

# Kỹ thuật phần mềm

## **Yêu cầu phần mềm**

# Nội dung chính

---

- Yêu cầu phần mềm là gì?
- Tầm quan trọng của yêu cầu phần mềm trong quá trình phát triển phần mềm
- Công nghệ yêu cầu

# Yêu cầu phần mềm - Requirements

---

- Tiêu chí gì quan trọng nhất đối với chất lượng phần mềm?

*Phần mềm thỏa mãn được yêu cầu của người dùng*

- Yêu cầu phần mềm?

*Những gì người ta muốn có trong phần mềm được phát triển*

# Ví dụ Travel Agency: Yêu cầu người dùng

---

- Hãng du lịch TravelGood đến gặp bạn (người làm phần mềm) và đề nghị làm dự án phần mềm sau:

- Mô tả bài toán / yêu cầu người dùng

*TravelGood muốn cung cấp cho khách hàng của họ một ứng dụng đặt vé và lập kế hoạch du lịch. Ứng dụng này cần cho phép khách lập kế hoạch về các chuyến bay và khách sạn. Đầu tiên, khách hàng có thể sắp xếp một chuyến đi, sau đó đặt vé và đặt phòng khách sạn cho chuyến đi đó. Người dùng có thể lập kế hoạch cho nhiều chuyến đi. Ngoài ra, phần mềm còn cho phép hủy các chuyến đã đặt.*

# Ví dụ Travel Agency: Yêu cầu hệ thống

---

- Sau khi nhận làm phần mềm cho TravelGood đội phát triển chi tiết hóa thành các **yêu cầu hệ thống**:
  1. *Người dùng có thể lập kế hoạch một chuyến đi bằng cách chọn một trình tự các điểm đến, rồi lưu lại. (kèm theo sơ đồ mô tả kịch bản ca sử dụng)*
  2. *Hệ thống cần là ứng dụng Web, chạy được tại tất cả các hệ điều hành và hầu hết các trình duyệt*
  3. *Ứng dụng Web phải triển khai được tại các server tiêu chuẩn như GlassFish hoặc Tomcat*
  4. *Hệ thống phải dễ sử dụng: đạt một test usability (kèm chi tiết cụ thể)*
  5. ...

# Ví dụ khác

---

## Đặc tả yêu cầu người dùng

1. Phần mềm phải cung cấp một phương tiện để biểu diễn và truy nhập các file bên ngoài được tạo bằng các công cụ khác.

## Đặc tả yêu cầu hệ thống

1.1. Người dùng cần được cung cấp tiện ích để định nghĩa kiểu của các file ngoài.

1.2 Mỗi kiểu file ngoài có thể được biểu diễn dưới dạng một biểu tượng trên phần hiển thị của người dùng.

1.3 Mỗi kiểu file ngoài có thể có một công cụ có thể dùng cho loại file đó.

1.4 Cần cung cấp các tiện ích để người dùng có thể định nghĩa biểu tượng cho file ngoài.

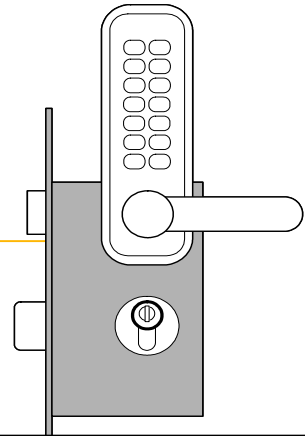
1.5 Khi một người dùng chọn một biểu tượng đại diện cho một file ngoài, hiệu ứng của việc chọn đó là gọi công cụ tương ứng với kiểu của file đó để chạy nó.

# Yêu cầu người dùng / Yêu cầu hệ thống

---

- Yêu cầu người dùng - User requirements
  - Các phát biểu bằng ngôn ngữ tự nhiên cộng với các sơ đồ về các dịch vụ mà hệ thống cung cấp và các ràng buộc về vận hành.
  - Được viết cho khách hàng.
- Yêu cầu hệ thống – System requirements
  - Một tài liệu có cấu trúc bao gồm các mô tả chi tiết về các chức năng và dịch vụ của hệ thống cùng với các ràng buộc về vận hành.
  - Định nghĩa cái gì cần được cài đặt
    - Có thể là một phần của một hợp đồng giữa khách hàng và người nhận thầu.

# Ví dụ yêu cầu hệ thống



Identifier	Priority	Requirement
REQ1	5	The system shall keep the door locked at all times, unless commanded otherwise by authorized user. When the lock is disarmed, a countdown shall be initiated at the end of which the lock shall be automatically armed (if still disarmed).
REQ2	2	The system shall lock the door when commanded by pressing a dedicated button.
REQ3	5	The system shall, given a valid key code, unlock the door and activate other devices.
REQ4	4	Hệ thống sẽ cho phép những sai sót trong khi nhập mã khóa. Tuy nhiên, để chống lại “các cuộc tấn công từ điển”, số lần thử không thành công được phép phải nhỏ, chẳng hạn như ba, sau đó hệ thống sẽ chặn và chuông cảnh báo sẽ vang lên.
REQ5	2	Hệ thống sẽ duy trì nhật ký của tất cả các truy cập đã cố gắng để xem xét sau này.
REQ6	2	Hệ thống nên cho phép thêm những người được ủy quyền mới trong thời gian chạy hoặc xóa những người hiện có.
REQ7	2	Hệ thống sẽ cho phép cấu hình các tùy chọn để kích hoạt thiết bị khi người dùng cung cấp mã khóa hợp lệ, cũng như khi phát hiện có hành vi trộm cắp.
REQ8	1	Hệ thống sẽ cho phép tìm kiếm nhật ký bằng cách chỉ định một hoặc nhiều thông số sau: khung thời gian, vai trò tác nhân, vị trí cửa hoặc loại sự kiện (mở khóa, khóa, mất điện, v.v.). Chức năng này sẽ khả dụng trên Web bằng cách trở trình duyệt đến một URL cụ thể.
REQ9	1	Hệ thống sẽ cho phép gửi các câu hỏi về các truy cập “đáng ngờ”. Chức năng này sẽ có sẵn trên Web.



# User Story

---

As a tenant, I can unlock the doors to enter my apartment.



- Tương tự với yêu cầu hệ thống, nhưng tập trung vào những gì người dùng nhận được từ hệ thống, thay vì các tính năng hệ thống.
- Được sử dụng phổ biến trong các phương pháp Agile.

# Example User Stories

Identifier	User Story	Size
<b>ST-1</b>	As an authorized person (tenant or landlord), I can keep the doors locked at all times.	<b>4 points</b>
<b>ST-2</b>	As an authorized person (tenant or landlord), I can lock the doors on demand.	<b>3 pts</b>
<b>ST-3</b>	The lock should be automatically locked after a defined period of time.	<b>6 pts</b>
<b>ST-4</b>	As an authorized person (tenant or landlord), I can unlock the doors. (Test: Allow a small number of mistakes, say three.)	<b>9 points</b>
<b>ST-5</b>	As a landlord, I can at runtime manage authorized persons.	<b>10 pts</b>
<b>ST-6</b>	As an authorized person (tenant or landlord), I can view past accesses.	<b>6 pts</b>
<b>ST-7</b>	As a tenant, I can configure the preferences for activation of various devices.	<b>6 pts</b>
<b>ST-8</b>	As a tenant, I can file complaint about “suspicious” accesses.	<b>6 pts</b>

# Yêu cầu chức năng / phi chức năng

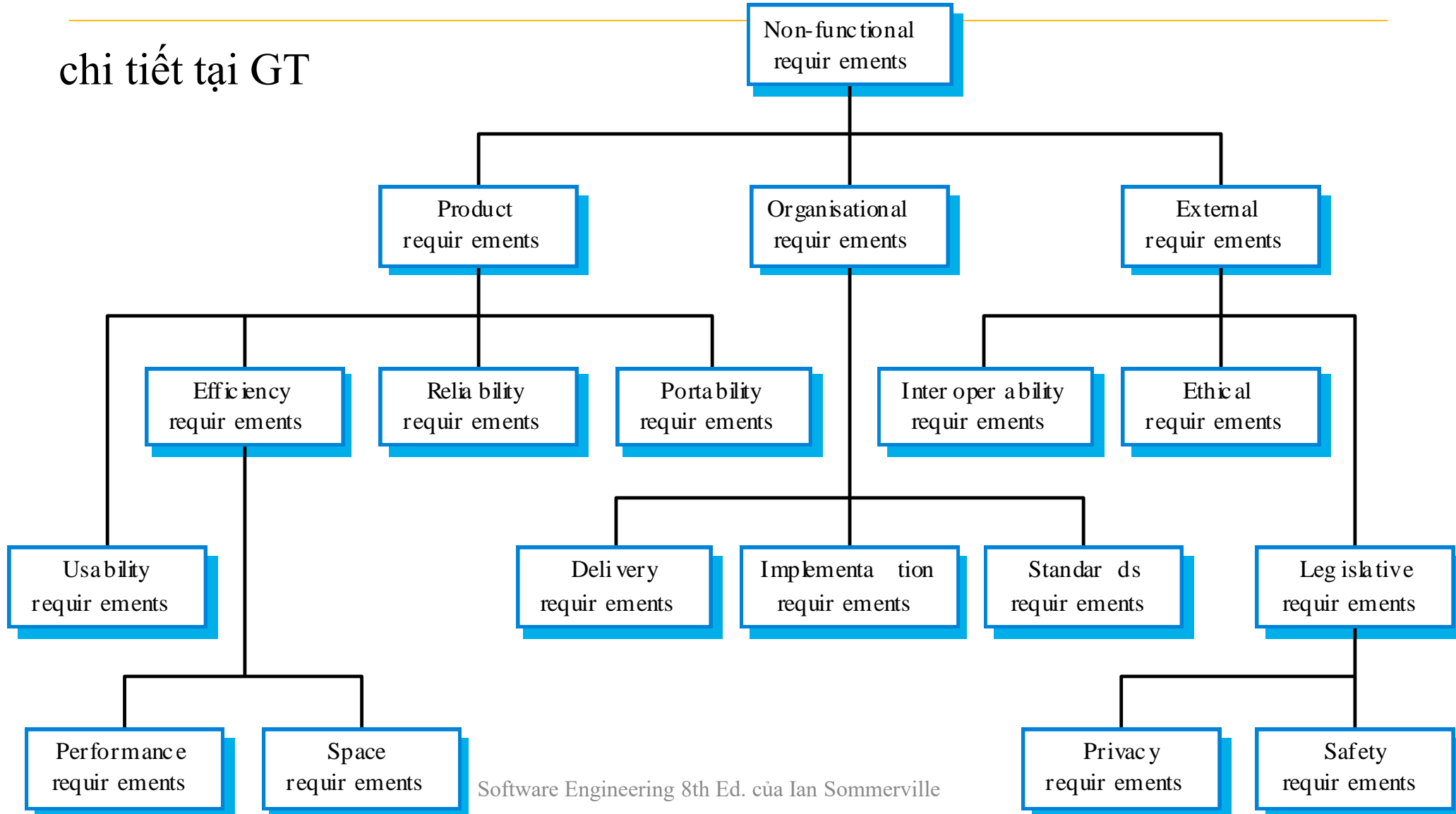
---

- Yêu cầu chức năng – functional requirement:
  - *Người dùng có thể lập kế hoạch một chuyến đi, đặt vé, đặt phòng, lưu một kế hoạch để sau này sẽ đặt vé đặt phòng...*
- Yêu cầu phi chức năng – non-functional requirement:
  - *Hệ thống cần là ứng dụng Web, chạy được tại tất cả các hệ điều hành và hầu hết các trình duyệt*
  - *Ứng dụng Web phải triển khai được tại các server tiêu chuẩn như GlassFish hoặc Tomcat*
  - *Hệ thống phải dễ sử dụng – phải đạt một test usability*

- 
- “Hệ thống cần là ứng dụng Web, chạy được tại tất cả các hệ điều hành và hầu hết các trình duyệt”
    - Không rõ ràng!!!!

# Các loại yêu cầu phi chức năng

chi tiết tại GT



# Yêu cầu chức năng và phi chức năng

---

- Yêu cầu chức năng
  - Phát biểu về các dịch vụ mà hệ thống cần cung cấp,
    - Hệ thống cần phản ứng như thế nào với các input cụ thể
    - Hệ thống cần ứng xử như thế nào trong các tình huống cụ thể.
- Yêu cầu phi chức năng
  - Ràng buộc về các dịch vụ hay chức năng của hệ thống
    - Chẳng hạn ràng buộc về thời gian, về quy trình phát triển, về các chuẩn v.v..

# Đặc điểm của yêu cầu được diễn đạt tốt

---

- Kiểm thử được – testability
  - Test được (thủ công hoặc tự động)
- Đo được
  - Ví dụ về yêu cầu không đo được:
    - *Hệ thống cần dễ sử dụng bởi các nhân viên và cần được tổ chức sao cho người dùng ít làm nhầm nhất*
  - Đo được:
    - *Nhân viên cần sử dụng được toàn bộ các chức năng của hệ thống sau 04 tiếng huấn luyện. Sau huấn luyện, số lỗi trung bình mà một người dùng có kinh nghiệm phạm phải trong mỗi giờ không vượt quá 02 lỗi*

# Các độ đo có thể sử dụng

Đặc điểm	Độ đo
Tốc độ	Số giao dịch được xử lý mỗi giây Thời gian đáp ứng mỗi sự kiện Tần xuất làm tươi màn hình
Kích thước	M Bytes Số lượng ROM chip
Dễ sử dụng	Thời gian huấn luyện Số trang tài liệu hướng dẫn sử dụng
Độ tin cậy Reliability	Khoảng thời gian trung bình giữa các sự cố Xác suất hệ thống không hoạt động tại một thời điểm Số lần xảy ra sự cố trong mỗi giờ
Vững mạnh - Robustness	Thời gian cần để hoạt động lại sau sự cố Phần trăm sự kiện gây sự cố Xác xuất hỏng dữ liệu do sự cố
Khả chuyển - Portability	Số lượng hệ thống đích



# Nội dung chính

---

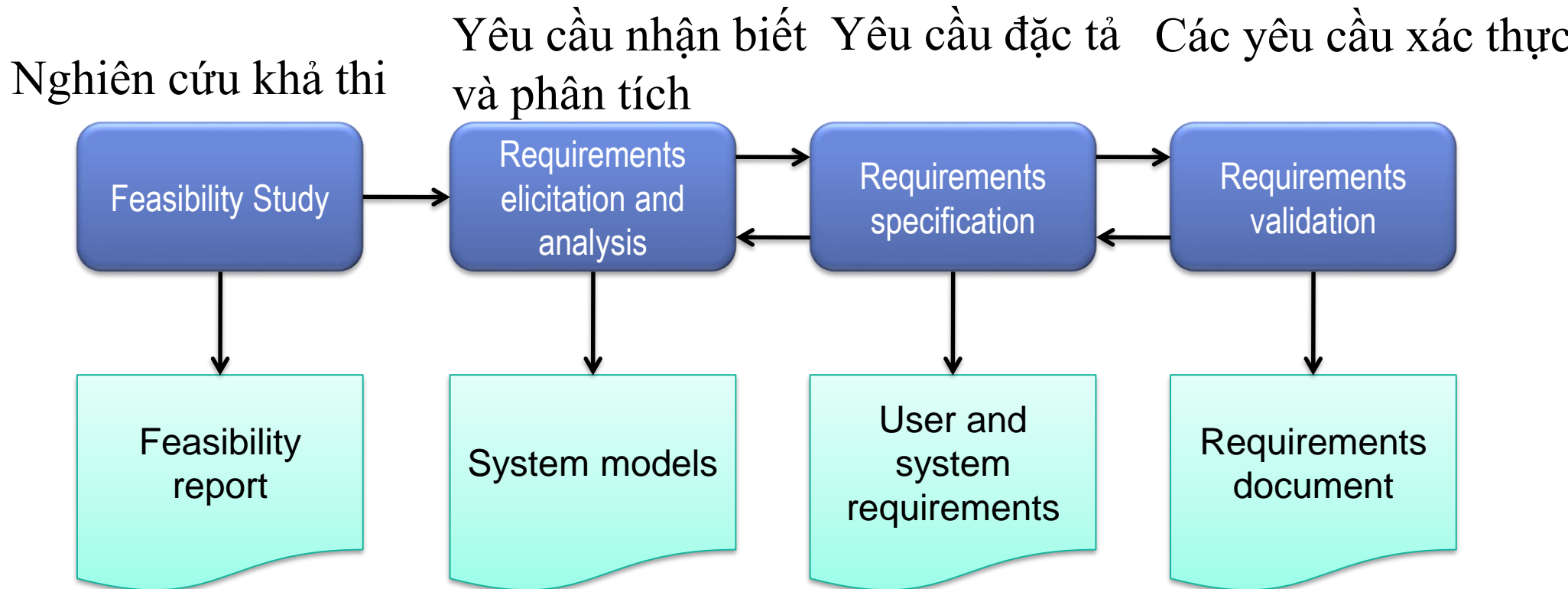
- Yêu cầu phần mềm là gì?
- Tầm quan trọng của yêu cầu phần mềm trong quá trình phát triển phần mềm
- Kỹ nghệ yêu cầu

# Nội dung chính

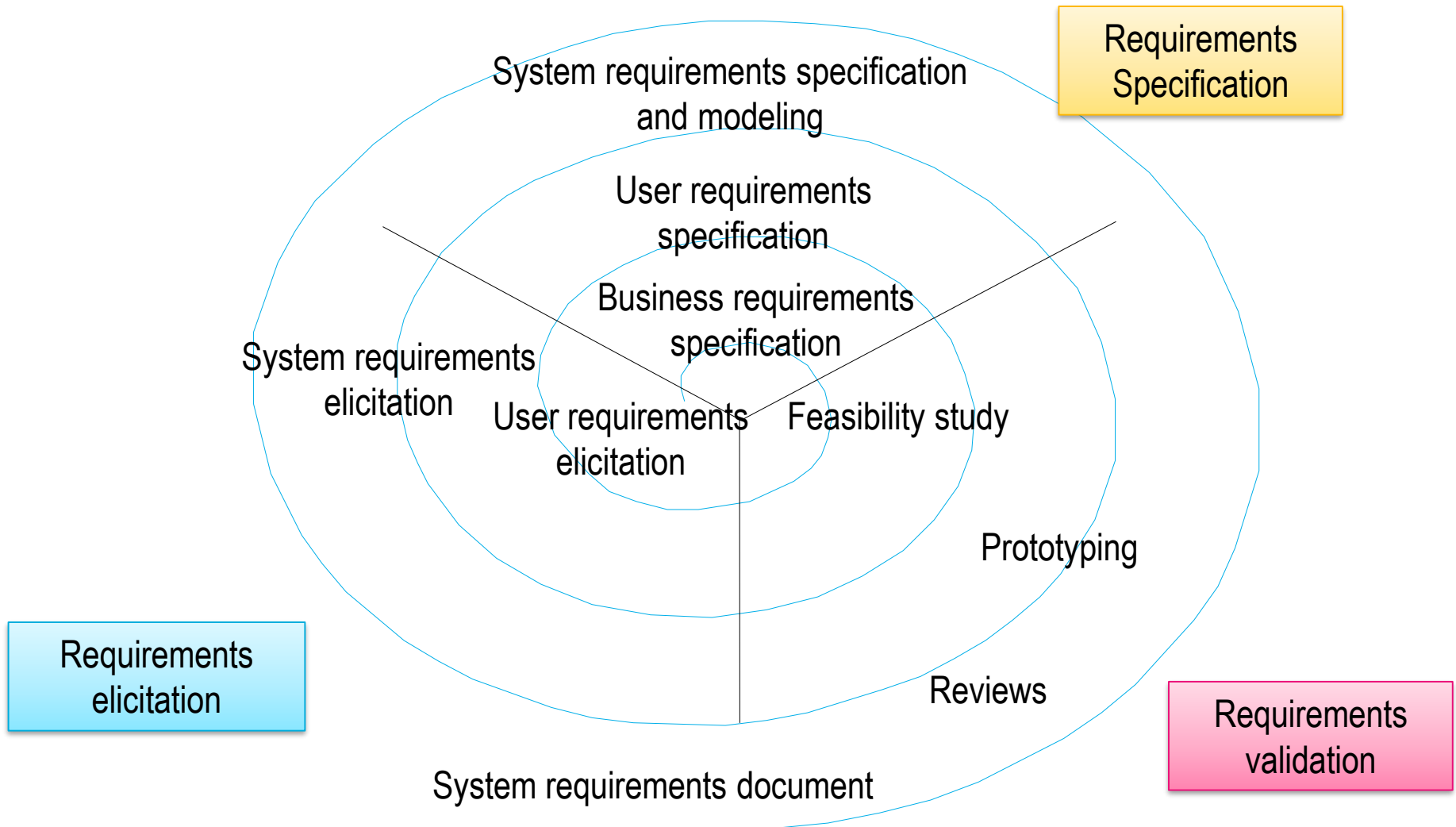
---

- Yêu cầu phần mềm là gì?
- Tầm quan trọng của yêu cầu phần mềm trong quá trình phát triển phần mềm
- **Kỹ nghệ yêu cầu**
  - Nghiên cứu khả thi
  - Thu thập và phân tích yêu cầu
  - Làm tài liệu yêu cầu
  - Thẩm định yêu cầu

# The requirements engineering process



# Kỹ nghệ yêu cầu



# Nghiên cứu khả thi

## Feasibility studies

---

- Một nghiên cứu ngắn, tập trung, nhằm kiểm tra xem
  - Hệ thống có đóng góp cho các mục tiêu của tổ chức hay không?
  - Hệ thống có thể được phát triển bằng công nghệ hiện hành và trong phạm vi ngân sách hay không?
  - Hệ thống có thể được tích hợp với các hệ thống khác đang được sử dụng hay không?

# Thực hiện nghiên cứu khả thi

---

- Dựa trên đánh giá thông tin (cái gì cần), thu thập thông tin và viết báo cáo.
- Các câu hỏi dành cho nhân viên của tổ chức
  - Nếu hệ thống không được cài đặt thì sao?
  - Quy trình hiện hành có những vấn đề gì?
  - Hệ thống được đề xuất sẽ giúp được gì và như thế nào?
  - Khi tích hợp sẽ gặp những rắc rối nào?
  - Có cần công nghệ mới hay không? Cần kĩ năng gì?
  - Hệ thống mới cần hỗ trợ những tiện ích nào?

# Nội dung chính

---

- Yêu cầu phần mềm là gì?
- Tầm quan trọng của yêu cầu phần mềm trong quá trình phát triển phần mềm
- Kỹ nghệ yêu cầu
  - Nghiên cứu khả thi
  - Thu thập và phân tích yêu cầu
  - Làm tài liệu yêu cầu
  - Thẩm định yêu cầu

# Các hoạt động quy trình

---

- Phát hiện
  - Tương tác với các stakeholder để tìm ra yêu cầu của họ.
  - Các yêu cầu về miền chuyên dụng cũng được phát hiện tại bước này.
- Phân loại và tổ chức
  - Phân nhóm các yêu cầu có liên quan đến nhau và tổ chức chúng thành các cụm có quan hệ gắn kết với nhau.
- Đặt thứ tự ưu tiên và giải quyết mâu thuẫn giữa các yêu cầu
  - Xếp thứ tự ưu tiên cho các yêu cầu và giải quyết các xung đột/mâu thuẫn giữa các yêu cầu.
- Documentation – Viết tài liệu
  - Ghi lại các yêu cầu làm tài liệu đầu vào cho vòng xoắn tiếp theo.



# Vòng xoắn ốc yêu cầu



# Phát hiện yêu cầu

---

- Quy trình **thu thập** thông tin về hệ thống đề xuất và các hệ thống sẵn có, **gạn lọc** ra các yêu cầu người dùng và yêu cầu hệ thống từ các thông tin này.

# Thu thập yêu cầu từ đâu?



- Làm việc với khách hàng để tìm hiểu thông tin về
  - Miền ứng dụng,
  - Các dịch vụ mà hệ thống cần cung cấp và
  - Các ràng buộc về vận hành hệ thống.
- Những người có thể cần tham gia: khách hàng, người sử dụng, lập trình viên, chuyên gia kỹ thuật,...
  - stakeholders.
- Tài liệu về hoạt động doanh nghiệp
- Đặc tả của các **hệ thống tương tự**.

# Ví dụ: ATM stakeholder



- Khách hàng của ngân hàng (người sử dụng dịch vụ)
- Đại diện của các ngân hàng khác (ATM của ngân hàng này có thể dùng để giao dịch với ngân hàng khác)
- Quản lý ngân hàng (dùng thông tin quản lý từ hệ thống ATM)
- Nhân viên tại các chi nhánh ngân hàng (vận hành hệ thống)
- Quản trị cơ sở dữ liệu (tích hợp hệ thống với CSDL của ngân hàng)
- Quản lý an ninh
- Phòng marketing (muốn dùng ATM để quảng cáo)
- Kỹ sư bảo trì phần mềm và phần cứng
- Những người điều phối hệ thống ngân hàng quốc gia (đảm bảo hệ thống tuân theo nguyên tắc chung)

# Các kĩ thuật

---

- Lấy yêu cầu
  - Phỏng vấn, điều tra bằng bảng câu hỏi
  - Danh mục khái niệm (glossary) để hiểu miền ứng dụng
  - Ca sử dụng / user story
  - Quan sát
  - Nghiên cứu tài liệu
  - Joint Application Design – JAD
  - Làm bản mẫu
- Đặc tả yêu cầu
  - Danh mục khái niệm
  - Use case / user story

# Khó khăn khi phân tích yêu cầu

---

- Stakeholder không biết họ thực sự muốn gì.
- Stakeholder diễn đạt yêu cầu bằng các thuật ngữ của họ.
- Các stakeholder khác nhau có thể có các yêu cầu xung đột.
- Các nhân tố tổ chức và **chính trị** có thể ảnh hưởng đến yêu cầu hệ thống.
- Các yêu cầu **thay đổi** ngay trong quá trình phân tích
  - Chẳng hạn khi môi trường doanh nghiệp thay đổi.

# Nội dung chính

---

- Yêu cầu phần mềm là gì?
- Tầm quan trọng của yêu cầu phần mềm trong quá trình phát triển phần mềm
- Kỹ nghệ yêu cầu
  - Nghiên cứu khả thi
  - Thu thập và phân tích yêu cầu
    - Use case – ca sử dụng
  - Làm tài liệu yêu cầu
  - Thẩm định yêu cầu

# Scenario

---

- Kịch bản (scenario) là các ví dụ đời thực về việc hệ thống có thể được sử dụng như thế nào.
- Các kịch bản nên bao gồm
  - Một miêu tả về tình huống ban đầu
  - Một miêu tả về luồng sự kiện thông thường
  - Một miêu tả về những trục trặc gì có thể xảy ra
  - Thông tin về các hoạt động xảy ra đồng thời
  - Một miêu tả về trạng thái khi kịch bản kết thúc



# Kịch bản LIBSYS (1)

**Initial Assumption:** Người dùng đã đăng nhập hệ thống LIBSYS và đã tìm thấy tạp chí có đăng tài liệu cần tìm.

**Normal:**

- Người dùng chọn tài liệu cần copy. Hệ thống sẽ yêu cầu người dùng nhập thông tin thuê bao hoặc chọn cách trả phí dùng tài liệu. Có thể thanh toán bằng thẻ tín dụng hoặc dùng số tài khoản của một tổ chức.
- Sau đó người dùng được yêu cầu điền một form bản quyền trong đó có chi tiết về giao dịch này, rồi submit form đó cho hệ thống LIBSYS.
- Hệ thống kiểm tra form bản quyền, nếu OK, bản PDF của tài liệu sẽ được tải xuống máy tính của người dùng và người dùng được thông báo về việc này. Sau đó người dùng được chọn một máy in, và tài liệu sẽ được in tại đó. Nếu tài liệu đã được gắn cờ 'print-only' thì nó sẽ được xóa khỏi máy của người dùng ngay sau khi người dùng khẳng định rằng đã in xong.

# Kịch bản LIBSYS (2)

## **What can go wrong:**

- Người dùng có thể điền form sai. Khi đó hệ thống cần hiện lại form để người dùng sửa lại. Nếu form được submit sau đó vẫn sai thì hủy yêu cầu đọc tài liệu của người dùng.
- Hệ thống có thể không chấp nhận giao dịch thanh toán tiền. Hủy yêu cầu đọc tài liệu của người dùng.
- Việc download tài liệu có thể thất bại. Làm lại cho đến khi thành công hoặc khi người dùng chấm dứt phiên làm việc.
- Có thể không in được tài liệu. Nếu bài báo không có gắn cờ 'print-only' thì giữ nó trong workspace của LIBSYS. Nếu không, xóa tài liệu và hoàn lại chi phí cho người dùng.

**Other activities:** Song song download các tài liệu khác nhau.

**System state on completion:** Người dùng đang ở trạng thái đăng nhập. Nếu tài liệu có gắn cờ 'print-only' thì nó đã bị xóa khỏi LIBSYS workspace.

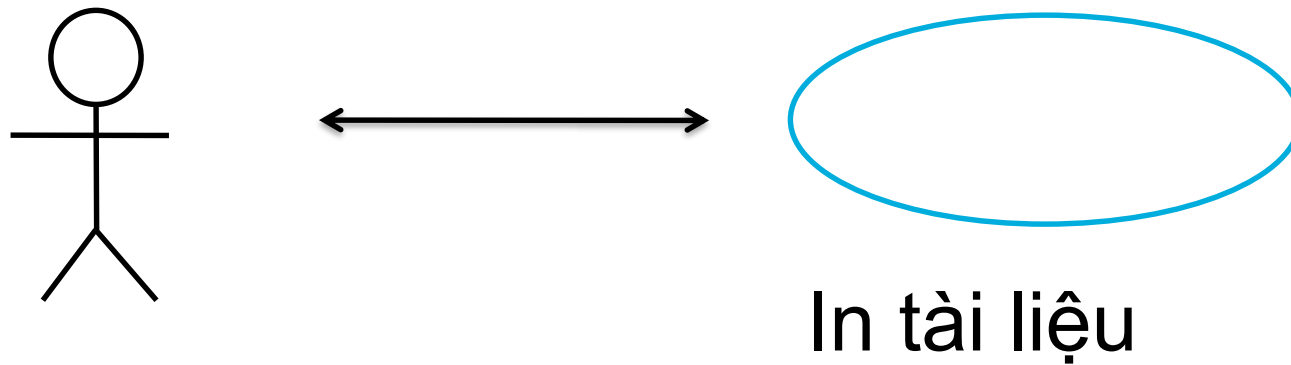
# Use case

---

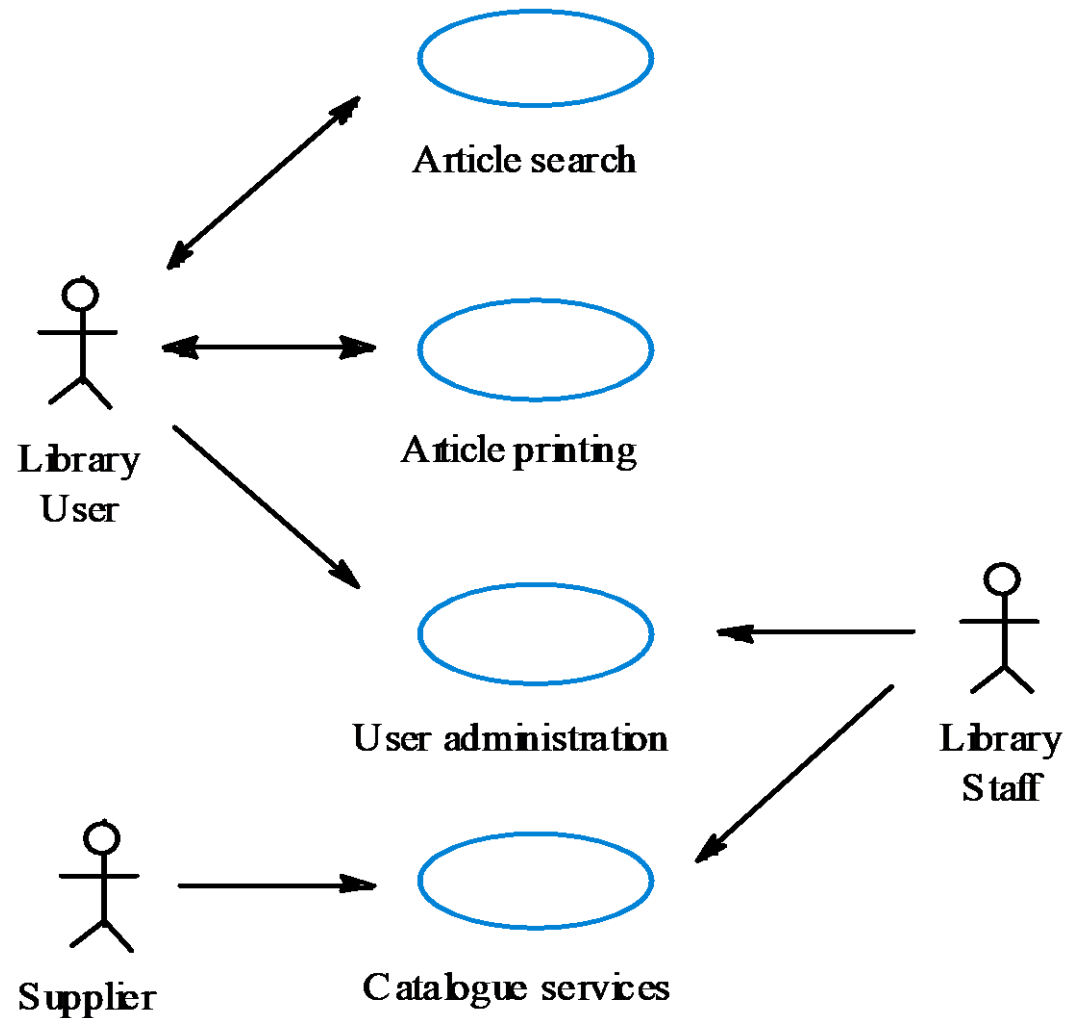
- Ca sử dụng (use-case) là một kĩ thuật kiểu kịch bản bằng ngôn ngữ UML
  - Chỉ rõ các actor trong một tương tác và
  - Mô tả chính tương tác đó.
- Một bộ ca sử dụng có thể mô tả được tất cả các tương tác có thể đối với hệ thống.
- Có thể dùng các sơ đồ tuần tự (sequence diagram) để bổ sung chi tiết cho các ca sử dụng
  - Minh họa chuỗi xử lý sự kiện.

# use-case in tài liệu

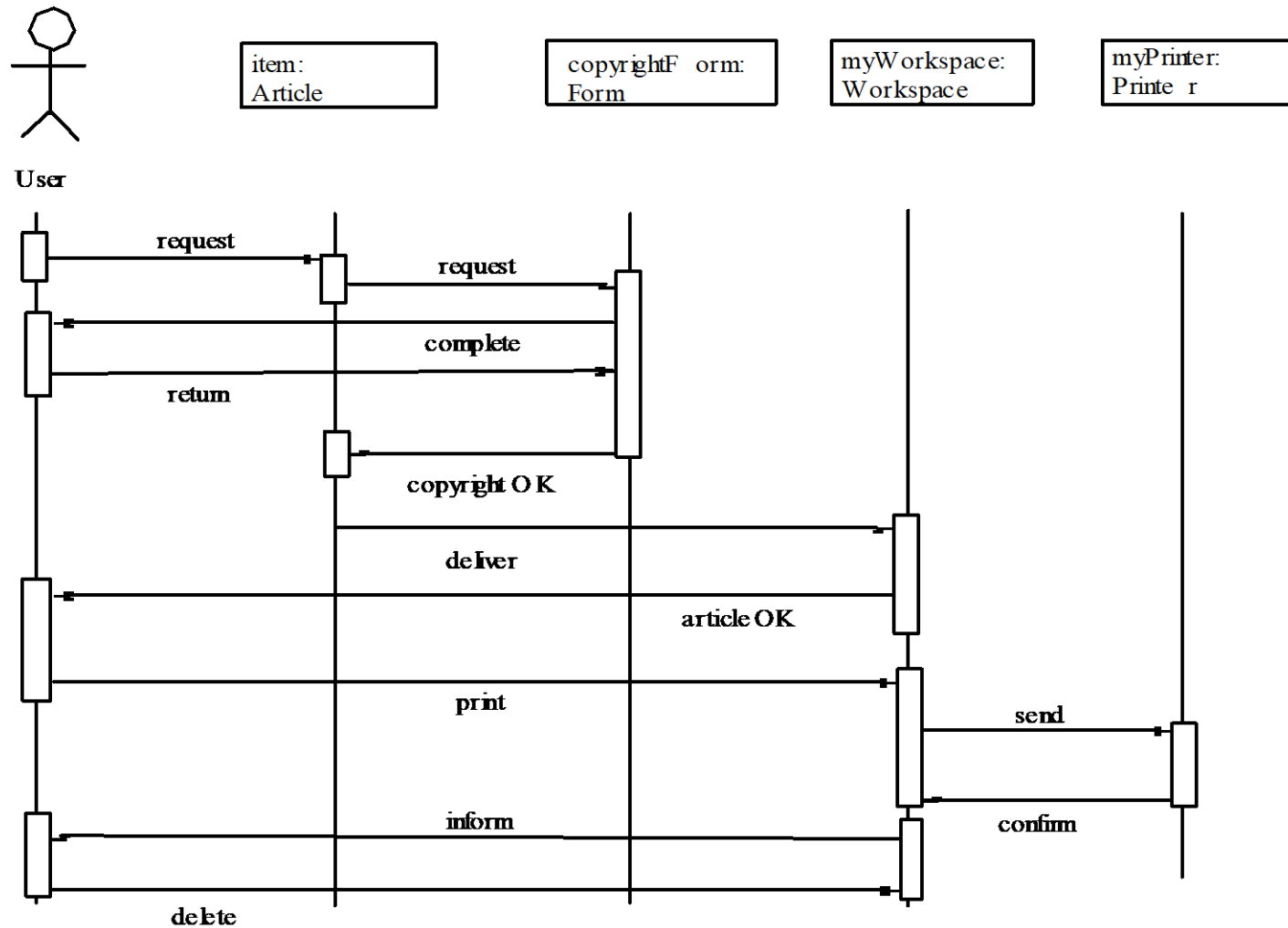
---



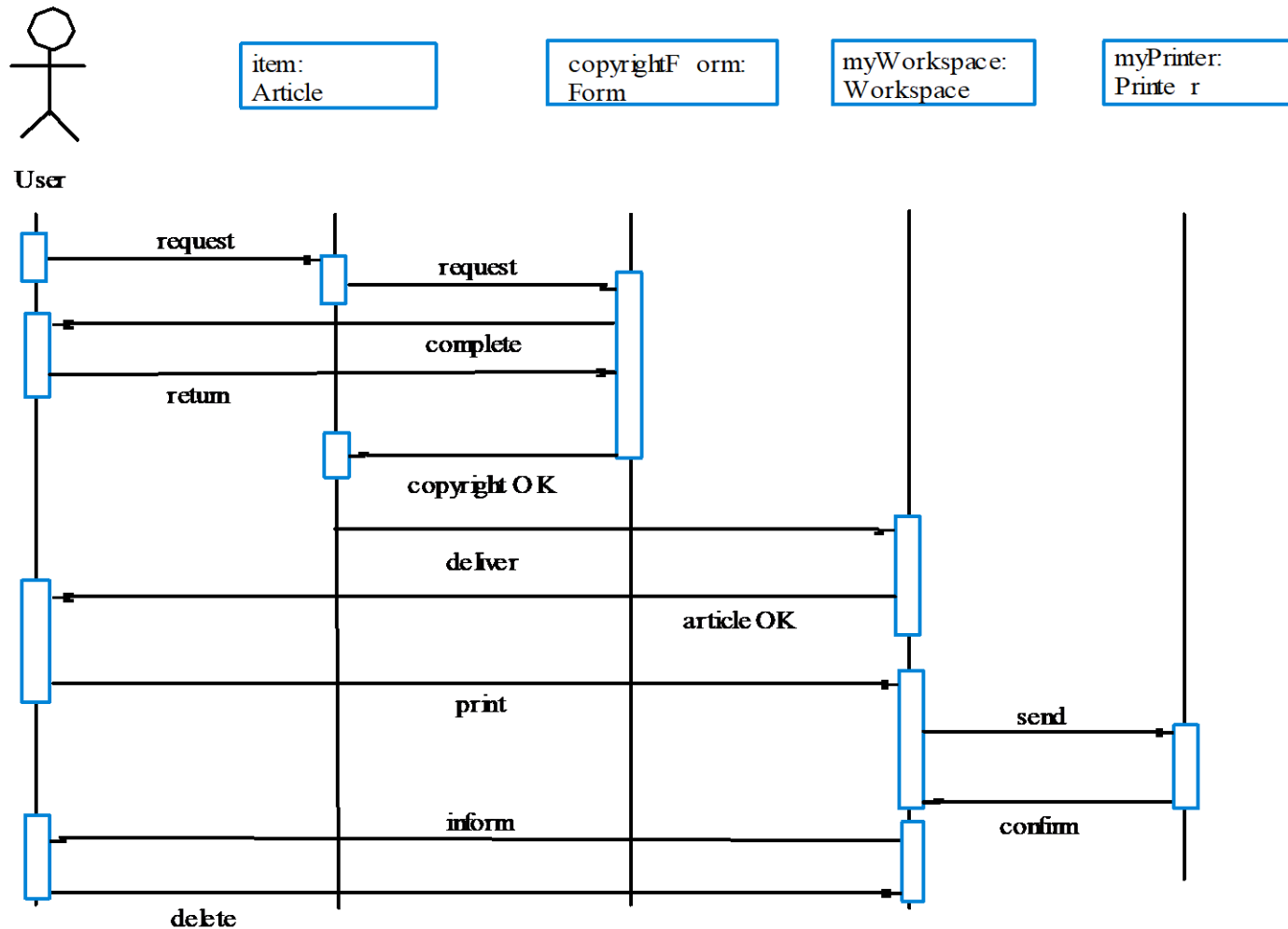
# LIBSYS use case



# Article printing



# Print article sequence



# Ca sử dụng – Use case

---

- Use case:
  - Là một tập các kịch bản tương tác giữa một hoặc vài actor với hệ thống nhằm thực hiện một mục tiêu chung
- Sơ đồ use case (đồ họa)
  - Sơ đồ mô tả tổng quan các ca sử dụng của một hệ thống và ai dùng chức năng nào
- Mô tả chi tiết use case (văn bản)
  - Mô tả chi tiết tương tác giữa người dùng và hệ thống trong một tập các kịch bản



# Ví dụ Use Case:

## Travel Agency use case list available flights

---

**Tên:** list available flights

**Mô tả:** người dùng xem danh sách các chuyến bay có thể đặt

**Actor:** người dùng

**Kịch bản chính:**

1. người dùng nhập thông tin về thành phố cần đến, ngày đi và ngày đến
2. hệ thống cung cấp một danh sách các chuyến bay phù hợp kèm theo giá vé và booking number

**Kịch bản phụ:**

- 1a. Dữ liệu vào không đúng
  2. Hệ thống báo lỗi và kết thúc, use case quay lại từ đầu
- 2.a. Không có chuyến bay nào phù hợp
  3. Use case quay lại từ đầu

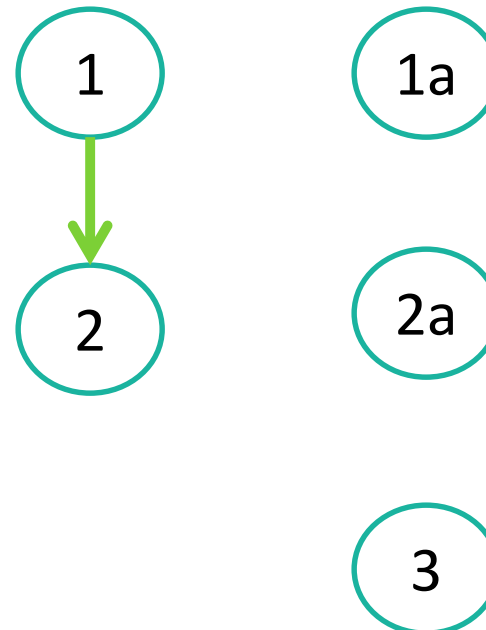
**Ghi chú:** Dữ liệu vào là đúng nếu tên thành phố đúng, ngày đi và ngày đến là các ngày hợp lệ, ngày đi sớm hơn ngày đến, ngày đến muộn hơn thời điểm hiện tại ít nhất 2 ngày, và ngày đi không muộn hơn một năm kể từ hiện tại

# Ví dụ Use Case:

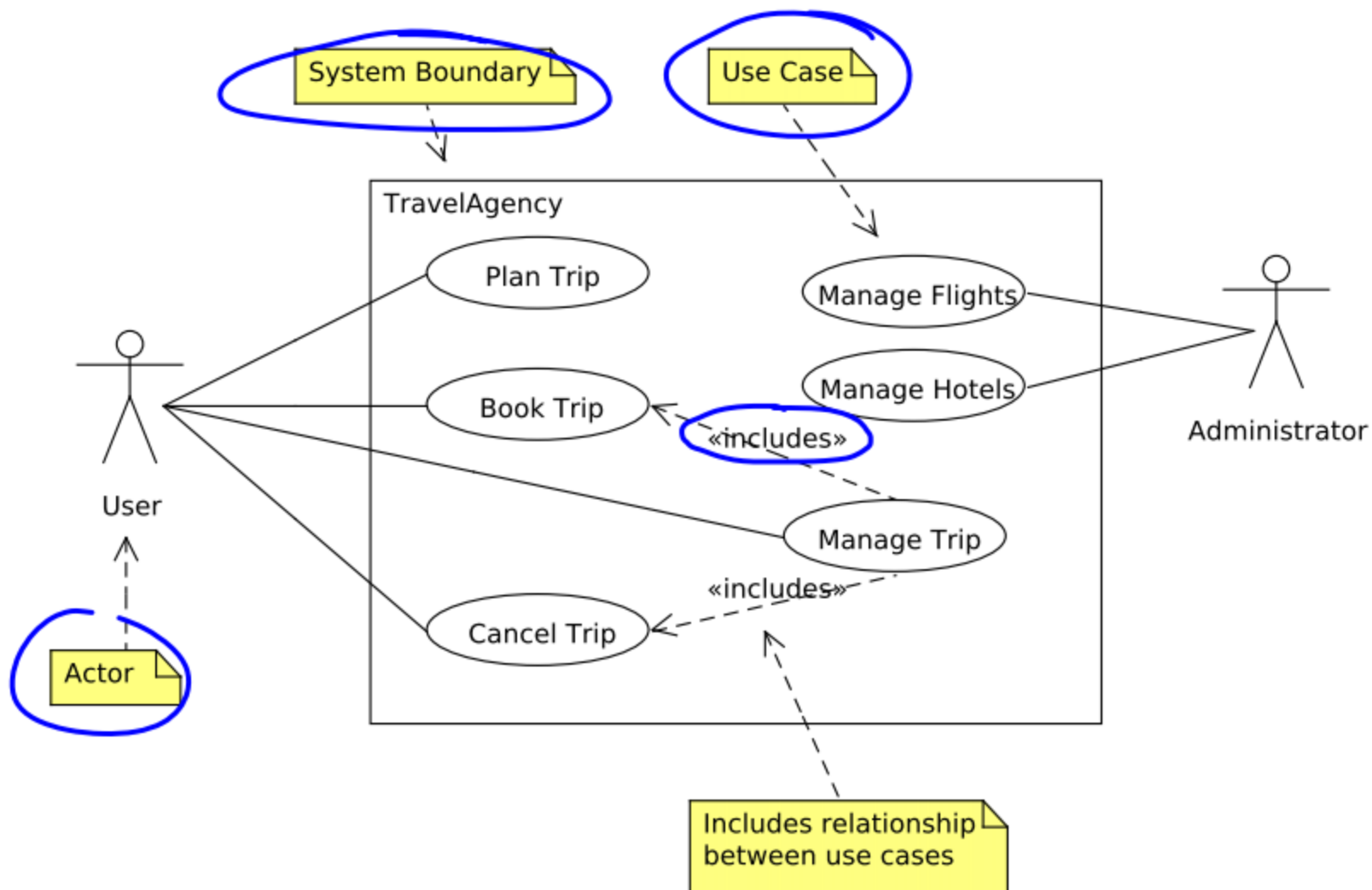
## Travel Agency use case list available flights

---

1. người dùng nhập thông tin về thành phố cần đến, ngày đi và ngày đến
2. hệ thống cung cấp một danh sách các chuyến bay phù hợp kèm theo giá vé và booking number
  - 1a. Dữ liệu vào không đúng
    2. Hệ thống báo lỗi và kết thúc, use case quay lại từ đầu
  - 2.a. Không có chuyến bay nào phù hợp
    3. Use case quay lại từ đầu



# Sơ đồ Use case

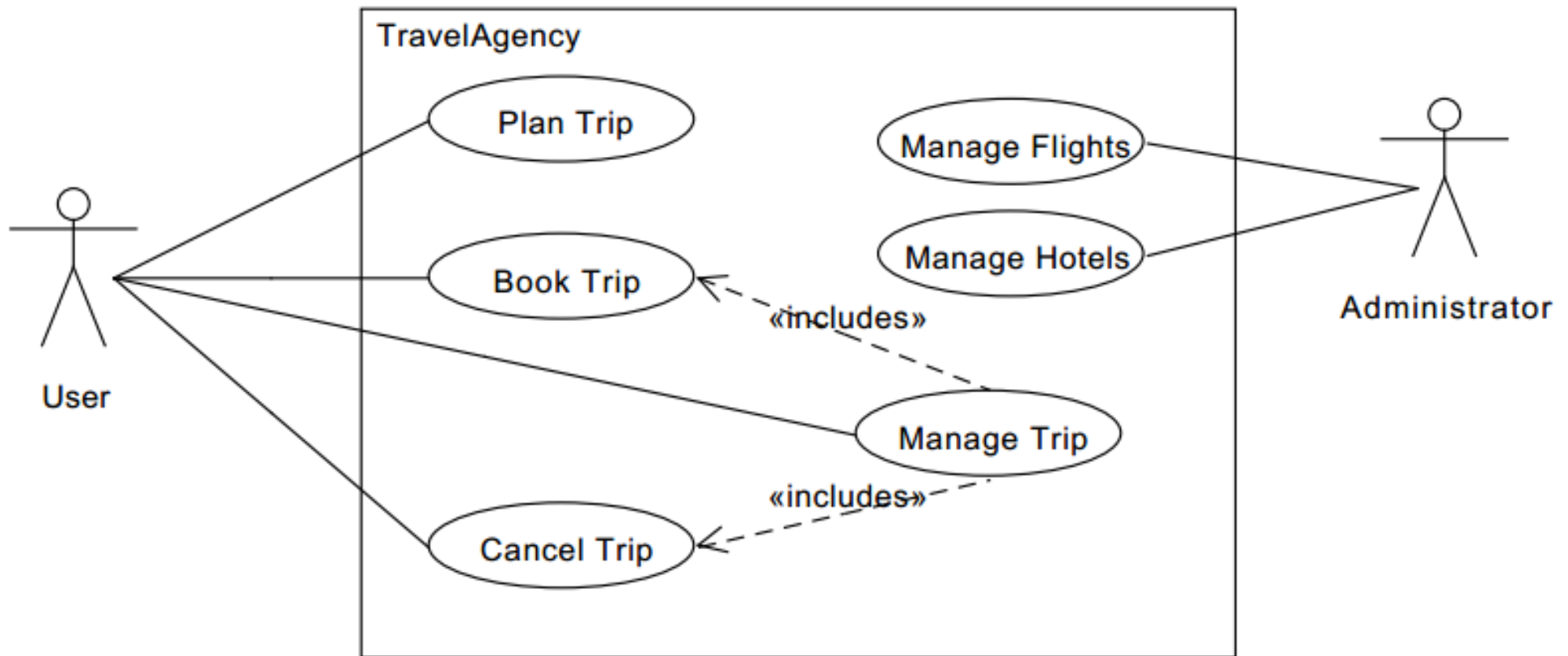


# Các loại sơ đồ use case

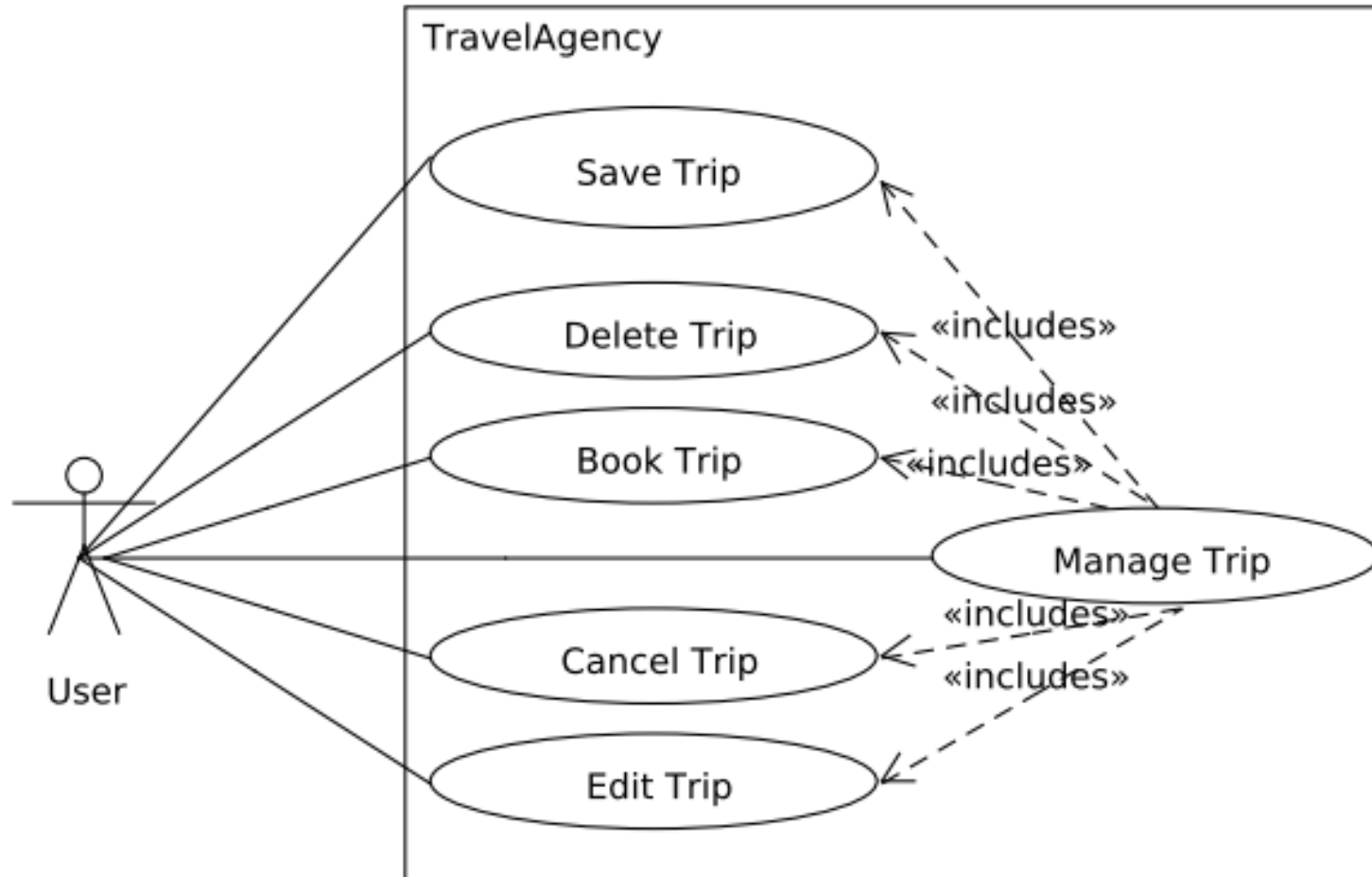
---

- Business use case
  - Một phần của tài liệu yêu cầu người dùng
  - Mô tả chức năng từ góc nhìn của người dùng
- System use case
  - Một phần của tài liệu kỹ thuật của đội phát triển
  - Mang tính kỹ thuật và chi tiết hơn
  - Tập trung vào mô tả những gì cần cài đặt

# Yêu cầu chức năng của TravelAgency: các business use case

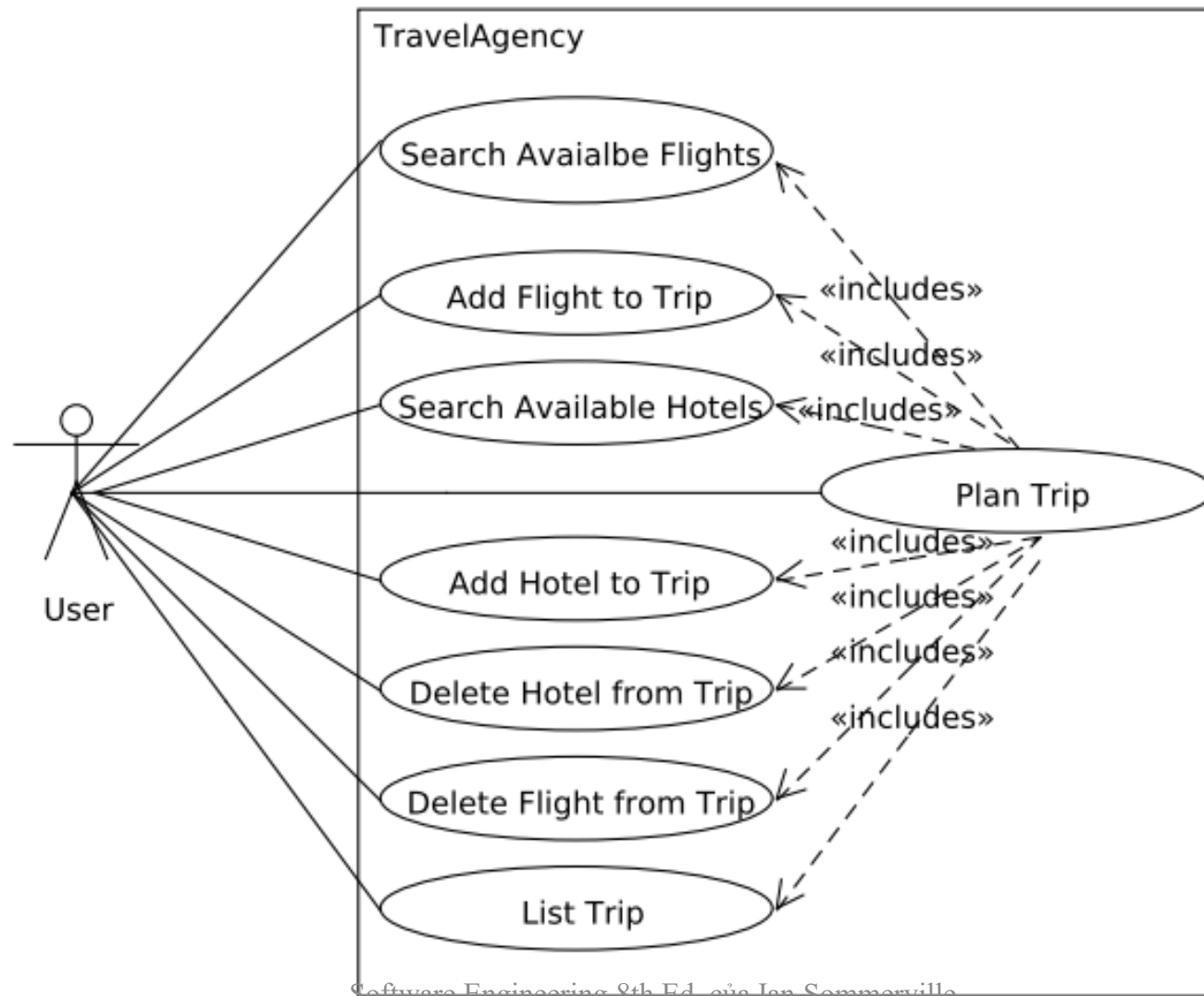


# Yêu cầu chức năng của TravelAgency: System use case, phần 1: *manage trip*



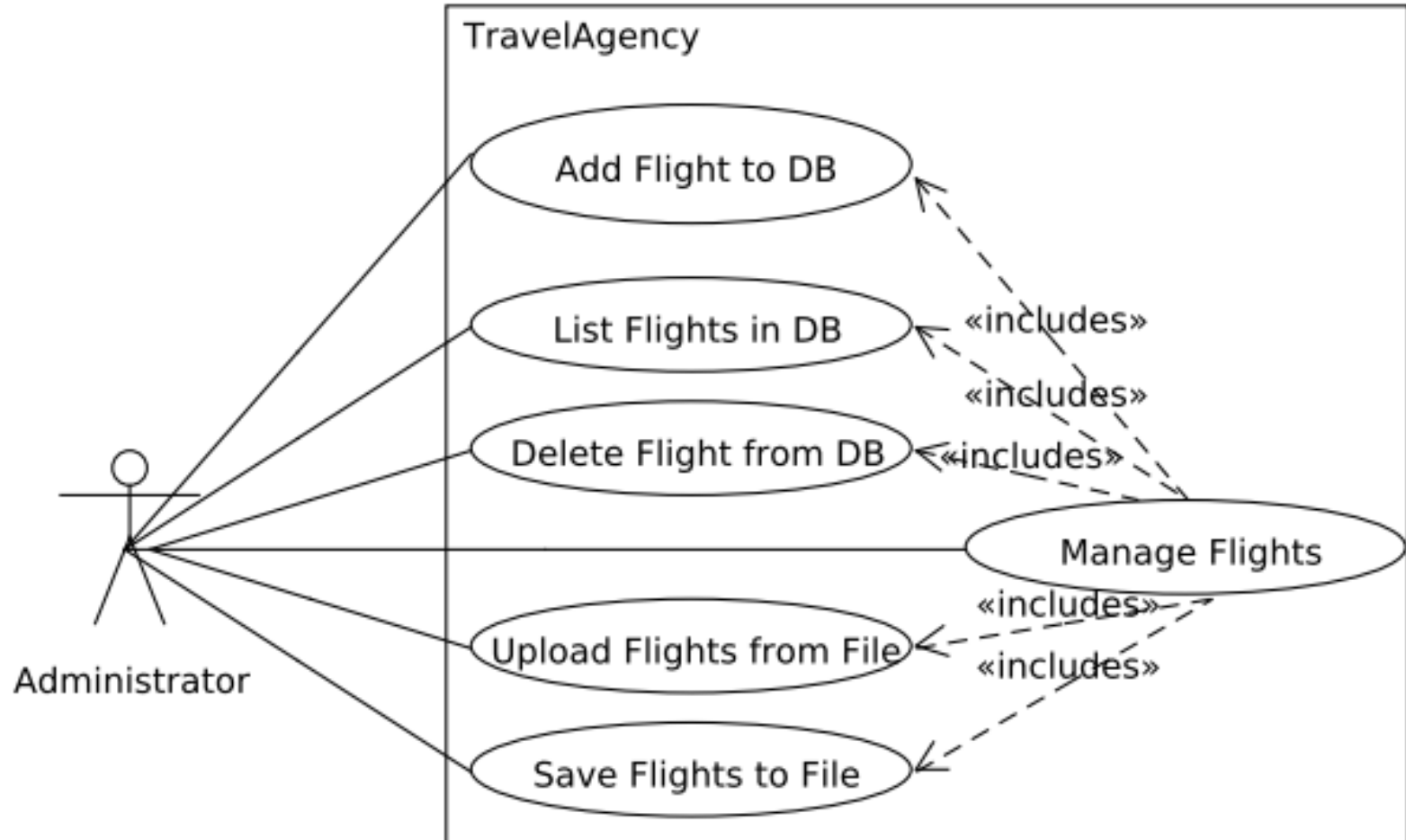
# Yêu cầu chức năng của TravelAgency:

## System use case, phần 2: *plan trip*



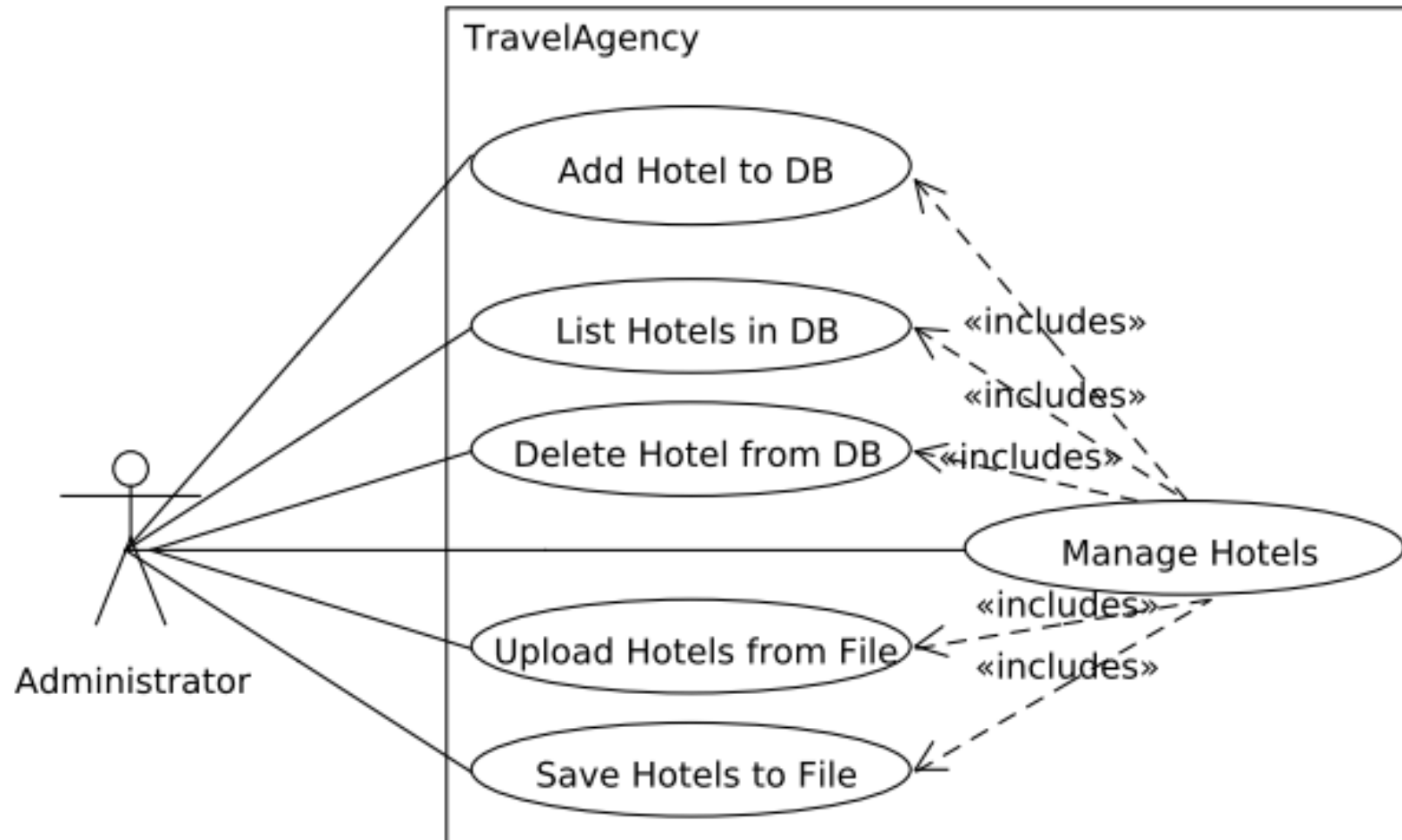
# Yêu cầu chức năng của TravelAgency:

## System use case, phần 3: *manage flights*





# Yêu cầu chức năng của TravelAgency: System use case, phần 4: *manage hotels*



# Mẫu tài liệu mô tả use case

---

Mẫu dùng cho môn học này

- **Tên:** tên của use case
- **Mô tả:** Mô tả ngắn gọn của use case – mục tiêu của ca sử dụng
- **Actor:** một hoặc vài actor – nhân tố tương tác với hệ thống
- **Tiền điều kiện:** các điều kiện hệ thống cần thỏa mãn để use case có thể hoạt động
- **Kịch bản chính:** Mô tả chuỗi tương tác chính giữa actor và hệ thống
  - Chú ý! Chỉ nên mô tả hệ thống từ góc nhìn của người sử dụng
- **Các kịch bản phụ:** Có thể chứa các kịch bản thất bại
- **Ghi chú:** Dùng cho tất cả những gì cần thiết nhưng lại không phù hợp với các thể loại trên

# Travel Agency. Mô tả chi tiết use case: list available flights

---

**Tên:** list available flights

**Mô tả:** người dùng xem danh sách các chuyến bay có thể đặt

**Actor:** người dùng

**Kịch bản chính:**

1. người dùng nhập thông tin về thành phố cần đến, ngày đi và ngày đến
2. hệ thống cung cấp một danh sách các chuyến bay phù hợp kèm theo giá vé và booking number

**Kịch bản phụ:**

- 1a. Dữ liệu vào không đúng
  2. Hệ thống báo lỗi và kết thúc, use case quay lại từ đầu
- 2.a. Không có chuyến bay nào phù hợp
  3. Use case quay lại từ đầu

**Ghi chú:** Dữ liệu vào là đúng nếu tên thành phố đúng, ngày đi và ngày đến là các ngày hợp lệ, ngày đi sớm hơn ngày đến, ngày đến muộn hơn thời điểm hiện tại ít nhất 2 ngày, và ngày đi không muộn hơn một năm kể từ hiện tại

# Kịch bản

---

- Tương tác giữa một actor và hệ thống
  - Người dùng tác động vào hệ thống
  - Hệ thống phản ứng
  - Các hiệu ứng quan trọng đối với người dùng / người dùng thấy
- Hoạt động *bên trong* của hệ thống **không phải** là một phần của tương tác

# Travel Agency. Mô tả chi tiết use case: cancel trip

---

**Tên:** cancel trip

**Mô tả:** người dùng hủy một chuyến đi đã đặt

**Actor:** người dùng

**Tiền điều kiện:**

- Chuyến đi đã được đặt chỗ
- Ngày đầu tiên của thời gian đặt phòng khách sạn hoặc chuyến bay phải muộn hơn thời điểm hiện tại ít nhất 1 ngày.

**Kịch bản chính:**

1. Người dùng chọn chuyến đi để hủy
2. Hệ thống thông báo chi phí của việc hủy chuyến đi
3. Chuyến đi được chọn sẽ bị hủy sau khi người dùng khẳng định việc hủy

# Travel Agency. Mô tả chi tiết use case: plan trip

---

Use case này chứa (include) các use case khác

**Tên:** plan trip

**Mô tả:** người dùng lập kế hoạch một chuyến đi bao gồm khách sạn và các chuyến bay

**Actor:** người dùng

**Kịch bản chính:** Lặp đi lặp lại hoạt động bất kì trong danh sách sau cho đến khi xong

1. list available flights (use case)
2. add flight to trip (use case)
3. list available hotels (use case)
4. add hotel to trip (use case)
5. list trip (use case)
6. delete hotel from trip (use case)
7. delete flight from tip (use case)

# Travel Agency. Mô tả chi tiết use case: save trip

---

Tên: save trip

Mô tả: người dùng đặt tên cho chuyến đi hiện hành và lưu lại cho sử dụng sau này

Actor: người dùng

Tiền điều kiện: chuyến đi hiện hành không rỗng

Kịch bản chính:

1. Người dùng nhập tên cho chuyến đi

Các kịch bản phụ:

- 1: tên nhập không hợp lệ

- 2: thông báo cho người dùng và kết thúc use case

- 1: tên trùng với tên của một chuyến khác

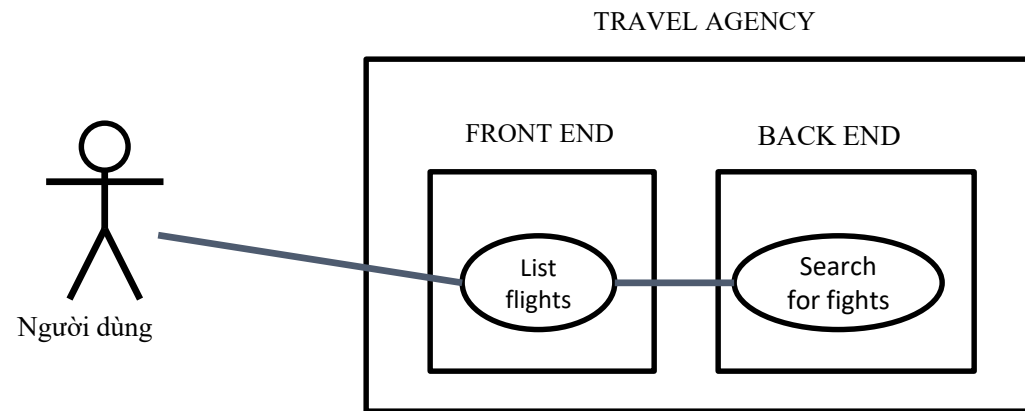
- 2: hỏi người dùng xem có nên ghi đè lên chuyến đã ghi lần trước hay không

- 3a: nếu người dùng đồng ý, ghi đè lên chuyến cũ

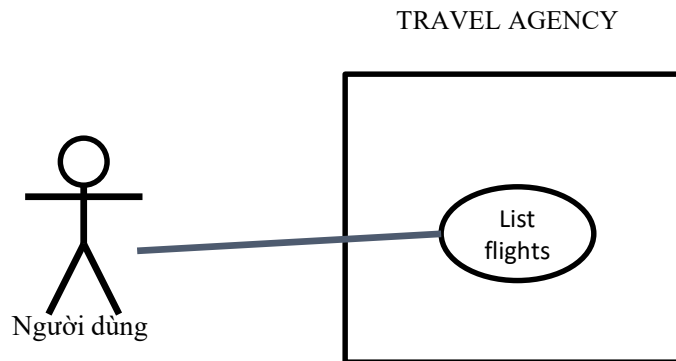
- 3b: nếu người dùng không đồng ý, kết thúc use case

# Use case và ranh giới hệ thống

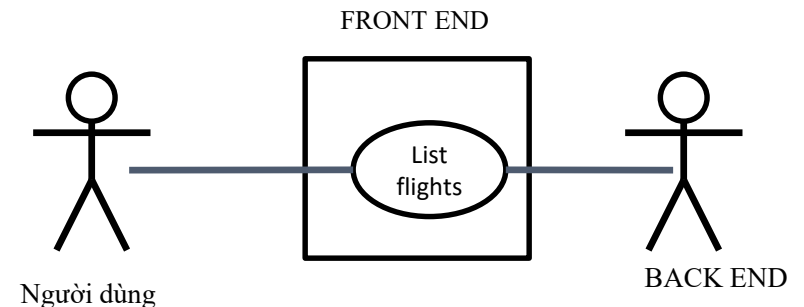
- Use case và actor được xác định tùy theo ranh giới hệ thống



- Biên hệ thống: TravelAgency



- Biên hệ thống: Front End





# Nội dung chính

---

- Yêu cầu phần mềm là gì?
- Tầm quan trọng của yêu cầu phần mềm trong quá trình phát triển phần mềm
- Kỹ nghệ yêu cầu
  - Nghiên cứu khả thi
  - Thu thập và phân tích yêu cầu
  - Làm tài liệu yêu cầu
  - **Thẩm định yêu cầu**

# Thẩm định yêu cầu

---

- (Requirement validation) quan tâm đến việc chứng tỏ rằng các yêu cầu định nghĩa được hệ thống mà **khách hàng thực sự muốn**.
- Chi phí của lỗi yêu cầu cao đến mức công đoạn thẩm định rất quan trọng
  - Việc sửa một lỗi yêu cầu sau khi đã bàn giao phần mềm có thể tốn kém gấp 100 lần chi phí cho việc sửa một lỗi cài đặt.

# Các tiêu chí cho thẩm định

---

- Hiệu lực – Validity
  - Hệ thống có cung cấp những chức năng phục vụ tốt nhất yêu cầu của khách hàng hay không?
- Nhất quán – Consistency
  - Có những yêu cầu nào xung đột nhau không?
- Đầy đủ – Completeness
  - Có đủ các chức năng mà khách hàng đòi hỏi hay không?
- Thực tế – Realism
  - Có thể cài đặt các yêu cầu trong phạm vi công nghệ và ngân sách cho phép hay không?
- Kiểm định được – Verifiability
  - Có cách kiểm tra các yêu cầu xem chúng đã được thỏa mãn chưa hay không?

# Kỹ thuật thẩm định yêu cầu

---

- Duyệt yêu cầu – Requirements reviews
  - Đọc và phân tích lại một cách có hệ thống (không dùng chương trình tự động).
- Phiên bản thử nghiệm – Prototyping
  - Dùng một mô hình chạy được của hệ thống để kiểm tra các yêu cầu (Chương 17)
- Tạo các ca kiểm thử (test case)
  - Viết các test dành cho các yêu cầu để kiểm tra khả năng kiểm thử được.

# Quản lý yêu cầu

---

***Yêu cầu phần mềm *luôn luôn thay đổi!****

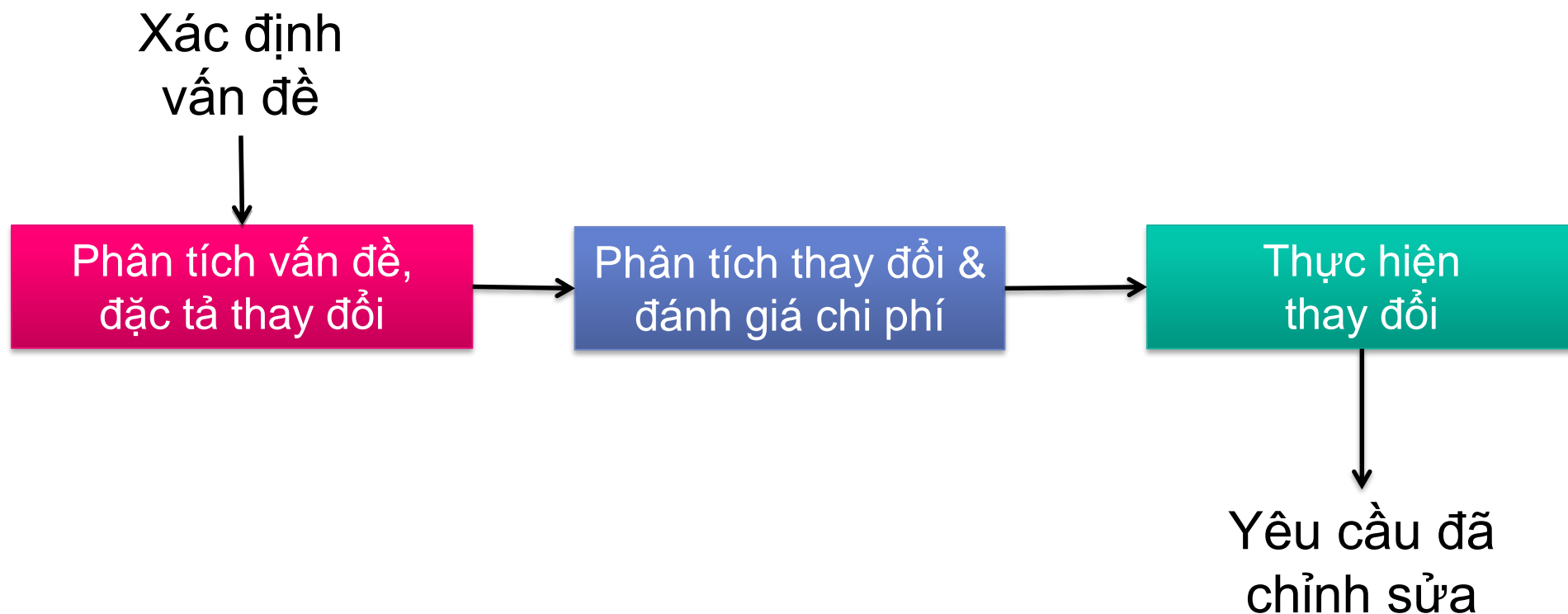
- Môi trường doanh nghiệp và kĩ thuật thay đổi
  - Phần cứng mới → giao diện mới.
  - Luật thay đổi, nhu cầu doanh nghiệp thay đổi → thay đổi chức năng
- Khách hàng, người sử dụng thay đổi
  - Thay đổi chức năng
- Xung đột giữa các yêu cầu mới nảy sinh, và giữa yêu cầu mới với yêu cầu cũ

# Quản lý yêu cầu

---

- Để quản lý yêu cầu, cần xác định
  - *Định danh yêu cầu*: Mỗi yêu cầu có định danh riêng để tiện cho việc tham chiếu giữa các yêu cầu và lần vết
  - *Quy trình quản lý thay đổi*: các hoạt động đánh giá ảnh hưởng và chi phí của thay đổi
  - *Chính sách lần vết*: cách ghi lại và lưu trữ quan hệ giữa các yêu cầu và giữa mỗi yêu cầu với thiết kế tương ứng với nó
  - *Công cụ hỗ trợ*: hỗ trợ thực hiện các công việc trên một cách có hiệu quả. CASE tool, spreadsheet, cơ sở dữ liệu....

# Quản lý thay đổi



# Quản lý thay đổi yêu cầu

---

- Nên áp dụng cho tất cả các thay đổi được đề xuất đối với bộ yêu cầu.
- Các giai đoạn chính
  - **Phân tích vấn đề**: Thảo luận về vấn đề của các yêu cầu và đề xuất thay đổi; Bổ sung chi tiết; Chốt lại những điểm sẽ thay đổi.
  - **Phân tích thay đổi và đánh giá chi phí**. Đánh giá hiệu ứng của thay đổi đối với các yêu cầu khác; Ra quyết định có thực hiện thay đổi hay không.
  - **Thực hiện thay đổi**. Cập nhật tài liệu yêu cầu và các tài liệu khác để thực hiện thay đổi đã xét.



# Bài tập lớn

---

- Thu thập và phân tích yêu cầu