



Kiến trúc phần mềm hiện đại

# Phân tích và thiết kế hướng đối tượng

# Nội dung chính

---

- Sự phức tạp của phát triển phần mềm
- Mô hình hướng đối tượng
- Ngôn ngữ mô hình thống nhất (UML)

# Tài liệu tham khảo

---

- Grady Booch et al., *Object-oriented analysis and design with applications*

# Sự phức tạp của phần mềm

---

- “Tính phức tạp của phần mềm là bản chất, không phải là ngẫu nhiên”
- Xuất phát từ
  - Sự phức tạp của miền vấn đề (problem domain)
  - Sự khó khăn của việc quản lý tiến trình phát triển
  - Vấn đề trong việc mô tả hành vi của hệ thống rời rạc

# Sự phức tạp của phần mềm

---

- Quản lý sự phức tạp
  - Phân rã: chia phần mềm thành các phần nhỏ
  - Trừu tượng: quan tâm đến các đặc điểm chính
  - Phân cấp: sắp xếp theo các đặc điểm

# Mô hình đối tượng

---

- Quản lý sự phức tạp bằng “đối tượng”
- Bao gồm các nguyên lý của trừu tượng hóa, đóng gói, mô đun, phân cấp, kiểu, đồng thời, và tính bền (persistence)

# Mô hình đối tượng

---

- Bốn nguyên tắc cơ bản
  - Trừu tượng hóa
  - Đóng gói
  - Môđun hóa
  - Phân cấp

# Đóng gói

---

- Đóng gói quy tụ các đặc trưng của một thực thể vào trong một hộp đen trừu tượng, cất giấu sự hiện thực các đặc trưng này sau một giao diện
- Thường được biết đến bởi khái niệm “che giấu thông tin”
- Ý kiến khác?



# Mô đun hóa

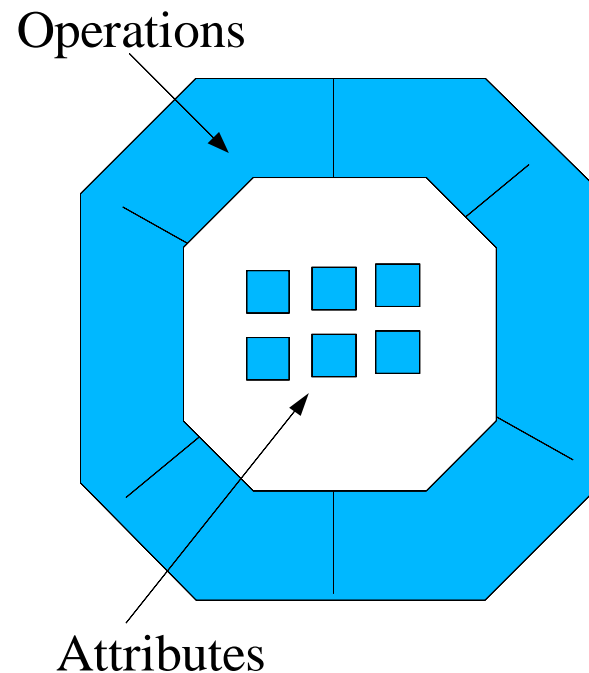
---

- Môđun hóa là sự phân rã về mặt vật lý hoặc logic một hệ thống lớn và phức tạp thành các thành phần quản lý được.
- Phân rã hệ thống thành các hệ thống con. Các hệ thống con này được phát triển độc lập và có thể tương tác được với nhau.

# Đối tượng và lớp

---

Đối tượng là một thực thể hoàn toàn xác định về ranh giới và định danh (id), nó chứa các trạng thái và phương thức

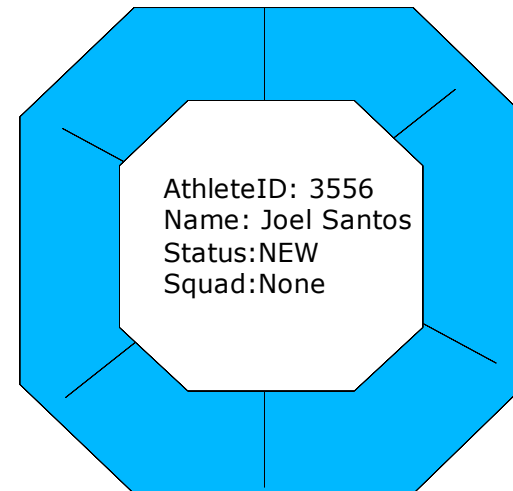


Lớp là tập hợp các đối tượng có chung cấu trúc, hành vi, ngữ nghĩa

# Trạng thái của đối tượng

---

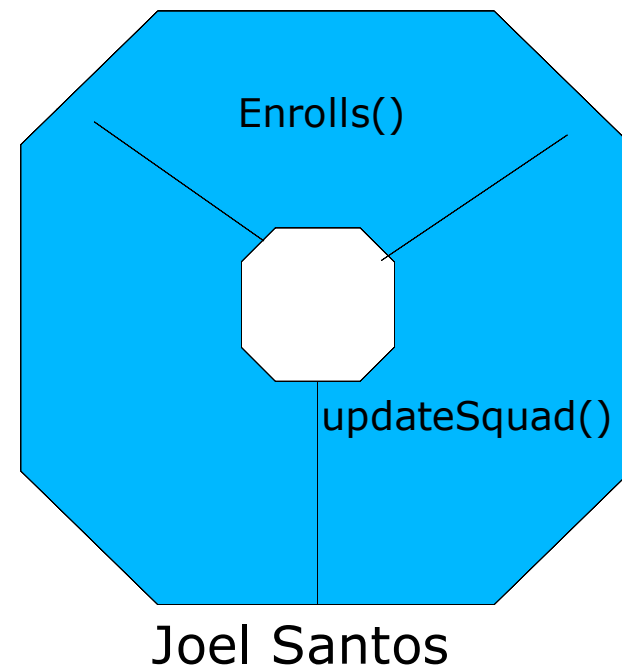
- Bao gồm các thuộc tính và giá trị của thuộc tính của đối tượng đó



# Phương thức của đối tượng

---

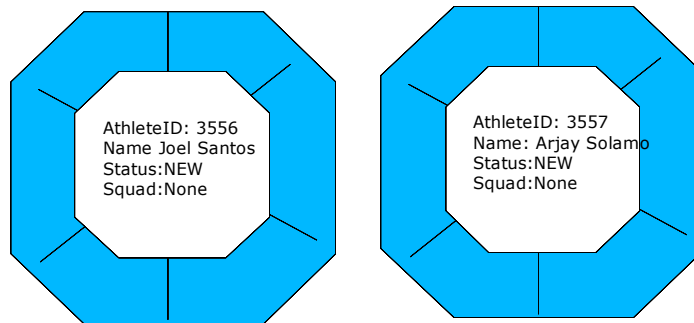
- Chỉ ra làm thế nào đối tượng phản ứng và phản ứng lại
- Biểu diễn bởi các thao tác đối tượng có thể thực hiện



# Định danh của đối tượng

---

- Mặc dù hai đối tượng có thể chia sẻ cùng trạng thái, hai đối tượng là riêng biệt và có định danh khác nhau



# Quan hệ giữa các đối tượng

---

- Liên kết
  - Một đối tượng gọi phương thức của một đối tượng khác
- Aggregation
  - Một đối tượng “chứa” các đối tượng khác

# Quan hệ giữa các lớp

---

- Kết hợp (association)
- Kế thừa
- Aggregation
- Phụ thuộc

# Tính kế thừa

---

- Ưu điểm
  - Tái sử dụng
  - Có thể làm cho việc phân rã rõ ràng
- Nhược điểm
  - Hiện thực của một lớp không còn gói gọn trong lớp đó
- 2 loại: đơn kế thừa và đa kế thừa
- Tính đa hình



# Mô hình hóa

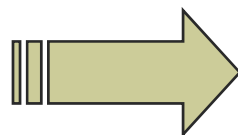
---

- Tồn tại nhiều cách mô hình hóa một hệ thống
  - Mô hình phù hợp sẽ làm cho việc giải bài toán dễ hơn
- Có nhiều mức chính xác của mô hình
  - Mô hình trừu tượng -> làm mịn -> mô hình chi tiết
- Không có mô hình nào là đầy đủ
  - Cần tiếp cận (hiểu) hệ thống thông qua nhiều mô hình khác nhau
- Mô hình tốt phải là mô hình phù hợp với thế giới thực

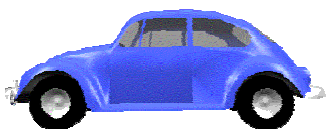
# Ví dụ về mô hình



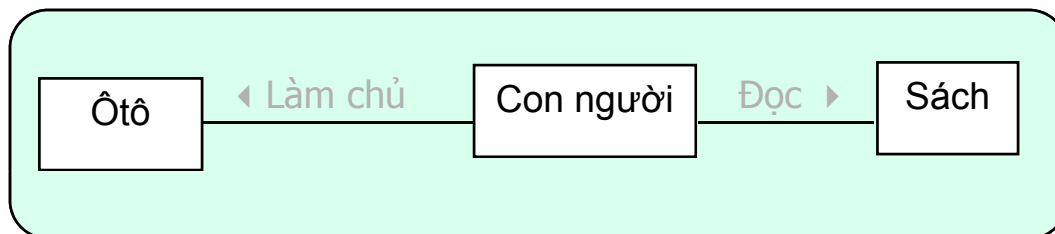
Thế giới thực



Mô hình: Quả địa cầu học sinh

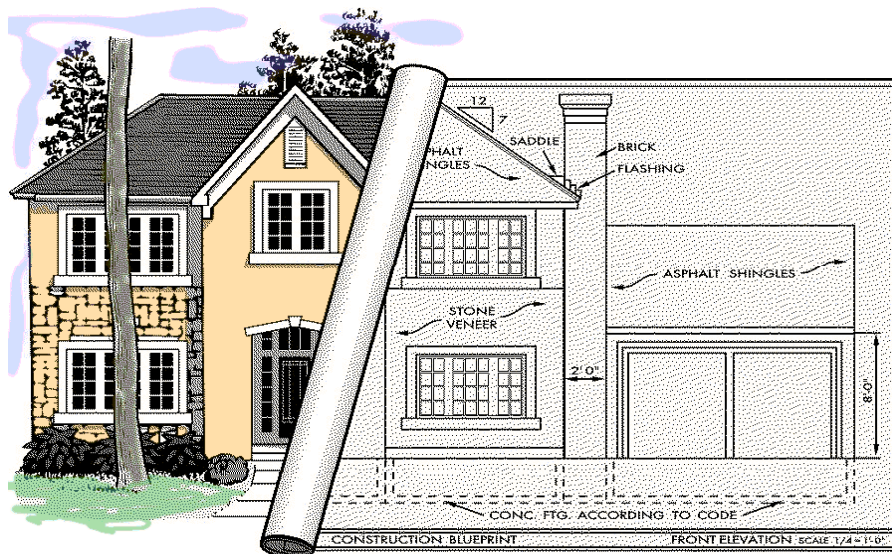


Thế giới thực



Mô hình

# Ví dụ về mô hình

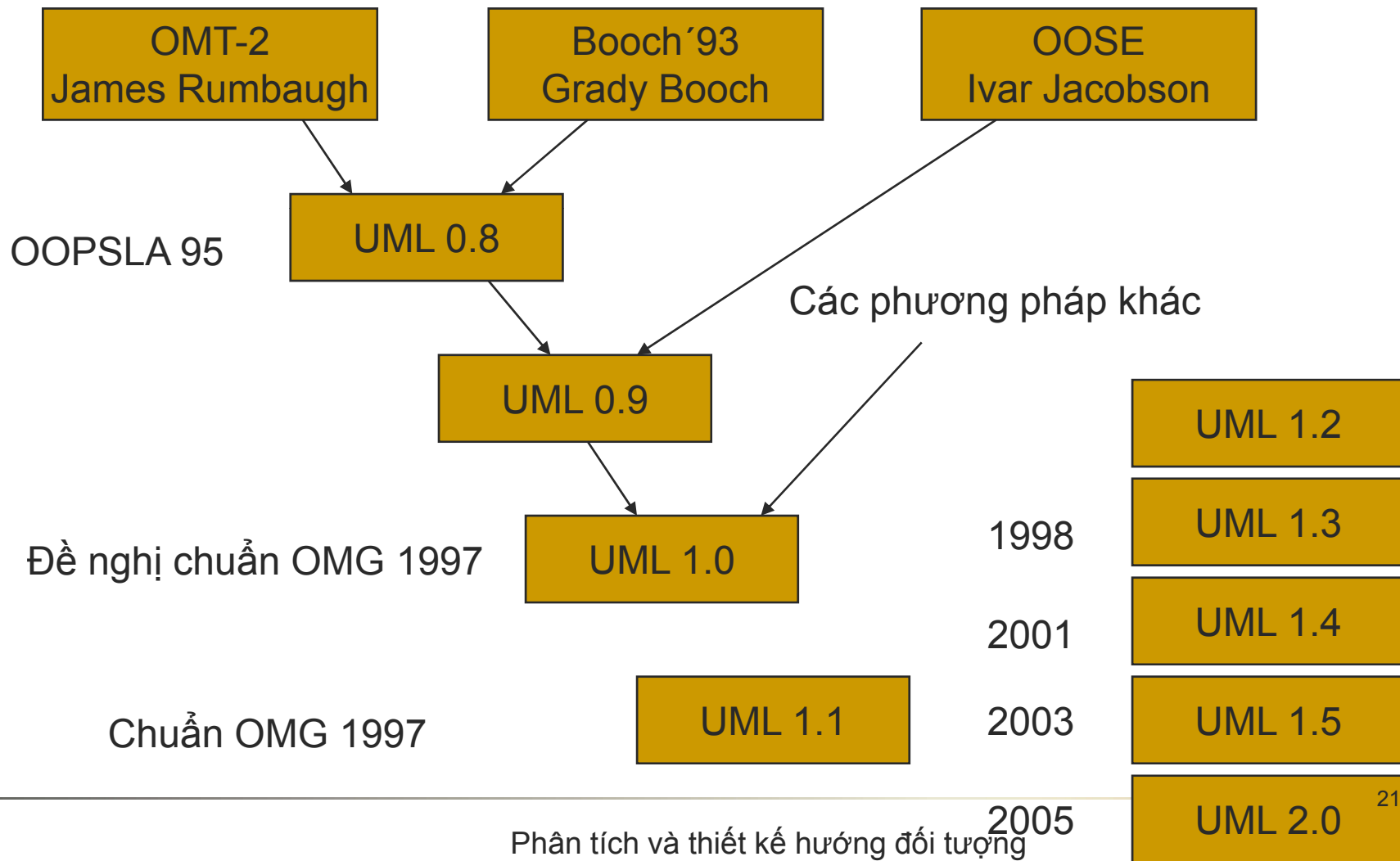


# Unified Modeling Language - UML

---

- Là ngôn ngữ mô hình hóa hướng đối tượng
- Được thừa nhận như một chuẩn mặc định của ngành CNTT
  - Real-time UML, Agent - UML
- Có nhiều tools và phương pháp dựa trên UML
  - Rational rose
  - Rational Unified Process (RUP)

# Unified Modeling Language



# Mục đích của UML

---

- Biểu diễn toàn bộ hệ thống
- Tạo sự liên kết giữa các khái niệm (concepts) về hệ thống và các artifacts thực hiện được
- Vừa trực quan (hướng người dùng) vừa hỗ trợ khả năng tự động hóa (máy)
- Một ngôn ngữ chung:
  - Sử dụng được với nhiều phương pháp khác nhau
  - Đáp ứng tất cả các giai đoạn trong phát triển phần mềm

# UML cho phép

---

- Hiện thị
  - Thông qua các ký pháp đồ họa có ngữ nghĩa xác định
- Đặc tả
  - Một cách chính xác và toàn diện
- Xây dựng
  - Các lớp, các quan hệ có thể được ánh xạ vào chương trình thực
- Làm tài liệu
  - Các biểu đồ, các ghi chú, các ràng buộc,...

# UML cho phép

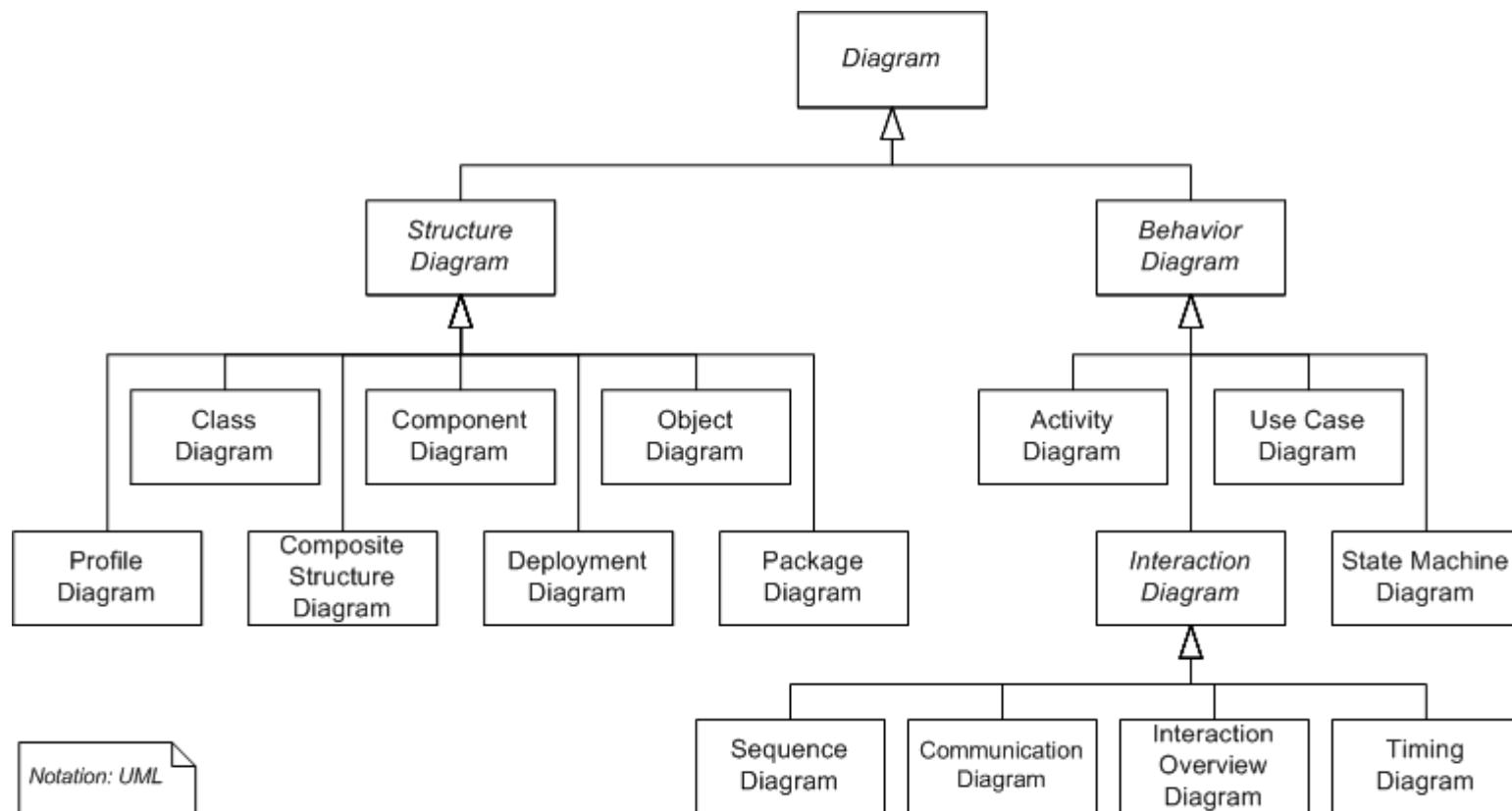
---

- Biểu diễn

- Khía cạnh tĩnh: tập trung vào cấu trúc của hệ thống thông qua các các đối tượng, thuộc tính, phương thức, và các mối quan hệ
- Khía cạnh động: tập trung vào hành vi của hệ thống thông qua sự tương tác giữa các đối tượng và sự thay đổi trạng thái của đối tượng

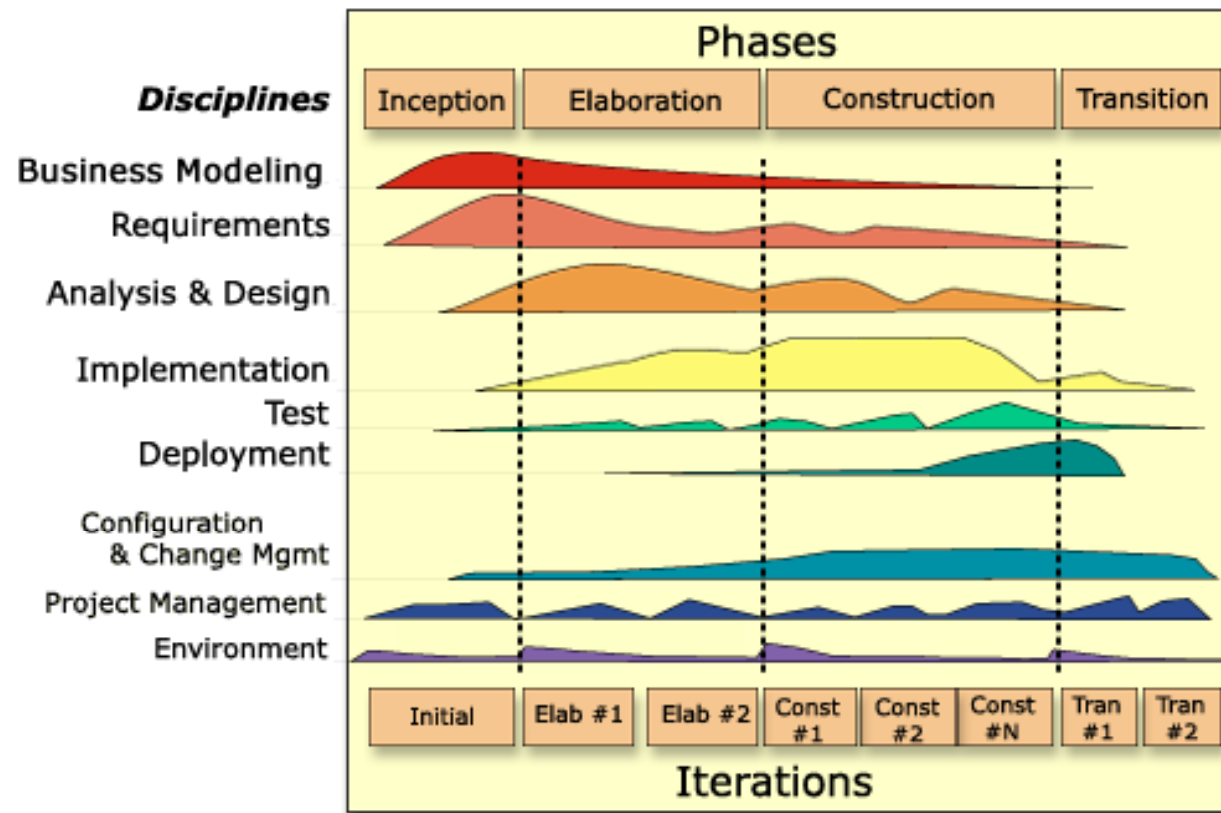


# Phân loại biểu đồ



# Quy trình phần mềm

- Là các hoạt động cần thiết để làm ra một phần mềm



# Phân tích hướng đối tượng

---

- Mục đích chính của phân tích hướng đối tượng là phát triển một chuỗi các mô hình mô tả phần mềm máy tính có thể làm việc thỏa mãn yêu cầu người dùng
- Định hướng của phân tích hướng đối tượng là định nghĩa một tập các lớp, các quan hệ và các hành vi liên quan đến hệ thống
- Bởi vì yêu cầu khách hàng ảnh hưởng đến việc xây dựng các mô hình, giai đoạn này cũng được gọi là kỹ nghệ yêu cầu (***requirements engineering***)

# Thiết kế hướng đối tượng

---

- Chuyển mô hình phân tích tạo ra từ phân tích hướng đối tượng sang mô hình thiết kế.
- Mô tả tổ chức dữ liệu, các thuộc tính, thủ tục chi tiết của các hành vi riêng biệt.

# Tóm tắt

---

- Sự phức tạp của phát triển phần mềm
- Mô hình hướng đối tượng
- Ngôn ngữ mô hình thống nhất (UML)

# Thảo luận

---

- So sánh giữa phát triển hướng đối tượng và phát triển hướng cấu trúc