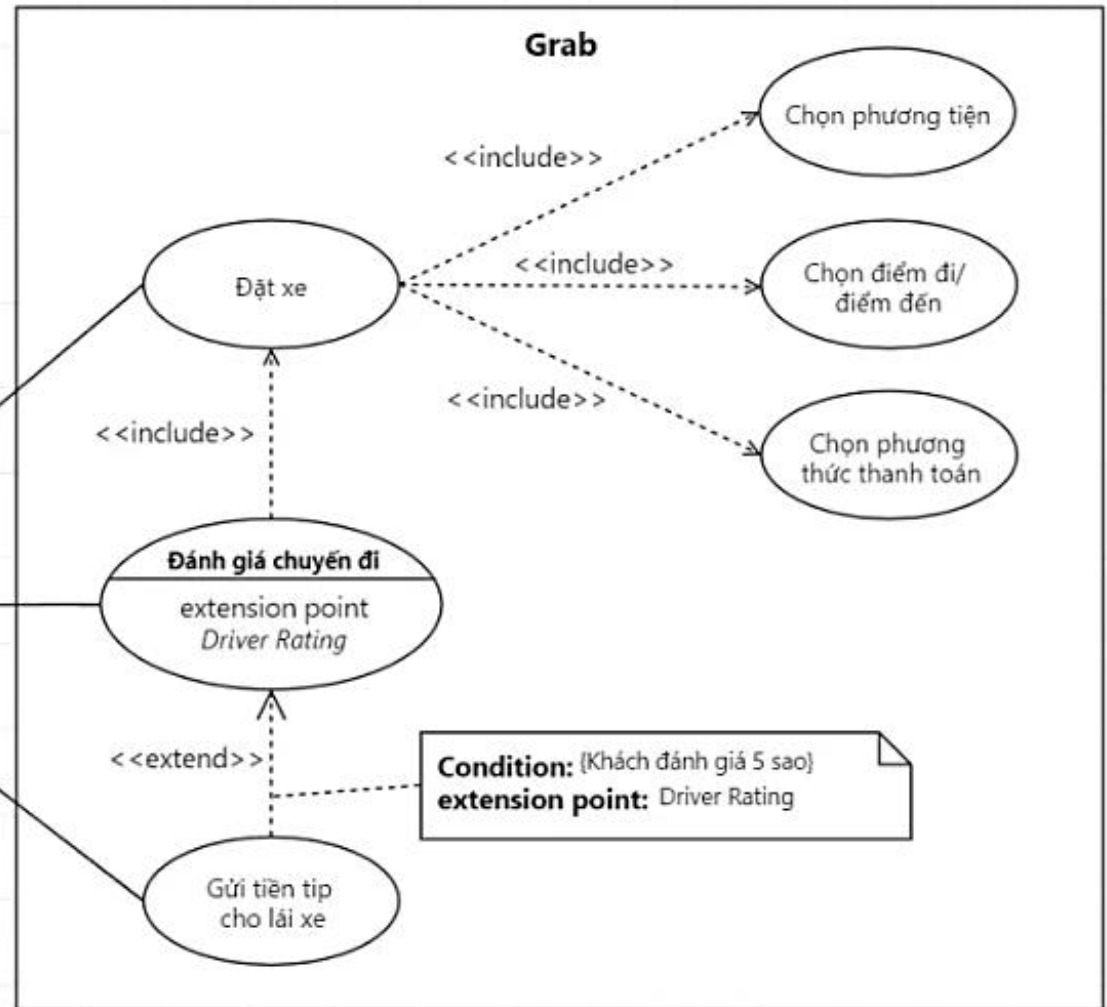
# Quan hệ <<extend>> và <<include>>



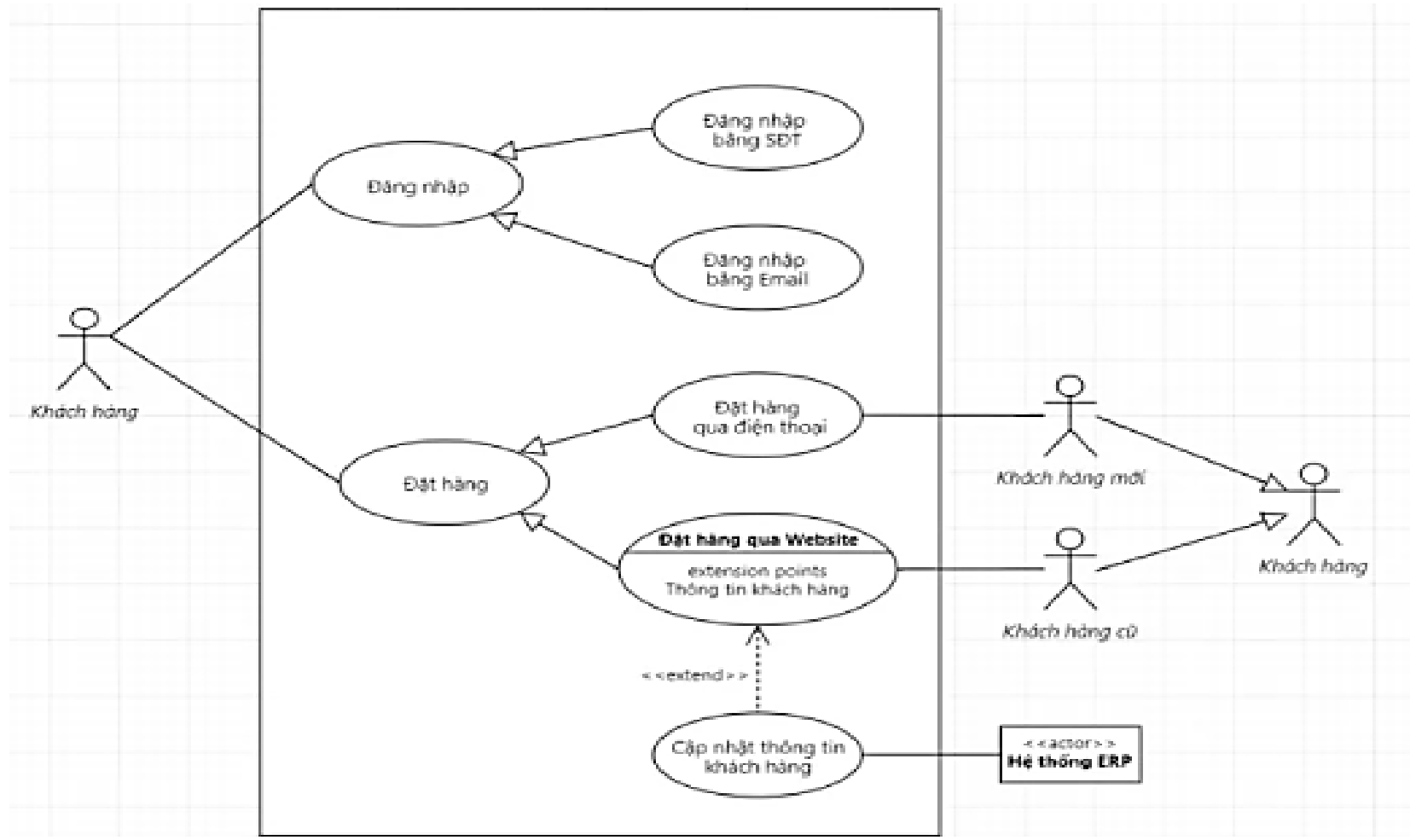
Include, extend???



**USE CASE  
DIAGRAM**



# Quan hệ <<Generalization>>



Ví dụ về quan hệ cha-con (Generalization) trong Use Case.

# MỘT SỐ CHÚ Ý - Đặt tên

- ✓ Tên Actor: là 1 danh từ, không dùng động từ, và không dùng mệnh đề quan hệ
- ✓ Tên Use Case: phải ghi rõ ràng, rành mạch, nên bao gồm Động từ + danh từ
- ✓ Ví dụ:

Tác nhân: Khách hàng, Nhân viên,...

UC là: Chuyển tiền nội địa, Thanh toán đơn hàng,...

# MỘT SỐ CHÚ Ý

