

# Kỹ thuật phần mềm

## **Yêu cầu phần mềm**

# Mục tiêu

---

- Giới thiệu các khái niệm về yêu cầu người dùng và yêu cầu hệ thống
- Miêu tả các yêu cầu chức năng (functional) và yêu cầu phi chức năng (non-functional)
- Giải thích cách tổ chức các yêu cầu phần mềm trong mọi tài liệu phần mềm

# Các chủ đề

---

1. Khái niệm và phân loại yêu cầu
  - Các yêu cầu chức năng và phi chức năng
  - Yêu cầu người dùng và yêu cầu hệ thống
2. Quy trình kỹ nghệ yêu cầu
3. Đặc tả yêu cầu
  - Mô hình hoá hệ thống
  - Tài liệu yêu cầu phần mềm

# Thế nào là một yêu cầu?

---

- Yêu cầu (requirement) có nhiều mức
  - Một mô tả trừu tượng về một dịch vụ rằng hệ thống phải chịu một ràng buộc nào đó
  - Một đặc tả chi tiết toán học về một chức năng.
- Các yêu cầu có thể phục vụ hai nhiệm vụ
  - Cơ sở để thương lượng một hợp đồng
    - Khi đó phải được viết một cách trừu tượng cần giải nghĩa thêm;
  - Cơ sở để viết hợp đồng
    - Khi đó phải được định nghĩa chi tiết;
  - Cả hai trường hợp đều có thể gọi là các requirement.

# Tính trừu tượng yêu cầu

“Nếu một công ty muốn có hợp đồng cho một dự án phát triển phần mềm lớn, nó phải định nghĩa các nhu cầu của nó một cách đủ trừu tượng để không xác định luôn một giải pháp cho nhu cầu đó. Các yêu cầu phải được viết sao cho những người đấu thầu khác nhau có thể đề xuất các cách khác nhau để đáp ứng nhu cầu của tổ chức mời thầu. Một khi đã thắng thầu, người đấu thầu sẽ phải viết cho khách hàng một định nghĩa hệ thống đủ chi tiết để khách hàng hiểu và có thể thẩm định được những hoạt động mà phần mềm sẽ thực hiện. Cả hai tài liệu này đều có thể được gọi là tài liệu yêu cầu (*requirement document*) của hệ thống.”

# Các loại yêu cầu

---

- 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.

# Định nghĩa và đặc tả

---

## Định nghĩa yêu cầu người dùng – User requirement definition

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 – System requirement specification

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.

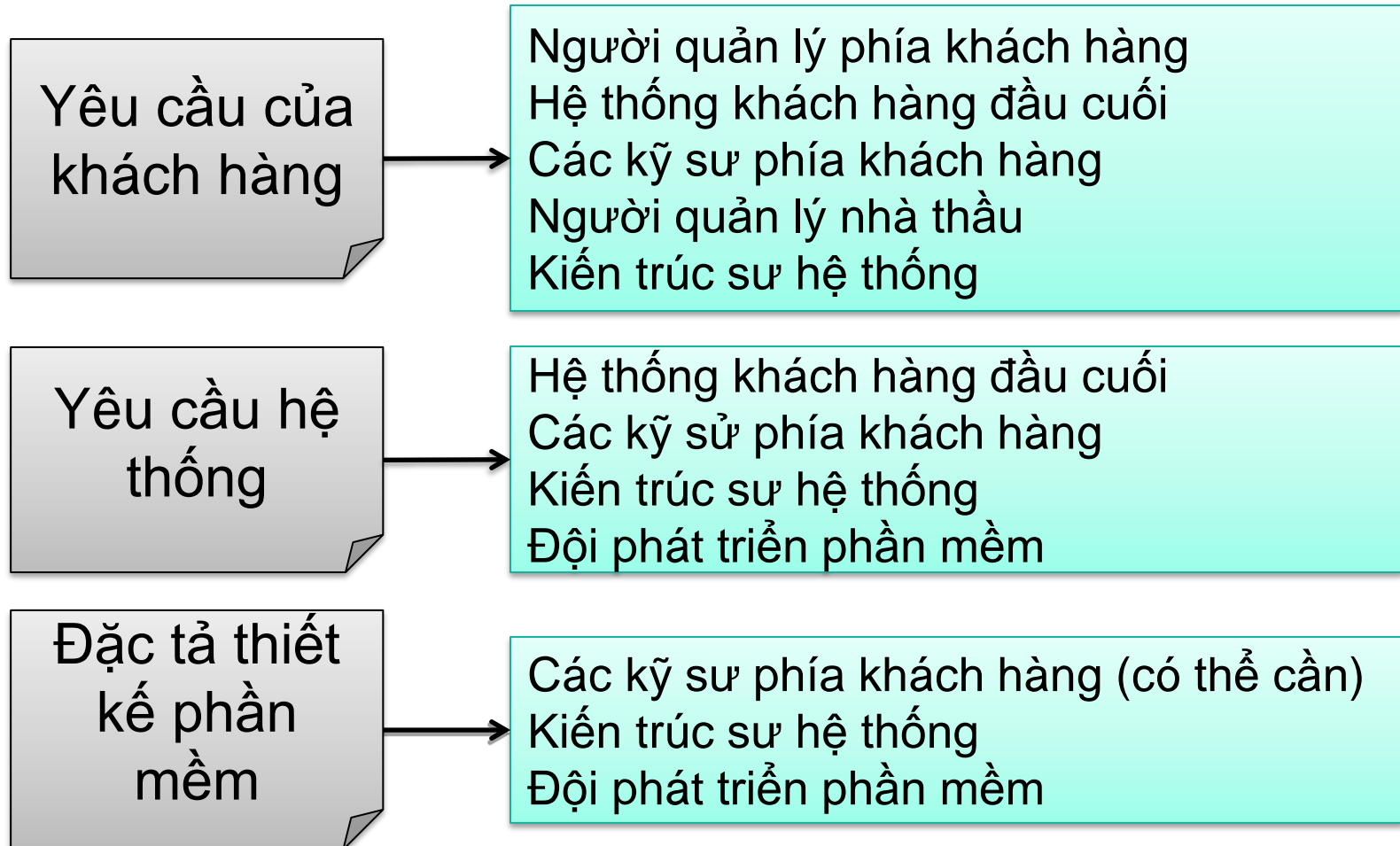
1.3 Mỗi kiểu file ngoài có thể có một công cụ có thể dùng cho loại file đó.

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.

# Những người đọc tài liệu yêu cầu

---





# Yêu cầu chức năng và phi chức năng

---

- Yêu cầu chức năng – Functional requirements
  - 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 và
    - Hệ thống cần ứng xử như thế nào trong các tình huống cụ thể.
- Yêu cầu phi chức năng – Non-functional requirements
  - 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..
- Yêu cầu miền – Domain requirements
  - Các yêu cầu phản ánh các đặc điểm của miền ứng dụng đó.

# Yêu cầu chức năng

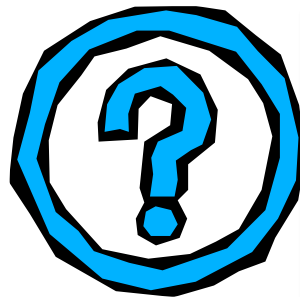
---

- Mô tả chức năng hoặc dịch vụ của hệ thống
- Tùy theo loại phần mềm, những người dự tính là sẽ sử dụng hệ thống, và loại hệ thống nơi sẽ triển khai phần mềm
- Các yêu cầu chức năng mức người dùng có thể là các phát biểu ở mức trừu tượng cao về những gì hệ thống nên làm, còn các yêu cầu chức năng mức hệ thống nên mô tả **chi tiết các dịch vụ của hệ thống**.

# Ví dụ hệ thống quản lý thư viện LIBSYS

---

- Một hệ thống thư viện cung cấp một giao diện duy nhất cho một loạt các cơ sở dữ liệu chứa các ấn phẩm tại các thư viện khác nhau.
- Người dùng có thể tìm kiếm, tải xuống, và in các ấn phẩm đó để phục vụ nghiên cứu của cá nhân.



Đâu là yêu cầu chức năng, đâu là yêu cầu phi chức năng?

# Ví dụ về các yêu cầu chức năng

---

- Người dùng phải có thể tìm kiếm trong toàn bộ tập hợp cơ sở dữ liệu ban đầu hoặc chọn một tập con của nó.
- Hệ thống cần cung cấp viewer thích hợp để người dùng đọc tài liệu trong kho tài liệu.
- Mỗi đơn hàng cần được cấp phát một định danh duy nhất (ORDER\_ID) mà người dùng có thể chép vào vùng lưu trữ dài hạn của tài khoản.

# Sự thiếu chính xác của các yêu cầu

---

- Rắc rối nảy sinh khi các yêu cầu không được phát biểu một cách chính xác.
  - Các yêu cầu mù mờ đa nghĩa có thể được nhóm phát triển và người dùng hiểu theo các cách khác nhau.
- Xét cụm từ ‘viewer thích hợp’
  - Chủ ý của người dùng – viewer đặc biệt dành cho mỗi loại tài liệu;
  - Cách hiểu của nhóm phát triển – cung cấp một text viewer hiển thị được nội dung của tài liệu.

# Tính đầy đủ và nhất quán của yêu cầu

---

- Về nguyên tắc, các yêu cầu phải đầy đủ cũng như nhất quán.
  - **Đầy đủ (complete)**: bao gồm miêu tả về tất cả các tiện ích được đòi hỏi.
  - **Nhất quán (consistent)**: không nên có mâu thuẫn hoặc xung đột trong các miêu tả về các tiện ích của hệ thống.
- Trong thực tiễn, không thể tạo ra được một tài liệu yêu cầu đầy đủ và nhất quán.
  - Rất khó kiểm tra/phát hiện các điểm còn thiếu hoặc không nhất quán

# Yêu cầu phi chức năng

---

- Yêu cầu phi chức năng quy định các tính chất và ràng buộc
  - Ví dụ độ tin cậy, thời gian phản ứng và dung lượng lưu trữ.
  - khả năng của thiết bị vào ra, biểu diễn dữ liệu dùng trong các giao diện hệ thống v.v..
- Các yêu cầu quy trình cũng có thể được dùng để quy định sử dụng một hệ thống CASE, một ngôn ngữ lập trình, hoặc một phương pháp phát triển cụ thể.
- Các yêu cầu phi chức năng có thể quan trọng hơn cả các yêu cầu chức năng.
  - Nếu không thỏa mãn các yêu cầu này thì hệ thống vô dụng.

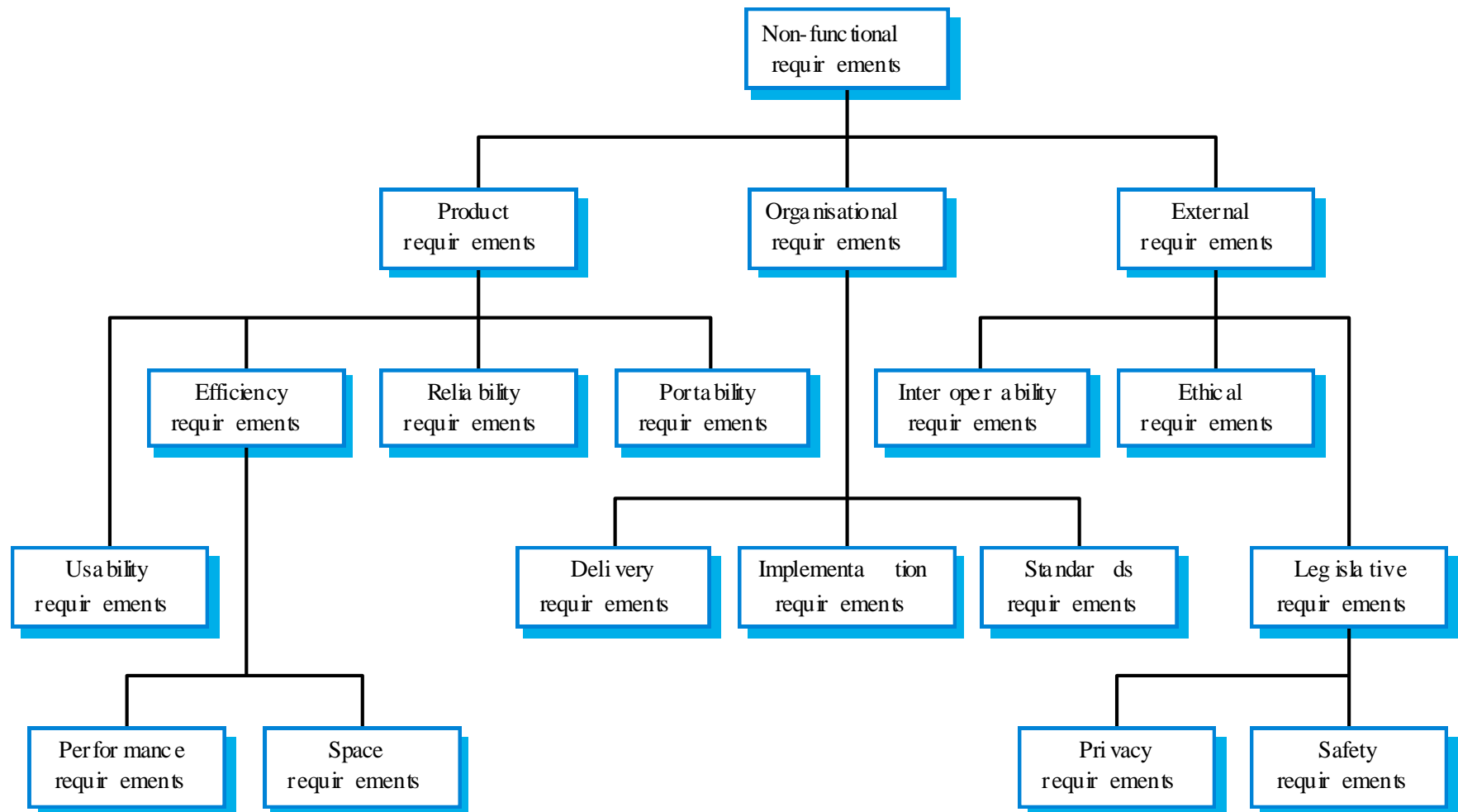
# Các phân loại phi chức năng

---

- Yêu cầu sản phẩm – Product requirements
  - Các yêu cầu quy định về cách hành xử của sản phẩm cần bàn giao
    - Ví dụ tốc độ thực thi, độ tin cậy (tỷ lệ chấp nhận được của các sự cố), v.v..
- Yêu cầu tổ chức – Organisational requirements
  - Các yêu cầu xuất phát từ các chính sách và quy trình của tổ chức của khách hàng cũng như nhóm phát triển
    - Ví dụ các chuẩn quy trình cần sử dụng, ngôn ngữ lập trình, phương pháp thiết kế, v.v...
- Yêu cầu bên ngoài – External requirements
  - Các yêu cầu nảy sinh từ các nhân tố bên ngoài hệ thống và quy trình phát triển nó
    - Ví dụ các yêu cầu về việc tương tác giữa hệ thống với các hệ thống của các tổ chức khác, yêu cầu pháp lý, v.v...



# Các loại yêu cầu phi chức năng



# Ví dụ về yêu cầu phi chức năng

---

- Product requirement
  - Giao diện người dùng của LIBSYS cần được cài đặt dạng HTML động đơn giản không chứa frame hay hiệu ứng phức tạp.
- Organisational requirement
  - Quy trình phát triển hệ thống và các tài liệu bàn giao cần tuân theo chuẩn quy trình và sản phẩm được qui định theo quy chuẩn.
- External requirement
  - Hệ thống sẽ không để lộ thông tin cá nhân của các khách hàng cho các nhân viên vận hành hệ thống, ngoại trừ tên và số tham chiếu (reference number) của khách hàng.

# Ví dụ

---

- Một mục tiêu hệ thống
  - Nên dễ dùng đối với những người dùng có kinh nghiệm và
  - Nên được tổ chức sao cho giảm thiểu được lỗi của người sử dụng.
- Một yêu cầu phi chức năng kiểm định được
  - Những người dùng có kinh nghiệm cần sử dụng được toàn bộ các chức năng của hệ thống sau **hai giờ tập huấn**.
  - Sau thời gian tập huấn đó, số lỗi trung bình mà một người sử dụng có kinh nghiệm phạm phải sẽ không vượt quá **hai lỗi mỗi ngày**.

# Mục tiêu và yêu cầu

---

- Một số yêu cầu phi chức năng có thể khó mà phát biểu được chính xác, và việc kiểm định các yêu cầu phi chức năng không chính xác có thể khó khăn.
  - **Tính dễ sử dụng.**
- Các yêu cầu phi chức năng kiểm định được (verifiable)
  - Một phát biểu sử dụng **một phép đo nào đó** mà có thể test được một cách khách quan.
  - Nên cố gắng diễn đạt các yêu cầu ở dạng kiểm định được

# Các phép đo đặc yêu cầu

Property	Measure
Speed (tốc độ)	Processed transactions/second User/Event response time Screen refresh time
Size (kích cỡ)	M Bytes Number of ROM chips
Ease of use (dễ sử dụng)	Training time Number of help frames
Reliability (thực tế)	Mean time to failure Probability of unavailability Rate of failure occurrence
Robustness (linh hoạt)	Time to restart after failure Percentage of events causing failure Probability of data corruption on failure
Portability (có khả năng di động)	Percentage of target dependent statements Number of target systems

# Tương tác giữa các yêu cầu

---

- Xung đột giữa các yêu cầu phi chức năng khác nhau là chuyện thường gặp trong các hệ thống phức tạp.



# Yêu cầu miền

## Domain requirements

---

- Các yêu cầu miền xuất phát từ miền ứng dụng (application domain), chúng mô tả các đặc điểm và tính chất hệ thống phản ánh miền ứng dụng đó.
- Các yêu cầu miền có thể là các yêu cầu chức năng mới, các ràng buộc về các yêu cầu đã có hoặc định nghĩa các tính toán cụ thể.
- Nếu các yêu cầu miền không được thỏa mãn, hệ thống có thể không hoạt động được.

# Yêu cầu miền của LIBSYS

---

- Cần có một chuẩn giao diện người dùng cho tất cả các cơ sở dữ liệu, chuẩn này cần dựa vào chuẩn **Z39.50**.
- Do các hạn chế về bản quyền, một số tài liệu phải được xóa ngay khi nhận được. Tùy theo các yêu cầu của người dùng, các tài liệu này sẽ được in tại chỗ tại máy chủ của hệ thống rồi chuyển bằng tay tới người dùng hoặc được in tại một máy in trong mạng.



# Rắc rối của các yêu cầu miền

---

- Mức độ hiểu được
  - Các yêu cầu được diễn đạt bằng ngôn ngữ của miền ứng dụng;
    - Thường khó hiểu đối với các kĩ sư phần mềm.
- Ấn ý
  - Các chuyên gia trong ngành hiểu rõ về lĩnh vực đang nói đến trong yêu cầu miền đến mức họ không nghĩ đến việc diễn đạt các yêu cầu miền một cách tường minh.

# Đặc điểm của yêu cầu người dùng tốt

---

- Nên mô tả các yêu cầu chức năng và phi chức năng sao cho người dùng hệ thống (những người không có kiến thức chi tiết về kỹ thuật) có thể hiểu được.
- Các yêu cầu người dùng được định nghĩa bằng ngôn ngữ tự nhiên, bảng biểu, sơ đồ mà tất cả người dùng đều hiểu được.

# Rắc rối với ngôn ngữ tự nhiên

---

- Thiếu rõ ràng
  - Khó diễn đạt chính xác mà không làm cho tài liệu trở nên khó đọc.
- Lẫn lộn yêu cầu
  - Các yêu cầu chức năng và phi chức năng có xu hướng bị **trộn lẫn**.
- Hỗn hợp yêu cầu
  - Vài yêu cầu khác nhau có thể được phát biểu đồng thời.

# Ví dụ: Yêu cầu của LIBSYS

---

LIBSYS cần cung cấp một hệ thống kế toán tài chính lưu trữ lại tất cả các khoản thanh toán do người sử dụng hệ thống thực hiện. Nhân viên quản lý hệ thống có thể cấu hình hệ thống này sao cho các khách quen có thể được hưởng khuyến mại.

# Yêu cầu về editor grid

---

**Tiện ích lưới** Để hỗ trợ việc định vị các thực thể trong một sơ đồ, người dùng có thể bật chế độ lưới (grid) theo đơn vị centimet hoặc inch, bằng một option tại control panel. Ban đầu, grid không ở trạng thái bật. Grid có thể được bật và tắt hoặc đổi giữa inch và centimet bất cứ lúc nào trong một phiên soạn thảo. Trong chế độ hiển thị reduce-to-fit (giảm kích thước cho vừa) cũng cần có option cho grid, nhưng số dòng grid được hiển thị sẽ được giảm để tránh tình trạng hiển thị lưới quá dày trong các sơ đồ nhỏ.

# Các rắc rối về yêu cầu

---

- Các yêu cầu cơ sở dữ liệu bao gồm cả thông tin mức khái niệm và thông tin chi tiết
  - Có miêu tả ở mức khái niệm về một hệ thống kế toán tài chính cần có trong hệ thống LIBSYS;
  - Tuy nhiên, nó chứa cả chi tiết rằng các nhân viên quản lý hệ thống có thể cấu hình hệ thống này – đây là chi tiết không cần thiết tại mức này.
- Yêu cầu về grid trộn lẫn ba loại yêu cầu khác nhau
  - Yêu cầu chức năng ở mức khái niệm (conceptual functional requirement): nhu cầu đối với grid;
  - Yêu cầu phi chức năng: đơn vị grid;
  - Yêu cầu phi chức năng về giao diện người dùng: bật tắt grid.

# Trình bày có cấu trúc

---

## Tiện ích grid

Trình soạn thảo sẽ cung cấp một tiện ích grid, trong đó, một lưới các đường thẳng dọc và ngang sẽ là nền cho cửa sổ soạn thảo. Grid có tính chất passive (bị động), việc cân chỉnh vị trí các thực thể thuộc về trách nhiệm của người dùng.

**Rationale:** Lưới vuông (grid) giúp người dùng vẽ sơ đồ với không gian hợp lý cho các thực thể. Mặc dù active grid (các thực thể có thể 'bắt' tọa độ lưới) có thể có ích, nhưng việc định vị lại không chính xác. Để cho người dùng quyết định vị trí của các thực thể là cách tốt nhất.

# Hướng dẫn viết tài liệu yêu cầu

---

- Đặt ra/chọn một định dạng chuẩn và dùng nó cho tất cả các yêu cầu.
- Sử dụng ngôn ngữ một cách nhất quán.
  - Dùng '**phải**'/'**sẽ**' cho các yêu cầu bắt buộc, '**nên**' cho các yêu cầu mong muốn (không bắt buộc).
- Dùng text highlighting để đánh dấu các phần quan trọng của yêu cầu.
- Tránh dùng từ lóng trong ngành IT (computer jargon).



# Yêu cầu hệ thống

---

- Là các đặc tả chi tiết hơn (so với yêu cầu người dùng) về các chức năng, dịch vụ và ràng buộc của hệ thống.
- Được dùng làm cơ sở cho việc thiết kế hệ thống.
- Có thể được tích hợp vào hợp đồng hệ thống.
- Có thể được định nghĩa hoặc minh họa bằng các mô hình hệ thống.

# Yêu cầu và thiết kế

---

- Về nguyên tắc, các yêu cầu nên phát biểu về những gì hệ thống cần làm (what), còn thiết kế mô tả cách thức hệ thống làm những việc đó (how).
- Trong thực tế, yêu cầu và thiết kế là không thể tách biệt
  - Một kiến trúc hệ thống có thể được thiết kế để cấu trúc hóa các yêu cầu;
  - Hệ thống có thể tương tác với các hệ thống khác và từ đó nảy sinh các yêu cầu về thiết kế;
  - Việc sử dụng một thiết kế cụ thể có thể chính là một domain requirement.

# Rắc rối với đặc tả bằng ngôn ngữ tự nhiên

---

- Mù mờ đa nghĩa – Ambiguity
  - Những người đọc và người viết tài liệu yêu cầu cần phải hiểu một nội dung theo cùng một kiểu. Ngôn ngữ tự nhiên có tính đa nghĩa cố hữu nên việc này rất khó.
- Quá linh động – Over-flexibility
  - Cùng một ý có thể được diễn đạt theo nhiều cách khác nhau trong đặc tả. Người đọc phải tự xác định xem hai yêu cầu có phải có cùng một nội dung
- Thiếu tính mô đun hóa
  - Các cấu trúc của ngôn ngữ tự nhiên không đủ để cấu trúc hóa các yêu cầu hệ thống. Ví dụ: khó tìm tất cả các yêu cầu có liên quan đến một điểm nào đó.

# Các lựa chọn khác cho đặc tả

---

Kí pháp	Miêu tả
Ngôn ngữ tự nhiên có cấu trúc	Cách này dựa vào việc định nghĩa các form hoặc template chuẩn để trình bày đặc tả yêu cầu.
Ngôn ngữ mô tả thiết kế ( <i>design description languages</i> )	Cách này sử dụng một ngôn ngữ tựa ngôn ngữ lập trình nhưng trừu tượng hơn để viết đặc tả cho các yêu cầu bằng cách định nghĩa một mô hình vận hành của hệ thống. Cách này ngày nay không còn thông dụng tuy nó có thể có ích cho các đặc tả giao diện.

# Các lựa chọn khác cho đặc tả (2)

Kí pháp	Miêu tả
Kí hiệu đồ họa	Một ngôn ngữ đồ họa, trợ giúp bởi các chú thích bằng text, được dùng để định nghĩa các yêu cầu chức năng của hệ thống. Ví dụ hiện đại là các mô tả ca sử dụng (use-case) và biểu đồ tuần tự (sequence diagram) đang được sử dụng rộng rãi.
Đặc tả bằng toán học	Đây là hệ thống kí hiệu dựa trên các khái niệm toán học như ô-tô-mát hữu hạn hoặc tập hợp. Các đặc tả đơn nghĩa này làm giảm tranh cãi giữa khách hàng và nhóm phát triển về chức năng hệ thống. Tuy nhiên, đa số khách hàng không hiểu các đặc tả hình thức và do dự khi chấp nhận dùng nó cho hợp đồng.

# Đặc tả bằng ngôn ngữ có cấu trúc

---

- Người viết yêu cầu được tự do trong phạm vi template định sẵn cho tài liệu yêu cầu.
- Tất cả các yêu cầu được viết theo một quy cách đã được chuẩn hóa.
- Các thuật ngữ sử dụng trong các miêu tả có thể bị giới hạn.
- Ưu điểm là giữ được hầu hết khả năng biểu đạt của ngôn ngữ, trong khi có thể đảm bảo áp đặt được một độ đồng nhất nào đó đối với tài liệu đặc tả.

# Đặc tả theo form

---

- Định nghĩa của chức năng (function) hoặc thực thể (entity).
- Mô tả về input và nguồn gốc input.
- Mô tả output và đích đến của output.
- Quy định về các thực thể khác cần đến.
- Các điều kiện – Pre and post conditions (nếu cần).
- Hiệu ứng phụ (nếu có) của chức năng.

# Đặc tả bằng bảng

---

- Dùng để bổ sung cho ngôn ngữ tự nhiên.
- Đặc biệt có ích khi ta cần định nghĩa một loạt các hướng có thể xảy ra.



# Mô hình đồ họa

---

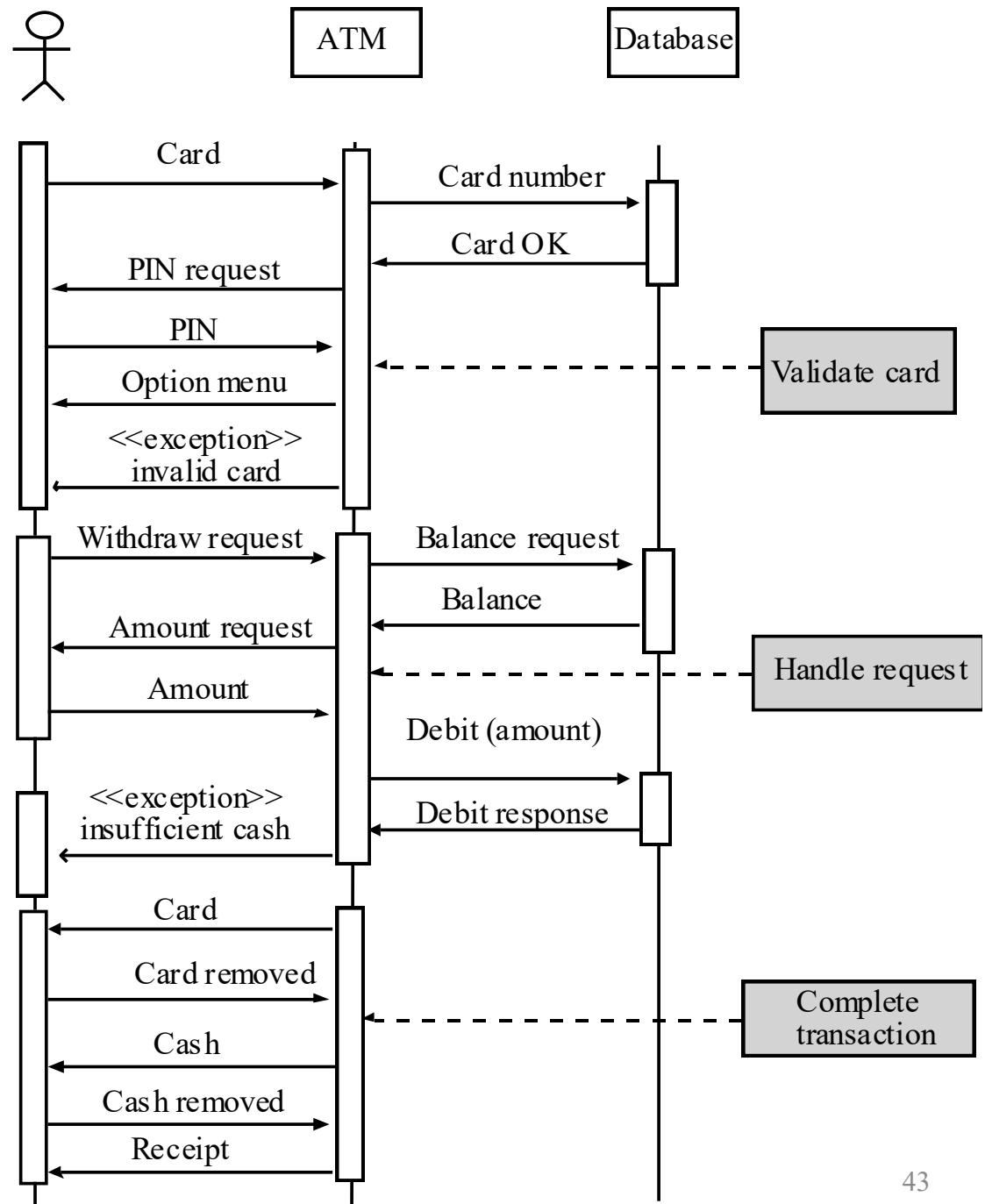
- Các mô hình đồ họa có ích nhất khi ta cần biểu diễn các thay đổi trạng thái hoặc khi cần mô tả một chuỗi hành động.

# Biểu đồ tuần tự - Sequence diagram

---

- Biểu diễn chuỗi các sự kiện xảy ra trong một quá trình tương tác giữa người dùng và một hệ thống.
- Ta đọc từ trên xuống dưới để thấy thứ tự các hành động xảy ra.
- Rút tiền từ một máy ATM
  - Validate card; (thẩm định thẻ)
  - Handle request; (xử lý yêu cầu)
  - Complete transaction. (hoàn thành giao dịch)

# Biểu đồ tuần tự cho việc rút tiền tại ATM



# Đặc tả giao diện

---

- Đa số các hệ thống phải tương tác với các hệ thống khác và các giao diện tương tác phải được đặc tả trong tài liệu yêu cầu.
- Có ba loại giao diện có thể phải định nghĩa
  - Procedural interfaces: quy cách gọi hàm giữa các hệ thống con (API – application programming interface)
  - Các cấu trúc dữ liệu sẽ được trao đổi giữa các hệ thống;
  - Các biểu diễn dữ liệu.
- Các kí pháp hình thức (formal notation) là một kĩ thuật hiệu quả cho đặc tả giao diện.

# PDL interface description

---

```
interface PrintServer {  
  
    // defines an abstract printer server  
    // requires: interface Printer, interface PrintDoc  
    // provides: initialize, print, displayPrintQueue,  
    cancelPrintJob, switchPrinter  
  
    void initialize ( Printer p ) ;  
    void print ( Printer p, PrintDoc d ) ;  
    void displayPrintQueue ( Printer p ) ;  
    void cancelPrintJob (Printer p, PrintDoc d) ;  
    void switchPrinter (Printer p1, Printer p2, PrintDoc d) ;  
} //PrintServer
```

# Tài liệu yêu cầu

---

- Tài liệu yêu cầu là phát biểu chính thức về những gì được đòi hỏi ở nhóm phát triển hệ thống.
- Cần bao gồm cả một định nghĩa yêu cầu người dùng và một đặc tả yêu cầu hệ thống.
- Nó KHÔNG phải là một tài liệu thiết kế. Nên cố gắng hết mức để nó nói về việc hệ thống cần làm CÁI GÌ thay vì hệ thống làm việc đó NHƯ THẾ NÀO.

# Ai dùng tài liệu yêu cầu?

Khách hàng hệ thống

Chỉ rõ chi tiết các yêu cầu và đọc chúng để kiểm tra xem chúng có thỏa mãn nhu cầu của họ hay không. Họ cũng chỉ ra các thay đổi đối với các yêu cầu

Nhà quản lý

Đề thương lượng giá cả cho hệ thống và lập kế hoạch quy trình phát triển hệ thống

Kỹ sư hệ thống

Để hiểu hệ thống cần được phát triển nó như thế nào

Kỹ sư thẩm định hệ thống

Để phát triển các test thẩm định dành cho hệ thống

Kỹ sư bảo trì hệ thống

Để hiểu hệ thống và quan hệ giữa các phần của hệ thống

# Chuẩn yêu cầu IEEE

---

- Định nghĩa một cấu trúc tổng quát cho một tài liệu yêu cầu cần được tạo cho mỗi hệ thống cụ thể.
  - Introduction. Giới thiệu
  - General description. Miêu tả chung
  - Specific requirements. Các yêu cầu cụ thể
  - Appendices. Phụ lục
  - Index. Chỉ mục



# Cấu trúc tài liệu yêu cầu

---

- Preface – lời nói đầu
- Introduction – giới thiệu
- Glossary – danh sách thuật ngữ
- User requirements definition – định nghĩa yêu cầu người dùng
- System architecture – kiến trúc hệ thống
- System requirements specification – đặc tả yêu cầu hệ thống
- System models – các mô hình hệ thống
- System evolution – sự tiến hóa của hệ thống
- Appendices – các phụ lục
- Index – chỉ mục

# Tổng kết

---

- Các yêu cầu trình bày về những gì hệ thống cần làm và quy định các ràng buộc về vận hành và cài đặt.
- Các yêu cầu chức năng trình bày các dịch vụ mà hệ thống cần cung cấp.
- Các yêu cầu phi chức năng quy định các ràng buộc đối với hệ thống đang được phát triển hoặc quy trình phát triển hệ thống.
- Các yêu cầu người dùng là các phát biểu ở mức cao về những gì hệ thống cần làm. Yêu cầu người dùng nên được viết bằng ngôn ngữ tự nhiên, bảng biểu và sơ đồ.

# Tổng kết (tiếp)

---

- Các yêu cầu hệ thống nhằm mô tả các chức năng mà hệ thống cần cung cấp.
- Một tài liệu yêu cầu phần mềm là một phát biểu đã được thống nhất về các yêu cầu hệ thống.
- Chuẩn IEEE là xuất phát điểm hữu ích cho việc định nghĩa các chuẩn yêu cầu cụ thể chi tiết hơn.