

Công nghệ phần mềm

Chương 03: Phân tích yêu cầu và đặc tả phần mềm



Giảng viên: Lê Thị Hoàng Anh

Email: anhlt@huce.edu.vn



Nội dung

→ 1. Phân tích yêu cầu và mô tả hoạt động

2. Đặc tả phần mềm

- Cách đặc tả và biểu diễn
- Các nguyên lý đặc tả
- Các mức trừu tượng của đặc tả
- Đặc tả yêu cầu
- Dàn bài đặc tả yêu cầu phần mềm
- Xét duyệt đặc tả

3. Kỹ nghệ hệ thống và tạo nguyên mẫu



Nội dung

4. Phân tích dữ liệu với mô hình thực thể liên kết và mô hình quan hệ

- Các khái niệm và biểu diễn đồ họa
- Phương pháp lập lược đồ dữ liệu
 - Mô tả bằng sơ đồ thực thể liên kết
 - Ánh xạ từ mô hình thực thể liên kết sang mô hình quan hệ
 - Xây dựng lược đồ CSDL quan hệ
- Biểu diễn lược đồ dữ liệu ở dạng bảng

Phân tích yêu cầu và mô tả hoạt động

- Quá trình xác định các chức năng và các ràng buộc của hệ thống gọi là tìm hiểu và xác định yêu cầu.
- Phân tích và đặc tả yêu cầu là:
 - Là bước kỹ thuật đầu tiên trong tiến trình kỹ nghệ phần mềm. Hoạt động phân tích và định rõ yêu cầu hướng tới đặc tả yêu cầu phần mềm.
 - Là sự phối hợp giữa nhà phát triển với khách hàng
 - Khâu quan trọng, quyết định chất lượng phần mềm đạt được với chi phí dự kiến và thời hạn cho trước.



Khái niệm yêu cầu phần mềm

- Là tất cả các mô tả từ trừu tượng đến chi tiết về dịch vụ mà hệ thống phần mềm cung cấp, do khách hàng - người sử dụng phần mềm - nêu ra
- Bao gồm:
 - Các chức năng của phần mềm.
 - Hiệu năng của phần mềm.
 - Các yêu cầu về thiết kế và giao diện.
 - Các yêu cầu đặc biệt khác ràng buộc đến hoạt động và sự phát triển của hệ thống.



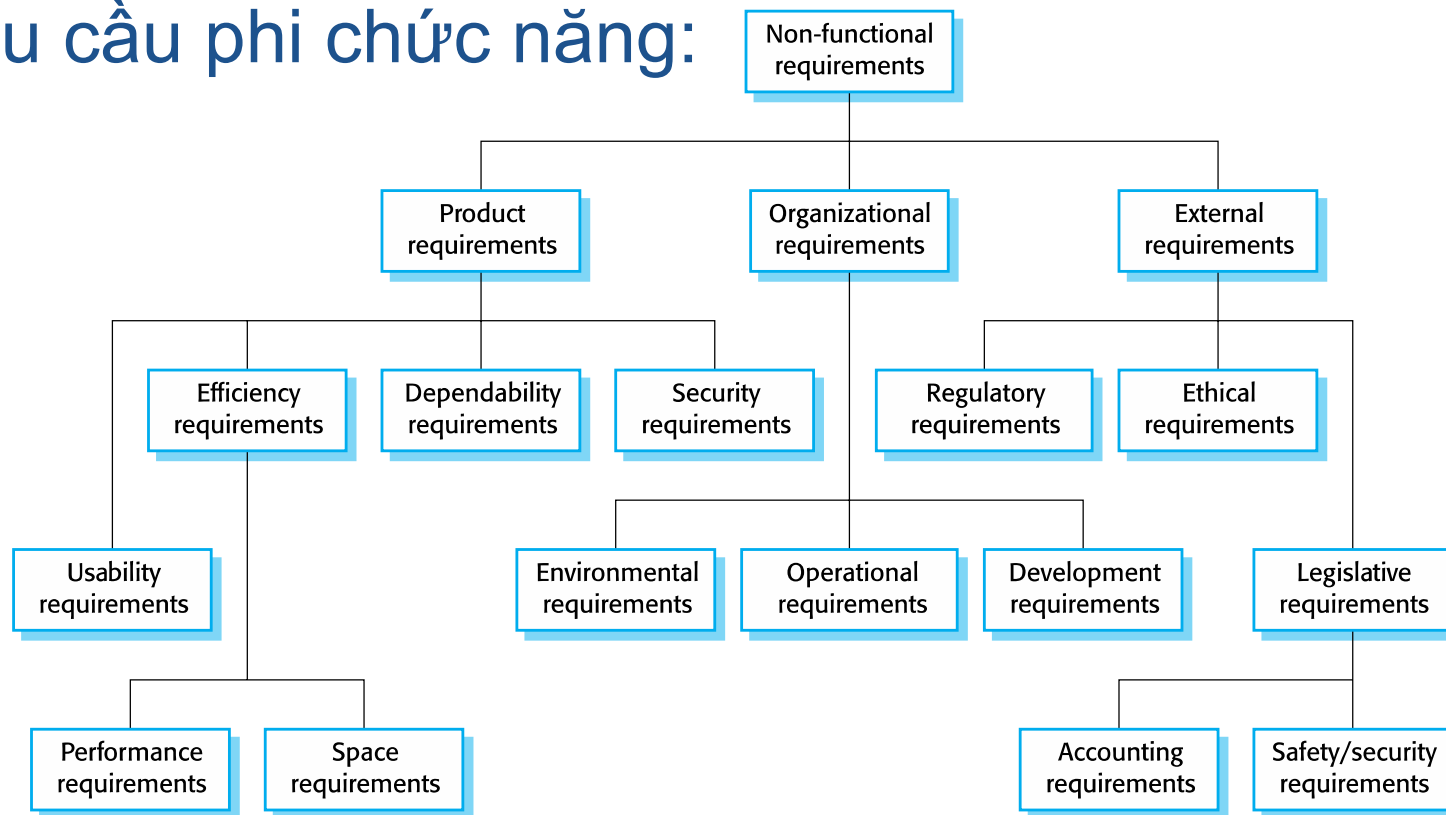
Khái niệm yêu cầu phần mềm

- Thông thường yêu cầu phần mềm được 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)



Khái niệm yêu cầu phần mềm

- Yêu cầu phần mềm chia thành 2 loại:
 - Yêu cầu chức năng: Cần mô tả chi tiết
 - Yêu cầu phi chức năng:





Tại sao cần đặt ra yêu cầu PM

- 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



Mục đích xác định và đặc tả các yêu cầu phần mềm

- Làm cơ sở cho việc mời thầu (cần có giải thích)
- Làm cơ sở ký kết hợp đồng thầu (cần đủ chi tiết)
- Làm tư liệu đầu vào cho thiết kế và triển khai (cần đầy đủ, chính xác, không mâu thuẫn)



Phân loại các tài liệu yêu cầu

- Xác định yêu cầu:
 - Là ***mô tả trừu tượng các dịch vụ*** mà hệ thống được mong đợi phải cung cấp và ***các ràng buộc*** mà hệ thống phải tuân thủ khi vận hành.
- Đặc tả yêu cầu phần mềm:
 - Là tài liệu cấu trúc mô tả hệ thống các dịch vụ được chi tiết hơn.
 - Là mô tả trừu tượng hơn của phần mềm làm cơ sở cho thiết kế và triển khai.



Yêu cầu đối với tư liệu các yêu cầu phần mềm

Heninger đòi hỏi 6 yêu cầu cho tư liệu các yêu cầu phần mềm:

1. Chỉ đặc tả tính chất bên ngoài của hệ thống
2. Đặc tả các ràng buộc về sự thực hiện
3. Phải là dễ thay đổi
4. Phải được dùng làm công cụ tham khảo cho người bảo trì hệ thống
5. Phải báo cáo dự tính trước về vòng đời của hệ thống
6. Phải đặc trưng hóa các đáp ứng chấp nhận được cho các sự kiện bất ngờ



Yêu cầu đối với cấu trúc của một tư liệu

Cấu trúc của một tư liệu yêu cầu được gợi ý theo kết cấu sau:

1. Phần dẫn nhập
2. Phần mô hình hệ thống
3. Phần tiến triển của hệ thống
4. Phần các yêu cầu chức năng
5. Phần từ điển thuật ngữ



Yêu cầu đối với cấu trúc của một tư liệu

Cấu trúc của một tư liệu yêu cầu được gợi ý theo kết cấu sau:

Chapter	Description
Preface	This should define the expected readership of the document and describe its version history, including a rationale for the creation of a new version and a summary of the changes made in each version.
Introduction	This should describe the need for the system. It should briefly describe the system's functions and explain how it will work with other systems. It should also describe how the system fits into the overall business or strategic objectives of the organization commissioning the software.
Glossary	This should define the technical terms used in the document. You should not make assumptions about the experience or expertise of the reader.
User requirements definition	Here, you describe the services provided for the user. The nonfunctional system requirements should also be described in this section. This description may use natural language, diagrams, or other notations that are understandable to customers. Product and process standards that must be followed should be specified.
System architecture	This chapter should present a high-level overview of the anticipated system architecture, showing the distribution of functions across system modules. Architectural components that are reused should be highlighted.



Nhiệm vụ phân tích yêu cầu

- Là một tiến trình khám phá, làm mịn, mô hình hóa và đặc tả. Phạm vi phần mềm ban đầu do người phân tích thiết lập sau đó được làm mịn dần trong việc lập kế hoạch dự án phần mềm.
- Cả người phát triển và khách hàng đều đóng vai trò quan trọng trong việc phân tích và đặc tả yêu cầu.



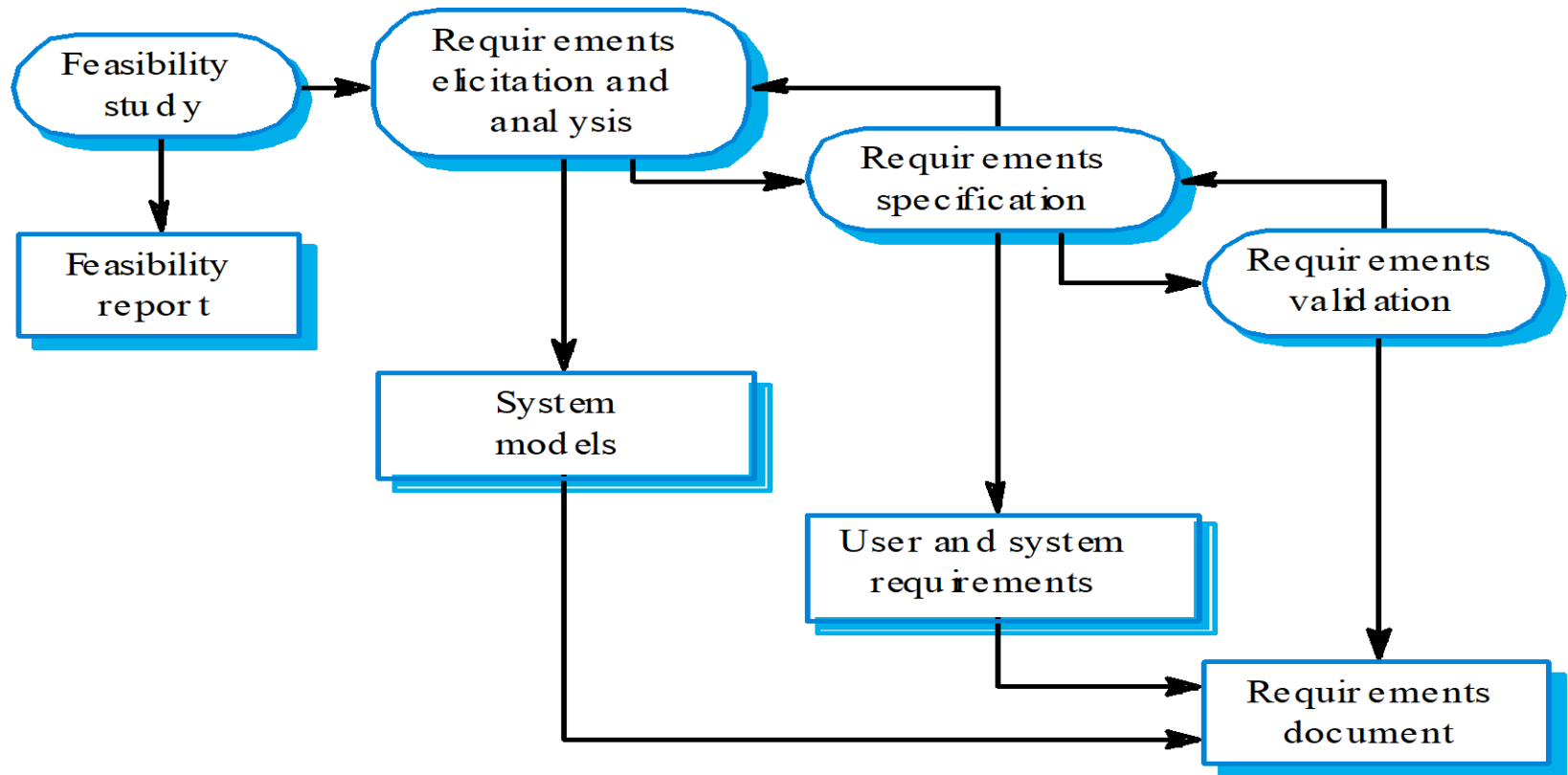
Nhiệm vụ phân tích yêu cầu

- Trong giai đoạn phân tích có hai đòi hỏi cơ bản để đạt được phần mềm tốt:
 - Các yêu cầu phần mềm phải bộc lộ theo phương thức "trên - xuống". Các chức năng, giao diện và luồng thông tin giữa các chức năng đó chủ yếu phải được hiểu hoàn toàn rõ trước khi xác định chi tiết các tầng lớp kế tiếp.
 - Người phân tích yêu cầu cũng cần hiểu rõ quy trình phát triển phần mềm và đánh giá được các bước kỹ nghệ phần mềm tổng quát, từ đó áp dụng được bất kể quy trình kỹ nghệ nào cần dùng.



Quy trình kỹ nghệ yêu cầu

- Quy trình kỹ nghệ yêu cầu (Requirements Engineering)





Quy trình kỹ nghệ yêu cầu

1. Nghiên cứu khả thi (Feasibility study);
2. Thu thập yêu cầu (Requirements elicitation);
3. Phân tích yêu cầu và thương lượng
(Requirements analysis and negotiation);
4. Đặc tả yêu cầu (Requirement specification)
5. Mô hình hóa yêu cầu hệ thống (System requirements modeling)
6. Thẩm định yêu cầu (Requirements validation);
7. Quản trị yêu cầu (Requirements management).



1. Nghiên cứu tính khả thi

- 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?



2. Thu thập yêu cầu

- Kỹ sư phần mềm làm việc với các stakeholder để tìm ra " Miền ứng dụng ".
- Các vấn đề của thu thập yêu cầu phần mềm:
 - Phạm vi của phần mềm (Scope)
 - Hiểu rõ phần mềm (Understanding)
 - Các thay đổi của hệ thống (Volatility)



2. Thu thập yêu cầu

- Phương pháp phát hiện yêu cầu phần mềm (**Requirements Elicitation Methodology**)
 - Xác định các phương pháp sử dụng phát hiện các yêu cầu phần mềm: phỏng vấn (closed interviews và open interviews) bằng điều tra câu hỏi, làm việc nhóm, các buổi họp, gặp gỡ đối tác, v.v.
 - Tìm kiếm các nhân sự (chuyên gia, người sử dụng)
 - Tự quan sát các quy trình nghiệp vụ.



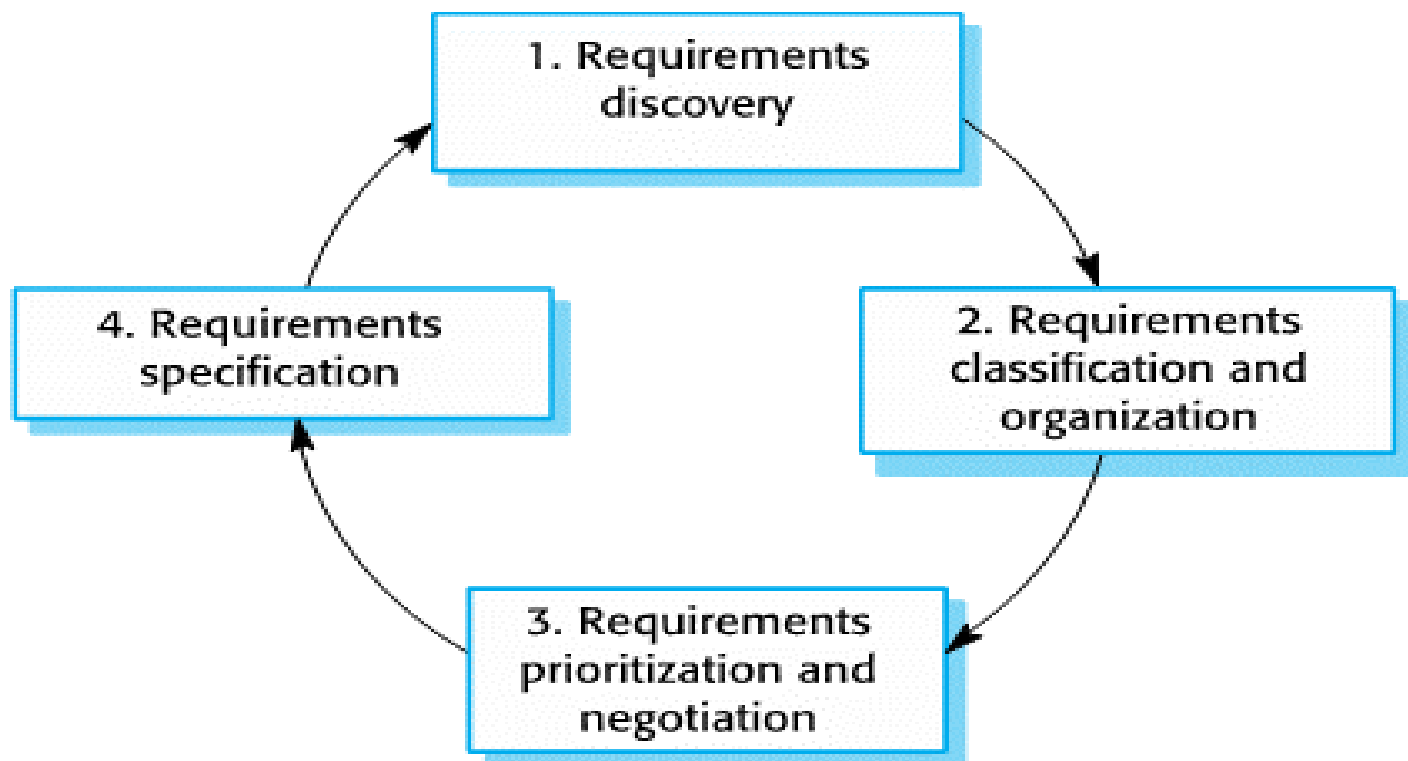
2. Thu thập yêu cầu

- Bao gồm các giai đoạn:
 - Phát hiện yêu cầu.
 - Phân loại các yêu cầu phần mềm và tổ chức chúng theo các nhóm liên quan.
(3) Chính là phân tích yêu cầu
 - Đàm phán, thương lượng và đánh thứ tự ưu tiên. *(3) Chính là phân tích yêu cầu và thương lượng*
 - Kết quả là bản đặc tả yêu cầu(4). *Chi tiết hơn về bản đặc tả yêu cầu chúng ta sẽ tìm hiểu sau.*



2. Thu thập yêu cầu

- Chúng ta có thể lặp lại nhiều lần quá trình phân tích và thu thập yêu cầu.





2. Thu thập yêu cầu

- Sản phẩm của việc 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

3. Phân tích yêu cầu và thương lượng



3. Phân tích yêu cầu và thương lượng

- 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.
- Phân cấp các yêu cầu phần mềm theo dựa trên nhu cầu và đòi hỏi khách hàng / người sử dụng

3. Phân tích yêu cầu và thương lượng

- Phân tích các rủi ro có thể xảy ra với từng yêu cầu phần mềm.
- Đánh giá thô 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 trên cơ sở thảo luận và thương lượng các yêu cầu đề ra



4. Đặc tả yêu cầu phần mềm

- Các khía cạnh phải đặc tả trong phát triển phần mềm:
 - Đặc tả vận hành chức năng (Operational specifications) mô tả các hoạt động của hệ thống phần mềm sẽ xây dựng.
 - Đặc tả mô tả/phi chức năng (Descriptive specifications) – đặc tả các đặc tính đặc trưng của phần mềm: Định nghĩa các tính chất của hệ thống, các ràng buộc, thí dụ như độ tin cậy, thời gian trả lời, dung lượng bộ nhớ,... Các yêu cầu do tổ chức qui định như qui định chuẩn về quá trình tiến hành, chuẩn tài liệu,...



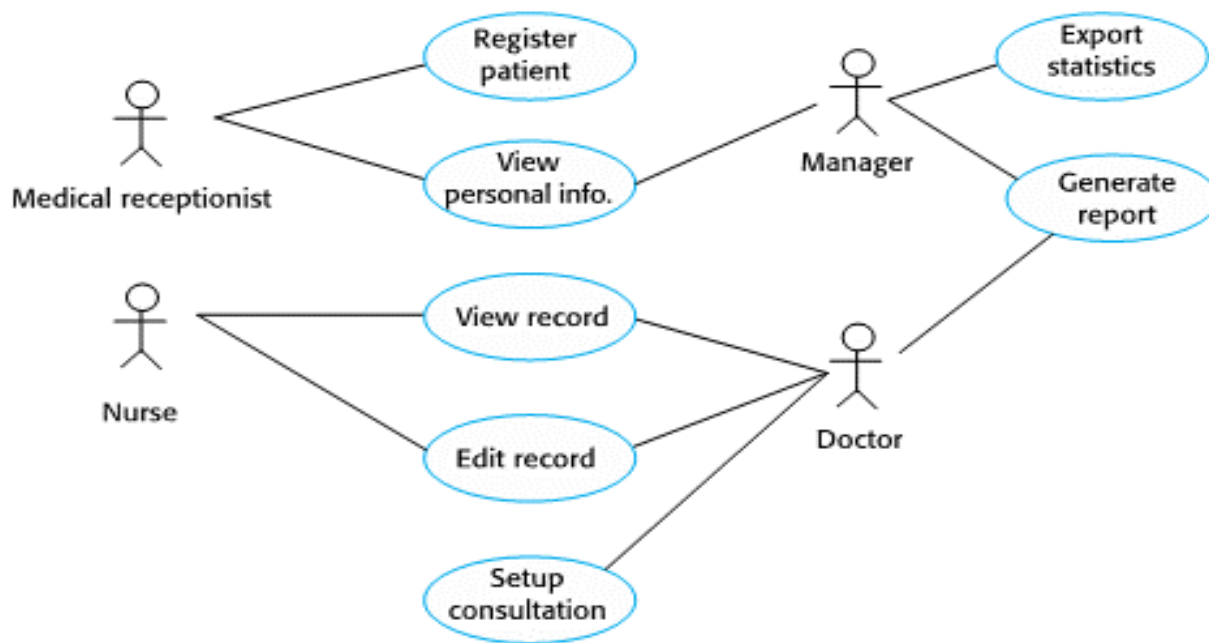
5. Mô hình hóa yêu cầu hệ thống

- Phải ước định được các thành phần của hệ thống trong mối quan hệ với nhau để xác định được yêu cầu như thế nào thì phù hợp với phần mềm và để đánh giá được mức chuyên nghiệp của hệ thống khi được hoàn thành.



5. Mô hình hóa yêu cầu hệ thống

- Use case: Là một dạng mô tả kịch bản trong UML. Một tập use case mô tả tất cả các tương tác có thể có với hệ thống.
Ví dụ: Hệ thống đăng ký khám bệnh





6. Thẩm định yêu cầu (Requirements Validation)

- Là việc kiểm tra rằng các yêu cầu được xác định ra ở trên có thực sự định nghĩa được hệ thống mà khách hàng cần.
- Vì chi phí để sửa lỗi yêu cầu cao, do đó việc thẩm định rất quan trọng. Việc kiểm tra bao gồm:
 - **Kiểm tra tính đúng đắn**
 - **Kiểm tra tính đầy đủ**
 - **Kiểm tra tính nhất quán**
 - **Kiểm tra tính hiện thực**
 - **Kiểm tra tính có thể kiểm tra được của yêu cầu.**



6. Thẩm định yêu cầu

1. Xem xét lại yêu cầu:

- Phân tích một cách có hệ thống
- Lấy ý kiến khách hàng
- Tiến hành thường xuyên

2. Làm bản mẫu:

- Sử dụng mô hình khả dụng
- Kiểm tra tính thực hiện được

3. Tạo ca kiểm thử (test case): kiểm tra tính kiểm tra được.

4. Sử dụng CASE: để kiểm tra tính nhất quán



7. Quản trị yêu cầu

- Quản trị yêu cầu (requirements management) là quy trình quản trị sự thay đổi yêu cầu trong suốt quá trình kỹ nghệ yêu cầu và phát triển hệ thống
 - Các yêu cầu mới phát sinh khi hệ thống đang được phát triển và cả khi nó được đưa vào sử dụng
 - Cần theo dõi những yêu cầu đơn lẻ và duy trì mối liên hệ giữa các yêu cầu phụ thuộc nhau để có thể đánh giá được ảnh hưởng khi thay đổi yêu cầu.
 - Cần thiết lập một quy trình hình thức cho những đề nghị thay đổi và tạo mối liên hệ giữa yêu cầu này với các yêu cầu hệ thống.



Nguyên nhân thay đổi yêu cầu

- Môi trường kinh doanh và công nghệ của hệ thống luôn thay đổi sau khi cài đặt xong phần mềm.
- Những người trả tiền cho hệ thống thường ít khi là những người dùng cuối của phần mềm. Do đó trong quá trình sử dụng có thể phát sinh ra các tính năng mới cần được thêm vào hệ thống để hỗ trợ người dùng.



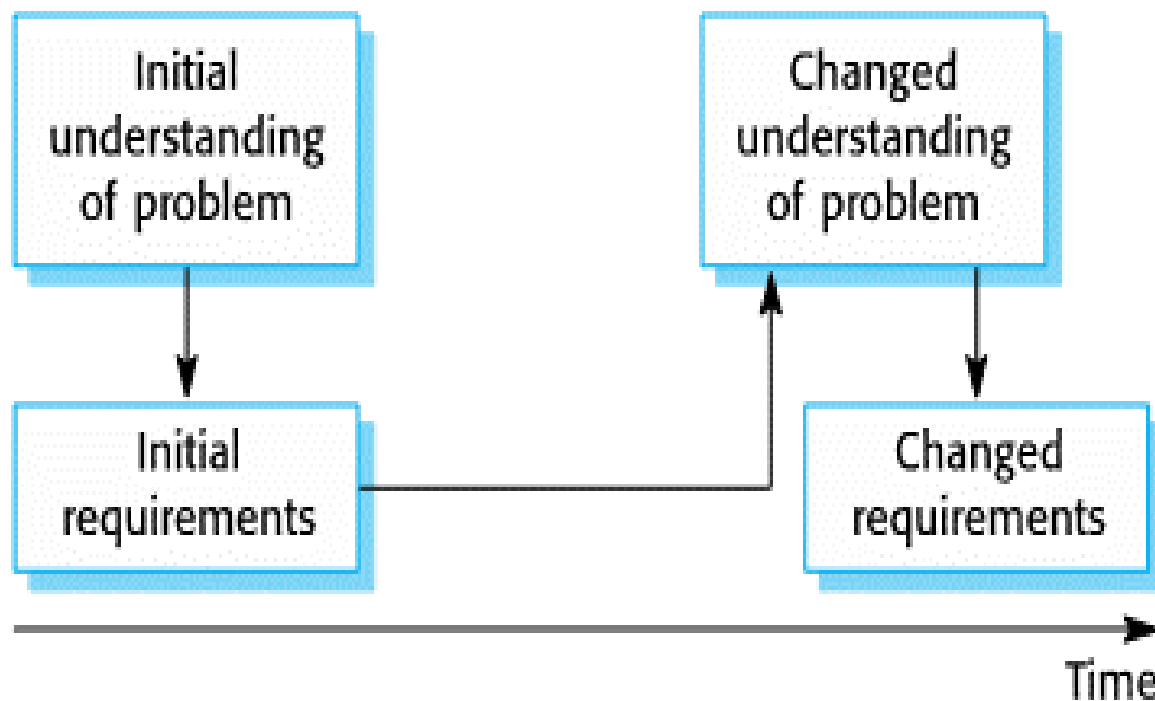
Nguyên nhân thay đổi yêu cầu

- Các hệ thống lớn thường có một cộng đồng người dùng đa dạng. Do vậy không tránh khỏi việc những người dùng này có yêu cầu và sự ưu tiên khác nhau có thể gây xung đột hoặc mâu thuẫn nên cần thỏa hiệp giữa họ. Và để cân bằng việc hỗ trợ các khách hàng khác nhau thì theo kinh nghiệm thường phải thay đổi yêu cầu phần mềm hệ thống.



7. Quản trị yêu cầu

- Cải tiến yêu cầu





7. Quản trị yêu cầu

- Quy tắc quản trị yêu cầu:
 - Mỗi yêu cầu phải là duy nhất.
 - Quá trình quản trị thay đổi là tập hợp của các hoạt động đánh giá tác động và chi phí thay đổi.
 - Chính sách truy đến yêu cầu: Phải có chính sách xác định mối quan hệ giữa mỗi yêu cầu và giữa các yêu cầu.
 - Phải có công cụ hỗ trợ quản trị yêu cầu.



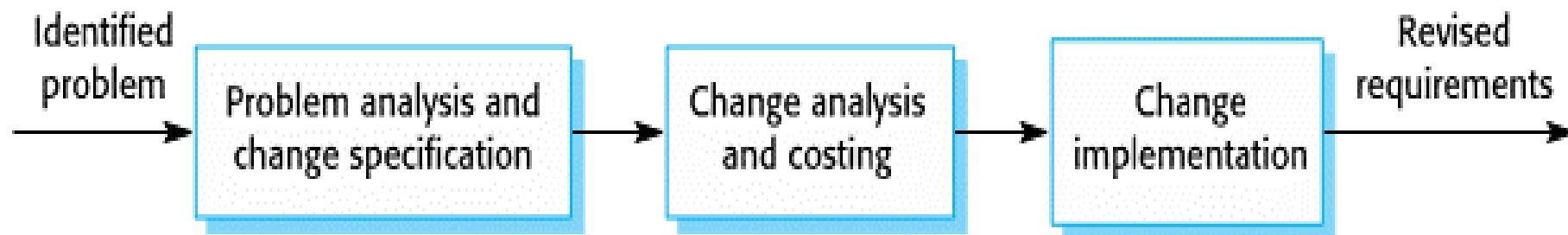
7. Quản trị yêu cầu

- Cân nhắc thay đổi yêu cầu:
 - Trong suốt quá trình này cần kiểm tra tính hợp lệ của những yêu cầu cần thay đổi, theo dõi phản ứng người dùng để quyết định thay đổi yêu cầu cụ thể hơn hay chỉ đơn thuần là thu hồi yêu cầu.
 - Phân tích sự thay đổi và chi phí.
 - Thay đổi việc thực thi hệ thống: Các tài liệu đặc tả yêu cầu được thay đổi và nếu cần thì việc thiết kế và thực thi hệ thống cũng được thay đổi.



7. Quản trị yêu cầu

- Quy trình quản trị yêu cầu





Các nguyên lý phân tích yêu cầu

NL1: Mô hình hóa dữ liệu:

- Phải hiểu và mô tả được miền thông tin
 - Xác định các đối tượng dữ liệu (thực thể)
 - Xác định các đặc tính của các đối tượng dữ liệu (thuộc tính)
 - Thiết lập các mối quan hệ giữa các đối tượng dữ liệu (quan hệ)



Các nguyên lý phân tích yêu cầu

NL2: Mô hình hóa các chức năng (Phải định nghĩa được các chức năng của phần mềm)

- Xác định các chức năng (biến đổi thông tin)
- Xác định cách thức dữ liệu (thông tin) di chuyển trong hệ thống (luồng dữ liệu)
- Xác định các tác nhân tạo dữ liệu (nguồn) và tác nhân tiếp nhận dữ liệu (đích)



Các nguyên lý phân tích yêu cầu

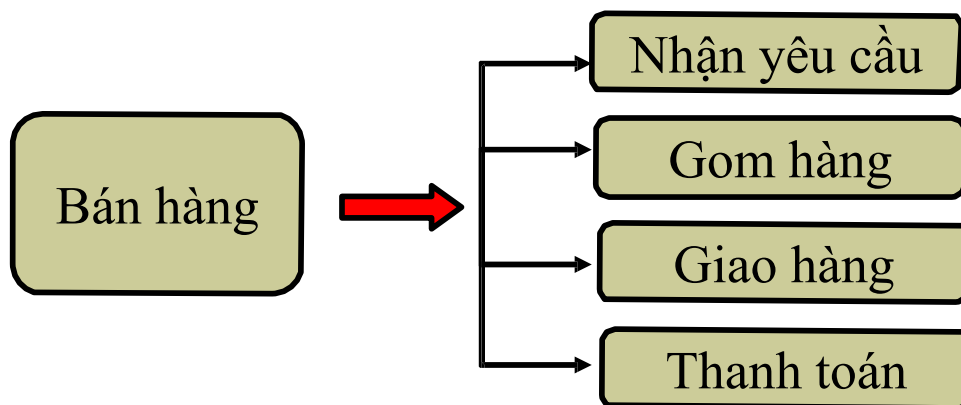
- NL3: Mô hình hóa hành vi (phải miêu tả được kết quả của các sự kiện bên ngoài)
- Xác định các trạng thái (states) khác nhau của hệ thống
 - Xác định các sự kiện (events) làm hệ thống thay đổi trạng thái



Các nguyên lý phân tích yêu cầu

NL4: Làm mịn, phân hoạch và biểu diễn các mô tả ở các mức chi tiết khác nhau

- Làm mịn các đối tượng dữ liệu
- Tạo ra biểu đồ phân cấp chức năng
- Biểu diễn hành vi (behavior) ở các mức chi tiết khác nhau. Ví dụ:





Các nguyên lý phân tích yêu cầu

NL5: Tập trung vào bản chất (Essence)

- Nhìn nhận bản chất của yêu cầu:
 - làm gì (chức năng)?
 - điều kiện gì (ràng buộc)?
- Không quan tâm đến cách thức cài đặt (làm như thế nào, khi nào, ở đâu? -> là quá trình vật lý)



Nội dung

1. Phân tích yêu cầu và mô tả hoạt động

→ 2. Đặc tả phần mềm

- Cách đặc tả và biểu diễn
- Các nguyên lý đặc tả
- Các mức trừu tượng của đặc tả
- Đặc tả yêu cầu
- Dàn bài đặc tả yêu cầu phần mềm
- Xét duyệt đặc tả

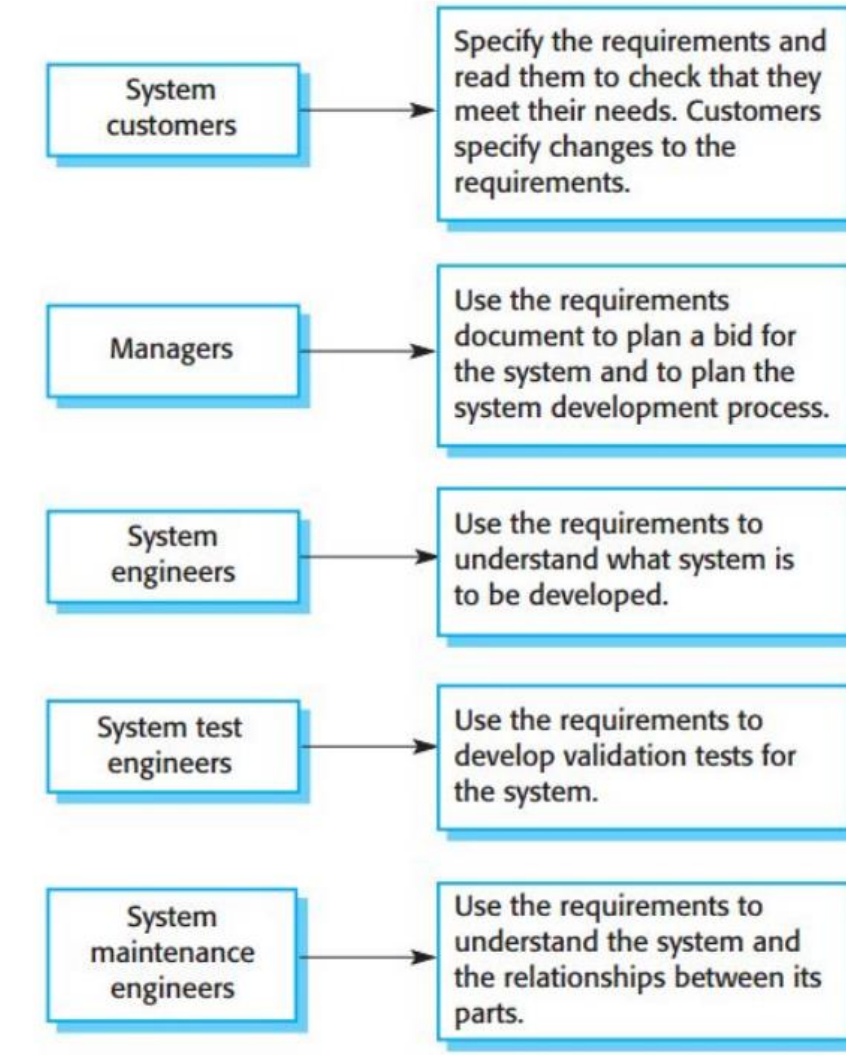


Đặc tả yêu cầu phần mềm

- Là công việc 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.
- Cách đặc tả và biểu diễn:
 - Đặc tả phi hình thức (Informal specifications) được viết bằng ngôn ngữ tự nhiên
 - Đặc tả hình thức (Formal specifications) được viết bằng tập các ký pháp có các quy định về *cú pháp (syntax)* và *ý nghĩa (sematic)* rất chặt chẽ



Đối tượng sử dụng của tài liệu yêu cầu





Cách đặc tả và biểu diễn

- Cách biểu diễn

- **Đặc tả chức năng (Operational Specifications):** thông thường khi đặc tả các chức năng của phần mềm người ta sử dụng các công cụ tiêu biểu sau
 - Biểu đồ phân rã chức năng (Function Decomposition Diagrams – FDD)
 - Biểu đồ luồng dữ liệu (Data Flow Diagrams – DFD)
 - Máy trạng thái hữu hạn (Finite State Machines – FSM)
 - Mạng Petri (Petri nets)
- **Đặc tả mô tả/phi chức năng (Descriptive Specifications)**
 - Biểu đồ thực thể liên kết (Entity-Relationship Diagrams - ERD)
 - Đặc tả Logic (Logic Specifications)
 - Đặc tả đại số (Algebraic Specifications)



Các mức trừu tượng của đặc tả

- Xác định yêu cầu
- Đặc tả yêu cầu
- Đặc tả phần mềm/đặc tả thiết kế (đây là một mô tả trừu tượng cho phần mềm)



Các nguyên lý đặc tả

- Nguyên lý 1: Đặc tả chức năng chứ không phải cài đặt.
- Nguyên lý 2: Ngôn ngữ đặc tả hệ thống hướng quy trình đôi khi được yêu cầu.
- Nguyên lý 3: Đặc tả phải bao gồm hệ thống trong đó phần mềm là một thành phần (đảm bảo tính hệ thống, thể hiện mối liên kết giữa phần mềm cần xây dựng với các thành phần khác)



Các nguyên lý đặc tả

- Nguyên lý 4: Đặc tả phải bao gồm cả môi trường mà hệ thống vận hành
- Nguyên lý 5: Đặc tả hệ thống phải là một mô hình nhận thức
- Nguyên lý 6: Đặc tả phải thể hiện tính vận hành
- Nguyên lý 7: Có khả năng cập nhật
- Nguyên lý 8: Đặc tả phải được cục bộ hóa và có khả năng lắp ghép – xuất phát từ lập trình hướng đối tượng



Đặc tả yêu cầu

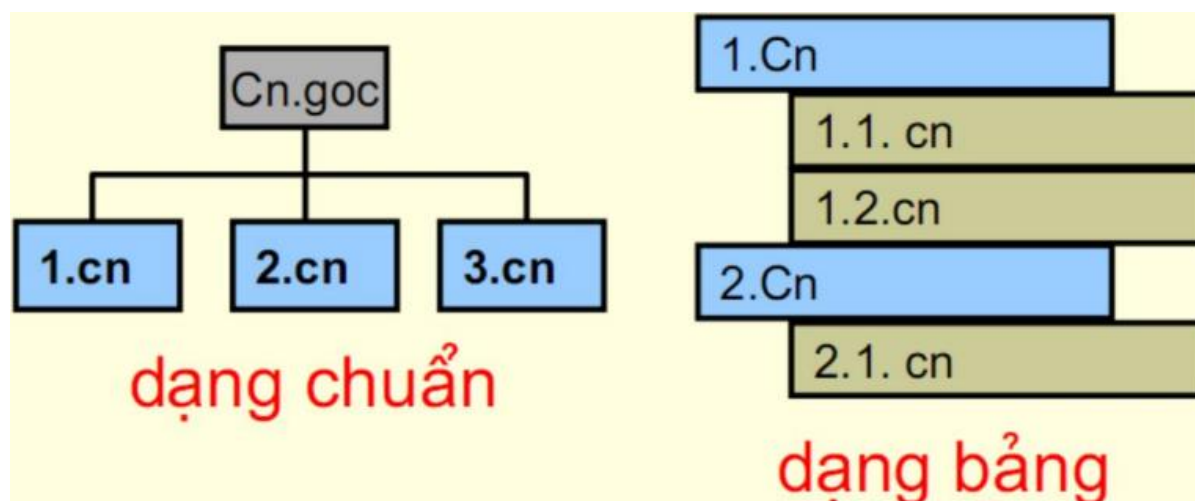
ĐT1: Biểu đồ phân rã chức năng

– FDD: Function Decomposition Diagram

- Xác định phạm vi của hệ thống
- Biểu thị sự phân hoạch chức năng
- Hệ thống có khả năng làm gì? Mức chi tiết đến đâu

Tên = động từ + bổ ngữ

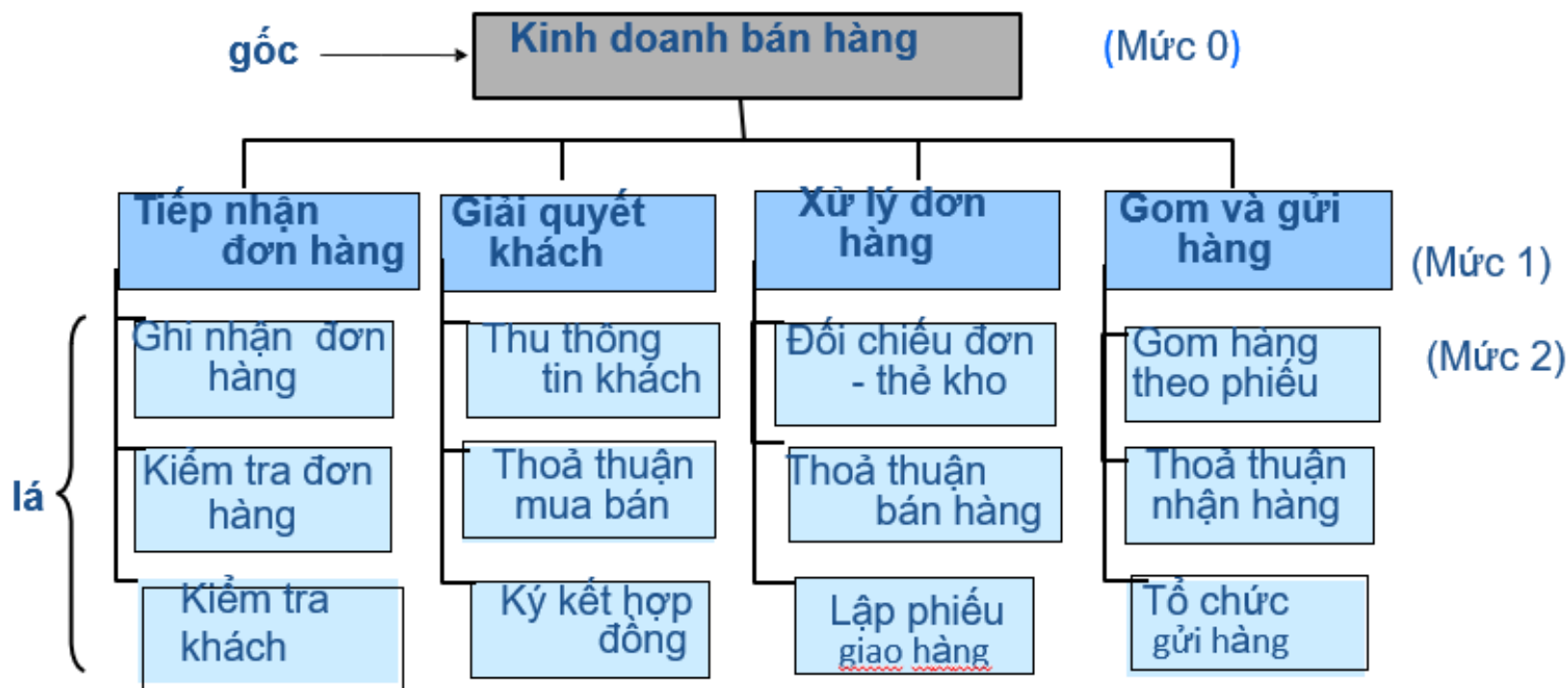
Chức năng





Đặc tả yêu cầu

- Biểu đồ phân rã chức năng- Ví dụ:



Biểu đồ phân rã chức năng nghiệp vụ kinh doanh bán hàng



Đặc tả yêu cầu

ĐT2: Biểu đồ luồng dữ liệu

– DFD: Data Flow Diagram

- Mô tả quá trình xử lý thông tin nghiệp vụ
- Biểu diễn cách thức dữ liệu di chuyển, được xử lý, lưu trữ trong hệ thống và trao đổi với môi trường
- Chỉ rõ các thông tin được chuyển giao giữa các chức năng và cho phép hình dung trình tự thực hiện của chúng

Trong kiểu đặc tả này, phần mềm được coi là một hệ thống gồm tập hợp các dữ liệu được xử lý bằng các chức năng tương ứng



Đặc tả yêu cầu

- Biểu đồ luồng dữ liệu: (tiếp)
 - Tác nhân:
 - Đối tượng (người, tổ chức) ngoài hệ thống
 - Phát sinh hoặc tiếp nhận dữ liệu
 - Tiến trình: chức năng nghiệp vụ
 - Dây hoạt động tác động lên dữ liệu
 - Luồng dữ liệu: di chuyển dữ liệu
 - Kho dữ liệu: nơi dữ liệu được lưu trữ

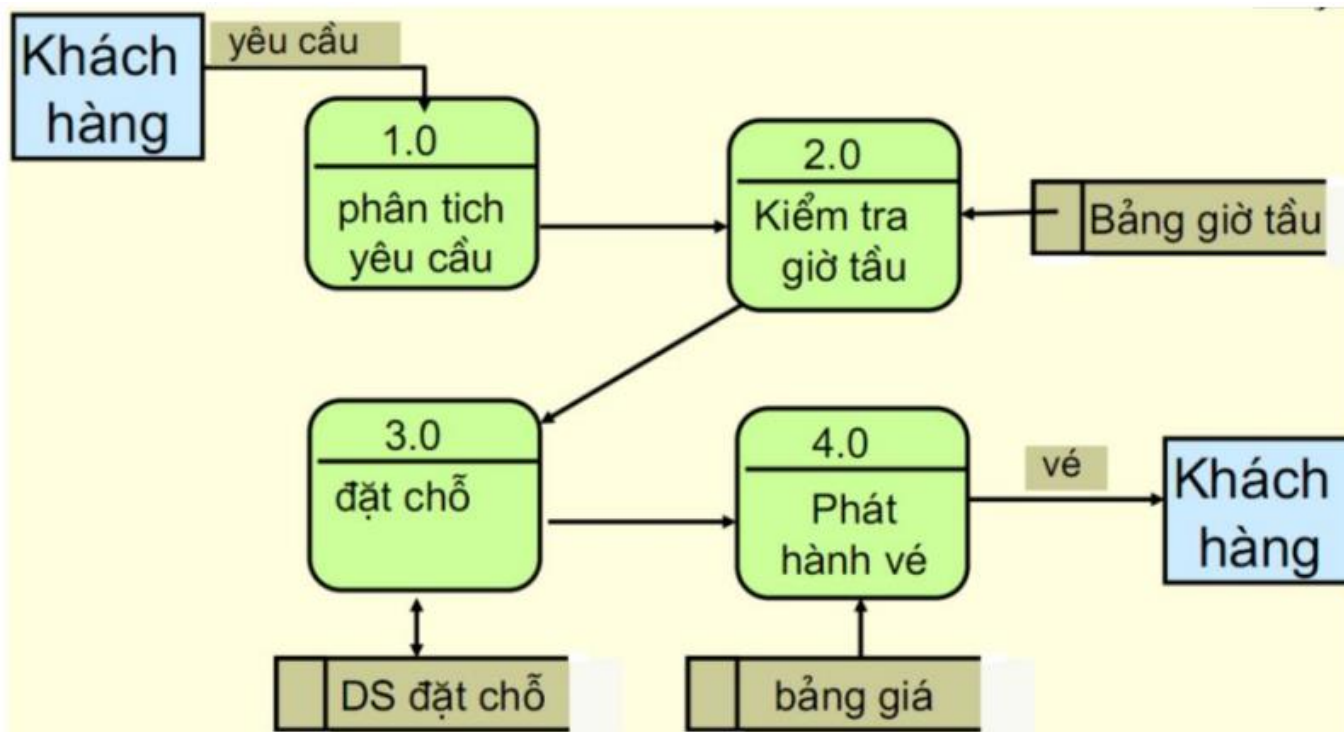




Đặc tả yêu cầu

- Biểu đồ luồng dữ liệu: Ví dụ

DFD của hệ thống bán vé tàu điện tử



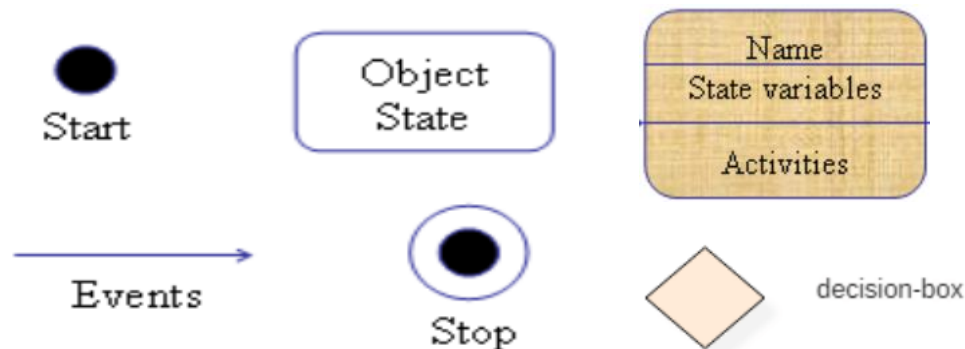


Đặc tả yêu cầu

ĐT3: Máy trạng thái hữu hạn

– FSM: Finite State Machine

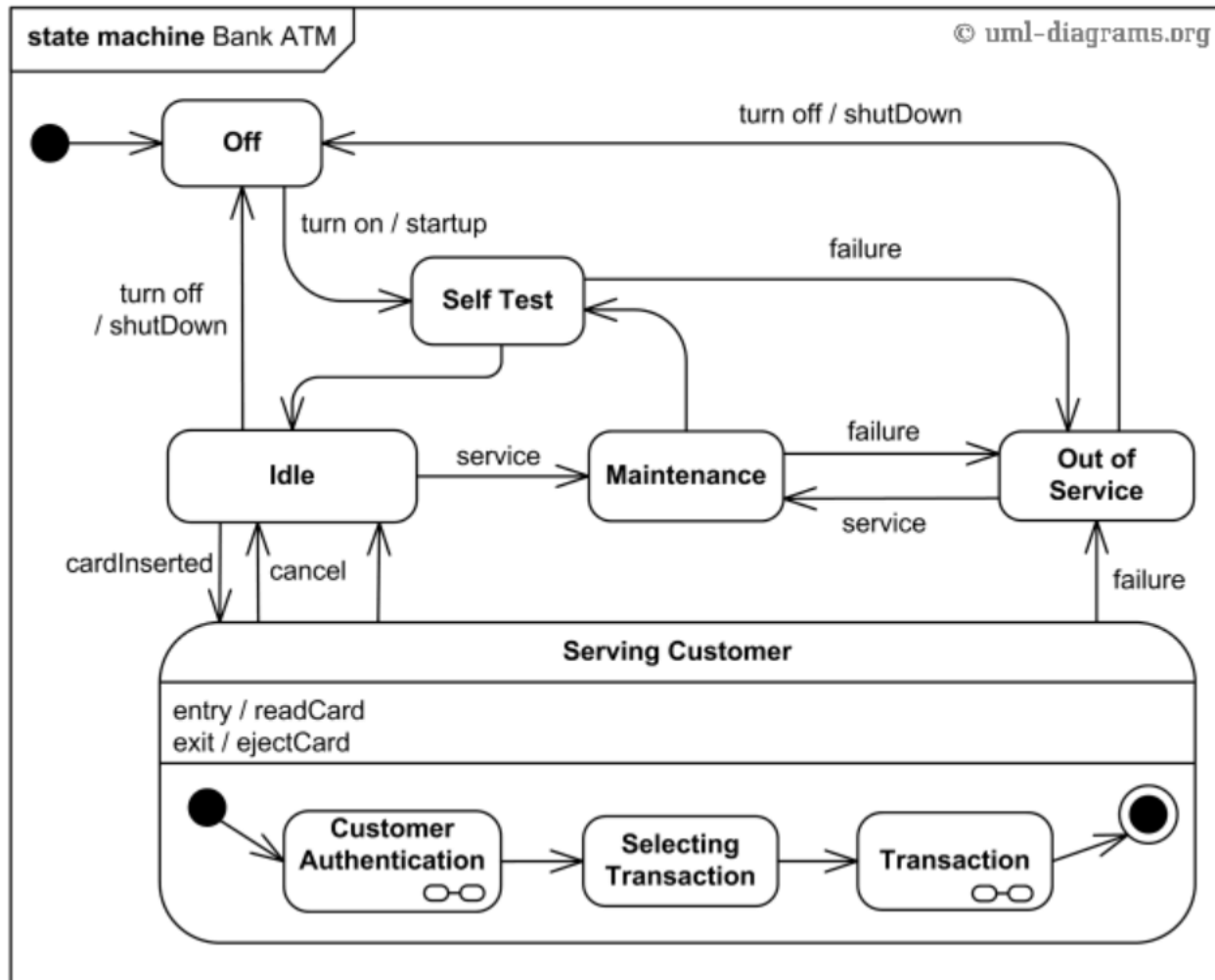
- Là một trong năm biểu đồ UML được sử dụng để mô hình hóa bản chất động của hệ thống.
- Chúng xác định các trạng thái khác nhau của cùng một đối tượng trong suốt vòng đời của nó. Các trạng thái này được thay đổi bởi các sự kiện.
- Các ký hiệu được sử dụng trong biểu đồ trạng thái





Đặc tả yêu cầu

- Máy trạng thái hữu hạn: Ví dụ

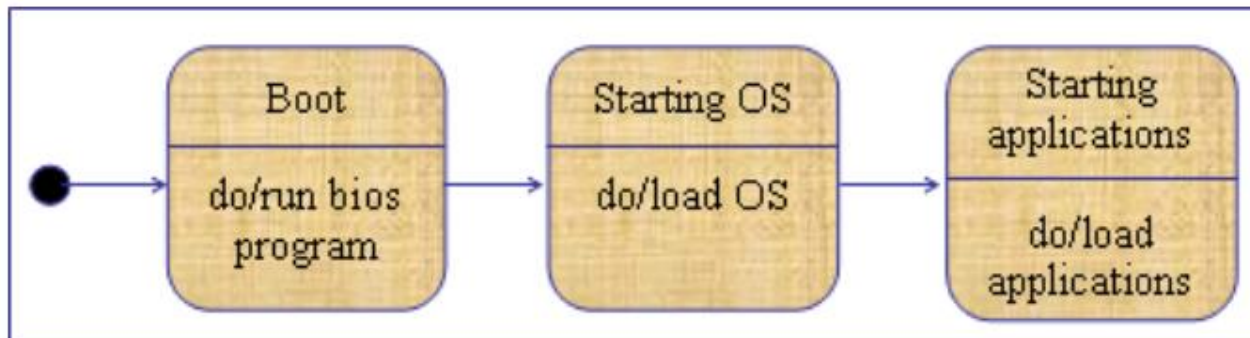




Đặc tả yêu cầu

- Máy trạng thái hữu hạn: Ví dụ (tiếp)

*Biến đổi trạng thái không có sự kiện từ bên ngoài.
Sự thay đổi trạng thái xảy ra khi các hoạt động
trong mỗi trạng thái được thực hiện xong*





Đặc tả yêu cầu

- Máy trạng thái hữu hạn (Tiếp)

Cách vẽ biểu đồ trạng thái trong UML

- Bước 1: Xác định trạng thái ban đầu và trạng thái kết thúc cuối cùng.
- Bước 2: Xác định các trạng thái khả dĩ mà đối tượng có thể tồn tại.
- Bước 3: Gán nhãn các sự kiện kích hoạt trạng thái này.



Đặc tả yêu cầu

- Máy trạng thái hữu hạn (Tiếp)

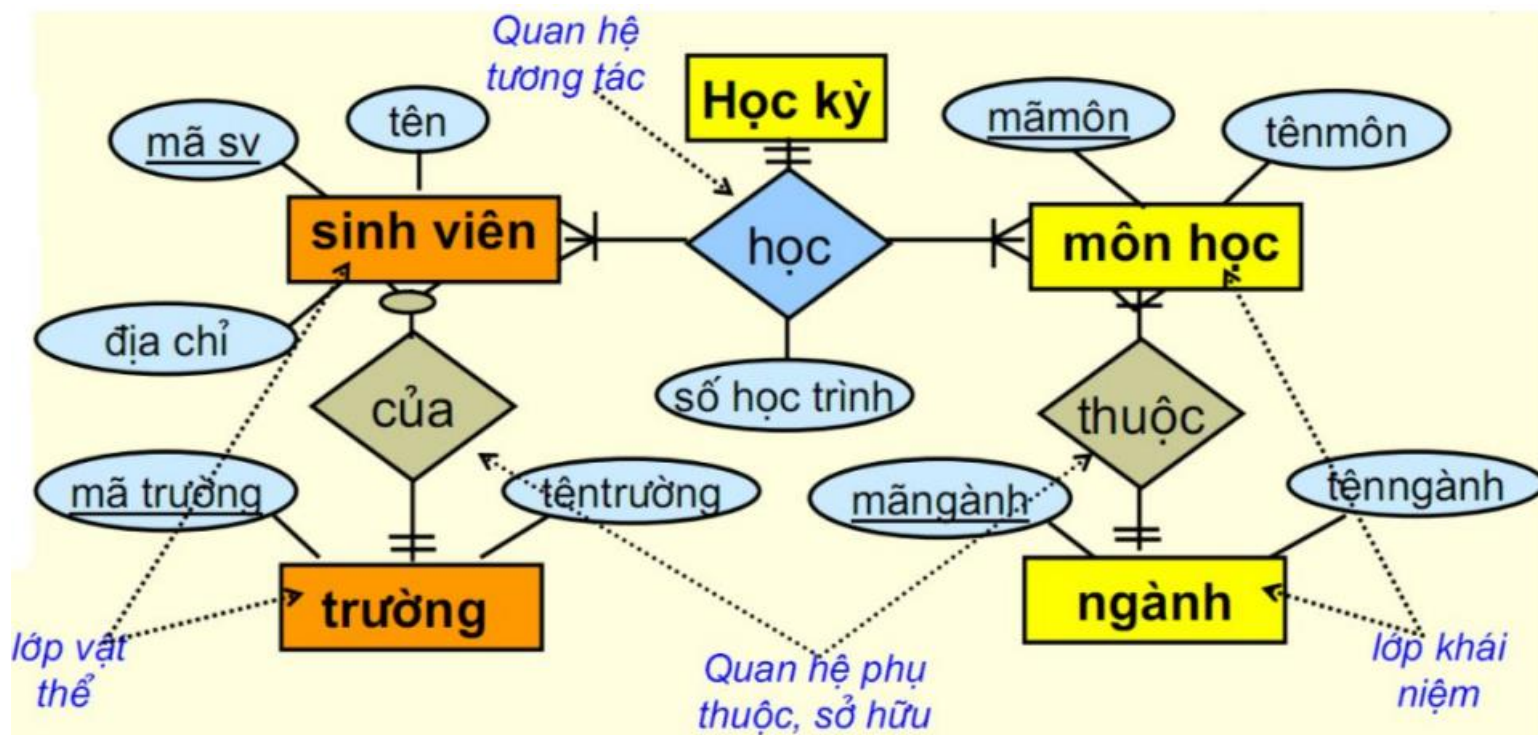
Khi nào sử dụng biểu đồ trạng thái trong UML

- Để mô hình hóa các trạng thái đối tượng của một hệ thống
- Để mô hình hóa hệ thống phản ứng
- Hệ thống phản ứng bao gồm các đối tượng phản ứng
- Để xác định các sự kiện chịu trách nhiệm cho các thay đổi trạng thái



Đặc tả yêu cầu

ĐT4: Biểu đồ thực thể liên kết





Đặc tả yêu cầu

ĐT5: Đặc tả logic

- Đặc tả logic giải quyết vấn đề thu thập các yêu cầu nghiệp vụ và chuyển đổi các yêu cầu đó thành một mô hình logic.
- Mô hình logic xoay quanh các yêu cầu của doanh nghiệp chứ không phải CSDL vật lý.
- Mô hình logic liên quan đến việc thu thập thông tin về các quy trình nghiệp vụ.
- Mô hình logic phải thể hiện chính xác hình ảnh trực quan của các hoạt động và dữ liệu liên quan đến một doanh nghiệp cụ thể.



Đặc tả yêu cầu

ĐT5: Đặc tả logic (tiếp)

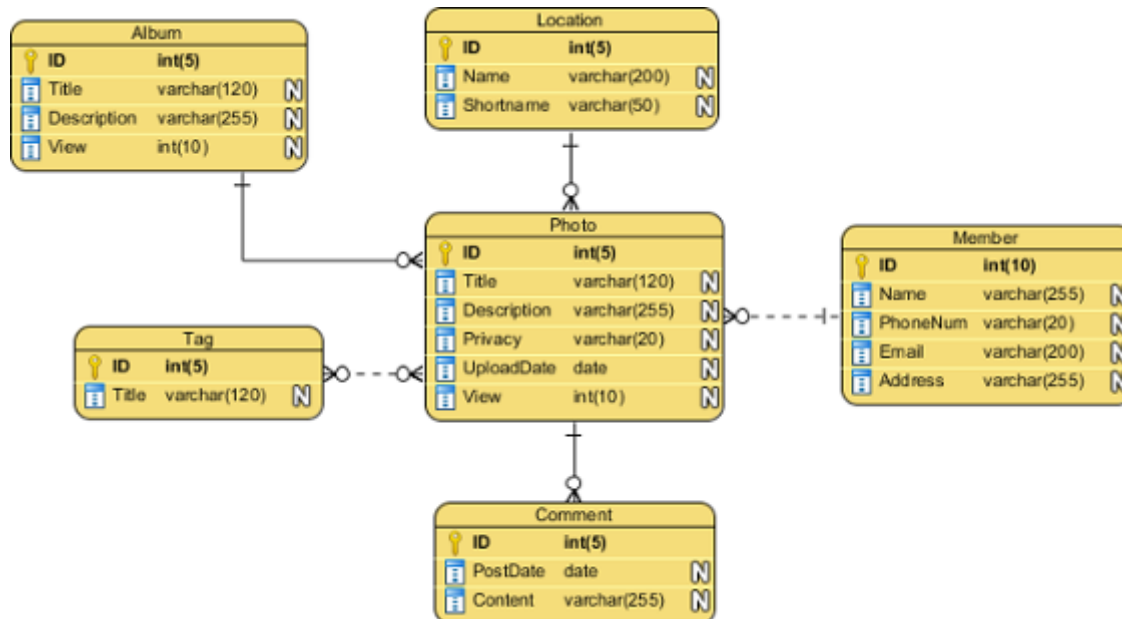
- Mô hình logic không chỉ ảnh hưởng đến hướng thiết kế CSDL mà còn ảnh hưởng gián tiếp đến hiệu suất và quản trị của một CSDL được triển khai.
- Các sơ đồ và tài liệu được tạo ra trong quá trình lập mô hình logic được sử dụng để xác định xem các yêu cầu của doanh nghiệp được thu thập đủ hay chưa? Có cần làm thêm công việc gì trước khi bắt đầu mô hình vật lý hay không?



Đặc tả yêu cầu

ĐT5: Đặc tả logic (tiếp)

- Các thể hiện điển hình của mô hình logic:
 - Mô hình thực thể quan hệ mức logic
 - Sơ đồ quy trình nghiệp vụ
 - Tài liệu phản hồi của người dùng





Đặc tả yêu cầu

ĐT5: Đặc tả đại số

- Được biểu diễn thông qua các hàm toán học.
- Nó chỉ ra điều mong muốn chứ không phải chỉ ra cách làm như thế nào.
- Có nhiều kiểu đặc tả biểu diễn khác nhau.
Ví dụ biểu diễn ràng buộc ngày vào công ty phải lớn hơn ngày sinh ít nhất 18 năm.

```
∀t ∈ NHANVIEN (  
    YEAR(t.NGAYSINH) ≤ YEAR(t.NGAYVAO) - 18 )
```



Dàn bài đặc tả yêu cầu phần mềm

Chuẩn IEEE 830-1984

1. Giới thiệu

2. Mô tả chung

3. Yêu cầu chi tiết

1. Giới thiệu

- 1. Mục đích**
- 2. Phạm vi**
- 3. Định nghĩa (định nghĩa, từ viết tắt)**
- 4. Tài liệu tham khảo**
- 5. Mô tả cấu trúc tài liệu**

2. Mô tả chung

- 1. Tổng quan về sản phẩm**
- 2. Chức năng sản phẩm**
- 3. Đối tượng người dùng**
- 4. Ràng buộc tổng thể**
- 5. Giả thiết về sự lệ thuộc**



Dàn bài đặc tả yêu cầu phần mềm

3. Yêu cầu chi tiết

–3.1 Yêu cầu chức năng

- 3.1.1 Yêu cầu chức năng
 - 3.1.1.1 Giới thiệu
 - 3.1.1.2 Dữ liệu
 - 3.1.1.3 Xử lý
 - 3.1.1.4 Kết quả
- 3.1.n Yêu cầu chức năng n
- 3.2 Yêu cầu giao diện ngoài
- 3.3 Yêu cầu hiệu suất
- 3.4 Ràng buộc thiết kế

3.5 Thuộc tính

3.5.1. Tính bảo mật

3.5.2 Tính bảo trì

3.6 Các yêu cầu khác

Tài liệu tham khảo

Phụ lục



Xét duyệt đặc tả

- Xét duyệt đặc tả là việc tiến hành khảo sát dựa trên tài liệu đặc tả để tìm lỗi, hoặc thông qua phỏng vấn (kiểm thử tĩnh).
- Có hai mức xét duyệt đặc tả:
 - Duyệt mức cao
 - Duyệt mức thấp



Xét duyệt đặc tả

- Duyệt mức cao:
 - Mục đích:
 - Giúp hiểu rõ hơn về phần mềm của mình và những yếu tố bên ngoài ảnh hưởng đến thiết kế của sản phẩm phần mềm.
 - Trong giai đoạn này “Nghiên cứu” để hiểu hơn là kiểm thử. Tìm ra các lỗi cơ bản lớn hay những lỗi về bỏ sót. Không đi sâu vào phân tích chi tiết lỗi.
 - Các kỹ thuật duyệt tài liệu đặc tả ở mức cao:
 - Hãy là khách hàng của sản phẩm
 - Hãy nghiên cứu các chuẩn và các hướng dẫn hiện hành.



Xét duyệt đặc tả

- Một số chuẩn nghiên cứu áp dụng trong duyệt đặc tả ở mức cao:
 - Hợp thức hóa các thuật ngữ và quy ước
 - Yêu cầu công nghiệp
 - Chuẩn quy định bởi chính phủ
 - Giao diện đồ họa với người sử dụng
 - Chuẩn bảo mật



Xét duyệt đặc tả

- Duyệt mức thấp:
 - Sau khi xét duyệt tổng quát ở mức cao, chúng ta sẽ chuyển sang bước xét duyệt đặc tả chi tiết ở mức thấp.
 - Kỹ thuật kiểm thử đặc tả ở mức thấp:
 - Thẩm định về các thuộc tính của đặc tả:
 - Đầy đủ
 - Trúng đích
 - Chính xác, rõ ràng, không nhập nhằng
 - Tương thích
 - Khả thi, khả kiểm
 - Phi mã lệnh
 - Thẩm định về thuật ngữ đặc tả



Nội dung

1. Phân tích yêu cầu và mô tả hoạt động
2. Đặc tả phần mềm
 - Cách đặc tả và biểu diễn
 - Các nguyên lý đặc tả
 - Các mức trừu tượng của đặc tả
 - Đặc tả yêu cầu
 - Dàn bài đặc tả yêu cầu phần mềm
 - Xét duyệt đặc tả
- 3. Kỹ nghệ hệ thống và tạo nguyên mẫu



Kỹ nghệ hệ thống

- Bước 1: Xác định nhu cầu
 - Người phân tích giúp khách hàng xác định các mục tiêu của hệ thống.
 - Người phân tích phải phân biệt được giữa “nhu cầu” của khách hàng và “điều mong muốn” của khách hàng.
 - Khi các mục tiêu tổng thể đã được xác định, người phân tích chuyển sang việc đánh giá các thông tin phụ.



Kỹ nghệ hệ thống

- Bước 2: Nghiên cứu khả thi
 - Khả thi về kinh tế: đánh giá về chi phí phát triển cần phải cân xứng với lợi tức cuối cùng hay lợi ích mà hệ thống được xây dựng đem lại.
 - Khả thi về kỹ thuật: khảo cứu về chức năng, hiệu suất và ràng buộc có thể ảnh hưởng tới khả năng đạt tới một hệ thống chấp nhận được.



Kỹ nghệ hệ thống

- Bước 2: Nghiên cứu khả thi (tiếp)
 - Khả thi về hợp pháp: quy định của pháp luật về sự xâm phạm, vi phạm hay khó khăn nào có thể gây ra từ việc xây dựng hệ thống.
 - Khả thi về phương án: đánh giá tính khả thi của phương án đã xác định để tiếp cận tới việc xây dựng hệ thống.



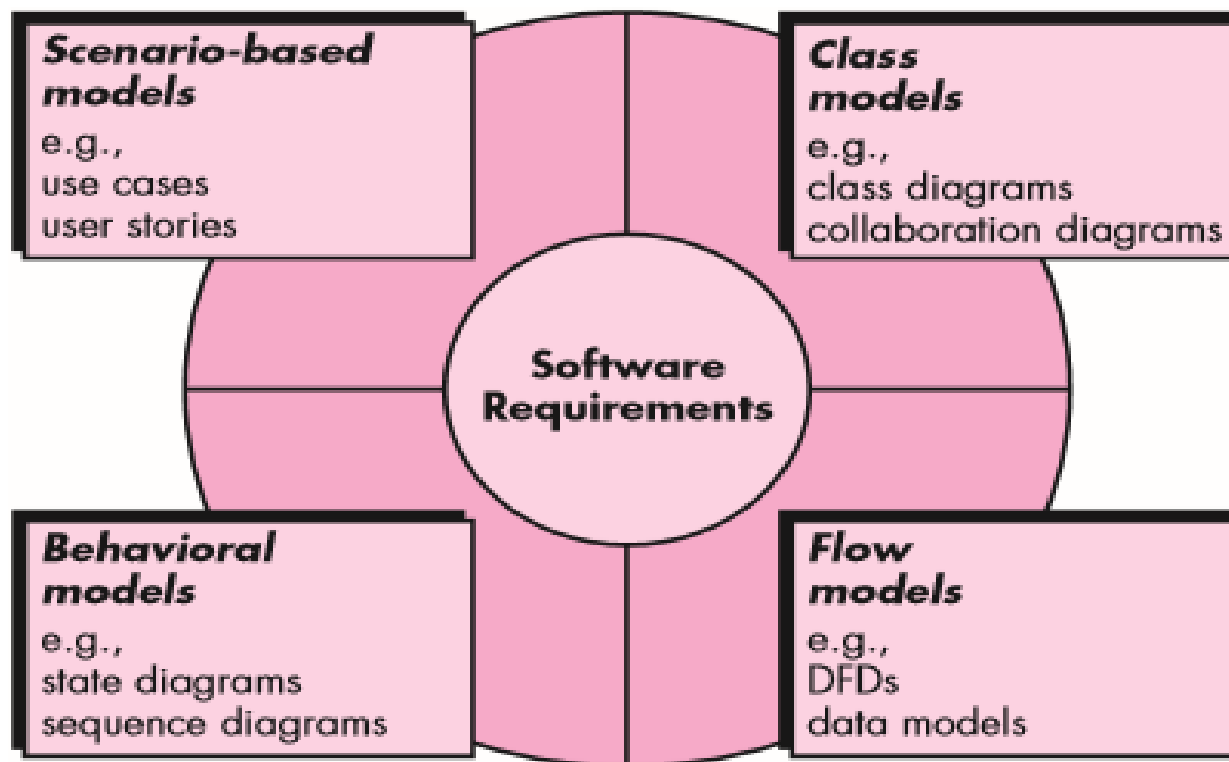
Kỹ nghệ hệ thống

- Bước 3: Mô hình hóa hệ thống
 - Việc mô hình hóa và mô phỏng hệ thống được sử dụng để loại bỏ những điều bất ngờ khi xây dựng hệ thống.
 - Những công cụ CASE được áp dụng trong các tiến trình kỹ nghệ hệ thống.



Mô hình hóa hệ thống

- Các yếu tố của mô hình phân tích: Phân tích cấu trúc và phân tích hướng đối tượng.





UML Diagram

Ví dụ:

- Use case diagram:
<https://www.smartdraw.com/use-case-diagram/>
- Activity Diagram:
<https://www.smartdraw.com/activity-diagram/>
- Class Diagram:
<https://www.smartdraw.com/class-diagram/>
- Sequence Diagram:
<https://www.smartdraw.com/sequence-diagram/>



Đặc tả hệ thống

- Bản đặc tả hệ thống:
 - Là một tài liệu làm nền tảng cho kỹ nghệ phần cứng, kỹ nghệ phần mềm, kỹ nghệ cơ sở dữ liệu và kỹ nghệ con người.
 - Mô tả về chức năng và hiệu suất của hệ thống và những ràng buộc hệ thống.
 - Quy định cả giới hạn cho từng phần tử hệ thống.
 - Mô tả thông tin (dữ liệu và điều khiển) vào/ ra khỏi hệ thống



Đặc tả hệ thống

- Xét duyệt đặc tả hệ thống: Tổ chức các cuộc họp cả hai bên.
 - Cuộc họp được tiến hành trong giai đoạn đưa ra các quan điểm quản lý áp dụng cho hệ thống, tiến hành đánh giá kỹ thuật về các phần tử và chức năng hệ thống
 - Cuộc họp phải đảm bảo: Tính đúng đắn của phạm vi dự án, chức năng, hiệu suất và giao diện. Phân tích rủi ro môi trường và tính khả thi. Người phát triển và khách hàng có cùng cảm nhận về mục tiêu hệ thống



Xét duyệt đặc tả hệ thống

- Về khía cạnh quản lý, tại giai đoạn đánh giá kỹ thuật, trong nhiệm vụ phân tích cần đều đặn trả lời các câu hỏi:
 - Nhu cầu kinh doanh của hãng đã được thiết lập chưa? Luận chứng hệ thống có nghĩa không?
 - Có cần môi trường (hay thị trường) riêng cho hệ thống đã được mô tả hay không?
 - Những phương án nào đã được xem xét?
 - Rủi ro phát triển cho từng phần tử hệ thống là gì?
 - Các tài nguyên đã có sẵn cho việc xây dựng hệ thống chưa?
 - Các giới hạn chi phí và lịch biểu có ý nghĩa gì không?



Xét duyệt đặc tả hệ thống

- Mức độ chi tiết trong giai đoạn đánh giá kỹ thuật phụ thuộc mức độ chi tiết trong nhiệm vụ xác định ban đầu nhưng phải bao gồm các vấn đề sau:
 - Về chức năng của hệ thống
 - Giao diện giữa các phần tử hệ thống và với môi trường
 - Các vấn đề độ tin cậy, hiệu suất và bảo trì
 - Cung cấp đủ nền tảng cho các bước kỹ nghệ phần cứng và phần mềm tiếp sau không?



Tạo nguyên mẫu

- Tạo nguyên mẫu là một trong năm giai đoạn của tư duy **thiết kế** để tạo ra giải pháp cho các vấn đề.
- Nó xây dựng một số phiên bản thu nhỏ của sản phẩm hoặc một tính năng cụ thể.
- Nó có thể đơn giản như phác thảo trên giấy hay tạo mockup, hoặc phức tạp như một phiên bản mã hóa đơn giản của một ứng dụng đang hoạt động.



Tạo nguyên mẫu

- Ví dụ tạo nguyên mẫu sử dụng mockup trên điện thoại



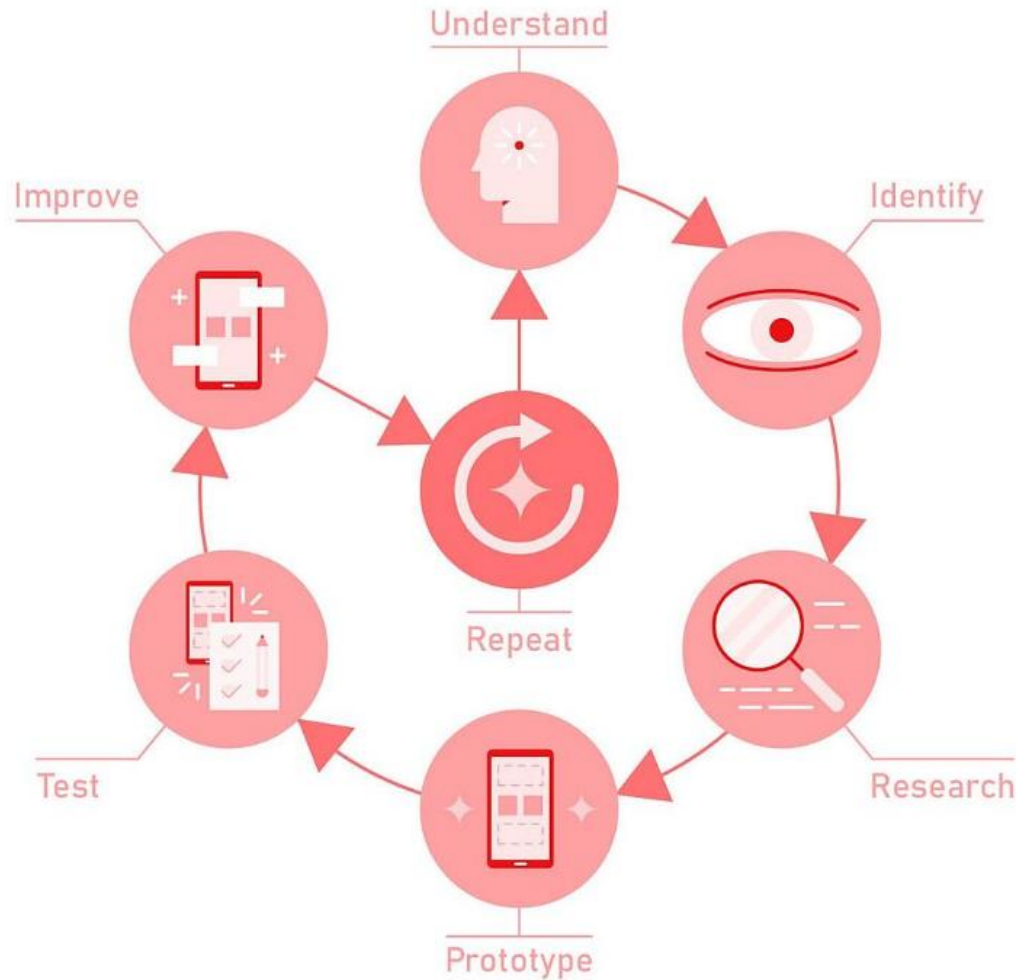


Tạo nguyên mẫu

- Lợi ích của tạo nguyên mẫu:
 - Giúp sớm kiểm tra ý tưởng và thực hiện các thay đổi trước khi thực hiện các công việc đắt tiền tiếp theo.
 - Giúp tiết kiệm thời gian và tiền bạc và có thêm cơ hội nghiên cứu nhu cầu của người dùng tốt hơn.
 - Xây dựng một nguyên mẫu chức năng để:
 - Làm công cụ giao tiếp với người dùng và khách hàng trước khi đưa ra sản phẩm cuối.
 - Có được thông tin phản hồi có giá trị với đủ thời gian và ngân sách để cải thiện sản phẩm.



Tạo nguyên mẫu



Quá trình tạo mẫu



Tạo nguyên mẫu

- Hiểu ngữ cảnh: Understand
 - Phân tích thị trường và xây dựng chiến lược.
- Xác định các yêu cầu cốt lõi: Identify
 - Nhìn vào các vấn đề từ quan điểm của người dùng để giải quyết chúng.
- Nghiên cứu sản phẩm: Research
 - Tìm hiểu các phần mềm tương tự hiện có để từ đó tạo ra sự khác biệt và cơ hội cho sản phẩm của mình.



Tạo nguyên mẫu

- Phát triển một nguyên mẫu ban đầu: Prototype
 - Sử dụng công cụ để tạo nguyên mẫu ban đầu phụ thuộc vào mục tiêu của nguyên mẫu. Giúp người dùng có sự trải nghiệm vào đúng khu vực có vấn đề cần khắc phục.
- Kiểm tra và xem xét các nguyên mẫu: Test
 - Đánh giá nguyên mẫu cùng người dùng và các bên liên quan. Sử dụng nguyên mẫu như một cơ hội để thúc đẩy sự tham gia của khách hàng và nhận sự góp ý từ những chuyên gia trải nghiệm người dùng.



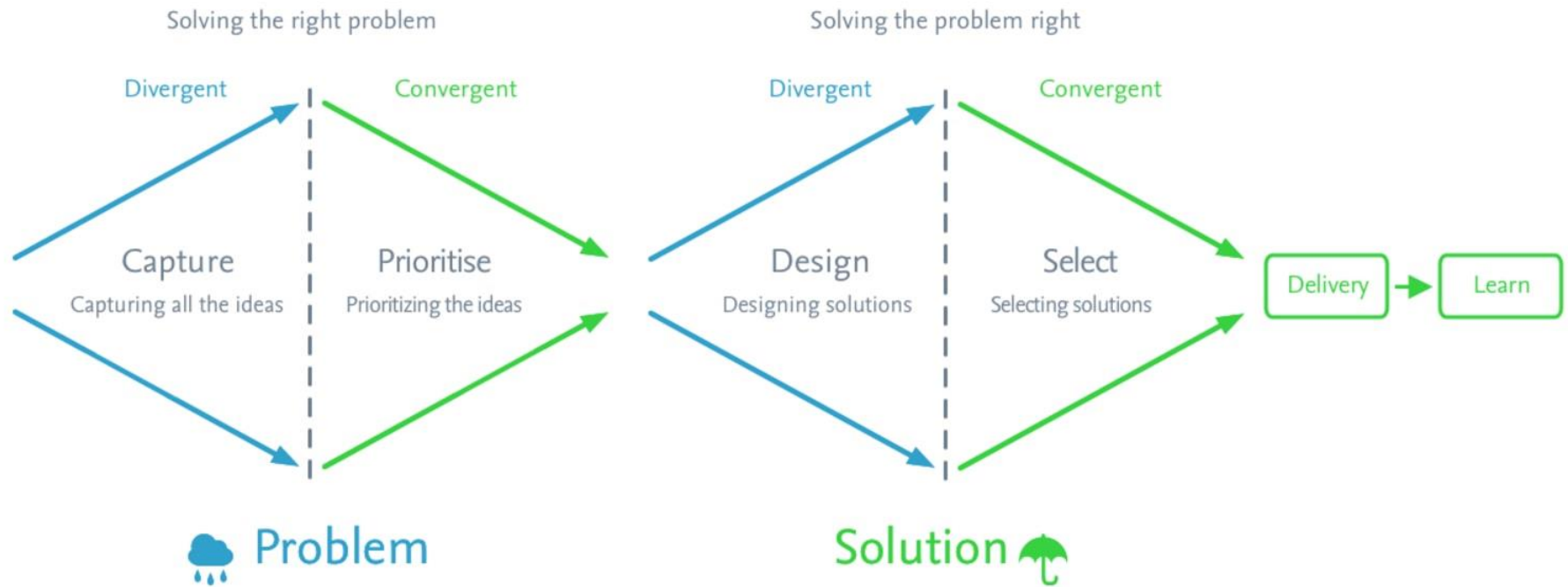
Tạo nguyên mẫu

- Sửa đổi và cải tiến nguyên mẫu: Improve
 - Sử dụng các phản hồi thu thập được để thay đổi và cải thiện nguyên mẫu.
 - Một lần nữa, hãy kiểm tra nguyên mẫu với người dùng cuối tiềm năng để đảm bảo rằng vấn đề cần giải quyết đã được hiểu đúng.
- Lặp lại: Repeat
 - Nếu trong giai đoạn cải tiến mà vẫn chưa làm rõ và chưa giải quyết được vấn đề, thì cần lặp lại toàn bộ quá trình.
 - Việc thực hiện lại quy trình nhằm kiểm tra các ý tưởng bổ sung để nâng cao sản phẩm.



Tạo nguyên mẫu

- Quan trọng của việc tạo nguyên mẫu là khám phá đưa ra được các ý tưởng cho sản phẩm.



Mô hình Double Diamond để khám phá và triển khai ý tưởng



Nội dung

4. Phân tích dữ liệu với mô hình thực thể liên kết và mô hình quan hệ

- Các khái niệm và biểu diễn đồ họa
- Phương pháp lập lược đồ dữ liệu
 - Mô tả bằng sơ đồ thực thể liên kết
 - Ánh xạ từ mô hình thực thể liên kết sang mô hình quan hệ
 - Xây dựng lược đồ CSDL quan hệ
- Biểu diễn lược đồ dữ liệu ở dạng bảng



Phân tích dữ liệu

- Sơ đồ thực thể liên kết bao gồm các bảng, thuộc tính và các liên kết.
- Các ký hiệu:

Thực thể



Thuộc tính



Liên kết





Mức tham gia liên kết

CARDINALITY

1/	—————+	Một
2/	—————++	Một và bắt buộc phải có một
3/	—————○+	Không hoặc một
4/	—————<	Nhiều
5/	—————≧	Một hoặc nhiều
6/	—————○≧	Không hoặc nhiều

CÓ THỂ HIỂU

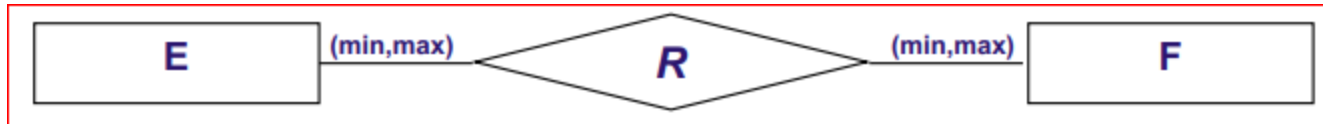
<i>Đơn thuần là một</i>
<i>Required (chỉ được một, và bắt buộc có một dữ liệu)</i>
<i>Optional (không có dữ liệu cũng được)</i>
<i>Đơn thuần là nhiều</i>
<i>Required (ít nhất phải có một dữ liệu)</i>
<i>Optional (không có dữ liệu cũng được)</i>

Mối liên kết – Tính Multiplicity



Ràng buộc mối liên kết

- Có hai loại chính: Cardinality ratio và Participation constraint
 - Ràng buộc dựa trên bản số
 - Thể hiện bởi bản số (min,max)



- Cardinality ratio:
 - Many to many
 - Many to one
 - One to many
 - One to one



Ràng buộc mối liên kết

- Participation Constraint: ràng buộc tham gia
 - *This constraint specifies the number of instances of an entity that **can** participate in a relationship type.*
 - Bao gồm:
 - Total participation: Biểu diễn bằng nét dày —
 - Partial participation: Biểu diễn bằng nét mảnh —

There are **two types** of participation constraints—**total** and **partial**.
The participation of an entity **set E** in a relationship **set R** is said to be **total** if **every entity in E participates in at least one relationship in R**.
If **only some entities** in **E** participate in relationships in **R**, the participation of entity set **E** in relationship **R** is said to be **partial**.



Phương pháp lập lược đồ dữ liệu

- Bước 1: Xác định và định danh thực thể
- Bước 2: Xác định thuộc tính mô tả cho các thực thể
- Bước 3: Xác định mối quan hệ giữa các thực thể và gán thuộc tính cho mỗi quan hệ
- Bước 4: Quyết định miền giá trị cho thuộc tính
- Bước 5: Quyết định thuộc tính khóa.



Phương pháp lập lược đồ dữ liệu

- Bước 6: Quyết định (min,max) cho mỗi quan hệ và thể hiện chúng trên sơ đồ thực thể liên kết

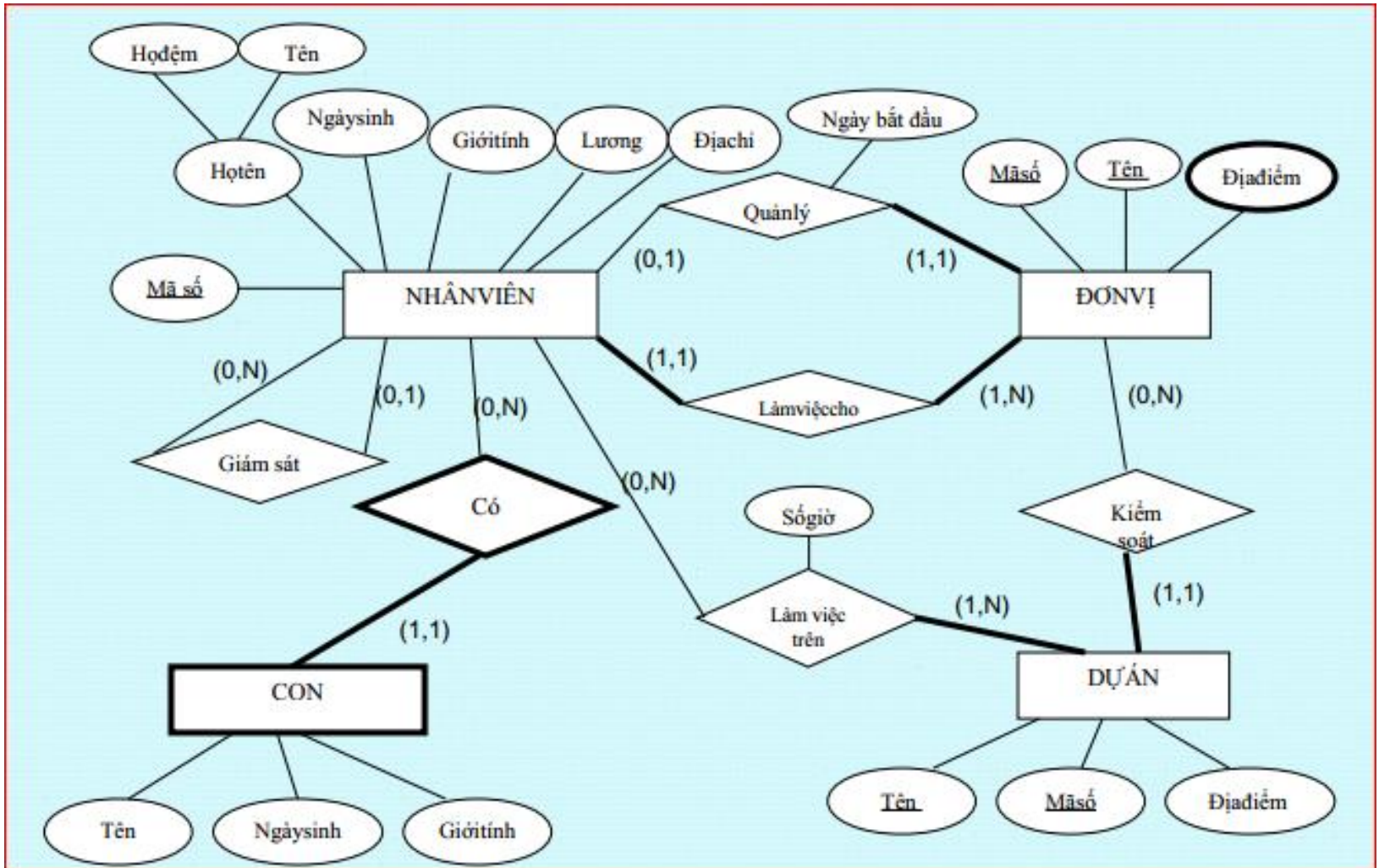
Bài tập

CSDL đề án công ty theo dõi các thông tin liên quan đến nhân viên, phòng ban và đề án

- Cty có nhiều đơn vị, mỗi đơn vị có tên duy nhất, mã đơn vị duy nhất, một trưởng phòng và ngày nhận chức. Mỗi đơn vị có thể ở nhiều địa điểm khác nhau.
- Dự án có tên duy nhất, mã duy nhất, do 1 một phòng ban chủ trì và được triển khai ở 1 địa điểm.
- Nhân viên có mã số, tên, địa chỉ, ngày sinh, giới tính và lương. Mỗi nhân viên làm việc ở 1 phòng ban, tham gia vào các đề án với số giờ làm việc khác nhau. Mỗi nhân viên đều có một người quản lý trực tiếp.
- Một nhân viên có thể có những người con được hưởng bảo hiểm theo nhân viên. Mỗi người con của nhân viên có tên, giới tính, ngày sinh.



Xây dựng sơ đồ ER





Biến đổi sơ đồ thực thể liên kết thành sơ đồ quan hệ

1. Biến đổi tập các thực thể
2. Biến đổi các liên kết
3. Các khóa của lược đồ quan hệ
4. Các lược đồ quan hệ với khóa chung



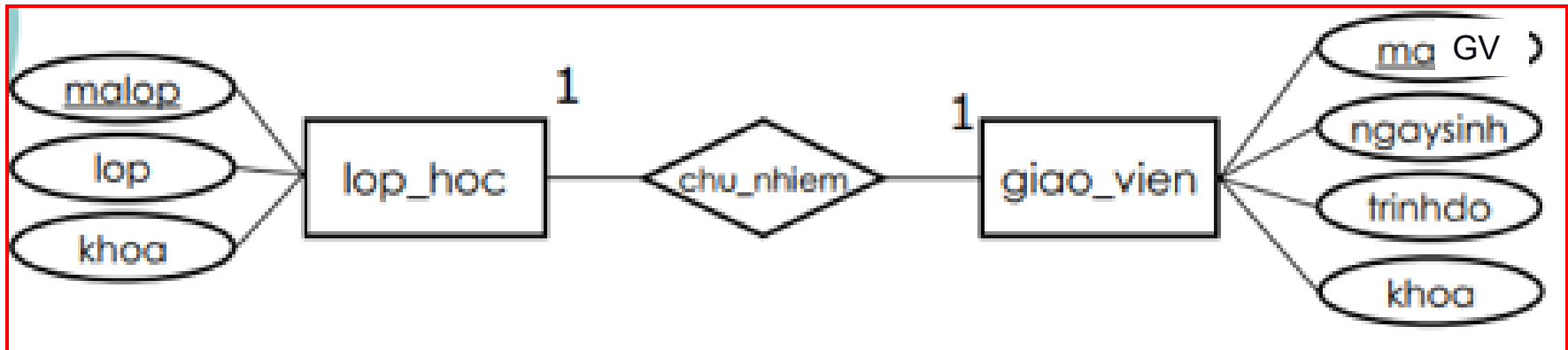
Các luật chuyển đổi

- Mỗi thực thể trong mô hình liên kết trở thành một quan hệ (bảng).
 - Thuộc tính -> thuộc tính của quan hệ (field)
 - Khóa (thuộc tính định danh) -> khóa của quan hệ (khóa chính)
- Thi hành các liên kết thông qua đặt khóa ngoại vào các quan hệ.
 - Với liên kết 1-1:
 - Đặt khóa chính của quan hệ này vào quan hệ kia làm khóa ngoại và ngược lại
 - Hoặc thêm thuộc tính khóa giống nhau vào cả hai quan hệ



Các luật chuyển đổi

– Liên kết 1-1



– Dùng khóa ngoại:

lop_hoc(malop, maGV, lop, khoa)

Giao_vien(maGV, malop, ngaysinh, trinhdo, khoa)



Các luật chuyển đổi

- Với liên kết 1-n: Đặt khóa chính của quan hệ đầu 1 vào quan hệ đầu nhiều làm khóa ngoài

Ví dụ: Với mô hình thực thể liên kết



Ta có các quan hệ:

SINH VIÊN (mã SV, họ tên, ngày sinh, tên lớp)

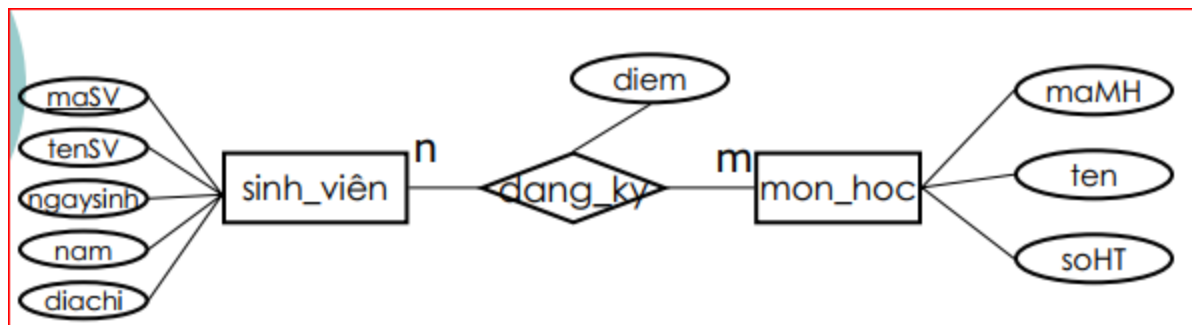
LỚP (tên lớp, phòng học)



Các luật chuyển đổi

– Với liên kết n-m:

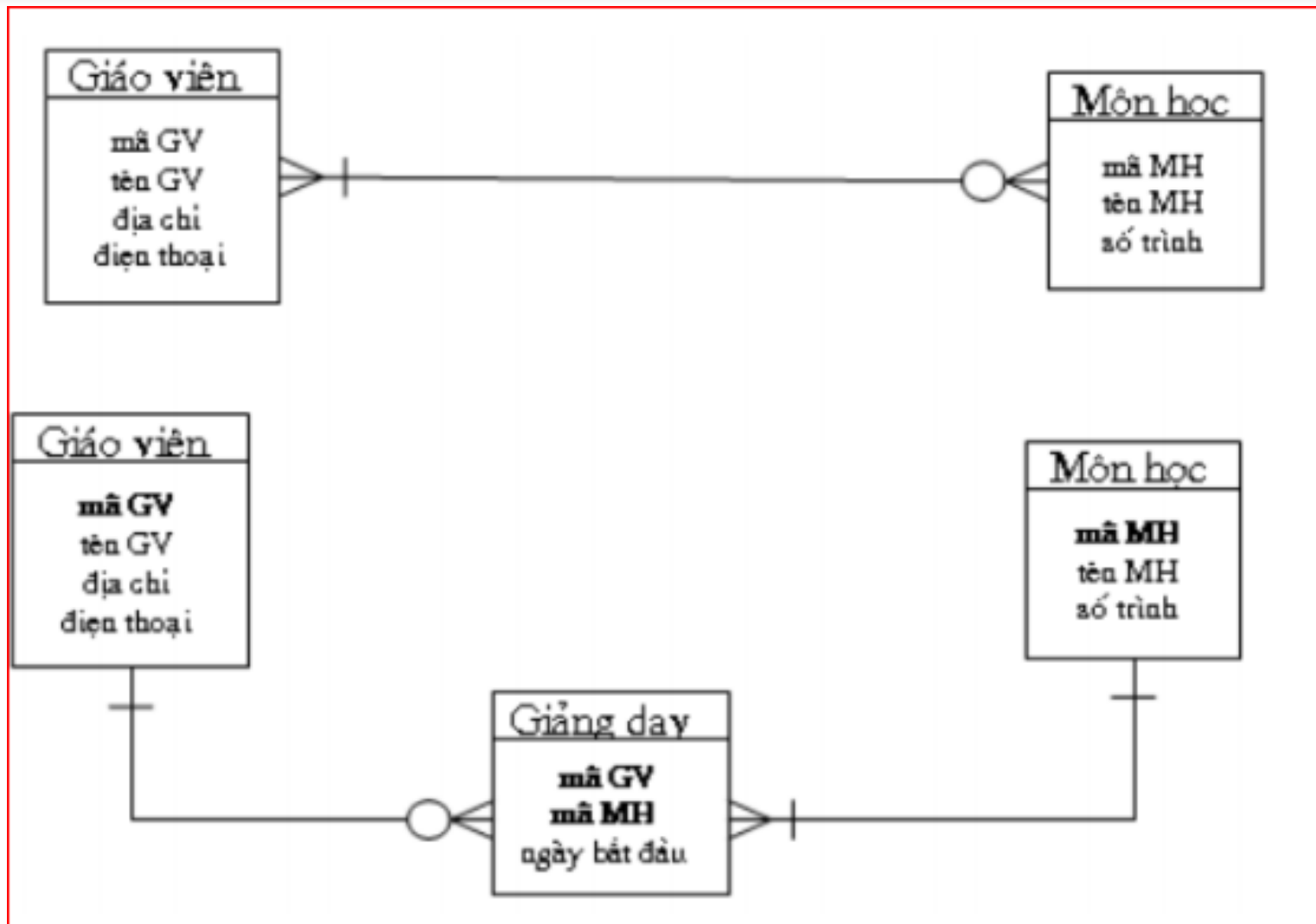
- Cần tách thành quan hệ 1-n rồi mới chuyển thành quan hệ(bảng)
- Thêm một quan hệ mới xác định bởi các thuộc tính là khóa của các thực thể có liên quan và các thuộc tính của liên kết.
- Sử dụng bộ khóa hoặc khóa giả cho quan hệ mới.



DANGKY (maSV,maMH,diem) hoặc

DANGKY (ID,maSV,maMH,diem)

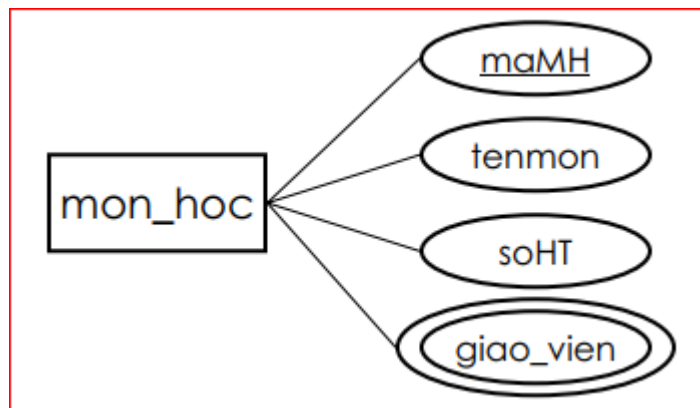
Các luật chuyển đổi





Các luật chuyển đổi

- Thuộc tính đa trị: Với mỗi thuộc tính đa trị:
 - Cần thêm một quan hệ mới
 - Bổ sung thêm khóa của thực thể tương ứng làm thuộc tính khóa cho quan hệ mới



MH_GV(maMH, giao_vien)

Cách vẽ mô hình thực thể quan hệ bằng visual paradigm

- Biểu diễn lược đồ dữ liệu ở dạng bảng

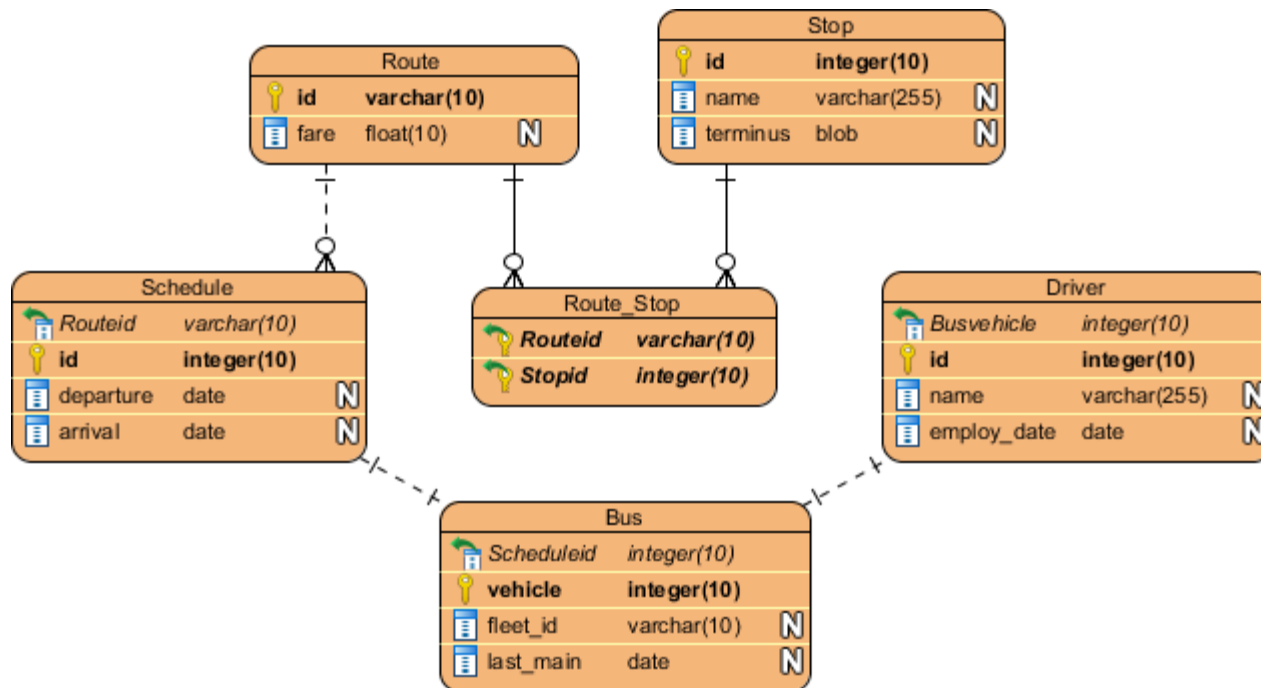
Tham khảo tại

https://www.visual-paradigm.com/support/documents/vpuserguide/3563/3564/85375_drawingentit.html

Và <https://www.visual-paradigm.com/tutorials/how-to-model-relational-database-with-erd.jsp>

Cách vẽ mô hình thực thể quan hệ bằng visual paradigm

- Ví dụ





Thảo luận

