



HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

Khoa Viễn thông 1

# Nội dung : Kỹ nghệ yêu cầu phần mềm (Requirement Engineering)

Giảng viên: TS. Lâm Đức Dương



## Nội dung: Kỹ nghệ yêu cầu phần mềm (Requirement Engineering)

---

Chương này bao gồm giai đoạn ban đầu quan trọng của phát triển phần mềm: hiểu và xác định những gì phần mềm nên làm.

1. Khái niệm và tầm quan trọng của yêu cầu phần mềm
2. Phân loại yêu cầu phần mềm
3. Tiến trình kỹ nghệ yêu cầu
4. Mô hình hóa yêu cầu

# 1. Khái niệm và tầm quan trọng

---

## **Yêu cầu phần mềm**

- Là khâu kỹ thuật đầu tiên của quá trình phát triển phần mềm. Thiếu nó không thể tiếp tục quá trình
- Là sự phối hợp của nhà phát triển và khách hàng
- Nó quyết định chất lượng phần mềm đạt được với chi phí dự kiến và thời hạn cho trước
- Quá trình thiết lập các dịch vụ mà khách hàng yêu cầu từ một hệ thống và các ràng buộc mà hệ thống đó vận hành và được phát triển theo.
- Các yêu cầu hệ thống là những mô tả về các dịch vụ và ràng buộc của hệ thống được tạo ra trong quá trình kỹ thuật yêu cầu.

## Kỹ nghệ yêu cầu là gì?

---

Có thể bao gồm từ một phát biểu trừu tượng ở mức cao về một dịch vụ hoặc một ràng buộc hệ thống cho đến một đặc tả chức năng toán học chi tiết.

Điều này là không thể tránh khỏi vì các yêu cầu có thể đóng vai trò kép:

- Có thể là cơ sở cho việc đấu thầu hợp đồng - do đó phải có thể được diễn giải theo nhiều cách;
- Có thể là cơ sở cho chính hợp đồng - do đó phải được định nghĩa chi tiết;
- Cả hai loại phát biểu này đều có thể được gọi là yêu cầu.



## 1. Khái niệm và tầm quan trọng

---

- **Các yêu cầu** là các mô tả trừu tượng đến chi tiết về **dịch vụ mà hệ thống cung cấp** cũng như các **ràng buộc** lên sự phát triển và hoạt động của nó
- Mục đích của các yêu cầu:
  - Làm cơ sở cho việc mời thầu (cần có giải thích từ phía chủ đầu tư)
  - Làm cơ sở cho việc ký hợp đồng thầu (cần đủ và chi tiết)
  - Chương này bao gồm giai đoạn ban đầu quan trọng của phát triển phần mềm: hiểu và xác định những gì phần mềm nên làm.



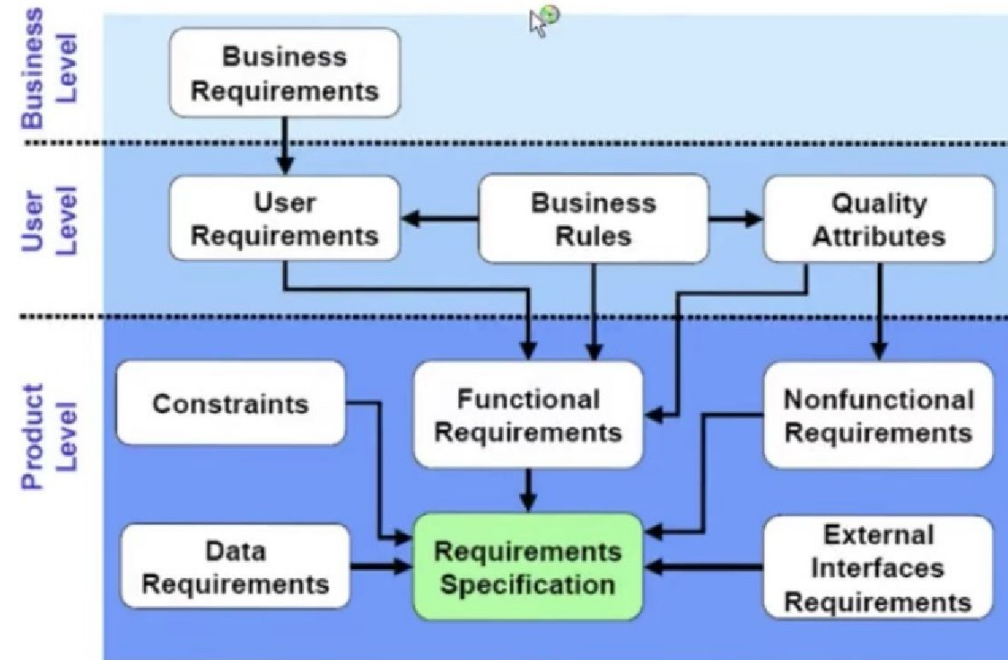
## 2. Phân loại yêu cầu phần mềm

### Business Level:

- Xác định các vấn đề kinh doanh hay cơ hội kinh doanh để giải quyết các sản phẩm phần mềm
- Xác định lý do tại sao các sản phẩm phần mềm đang được phát triển

### User Level:

- Trình bày tổng quát các dịch vụ và ràng buộc của hệ thống
- Tập trung vào các chức năng của hệ thống từ quan điểm của người sử dụng
- Xác định những gì hệ thống sẽ cung cấp để đạt được mục tiêu của người sử dụng
- Nhiều yêu cầu người sử dụng để thực hiện một yêu cầu doanh nghiệp duy nhất
- Chủ yếu cho người dùng cuối và viết bằng ngôn ngữ tự nhiên cộng với sơ đồ



### Product Level:

- Mô tả chi tiết các dịch vụ và ràng buộc của hệ thống
- Xác định các chức năng đó phải được xây dựng vào phần mềm để thực hiện nhiệm vụ của người sử dụng
- Nhiều yêu cầu cấp độ sản phẩm
- Viết bằng sơ đồ + ngôn ngữ tự nhiên có cấu trúc và ký hiệu toán học



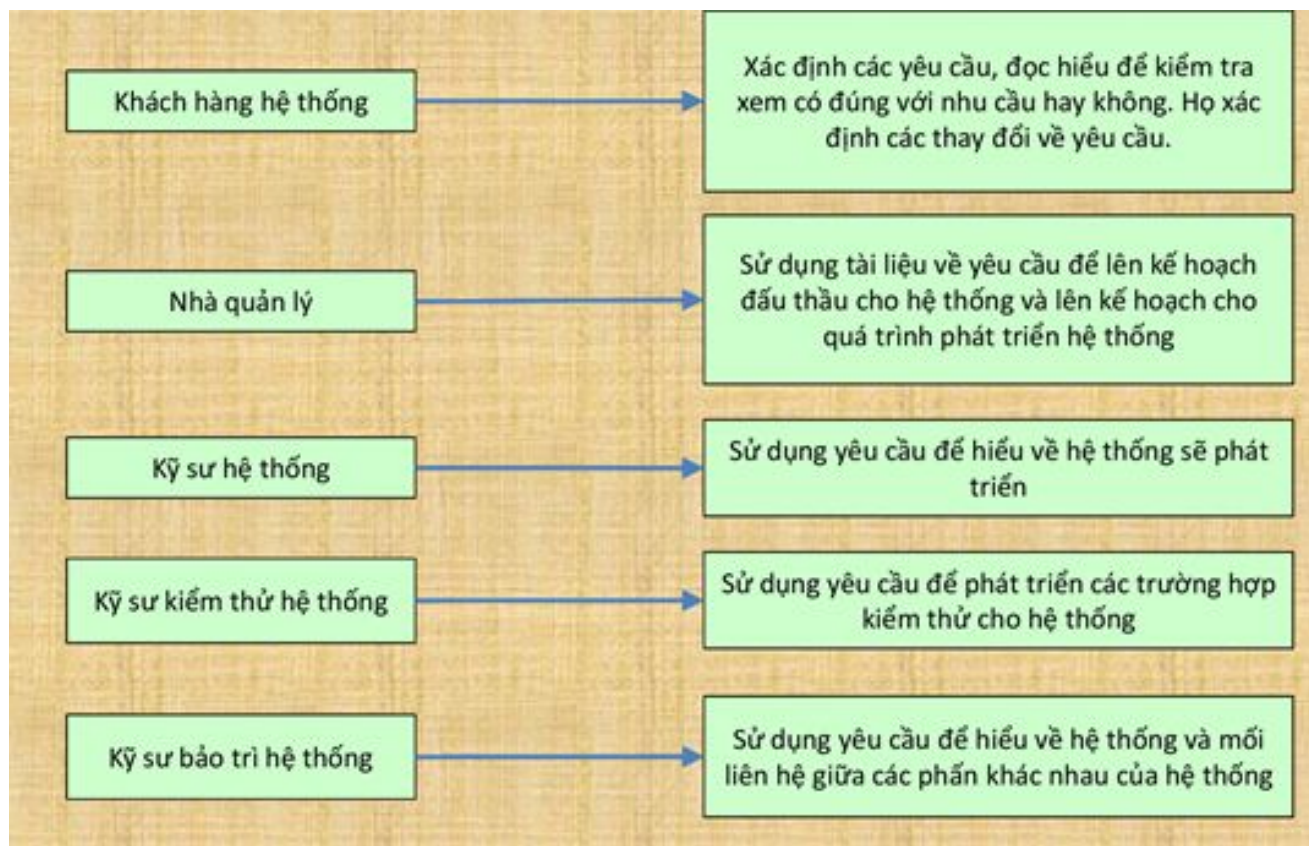
## 2. Phân loại yêu cầu phần mềm

---

- 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 phi chức năng
  - Các ràng buộc khác



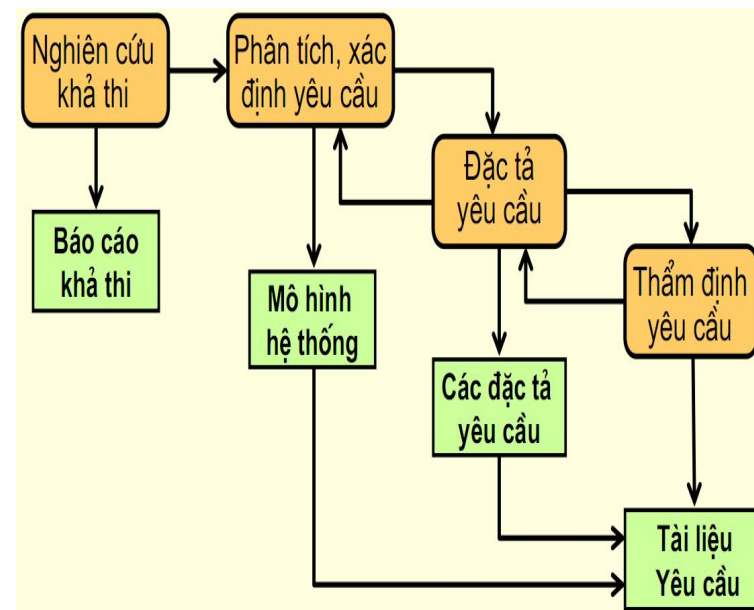
# Tầm ảnh hưởng của tài liệu yêu cầu





### 3. Tiến trình kỹ nghệ yêu cầu

- Các hoạt động của tiến trình kỹ nghệ
  - Nghiên cứu khả thi
  - Xác định, phân tích yêu cầu
  - Đặc tả yêu cầu
  - Xác nhận yêu cầu





## Quy trình Kỹ thuật Yêu cầu

---

- Quy trình này bao gồm một số hoạt động chính:
  - Nghiên cứu khả thi,
  - Xác định/Thu thập yêu cầu,
  - Phân tích yêu cầu,
  - Đặc tả yêu cầu
  - Xác nhận yêu cầu.

Thứ tự điển hình là Nghiên cứu Khả thi, Xác định/Thu thập, Phân tích và Đặc tả.



## Nghiên cứu Khả thi

---

- Đánh giá liệu dự án có khả thi hay không.
- Bao gồm việc xem xét các yếu tố như công nghệ, tiền bạc, thời gian và tài nguyên.
- Nó kiểm tra xem khả năng tài chính và công nghệ hiện tại của tổ chức có đủ hay không.
- Khả thi về kinh tế, kỹ thuật và triển khai là những khía cạnh chính. Khả thi về chất lượng cũng được đề cập.

## Xác định/Thu thập Yêu cầu

---

- Quá trình giao tiếp với khách hàng và người dùng để xác định các yêu cầu của họ.
- Các yêu cầu thường khó khám phá và dễ thay đổi. Chúng không phải lúc nào cũng được biết chính xác ngay từ đầu.
- Các kỹ thuật bao gồm:
  - **Phỏng vấn.**
  - **Quan sát** (người dùng thực hiện các tác vụ).
  - **Hội thảo được tạo điều kiện (FAST - Facilitated Application Specification Techniques):**  
Bao gồm các nhà phát triển và khách hàng làm việc cùng nhau. Một kết quả chính là phát triển một bộ yêu cầu ban đầu. Các kỹ sư phần cứng/phần mềm, đại diện thị trường và nhân viên tài chính cấp cao có thể là những người tham gia, nhưng đại diện của nhà sản xuất thường không phù hợp.
  - **Bảng câu hỏi & Khảo sát:** Hữu ích để thu thập ý kiến từ một số lượng lớn người dùng hoặc khách hàng.
  - **Kỹ thuật Delphi:** Yêu cầu sự đồng thuận của chuyên gia.
  - **Tạo nguyên mẫu:** Xây dựng một mô hình hoạt động để làm rõ yêu cầu với người dùng.
  - **Trường hợp sử dụng:** Mô tả các tương tác của người dùng để xác định yêu cầu.

## Phân tích Yêu cầu

---

- Bao gồm việc làm rõ và tổ chức các yêu cầu đã thu thập.
- Bao gồm các hoạt động như định nghĩa, tổng hợp, mô hình hóa, xác nhận và xem xét vấn đề.
- Phân tích dữ liệu xem xét các thuộc tính như cấu trúc dữ liệu, tính đầy đủ, bảo mật và kích thước (khối lượng).
- **Khối lượng** đề cập đến lượng dữ liệu được tạo ra hoặc các giao dịch kinh doanh được xử lý trong một khoảng thời gian nhất định.
- Kết quả phân tích là các mô hình biểu diễn chức năng và hành vi. Các mô hình khác như kiến trúc, cấu trúc, dữ liệu, độ tin cậy và khả năng sử dụng cũng được đề cập trong bối cảnh mô hình hóa yêu cầu, nhưng chức năng và hành vi là trọng tâm.
- Phân rã vấn đề có thể được xem là từ dưới lên và từ trên xuống hoặc ngang và dọc hoặc phụ thuộc và cấp trên.
- Quan điểm tác nhân được coi là quan điểm đầu tiên trong phân tích yêu cầu phần mềm.

## Đặc tả Yêu cầu

- Quá trình ghi lại các yêu cầu của người dùng và hệ thống một cách rõ ràng và chính xác.
- Đầu ra chính là tài liệu **Đặc tả Yêu cầu Phần mềm (SRS)**.
- SRS là một tuyên bố chính thức về các yêu cầu phần mềm từ góc nhìn của các nhà phát triển hệ thống.
- Nó mô tả những gì hệ thống phải làm, bao gồm các yêu cầu chức năng, phi chức năng và miền.
- Tài liệu SRS có thể đóng vai trò là cơ sở cho một hợp đồng giữa nhà phát triển và khách hàng.
- IEEE 830-1998 là một khuyến nghị tiêu chuẩn về cấu trúc và nội dung của một SRS. Tài liệu SRS thường bao gồm các yêu cầu chức năng và phi chức năng và mục tiêu hiệu suất nhưng không bao gồm hướng dẫn sử dụng.
- Các yêu cầu có thể được mô tả ở các cấp độ khác nhau: **Xác định Yêu cầu** và **Đặc tả Yêu cầu** (Tài liệu hóa). Đặc tả cung cấp một mô tả chi tiết.
- Các phương pháp để đặc tả bao gồm:
  - **Đặc tả Hình thức:** Sử dụng các ký hiệu toán học. Ưu điểm: Tăng độ chính xác. Nhược điểm: Khó đọc và sử dụng, tốn kém.
  - **Đặc tả Phi hình thức:** Sử dụng ngôn ngữ tự nhiên (như tiếng Việt). Ưu điểm: Dễ đọc và sử dụng, linh hoạt. Nhược điểm: Thiếu độ chính xác, dễ mơ hồ.
  - **Đặc tả Bán hình thức:** Sử dụng các sơ đồ và văn bản có cấu trúc. Đặc tả dựa trên sơ đồ phổ biến trong các tài liệu SRS.
  - Các kiểu kiến trúc (Luồng dữ liệu, Gọi và trả về, Phân lớp) là các loại mô hình kiến trúc, không phải là các phương pháp đặc tả.
- Các biểu diễn yêu cầu phải nhất quán, phù hợp về hình thức và nội dung, và có thể xem xét lại.

## Xác nhận Yêu cầu

---

- Kiểm tra xem các yêu cầu đã xác định có đầy đủ, nhất quán và phản ánh chính xác những gì khách hàng cần hay không.
- Nó được thực hiện bởi nhà phân tích và các bên liên quan (bao gồm khách hàng và người dùng).
- Các hoạt động bao gồm làm rõ yêu cầu, xem xét yêu cầu và tài liệu hóa yêu cầu. Thứ tự là làm rõ, tài liệu hóa, xem xét.
- Các kỹ thuật bao gồm xem xét yêu cầu, kiểm tra với khách hàng, tạo nguyên mẫu và tạo các trường hợp kiểm thử.
- Trong quá trình xem xét, các nhà phân tích nên làm rõ các thuật ngữ mơ hồ như "một số", "đôi khi", "thường", "thường xuyên", "bình thường", "chủ yếu", "đa số".
- Bảng truy vết (hoặc ma trận) được sử dụng để xác định, kiểm soát và truy vết các thay đổi đối với yêu cầu.
- Các yêu cầu phải chính xác và nhất quán. Chúng cũng phải đầy đủ và thực tế.
- Chi phí do lỗi yêu cầu thì cao, vì vậy việc xác nhận là rất quan trọng.
- Việc sửa một lỗi yêu cầu sau khi đã bàn giao có thể tốn kém gấp đến 100 lần chi phí sửa một lỗi triển khai.



## Trừu tượng hóa yêu cầu

---

- Nếu một công ty muốn giao thầu cho một dự án phát triển phần mềm lớn, họ phải xác định nhu cầu của mình một cách đủ trừu tượng để giải pháp không bị định trước.
- Các yêu cầu phải được viết ra sao cho nhiều nhà thầu có thể đấu thầu hợp đồng, có lẽ đề xuất những cách khác nhau để đáp ứng nhu cầu của tổ chức khách hàng.
- Một khi hợp đồng đã được trao, nhà thầu phải viết một bản định nghĩa hệ thống chi tiết hơn cho khách hàng để khách hàng hiểu và có thể xác nhận những gì phần mềm sẽ làm.
- Cả hai tài liệu này đều có thể được gọi là tài liệu yêu cầu cho hệ thống.



## 1. Yêu cầu người dùng (User requirements)

- Các phát biểu bằng ngôn ngữ tự nhiên kèm theo các biểu đồ về những dịch vụ mà hệ thống cung cấp và các ràng buộc vận hành của nó. Được viết cho khách hàng.

## 2. Yêu cầu hệ thống (System requirements)

- Một tài liệu có cấu trúc trình bày các mô tả chi tiết về chức năng, dịch vụ và ràng buộc vận hành của hệ thống. Nó định nghĩa những gì cần được triển khai, vì vậy có thể là một phần của hợp đồng giữa khách hàng và nhà thầu.



## Phương pháp Agile và yêu cầu

---

- Nhiều phương pháp linh hoạt cho rằng việc tạo ra các yêu cầu hệ thống chi tiết là lãng phí thời gian vì các yêu cầu thay đổi quá nhanh.
- Do đó, tài liệu yêu cầu luôn bị lỗi thời.
- Các phương pháp linh hoạt thường sử dụng kỹ thuật yêu cầu tăng trưởng và có thể diễn đạt yêu cầu dưới dạng ‘câu chuyện người dùng’.
- Điều này là thiết thực đối với các hệ thống nghiệp vụ nhưng lại có vấn đề đối với các hệ thống đòi hỏi phân tích trước khi bàn giao (ví dụ: các hệ thống trọng yếu) hoặc các hệ thống được phát triển bởi nhiều nhóm



# Yêu cầu chức năng và phi chức năng

---

## 1. Yêu cầu chức năng (Functional requirements)

- Các phát biểu về dịch vụ mà hệ thống cần cung cấp, cách hệ thống cần phản ứng với các đầu vào cụ thể và cách hệ thống cần hoạt động trong các tình huống cụ thể. Có thể nêu rõ những gì hệ thống *không* nên làm.

## 2. Yêu cầu phi chức năng (Non-functional requirements)

- Các ràng buộc đối với các dịch vụ hoặc chức năng do hệ thống cung cấp, ví dụ như ràng buộc về thời gian, ràng buộc về quy trình phát triển, các tiêu chuẩn, v.v. Thường áp dụng cho toàn bộ hệ thống thay vì các tính năng hoặc dịch vụ riêng lẻ.

## 3. Yêu cầu miền lĩnh vực (Domain requirements)

- Các ràng buộc đối với hệ thống xuất phát từ lĩnh vực vận hành của nó.

Các yêu cầu cũng có thể được phân biệt là **Yêu cầu Người dùng** (trừu tượng) và **Yêu cầu Hệ thống**.



## Yêu cầu chức năng

---

- Yêu cầu chức năng mô tả chức năng hoặc các dịch vụ hệ thống.
- Chúng phụ thuộc vào loại phần mềm, người dùng dự kiến và loại hệ thống nơi phần mềm được sử dụng.
- Các yêu cầu chức năng của người dùng có thể là các phát biểu mức cao về những gì hệ thống cần làm.
- Các yêu cầu chức năng của hệ thống cần mô tả chi tiết các dịch vụ hệ thống.



## Sự thiếu chính xác của yêu cầu

---

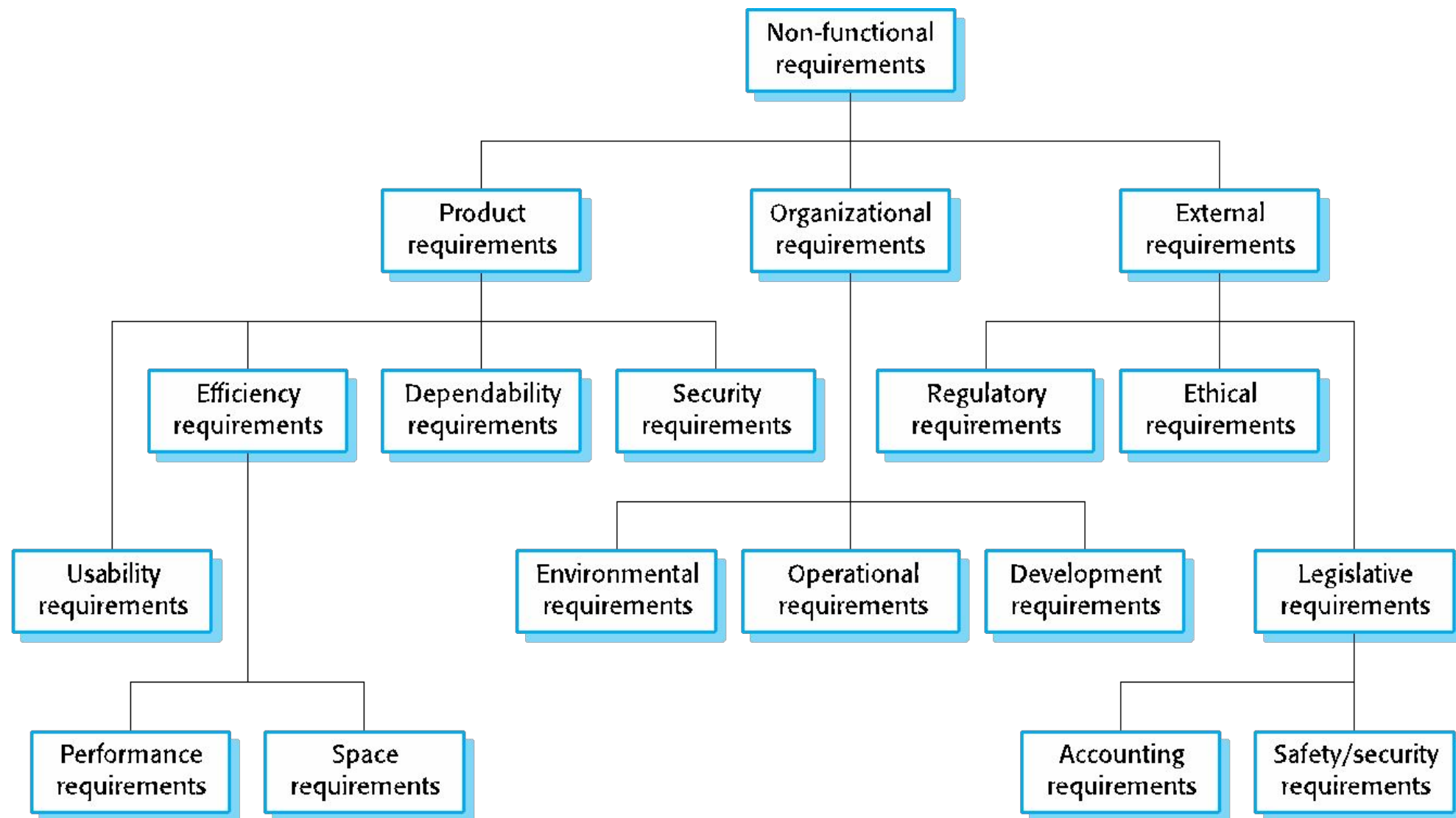
- Các vấn đề nảy sinh khi yêu cầu chức năng không được phát biểu một cách chính xác.
- Các yêu cầu mơ hồ có thể được các nhà phát triển và người dùng diễn giải theo những cách khác nhau.

## Yêu cầu phi chức năng

---

- Các yêu cầu phi chức năng định nghĩa các thuộc tính và ràng buộc của hệ thống, ví dụ: độ tin cậy, thời gian phản hồi và yêu cầu về lưu trữ. Các ràng buộc này có thể là khả năng của thiết bị Nhập/Xuất (I/O), các biểu diễn hệ thống, v.v.
- Các yêu cầu về quy trình cũng có thể được chỉ định, bắt buộc một IDE, ngôn ngữ lập trình hoặc phương pháp phát triển cụ thể.
- Yêu cầu phi chức năng có thể quan trọng hơn yêu cầu chức năng. Nếu chúng không được đáp ứng, hệ thống có thể trở nên vô dụng.

# Các loại yêu cầu Phi chức năng





## Việc triển khai yêu cầu phi chức năng

---

- Yêu cầu phi chức năng có thể ảnh hưởng đến kiến trúc tổng thể của hệ thống thay vì các thành phần riêng lẻ.

Ví dụ, để đảm bảo các yêu cầu về hiệu năng được đáp ứng, bạn có thể phải tổ chức hệ thống để giảm thiểu sự giao tiếp giữa các thành phần.

- Một yêu cầu phi chức năng duy nhất, ví dụ như yêu cầu về bảo mật, có thể tạo ra một số yêu cầu chức năng liên quan nhằm định nghĩa các dịch vụ hệ thống cần thiết. Nó cũng có thể tạo ra các yêu cầu nhằm hạn chế các yêu cầu hiện có



# Phân loại Yêu cầu phi chức năng

---

## 1. Yêu cầu sản phẩm (Product requirements)

- Các yêu cầu quy định rằng sản phẩm được bàn giao phải hoạt động theo một cách cụ thể, ví dụ: tốc độ thực thi, độ tin cậy, v.v.

## 2. Yêu cầu tổ chức (Organisational requirements)

- Các yêu cầu là hệ quả của các chính sách và quy trình của tổ chức, ví dụ: các tiêu chuẩn quy trình được sử dụng, các yêu cầu về triển khai, v.v.

## 3. Yêu cầu bên ngoài (External requirements)

- Các yêu cầu này sinh từ các yếu tố bên ngoài hệ thống và quy trình phát triển của nó, ví dụ: yêu cầu về khả năng tương tác, yêu cầu về luật pháp, v.v.



## Ví dụ về yêu cầu phi chức năng trong hệ thống Mentcare

---

### 1. Yêu cầu sản phẩm (Product requirement)

- Hệ thống Mentcare phải khả dụng cho tất cả các phòng khám trong giờ làm việc bình thường (Thứ Hai–Thứ Sáu, 08:30–17:30). Thời gian ngừng hoạt động trong giờ làm việc bình thường sẽ không được vượt quá năm giây trong bất kỳ ngày nào.

### 2. Yêu cầu tổ chức (Organizational requirement)

- Người dùng của hệ thống Mentcare phải xác thực bản thân bằng thẻ định danh do cơ quan y tế cấp.

### 3. Yêu cầu bên ngoài (External requirement)

- Hệ thống phải triển khai các điều khoản về quyền riêng tư của bệnh nhân như được quy định trong luật



## Mục tiêu và yêu cầu

---

**Yêu cầu phi chức năng** có thể rất khó để phát biểu một cách chính xác và các yêu cầu thiếu chính xác có thể khó kiểm chứng.

### **Mục tiêu (Goal)**

Một ý định chung chung của người dùng, ví dụ như tính dễ sử dụng.

### **Yêu cầu phi chức năng có thể kiểm chứng được (Verifiable non-functional requirement)**

Một phát biểu sử dụng một thước đo nào đó mà có thể được kiểm tra một cách khách quan.

Các mục tiêu hữu ích cho nhà phát triển vì chúng truyền đạt được ý định của người dùng hệ thống.



## Các yêu cầu về tính dễ sử dụng

---

- Hệ thống nên dễ sử dụng bởi nhân viên y tế và nên được tổ chức sao cho giảm thiểu lỗi của người dùng. (Mục tiêu)
- Nhân viên y tế phải có khả năng sử dụng tất cả các chức năng hệ thống sau bốn giờ đào tạo. Sau khóa đào tạo này, số lỗi trung bình mắc phải bởi người dùng có kinh nghiệm sẽ không vượt quá hai lỗi mỗi giờ sử dụng hệ thống. (Yêu cầu phi chức năng có thể kiểm chứng được)



## Các chỉ số để đặc tả yêu cầu phi chức năng

---

- Tốc độ (Speed)
- Kích thước (Size)
- Tính dễ sử dụng (Ease of use)
- Độ tin cậy (Reliability)
- Tính mạnh mẽ (Robustness)
- Khả năng di chuyển (Portability)



# Các chỉ số để đặc tả yêu cầu phi chức năng

---

## Thuộc tính (Property)

### Tốc độ (Speed)

## Thước đo (Measure)

Số giao dịch xử lý/giây (Processed transactions/second)

Thời gian phản hồi người dùng/sự kiện (User/event response time)

Thời gian làm mới màn hình (Screen refresh time)

### Kích thước (Size)

Mbytes (Megabytes)

Số lượng chip ROM (Number of ROM chips)

### Tính dễ sử dụng (Ease of use)

Thời gian đào tạo (Training time)

Số lượng khung/màn hình trợ giúp (Number of help frames)

### Độ tin cậy (Reliability)

Thời gian trung bình đến khi xảy ra lỗi (Mean time to failure - MTTF/MTBF)

Xác suất không khả dụng (Probability of unavailability)

Tần suất xảy ra lỗi / Tỷ lệ lỗi (Rate of failure occurrence)

Tính sẵn sàng / Độ khả dụng (Availability)

### Tính mạnh mẽ (Robustness)

Thời gian khởi động lại sau lỗi (Time to restart after failure)

Tỷ lệ phần trăm sự kiện gây ra lỗi (Percentage of events causing failure)

Xác suất hỏng dữ liệu khi xảy ra lỗi (Probability of data corruption on failure)



# Các chỉ số để đặc tả yêu cầu phi chức năng - Ví dụ

Thuộc tính (Property)	Thước đo (Measure / Metric)	Ví dụ yêu cầu sử dụng thước đo
Tốc độ (Speed)	Thời gian phản hồi (Response time) - tính bằng giây, mili giây	"Thời gian phản hồi cho chức năng tìm kiếm phải dưới 2 giây."
	Thông lượng (Throughput) - giao dịch/giây, yêu cầu/phút	"Hệ thống phải xử lý được ít nhất 500 giao dịch đặt hàng mỗi phút."
Kích thước (Size)	KBytes, MBytes (dung lượng bộ nhớ hoặc ổ cứng)	"Dung lượng bộ nhớ RAM tối đa mà ứng dụng sử dụng không được vượt quá 512 MBytes."
Tính dễ sử dụng (Ease of use)	Thời gian hoàn thành tác vụ (Task completion time) - phút, giây	"Người dùng mới phải hoàn thành việc đăng ký tài khoản trong vòng 3 phút."
	Tỷ lệ lỗi (Error rate) - số lỗi/tác vụ, tỷ lệ %	"Tỷ lệ lỗi khi nhập dữ liệu bệnh nhân không được vượt quá 1%."
	Thời gian đào tạo (Training time) - giờ	"Nhân viên mới cần tối đa 4 giờ đào tạo để sử dụng thành thạo các chức năng cơ bản."
Độ tin cậy (Reliability)	Thời gian trung bình giữa các lỗi (Mean Time Between Failures - MTBF)	"MTBF của hệ thống phải đạt ít nhất 1000 giờ hoạt động liên tục."
	Tính sẵn sàng (Availability) - tỷ lệ %	"Hệ thống phải đảm bảo tính sẵn sàng 99.95% trong giờ hành chính."
Tính mạnh mẽ (Robustness)	Thời gian phục hồi sau lỗi (Time to recover from failure)	"Hệ thống phải tự động khởi động lại và phục hồi hoạt động trong vòng 5 phút sau khi gặp sự cố."
Khả năng di chuyển	Số bài điều hành/pấn tăng hỗ trợ	"Ứng dụng phải chạy được trên cả Windows 10, Windows 11 và



## Các quy trình kỹ thuật yêu cầu

---

Các quy trình được sử dụng cho Kỹ thuật Yêu cầu (RE) rất đa dạng, phụ thuộc vào lĩnh vực ứng dụng, những người tham gia và tổ chức phát triển các yêu cầu. Tuy nhiên, có một số hoạt động chung phổ biến cho tất cả các quy trình:

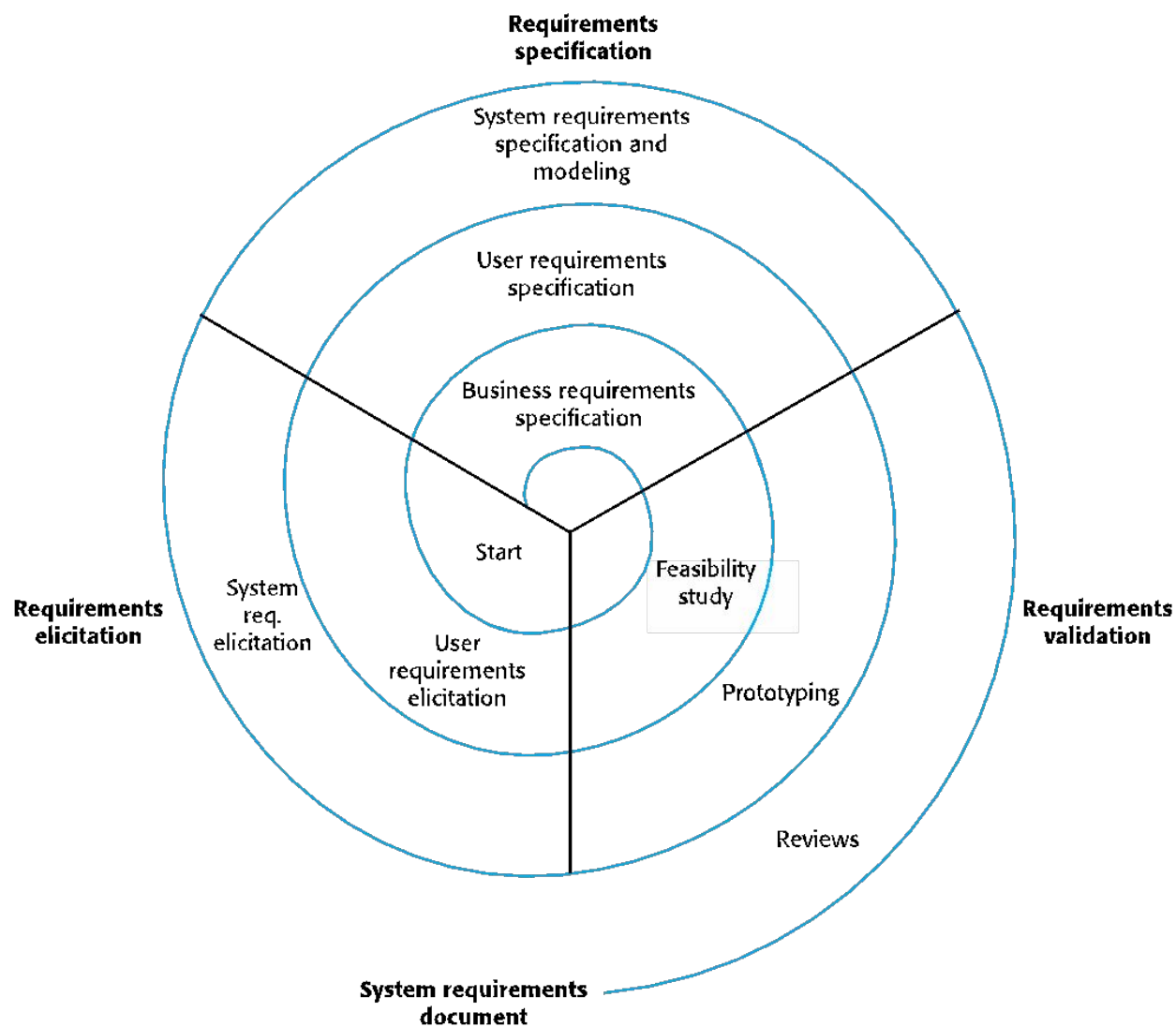
- Thu thập yêu cầu (Requirements elicitation);
- Phân tích yêu cầu (Requirements analysis);
- Xác nhận yêu cầu (Requirements validation);
- Quản lý yêu cầu (Requirements management).

Trong thực tế, Kỹ thuật Yêu cầu là một hoạt động có tính lặp, trong đó các hoạt động này được thực hiện xen kẽ nhau





# Mô hình xoắn ốc của quy trình kỹ thuật yêu cầu



## Thu thập yêu cầu

---

- Đôi khi được gọi là thu thập yêu cầu (requirements elicitation) hoặc khám phá yêu cầu (requirements discovery).
- Hoạt động này bao gồm việc nhân viên kỹ thuật làm việc với khách hàng để tìm hiểu về lĩnh vực ứng dụng, các dịch vụ mà hệ thống cần cung cấp và các ràng buộc vận hành của hệ thống.
- Có thể bao gồm sự tham gia của người dùng cuối, các nhà quản lý, các kỹ sư bảo trì, các chuyên gia lĩnh vực, các công đoàn, v.v. Những đối tượng này được gọi là các bên liên quan (stakeholders)



## Thu thập yêu cầu

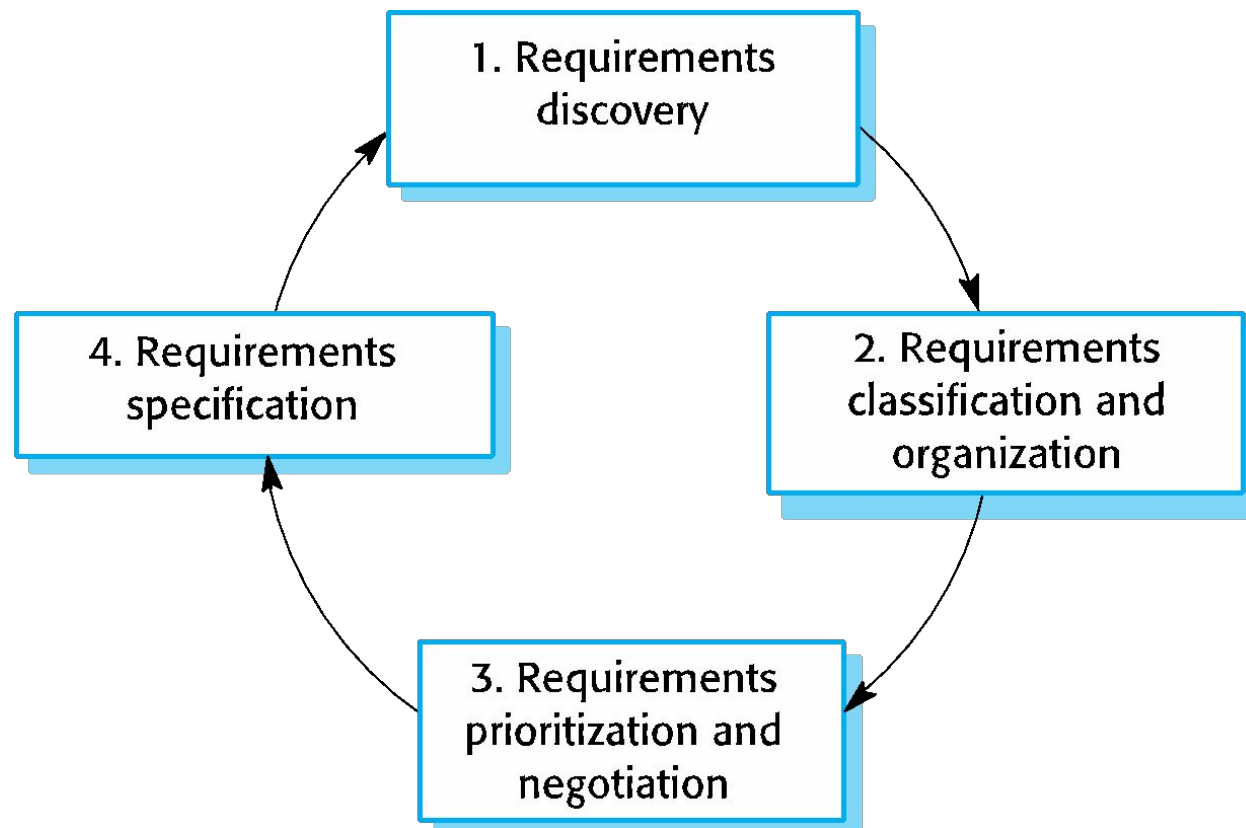
---

Các kỹ sư phần mềm làm việc với một loạt các bên liên quan của hệ thống để tìm hiểu về lĩnh vực ứng dụng, các dịch vụ mà hệ thống cần cung cấp, hiệu năng hệ thống được yêu cầu, các ràng buộc phần cứng, các hệ thống khác, v.v. Các giai đoạn bao gồm:

- Khám phá yêu cầu (Requirements discovery),
- Phân loại và tổ chức yêu cầu (Requirements classification and organization),
- Ưu tiên hóa và đàm phán yêu cầu (Requirements prioritization and negotiation),
- Đặc tả yêu cầu (Requirements specification)



# Quy trình thu thập và phân tích yêu cầu





## Nghiên cứu tổ chức

---

- Một nhà khoa học xã hội dành một khoảng thời gian đáng kể để quan sát và phân tích cách mọi người làm việc trong thực tế.
- Mọi người không cần phải giải thích hay diễn đạt rõ ràng về công việc của họ.
- Các yếu tố xã hội và tổ chức quan trọng có thể được quan sát thấy.
- Các nghiên cứu dân tộc học đã chỉ ra rằng công việc thường phong phú và phức tạp hơn so với những gì được thể hiện bởi các mô hình hệ thống đơn giản.



## Đặc tả yêu cầu

---

- Đặc tả yêu cầu là quy trình viết ra các yêu cầu người dùng và yêu cầu hệ thống vào trong một tài liệu yêu cầu.
- Yêu cầu người dùng phải có thể hiểu được bởi người dùng cuối và khách hàng, những người không có chuyên môn kỹ thuật.
- Yêu cầu hệ thống là các yêu cầu chi tiết hơn và có thể bao gồm nhiều thông tin kỹ thuật hơn.
- Các yêu cầu có thể là một phần của hợp đồng cho việc phát triển hệ thống. Do đó, điều quan trọng là chúng phải đầy đủ nhất có thể



## Các cách thức viết một bản đặc tả yêu cầu hệ thống

---

**Ngôn ngữ tự nhiên** (Natural language)

**Ngôn ngữ tự nhiên có cấu trúc** (Structured natural language)

**Ngôn ngữ mô tả thiết kế** (Design description languages - DDLs)

**Ký hiệu đồ họa** (Graphical notations)

**Đặc tả toán học / Đặc tả hình thức** (Mathematical specifications / Formal Specifications)



# Các cách thức viết một bản đặc tả yêu cầu hệ thống

Ký hiệu (Notation)	Mô tả (Description)
<b>Ngôn ngữ tự nhiên</b> (Natural language)	Các yêu cầu được viết bằng các câu được đánh số bằng ngôn ngữ tự nhiên. Mỗi câu nên diễn đạt một yêu cầu.
<b>Ngôn ngữ tự nhiên có cấu trúc</b> (Structured natural language)	Các yêu cầu được viết bằng ngôn ngữ tự nhiên trên một biểu mẫu (form) hoặc khuôn mẫu (template) chuẩn. Mỗi trường (field) cung cấp thông tin về một khía cạnh (aspect) của yêu cầu.
<b>Ngôn ngữ mô tả thiết kế</b> (Design description languages - DDLs)	Phương pháp này sử dụng một ngôn ngữ giống như ngôn ngữ lập trình, nhưng với các đặc trưng trừu tượng (abstract features) hơn để đặc tả yêu cầu bằng cách định nghĩa một mô hình hoạt động (operational model) của hệ thống. Phương pháp này hiện nay ít được sử dụng mặc dù nó có thể hữu ích cho việc đặc tả giao diện (interface specifications)
<b>Ký hiệu đồ họa</b> (Graphical notations)	Các mô hình đồ họa (graphical models), được bổ sung bởi các chú thích văn bản (text annotations), được sử dụng để định nghĩa các yêu cầu chức năng cho hệ thống; Sơ đồ use case và sơ đồ tuần tự UML thường được sử dụng phổ biến.
<b>Đặc tả toán học / Đặc tả hình thức</b> (Mathematical specifications / Formal Specifications)	Các ký hiệu này dựa trên các khái niệm toán học (mathematical concepts) như máy trạng thái hữu hạn (finite-state machines) hoặc tập hợp (sets). Mặc dù các đặc tả không mơ hồ (unambiguous specifications) này có thể làm giảm sự mơ hồ trong tài liệu yêu cầu, hầu hết khách hàng không hiểu được một đặc tả hình thức (formal specification). Họ không thể kiểm tra xem nó có đại diện cho những gì họ muốn hay không và ngần ngại (reluctant) chấp nhận nó như một hợp đồng hệ thống (system contract).





## Triển khai Chức năng Chất lượng (QFD)

---

- Triển khai Chức năng Chất lượng (Quality Function Deployment - QFD) là một kỹ thuật được sử dụng trong phân tích yêu cầu để hiểu nhu cầu của khách hàng.
- Nó bao gồm phân tích các yêu cầu:
  - Bình thường: những đặc điểm **cơ bản, tối thiểu** mà khách hàng mặc định mong đợi
  - Mong đợi: những đặc điểm mà **càng có nhiều, khách hàng càng hài lòng hơn**
  - Thú vị: những đặc điểm **độc đáo, bất ngờ** mà khách hàng không hề mong đợi. Khi chúng được cung cấp, khách hàng sẽ cực kỳ hài lòng và thích thú, tạo nên sự khác biệt lớn cho sản phẩm
- Phân tích giá trị trong QFD giúp xác định mức độ ưu tiên tương đối của các yêu cầu.



## Biểu diễn Yêu cầu

---

- Có thể sử dụng các mô hình và sơ đồ khác nhau như:
  - Sơ đồ Trường hợp sử dụng,
  - Sơ đồ Thực thể - Mối quan hệ,
  - Sơ đồ Chuyển trạng thái,
  - Sơ đồ Hoạt động.
- Việc lựa chọn các mô hình được tạo ra trong quá trình phân tích yêu cầu thường tập trung vào việc biểu diễn chức năng và hành vi của hệ thống.
- Các yêu cầu cũng có thể được biểu diễn bằng các phương pháp hình thức (toán học) hoặc bán hình thức (sơ đồ, văn bản có cấu trúc).



## Thẩm định yêu cầu

---

- Liên quan đến việc kiểm tra tính đúng đắn, tính đầy đủ, tính nhất quán, tính hiện thực và kiểm tra được của yêu cầu. Cụ thể trả lời được các câu hỏi:
  - Còn nhu cầu nào của người dùng chưa kể đến?
  - Có mâu thuẫn gì giữa các yêu cầu?
  - Chức năng, ràng buộc gì chưa kể?
  - Có thực hiện được không?
  - Có thể kiểm tra nó như thế nào?

## Các kỹ thuật thẩm định yêu cầu

---

- Xem xét lại yêu cầu:
  - Phân tích một cách có hệ thống
  - Lấy ý kiến khách hàng
  - Tiến hành thường xuyên
- Làm bản mẫu
  - Sử dụng mô hình khả dụng
  - Kiểm tra tính thực hiện được
- Tạo ca kiểm thử (test case)
- Sử dụng CASE: kiểm tra tính nhất quán



## 4. Mô hình hóa yêu cầu

### Vì sao phải mô hình hóa yêu cầu?



Để hiểu rõ các chức năng của hệ thống

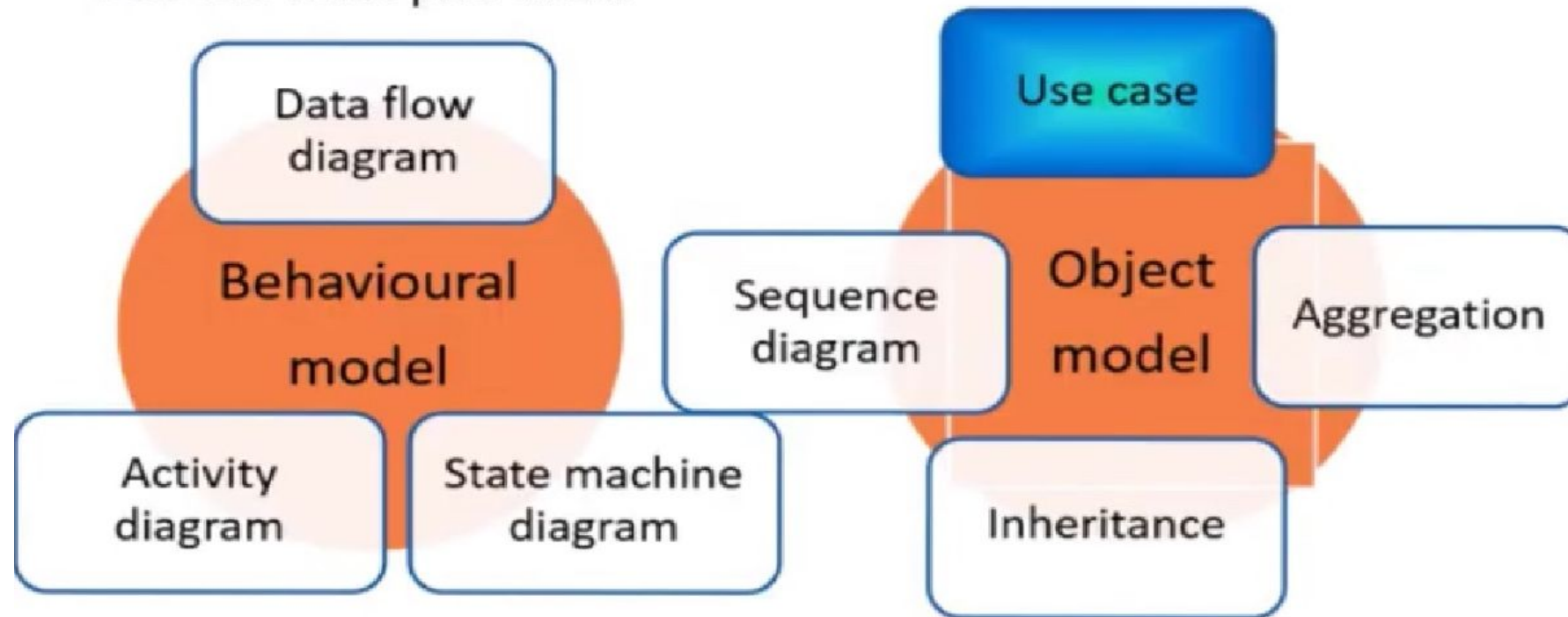
Để trình bày các hệ thống từ những quan điểm khác nhau



## 4. Mô hình hóa yêu cầu

Các mô hình hệ thống trình bày một sự trừu tượng của hệ thống trong lĩnh vực phần mềm, giúp hiểu biết về các yêu cầu chức năng ở dạng sơ đồ khối, và giúp xác định tất cả các yếu tố phần mềm cần thiết và nhiệm vụ

Các mô hình phổ biến:





## 4.1. Mô hình hóa yêu cầu

### Sử dụng UML:

✓ Sơ đồ Use case nhằm thể hiện:

- Cho thấy một tập hợp các chức năng, vai trò người dùng và các mối quan hệ của chúng.
- Chức năng được cung cấp (Use Case)
- Các roles (actors) cho người dùng mà các chức năng này phải thực hiện.

### Actor - Tác nhân

Actor được dùng để chỉ người sử dụng hoặc một đối tượng nào đó bên ngoài tương tác với hệ thống chúng ta đang xem xét

Tác nhân có thể là con người, thành phần, hay hệ thống khác



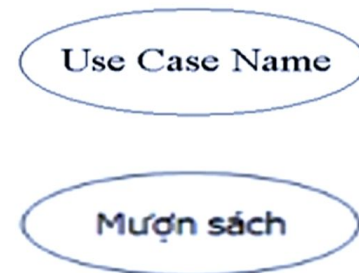
- Use Case Diagram là rất quan trọng, nó giúp chúng ta hiểu yêu cầu, kiến trúc chức năng của hệ thống và chi phối tất cả các bản vẽ còn lại

### Use case

Use Case là chức năng mà các Actor sẽ sử dụng

Mỗi Use case biểu diễn một chức năng chính của hệ thống

Tên use case phải là động từ



### Relationship:

Relationship được sử dụng để kết nối giữa các đối tượng với nhau tạo nên bản vẽ Use Case

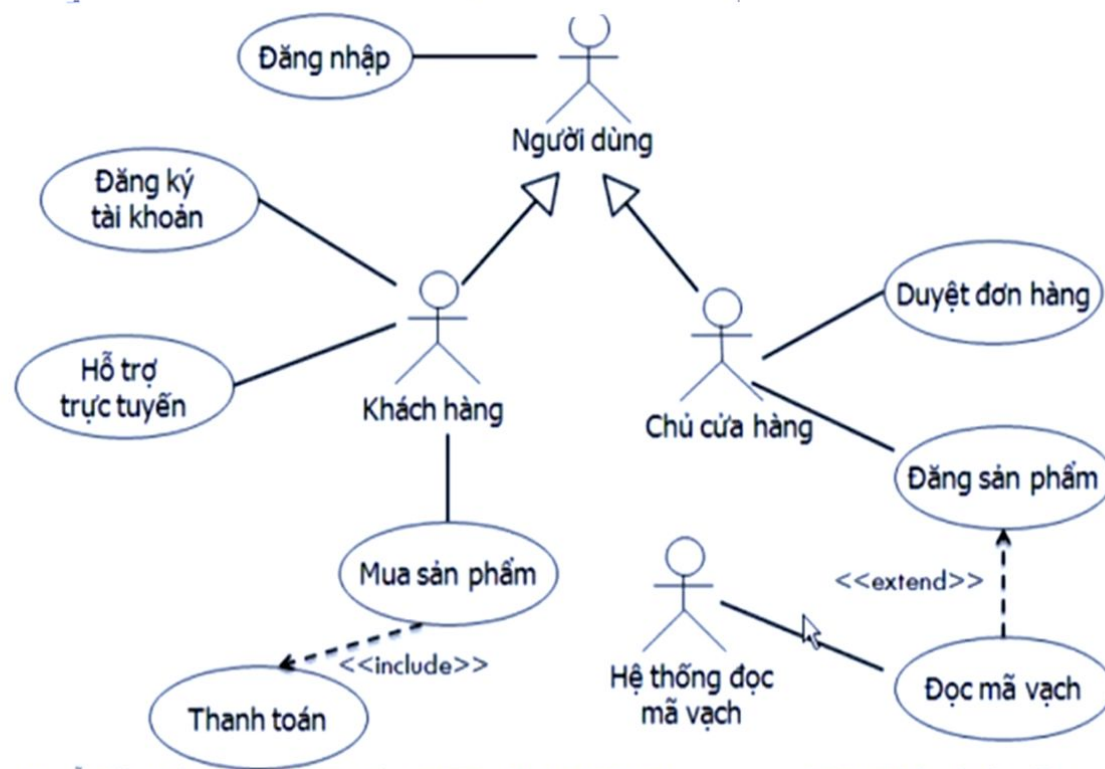
Có các kiểu quan hệ cơ bản sau:

- Association: Liên hệ
- Include: Gộp, bao gồm
- Generalization: Tổng quát hóa





## 4.1. Mô hình hóa yêu cầu - Ví dụ

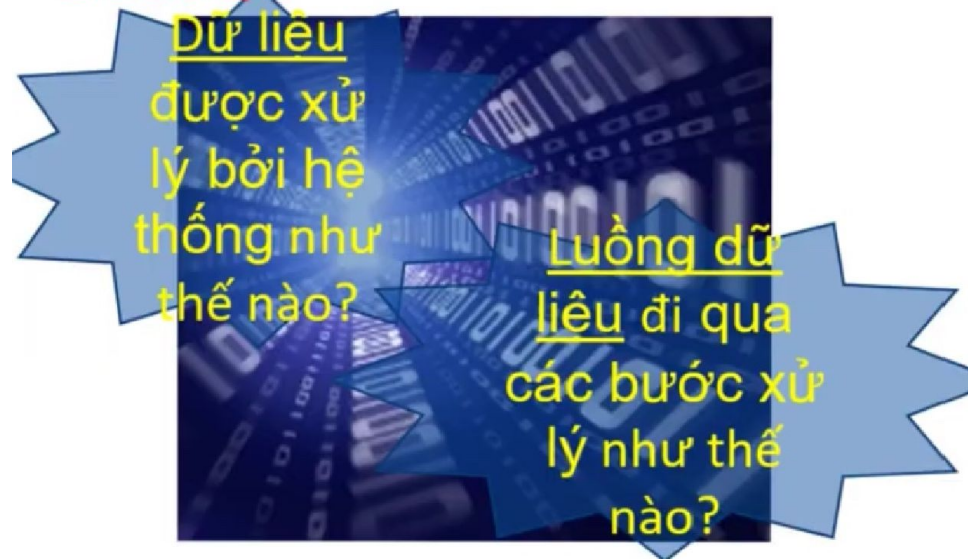






## 4.2 Mô hình hóa yêu cầu

DFD thể hiện:

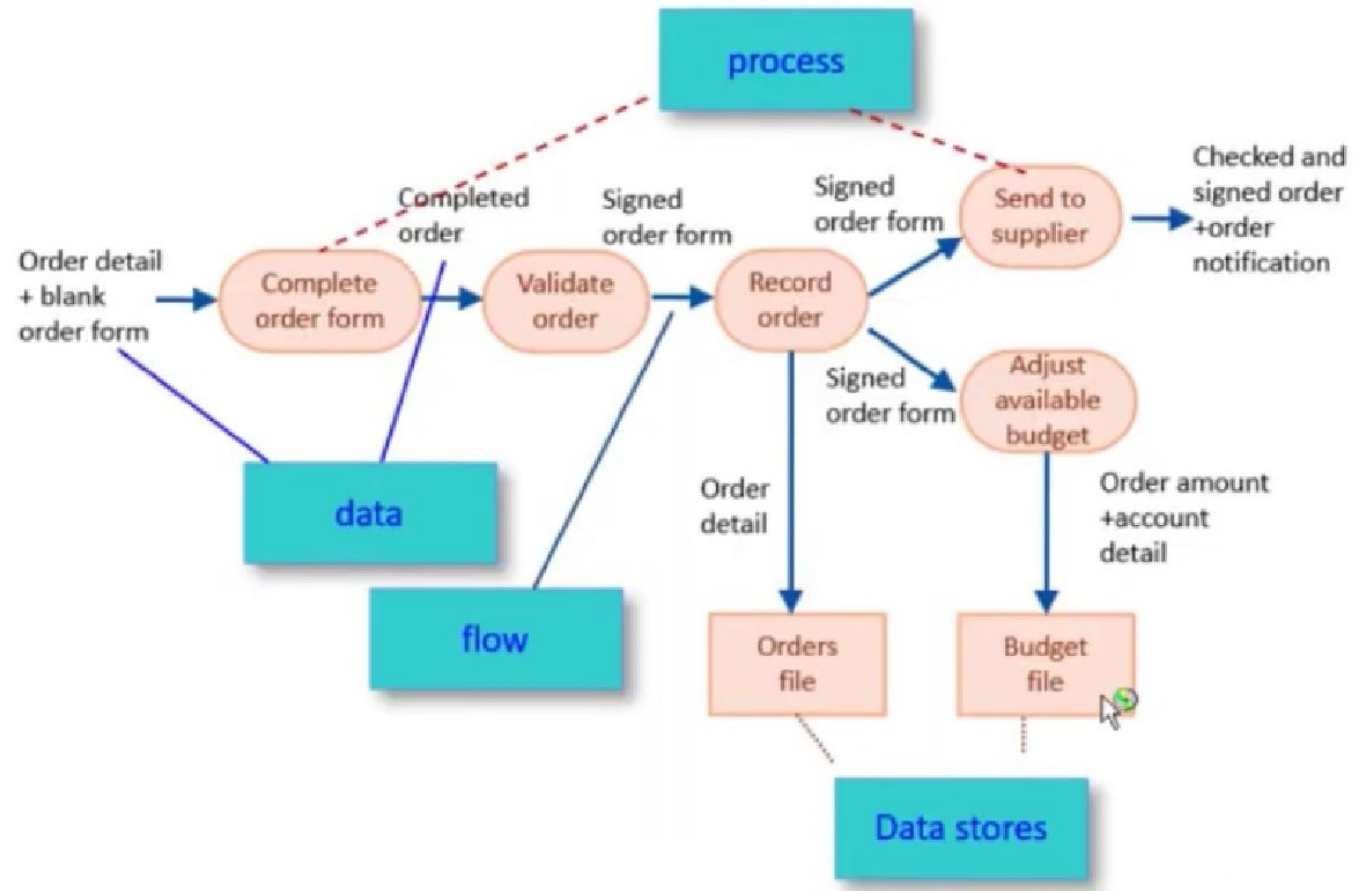


Sơ đồ luồng dữ liệu Data-flow diagram (DFD) là một cách biểu diễn luồng dữ liệu thông qua một quy trình hoặc một hệ thống (thường là một hệ thống thông tin). DFD cũng cung cấp thông tin về đầu ra và đầu vào của từng thực thể và chính trong quy trình đó

Lược đồ luồng dữ liệu DFD- Data Flow Diagram



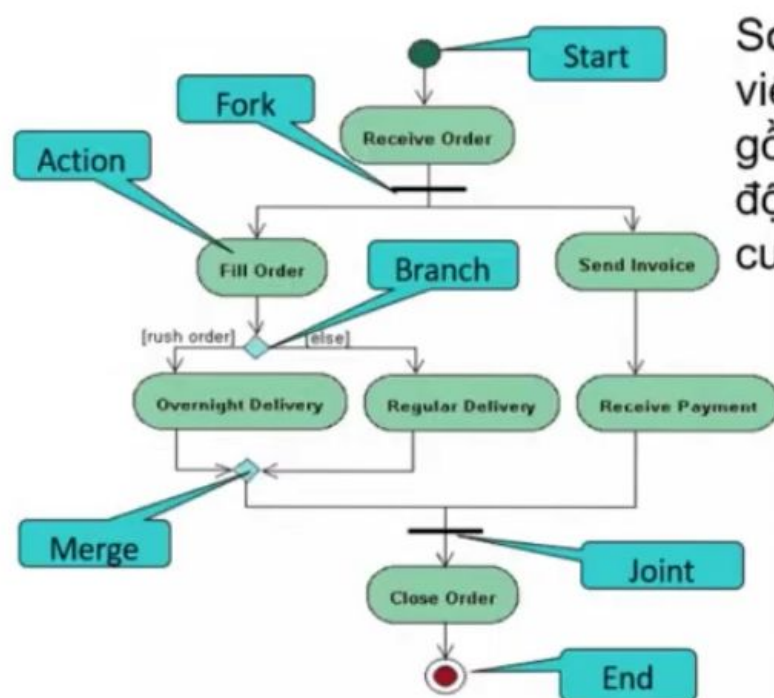
## 4.2 Mô hình hóa yêu cầu



Sơ đồ luồng dữ liệu DFD- Data Flow Diagram



## 4.3 Mô hình hóa yêu cầu



Sơ đồ hoạt động mô tả công việc của một hệ thống bao gồm một chuỗi các hoạt động thực hiện từ đầu đến cuối

Các hoạt động có thể được thực hiện:

- Thứ tự tuần tự
- Song song
- Chuyển tiếp có điều kiện

Sơ đồ hoạt động – Activities Diagram



## Tổng kết

---

Sau khi học nội dung 2, sinh viên được trang bị các kiến thức sau :

- Các khái niệm và tầm quan trọng liên quan tới kỹ nghệ YCPM
- Phân loại YCPM
- Các hoạt động chính trong kỹ nghệ YCPM
- Một số mô hình thường dùng trong quá trình đặc tả YCPM