

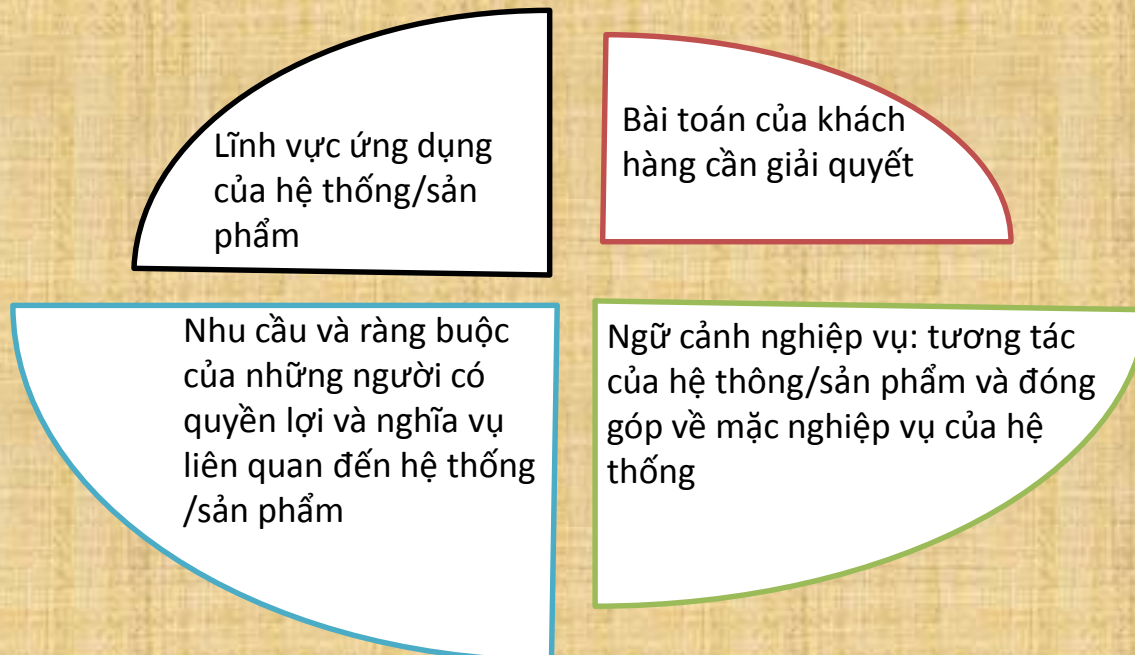
# **NHẬP MÔN CÔNG NGHỆ PHẦN MỀM (INTRODUCTION TO SOFTWARE ENGINEERING)**

# Chương 5: Kỹ nghệ yêu cầu phần mềm (Requirement Engineering)

- 1. Tổng quan về yêu cầu phần mềm
- 2. Quy trình xác định yêu cầu phần mềm
- 3. Phương pháp và công cụ đặc tả yêu cầu phần mềm
- 4. Nguyên lý phân tích yêu cầu sử dụng

# Khái niệm

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



# Mục đích xác định 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.
- Cho nên 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



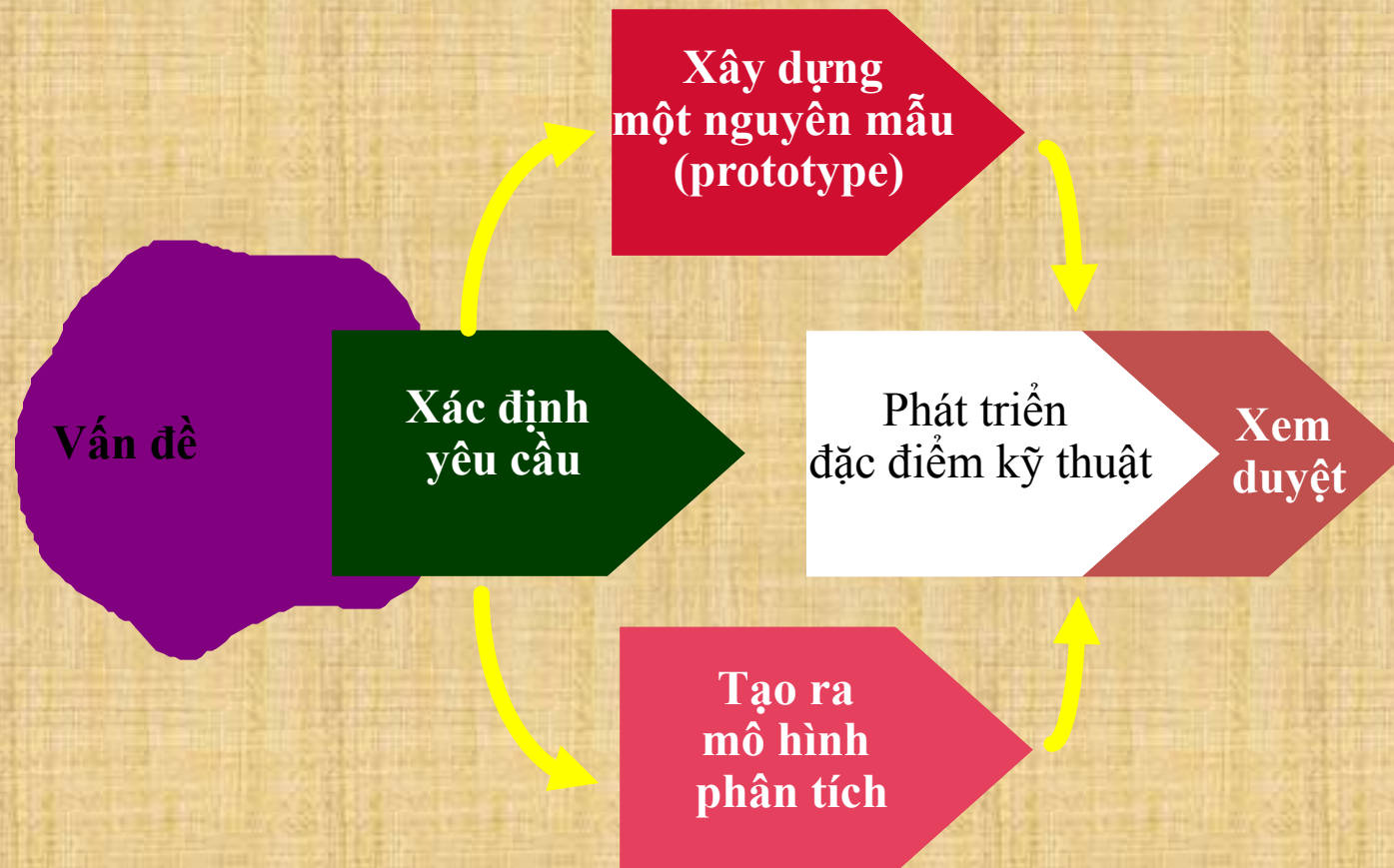
# Phân loại yêu cầu

- 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

## 2. 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 (tiếp)



# 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)



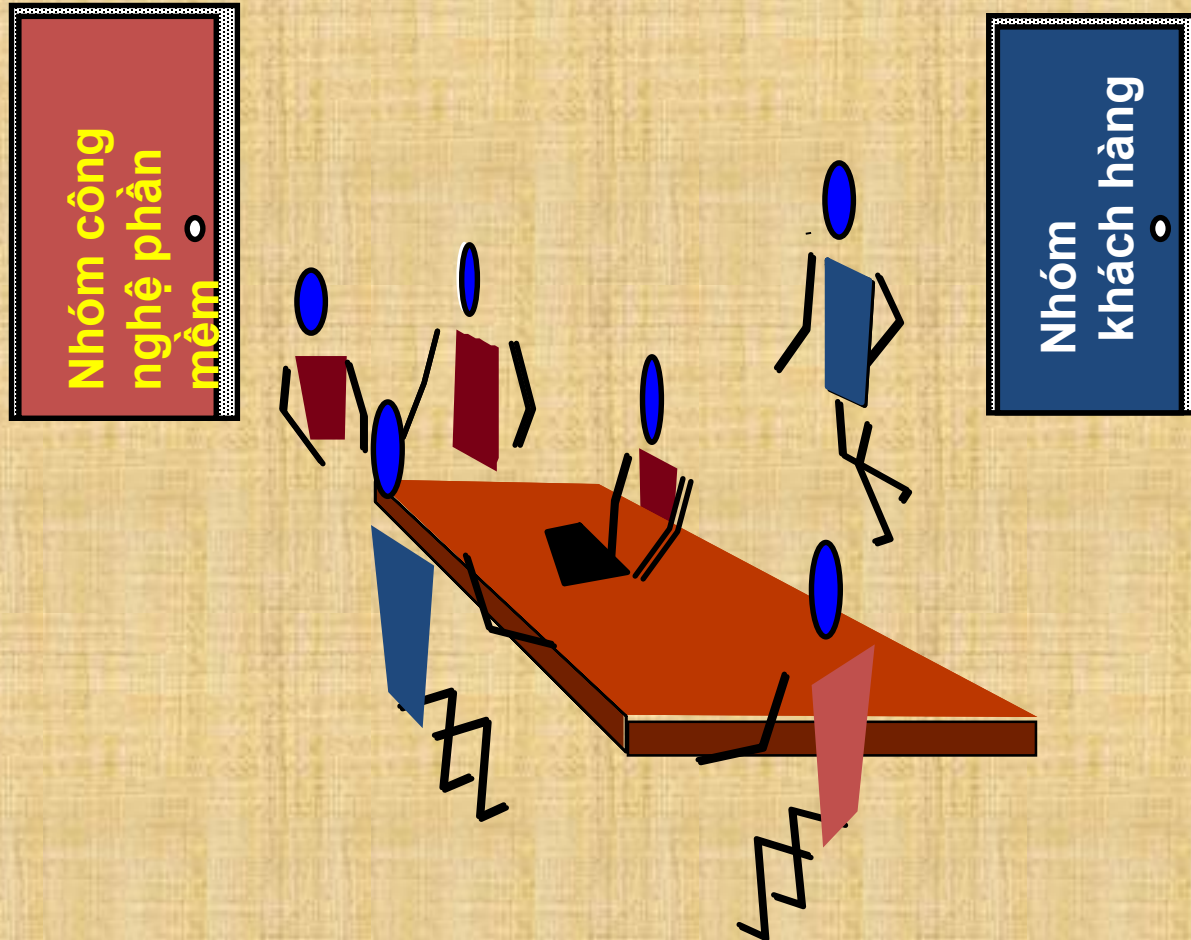
# Phát hiện yêu cầu phần mềm

- 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.

# Đầu ra của bước phát hiện yêu cầu phần mềm

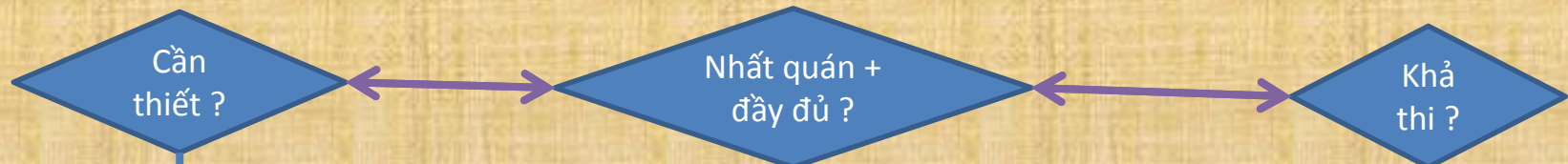
- Bảng kê (statement) các đòi hỏi và chức năng khả thi của phần mềm
- Bảng kê phạm vi ứng dụng của phần mềm
- Mô tả môi trường kỹ thuật của phần mềm
- Bảng kê tập hợp các kịch bản sử dụng của phần mềm
- Các nguyên mẫu xây dựng, phát triển hay sử dụng trong phần mềm (nếu có)
- Danh sách nhân sự tham gia vào quá trình phát hiện các yêu cầu phần mềm - kể cả các nhân sự từ phía công ty- khách hàng

# Phân tích các yêu cầu PM và thương lượng với khách hàng



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

## Phân tích



Yêu cầu không cần thiết

Yêu cầu không đầy đủ,  
yêu cầu mâu thuẫn

Yêu cầu không khả thi

## Thương lượng

Thảo luận  
về các  
yêu cầu

Đặt thứ tự ưu  
tiên cho các yêu  
cầu

Nhất trí  
về các  
yêu cầu



# 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

# Ví dụ: Các yêu cầu về hồ sơ đặc tả

- Đặc tả hành vi bên ngoài của HT
- Đặc tả các ràng buộc về cài đặt
- Dễ thay đổi
- Dùng như công cụ tham khảo cho bảo trì
- Sự ghi chép cẩn thận về vòng đời của HT, nghĩa là dự đoán các thay đổi
- Các đáp ứng với các sự cố không mong đợi



# 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)

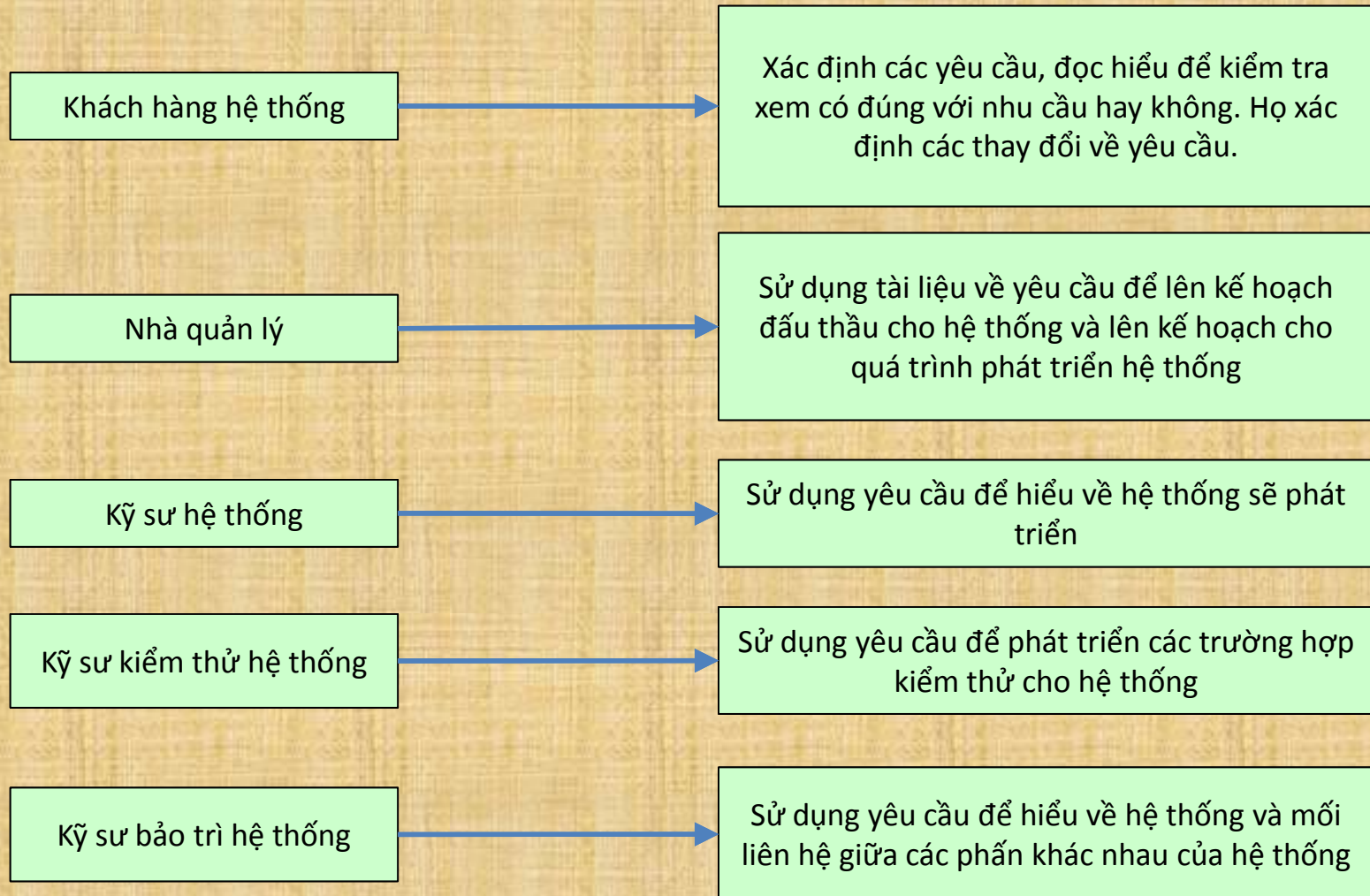
→ Khó phát biểu chính xác, Rất khó kiểm tra

# 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



### 3. Phương pháp và công cụ đặc tả yêu cầu phần mềm

- Biểu đồ phân cấp chức năng - WBS (work break down structure)
- Biểu đồ luồng dữ liệu – DFD (data flow diagram)
- Máy trạng thái – FSM (Finite state machine)
- Sơ đồ thực thể liên kết – ERD (entity relation diagram)

# Đặc tả chức năng với DFD

- Hệ thống (System): tập hợp các dữ liệu (data) được xử lý bằng các chức năng tương ứng (functions)
- Các ký pháp sử dụng:



Thể hiện các chức năng (functions)



Thể hiện luồng dữ liệu



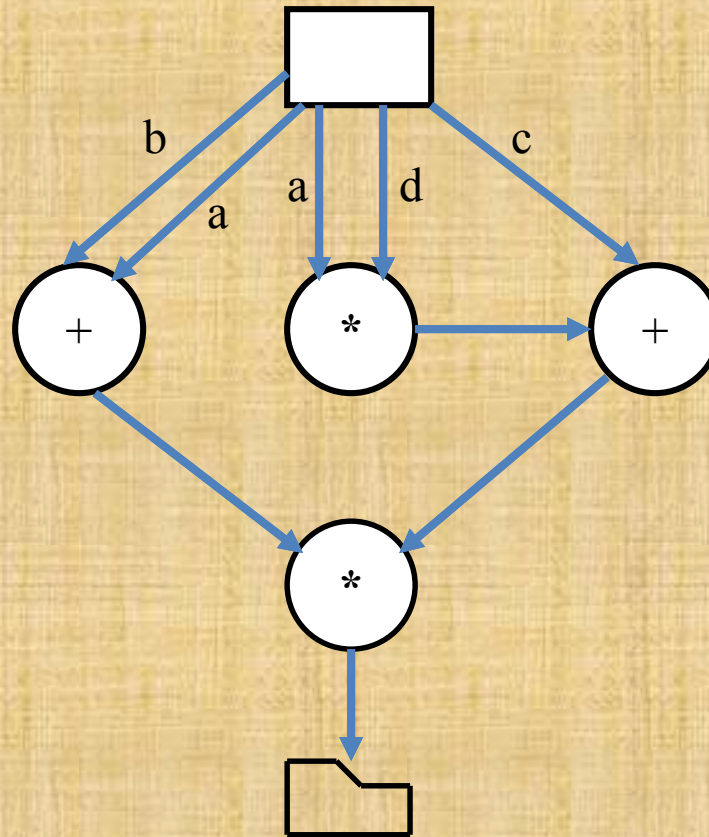
Kho dữ liệu



Vào ra dữ liệu và tương tác giữa hệ thống và người sử dụng

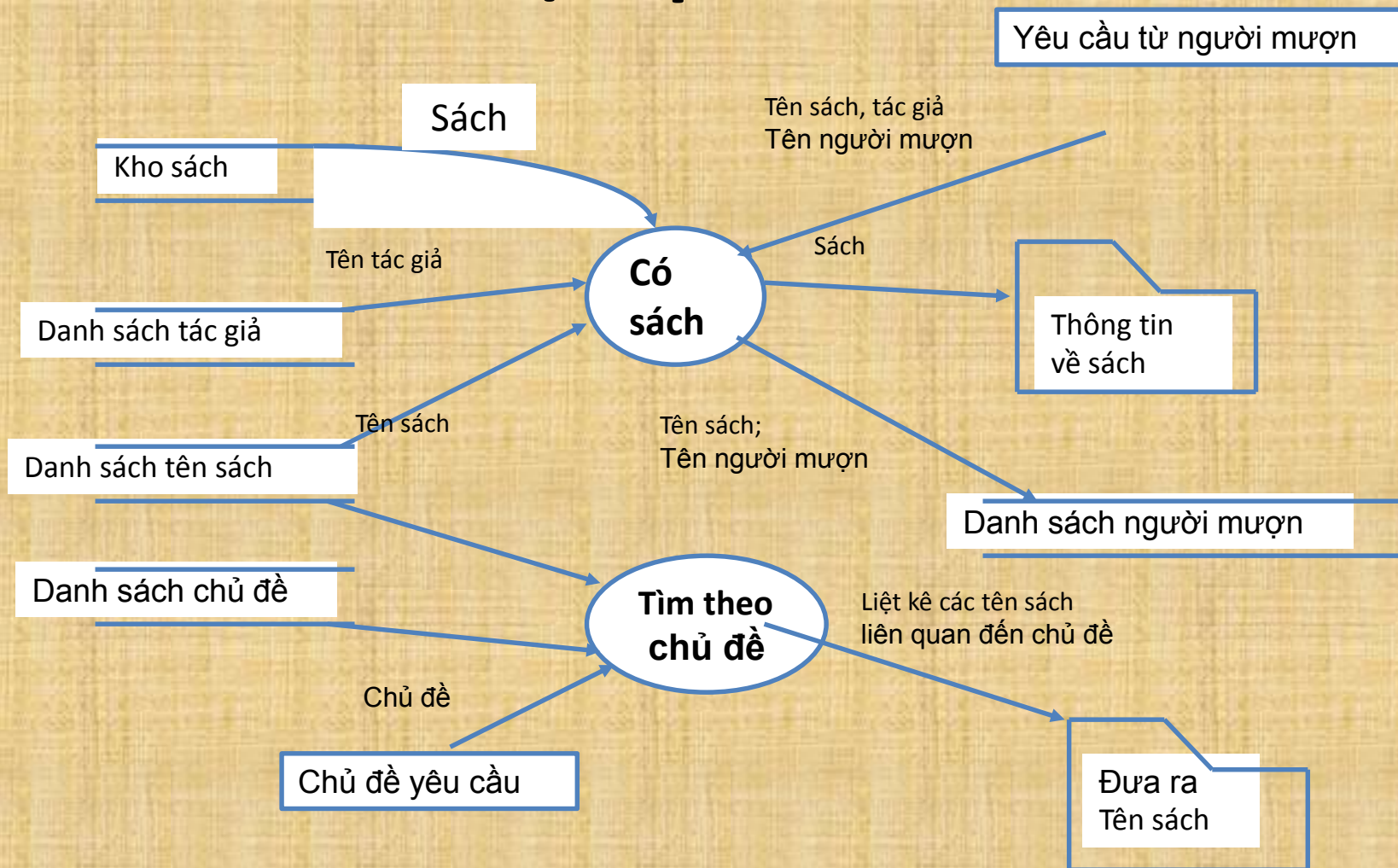
# Ví dụ: mô tả biểu thức toán học bằng DFD

$(a+b)*(c+a*d)-e*(a+b)$





# Ví dụ đặc tả các chức năng của thư viện qua DFD

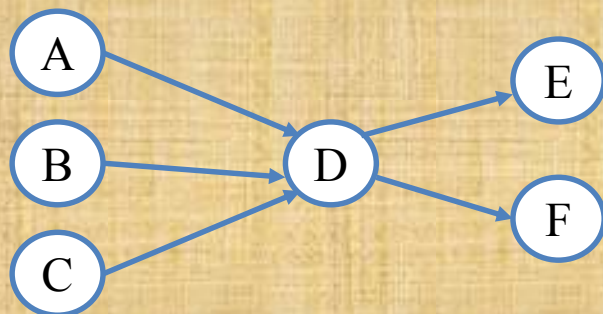


# Các hạn chế của DFD

- Ý nghĩa của các ký pháp sử dụng được xác định bởi các định danh lựa chọn của NSD
- Ví dụ: DFD của chức năng tìm kiếm sách:  
If NSD nhập vào cả tên tác giả và tiêu đề sách Then  
tìm kiếm sách tương ứng, không có thì thông báo lỗi  
Elseif chỉ nhập tên tác giả Then  
hiển thị danh sách các sách tương ứng với  
tên tác giả đã nhập và yêu cầu NSD lựa chọn sách  
Elseif chỉ nhập tiêu đề sách Then  
...  
Endif

# Các hạn chế của DFD

- Trong DFD không xác định rõ các hướng thực hiện (control aspects)



- Chức năng D có thể cần cả A, B và C
  - Chức năng D có thể chỉ cần một trong A, B và C để thực hiện
  - Chức năng D có thể kết xuất kết quả cho một trong E và F
  - Chức năng D có thể kết xuất kết quả chung cho cả E và F
  - Chức năng D có thể kết xuất kết quả riêng cho cả E và F
- Biểu đồ DFD này không chỉ rõ đầu vào là gì để thực hiện chức năng D và đầu ra là gì sau khi thực hiện chức năng D.

# Các hạn chế của DFD

- DFD không xác định sự đồng bộ giữa các chức năng / mô-đun
  - A xử lý dữ liệu và B được hưởng (nhận) các kết quả được xử lý từ A
  - A và B là các chức năng không đồng bộ (asynchronous activities) vì thế cần có buffer để ngăn chặn tình trạng mất dữ liệu





# Đặc tả trạng thái với FSM - Finite State Machines

- FSM chứa
  - Tập hữu hạn các trạng thái  $Q$
  - Tập hữu hạn các đầu vào  $I$
  - Các chức năng chuyển tiếp

$$\delta : Q \times I \rightarrow Q$$



# Ví dụ: quản lý thư viện

- Xét các giao dịch:
  - Mượn sách / Trả sách
  - Thêm đầu sách / Loại bỏ đầu sách
  - Liệt kê danh sách các đầu sách theo tên tác giả hay theo chủ đề
  - Tìm kiếm sách theo các yêu cầu của người mượn
  - Tìm kiếm sách quá hạn trả, . . .

# Đặc tả yêu cầu

- Độc giả không được mượn quá một số lượng sách nhất định, trong một thời gian nhất định
- Một số sách không được mượn về
- Một số người không được mượn một số loại sách nào đó, . . .

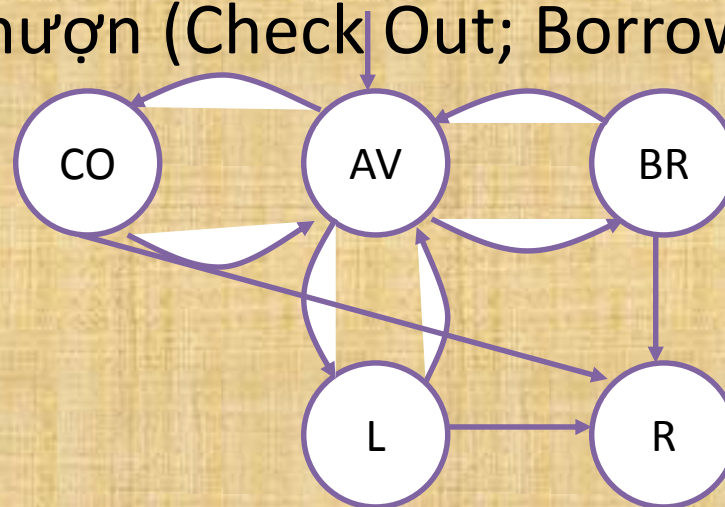
# Đặc tả các đối tượng

- Các đối tượng:
  - Tên sách
  - Mã quyền
  - Nhân viên phục vụ
  - Người mượn
- Cần có:
  - tập hợp (danh sách) các tiêu đề sách
  - danh sách các tác giả cho từng quyền sách,
  - danh sách các chủ đề liên quan của các quyền sách



# FSM đặc tả trạng thái

- Ta có tập hợp các sách (mỗi đầu sách có thể có nhiều quyển sách trong thư viện).
- Mỗi quyển sách có thể có 1 trong 5 trạng thái sau:
  - (AV) Available: được phép mượn,
  - (CO) - (BR): đã mượn (Check Out; Borrow),
  - (L): Last,
  - (R): Remove



# Đặc tả dữ liệu với Mô hình thực thể liên kết -ERD

- Mô hình khái niệm cho phép đặc tả các yêu cầu logic của hệ thống, thường được sử dụng trong các hệ thống dữ liệu lớn
- ER Model
  - Thực thể
  - Quan hệ
  - Thuộc tính
- Biểu đồ thực thể

# Thực thể

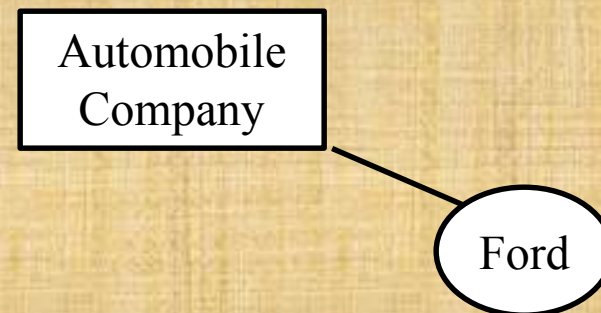
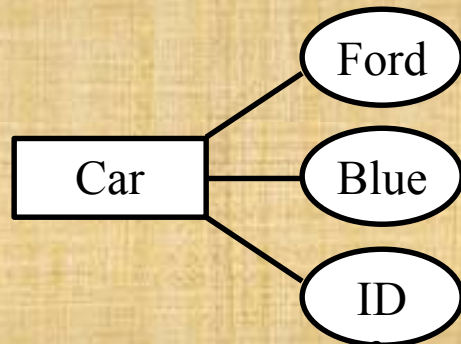
- Thực thể : tập hợp các thông tin liên quan cần được xử lý trong phần mềm
- Thực thể có thể có mối quan hệ:
  - người sở hữu ô tô



- Thực thể có các thuộc tính

# Thuộc tính

- Tính chất của một thực thể hoặc một đối tượng dữ liệu
  - đặt tên cho 1 mẫu (instance) của đối tượng dữ liệu
  - mô tả mẫu (instance)
  - tạo liên kết (reference) đến các mẫu khác

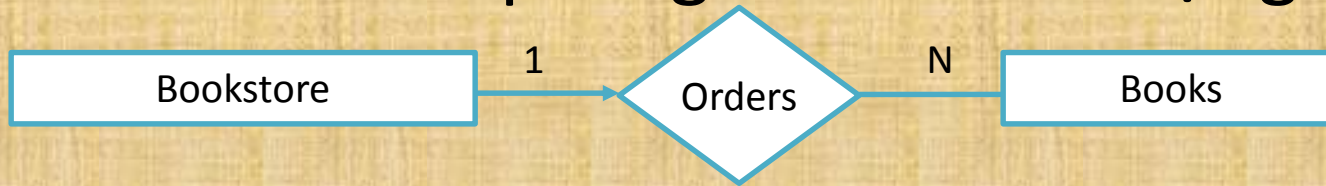


*Tập các thuộc tính của 1 đối tượng dữ liệu được xác định thông qua ngữ cảnh của bài toán.*



# Quan hệ

- Chỉ ra mối liên quan giữa các đối tượng dữ liệu

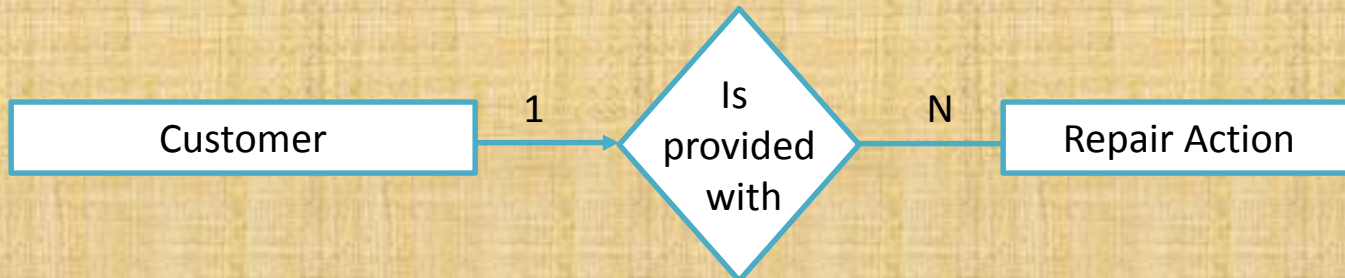


➤ Cardinality : chỉ ra định lượng của mối quan hệ

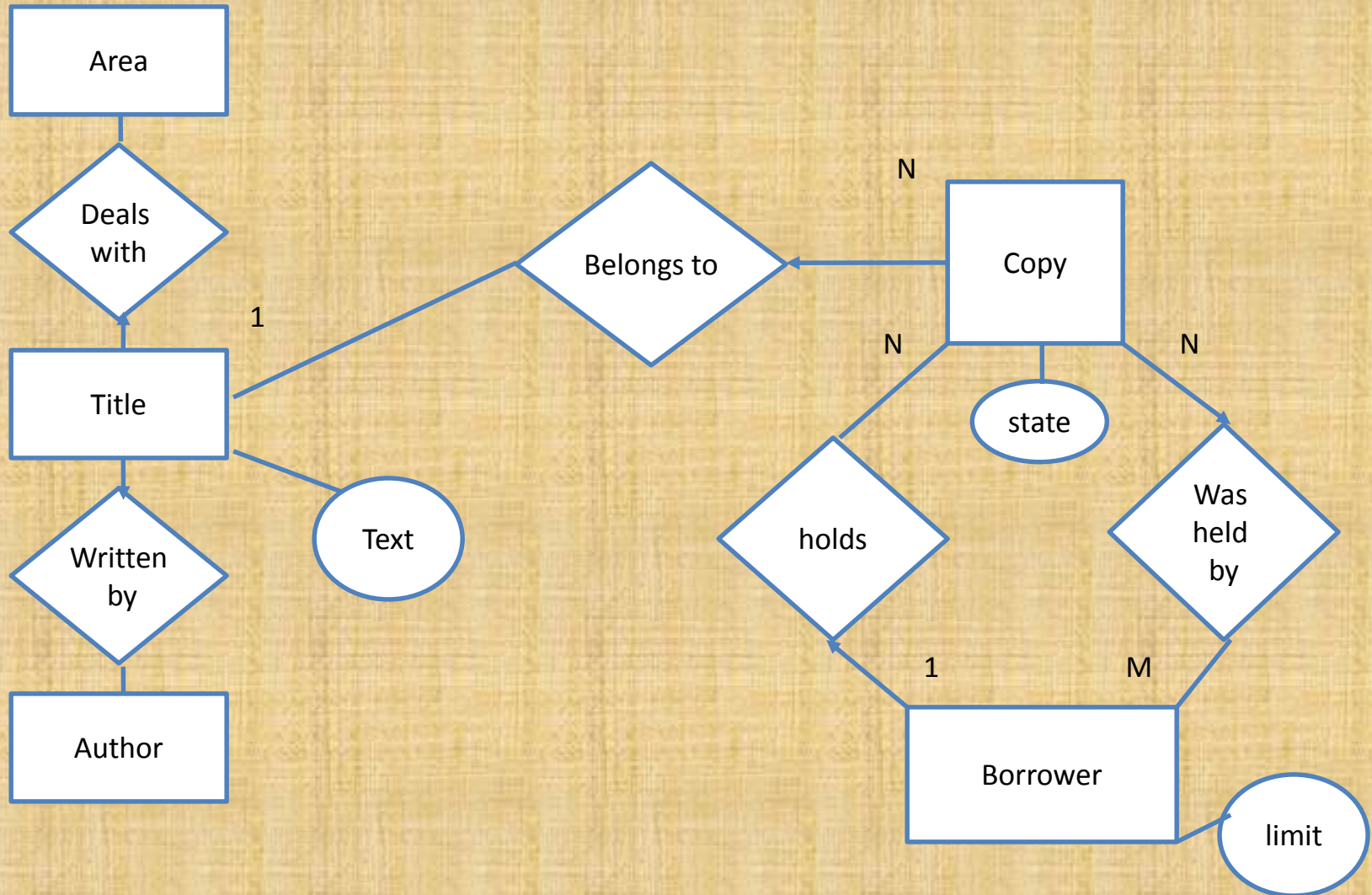
1:1 one-to-one 1:N one-to-many M:N many-to-many

➤ Modality : 0 – có thể có, có thể không có quan hệ

1 – bắt buộc có quan hệ



# Ví dụ: ERD mô tả thư viện



# So sánh

DFD	FSM	ERD
Đơn giản, dễ hiểu.	Có thể phức tạp với số lượng trạng thái lớn	Đơn giản, dễ hiểu
Mô tả luồng dữ liệu	Mô tả trạng thái của thực thể	Mô tả trừu tượng cơ sở dữ liệu
Không xác định rõ hướng thực hiện	Xác định rõ hướng thực hiện	Không xác định rõ hướng thực hiện
Không thể hiện tính tuần tự hay song song của tiến trình	Thể hiện tốt tính song song và tuần tự	Không thể hiện tính tuần tự hay song song

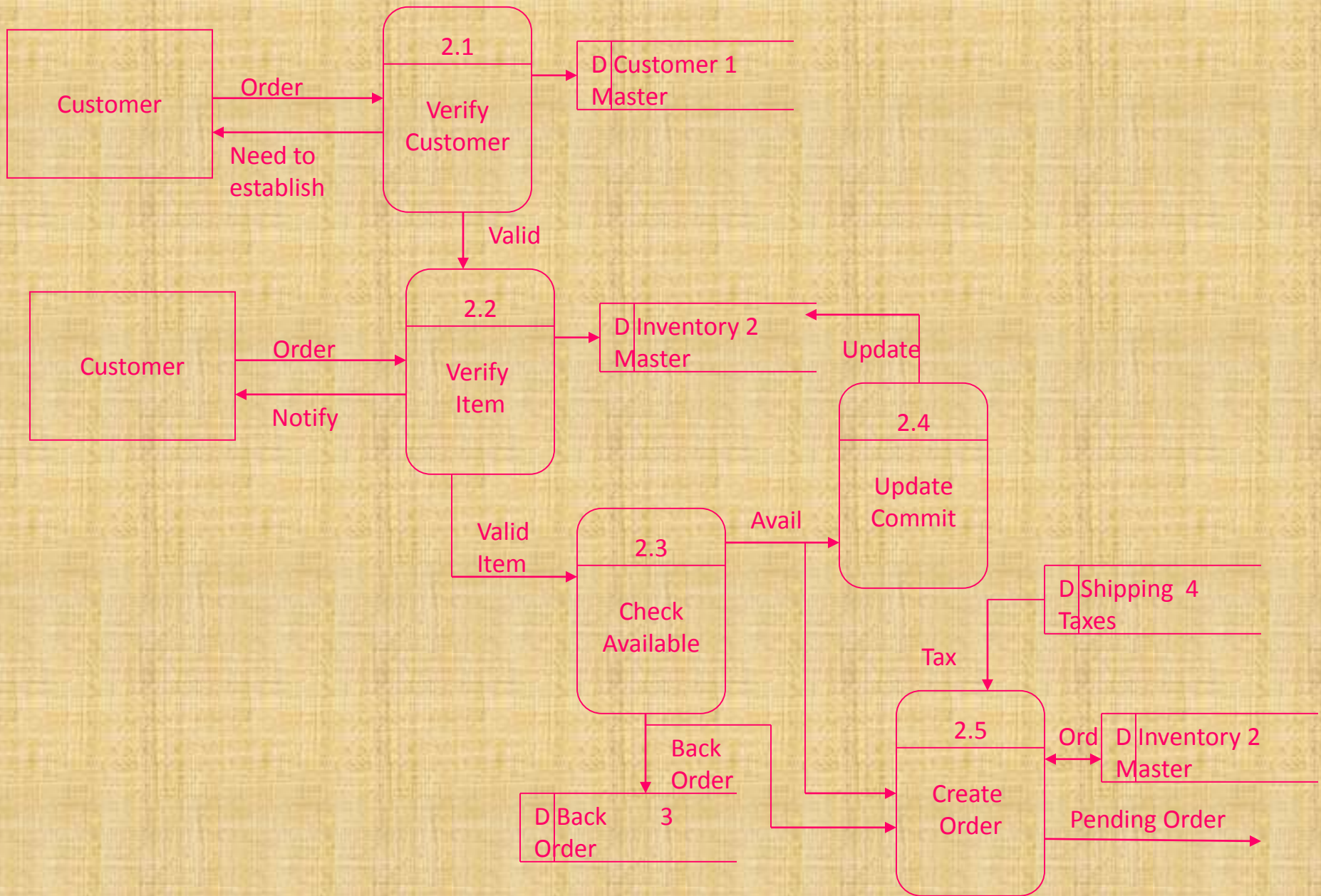
# Thế nào là một đặc tả tốt?

- Dễ hiểu với người dùng
- Có ít điều nhập nhằng
- Có ít quy ước khi mô tả, có thể tạo đơn giản
- Với phong cách từ trên xuống (topdown)
- Dễ triển khai cho những pha sau của vòng đời: thiết kế hệ thống và thiết kế chương trình và giao diện dễ làm, đảm bảo tính nhất quán, . . .



# Thế nào là một đặc tả tốt?





# 4. Nguyên lý phân tích yêu cầu sử dụng

## Mô hình hóa dữ liệu

- Xác định các đối tượng dữ liệu
- Xác định các đặc tính của các đối tượng dữ liệu
- Thiết lập các mối quan hệ giữa các đối tượng dữ liệu

# Mô hình hóa các chức năng

- Xác định các chức năng chuyển đổi đối tượng dữ liệu
- Chỉ ra luồng dữ liệu đi qua hệ thống như thế nào
- Biểu diễn bộ phận sản sinh dữ liệu và bộ phận tiêu thụ dữ liệu



# Mô hình hóa hành vi

- Chỉ ra các trạng thái (states) khác nhau của hệ thống
- Đặc tả các hiện tượng (events) làm hệ thống thay đổi trạng thái

# Phân mảnh các mô hình

- Tinh lọc từng mô hình để biểu diễn các mức trừu tượng thấp hơn
- Lọc đối tượng dữ liệu
- Tạo ra phân cấp chức năng
- Biểu diễn hành vi (behavior) ở các mức chi tiết khác nhau

# Bản chất của việc phân tích

- Hãy bắt đầu bằng cách tập trung vào bản chất của vấn đề chứ không xem xét những chi tiết cài đặt