

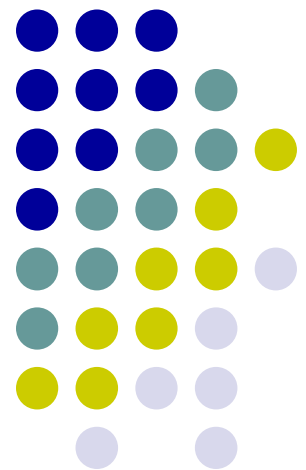
# Phân tích hướng đối tượng UML

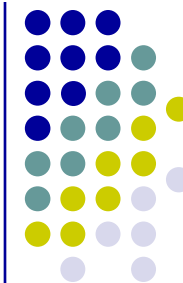
---

**Giáo viên: Đỗ Thị Mai Hương**

**Bộ môn : Các hệ thống thông tin**

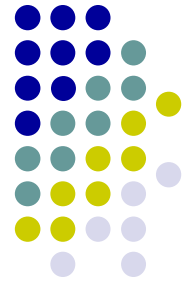
**Khoa : CNTT - Học viện kỹ thuật quân sự**





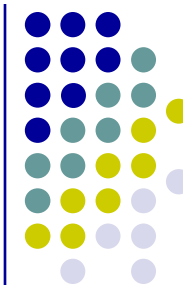
# Bài 1

## Tổng quan



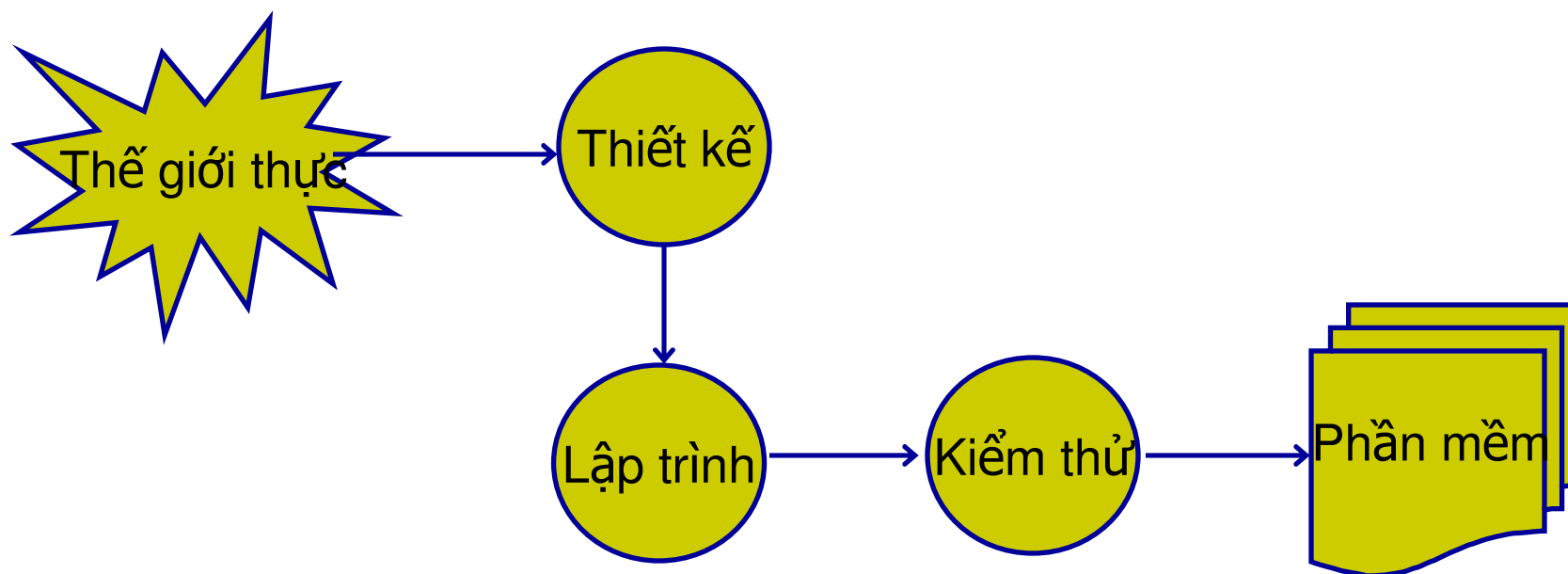
# Nội dung

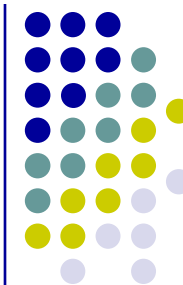
- Phân tích thiết kế là gì?
- Tại sao phải phân tích thiết kế?
- Tầm quan trọng của phân tích thiết kế trong công nghệ phần mềm
- Các cách tiếp cận phân tích và thiết kế hệ thống
- Các khái niệm cơ bản của hướng đối tượng
- Khái quát qui trình phát triển hệ thống thông tin
- Tiến trình RUP



# Phân tích thiết kế là gì?

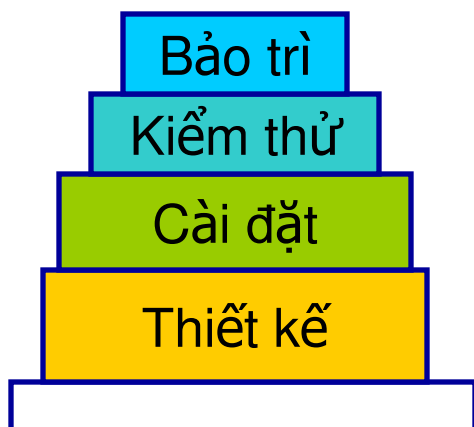
- Phân tích thiết kế **phần mềm**:
  - Quá trình **tìm hiểu** và **mô phỏng lại** hiện tượng, quy trình nghiệp vụ trong thế giới thực từ đó xây dựng hệ thống để giải quyết bài toán đặt ra **trên máy tính**.



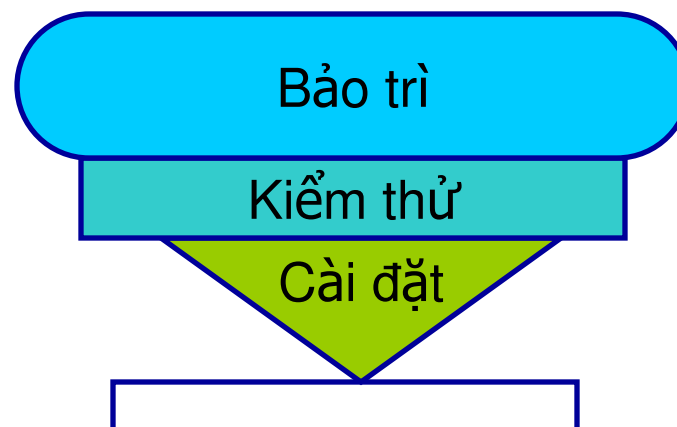


# Tại sao phải phân tích thiết kế?

- Tầm quan trọng của thiết kế

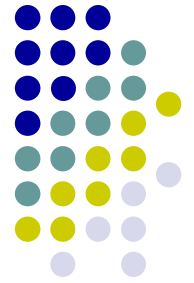


Có thiết kế



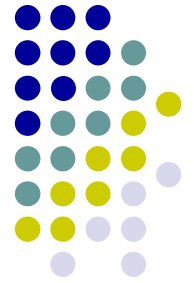
Không thiết kế

# Tầm quan trọng của phân tích thiết kế



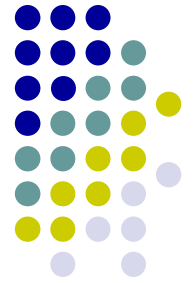
- **Chất lượng thiết kế** là nhân tố chính **quyết định** chất lượng phần mềm
- **Không thiết kế - hoặc thiết kế không tốt** dẫn đến phần mềm chất lượng thấp
  - Không quản lý được những thay đổi yêu cầu
  - Khó kiểm thử
  - Khó bảo trì
  - Không có tính tiến hóa
  - Không tái sử dụng được

# Tầm quan trọng của phân tích thiết kế



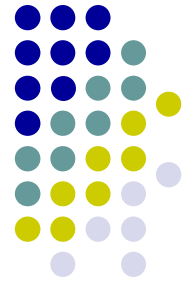
- Thiết kế tốt mang lại phần mềm chất lượng tốt:
  - Dễ dàng thay đổi yêu cầu
  - Dễ kiểm thử
  - Dễ bảo trì
  - Có tính tiến hóa cao
  - Có khả năng tái sử dụng cao

# Các cách tiếp cận phân tích và thiết kế hệ thống



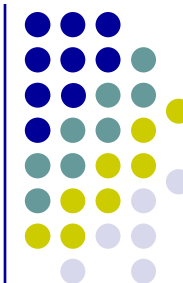
- Có 2 cách:
  - Hướng chức năng/ cấu trúc
  - Hướng đối tượng





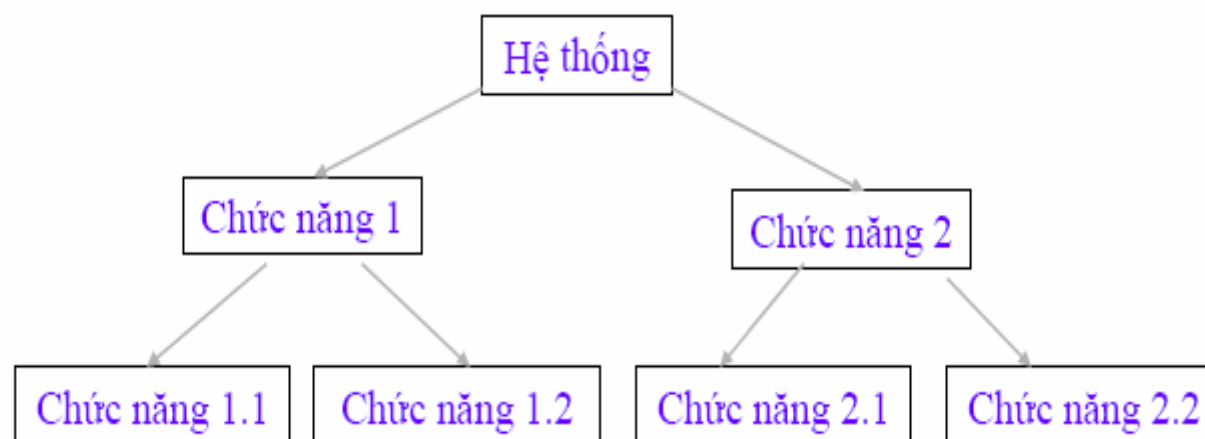
# Hướng chức năng

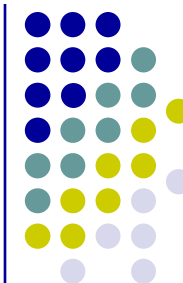
- Đặc trưng của phương pháp hướng cấu trúc là phân chia chương trình chính thành nhiều chương trình con, mỗi chương trình con nhằm đến thực hiện một công việc xác định.
- Cách thức thực hiện của phương pháp hướng cấu trúc là ***phương pháp thiết kế từ trên xuống (top-down)***. ***Phương pháp này tiến hành phân rã bài toán thành các bài*** toán nhỏ hơn, rồi tiếp tục phân rã các bài toán con cho đến khi nhận được các bài toán có thể cài đặt được ngay sử dụng các hàm của ngôn ngữ lập trình hướng cấu trúc.



# Hướng chức năng...

## ○ Phân cấp chức năng

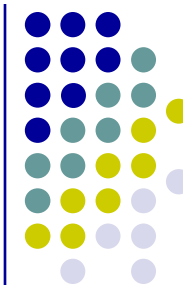




## Hướng chức năng...

- Đây là lối tiếp cận truyền thống của ngành Công nghệ phần mềm, quan tâm chủ yếu tới những thông tin mà hệ thống sẽ giữ gìn.
- Căn cứ vào thông tin người dùng cần => thiết kế dữ liệu để chứa những thông tin đó, cung cấp Forms để nhập thông tin và in báo cáo để trình bày các thông tin.

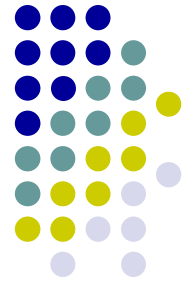
=> Tập trung vào thông tin.



# Hướng đối tượng

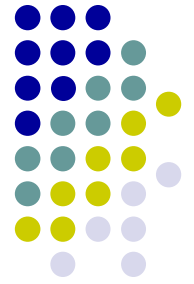
- Lấy đối tượng làm trung tâm
- Đối tượng = chức năng + dữ liệu
- Hệ thống = tập hợp các đối tượng + quan hệ giữa các đối tượng
- Cách tiếp cận hướng đối tượng là một lối tư duy theo cách ánh xạ các thành phần trong bài toán vào các đối tượng ngoài đời thực. Với cách tiếp cận này, một hệ thống được chia tương ứng thành các thành phần nhỏ gọi là các *đối tượng*, mỗi đối tượng bao gồm đầy đủ cả dữ liệu và hành động liên quan đến đối tượng đó.

# Ưu điểm OOA



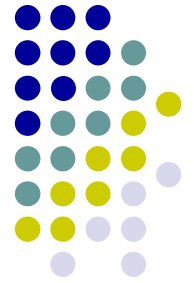
- Ưu điểm
  - Gần gũi với thế giới thực
  - Tái sử dụng dễ dàng
  - Đóng gói, che dấu thông tin làm cho hệ thống tin cậy hơn
  - Thừa kế giảm chi phí, hệ thống có tính mở cao
  - Phù hợp với hệ thống lớn và phức tạp

# Các khái niệm cơ bản của hướng đối tượng

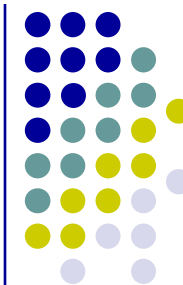


- Đối tượng
- Lớp
- Gói
- Kế thừa

# Đối tượng



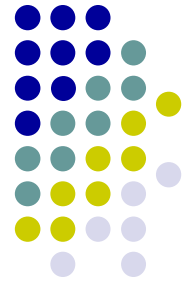
- Đối tượng là khái niệm cho phép mô tả các sự vật/thực thể trong thế giới thực
- Các đối tượng duy trì mối quan hệ giữa chúng
- Ví dụ: Nguyễn Văn A là một đối tượng



# Đối tượng..

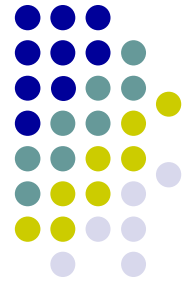
- Các tính chất của đối tượng
  - Đối tượng = trạng thái + hành vi + định danh
    - Trạng thái là các đặc tính của đối tượng tại một thời điểm
    - Hành vi thể hiện các chức năng của đối tượng
    - Định danh thể hiện sự tồn tại duy nhất của đối tượng
- Trạng thái = tập hợp các thuộc tính
  - Mỗi thuộc tính mô tả một đặc tính
  - Tại một thời điểm cụ thể, các thuộc tính mang các giá trị trong miền xác định
  - Ví dụ
    - Một chiếc xe máy: màu xanh, 110 cm<sup>3</sup>, dream, 12000km,...





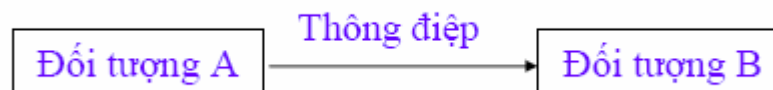
# Đối tượng..

- Hành vi = tập hợp các phương thức
  - Phương thức: là một thao tác hoặc được thực hiện bởi chính nó, hoặc thực hiện khi có yêu cầu từ môi trường (thông điệp từ đối tượng khác)
  - Hành vi phụ thuộc vào trạng thái
  - Ví dụ  
một xe máy có các hành vi: khởi động, chạy, ...

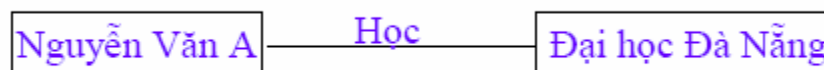


# Giao tiếp giữa các đối tượng

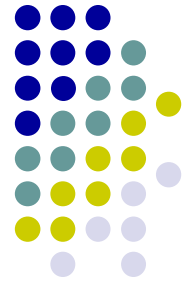
- Các đối tượng giao tiếp với nhau
  - Gửi các thông điệp (message) cho nhau



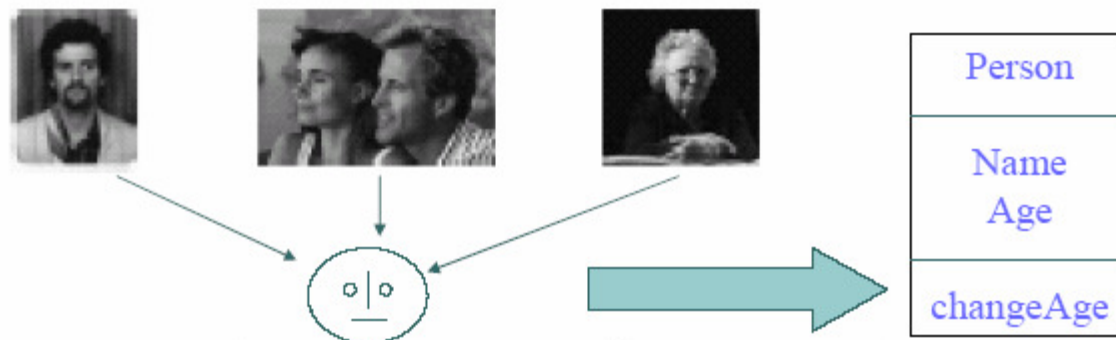
- Các loại thông điệp
  - Hàm dựng (constructor)
  - Hàm hủy (destructor)
  - Hàm chọn lựa (get)
  - Hàm sửa đổi (set)
  - Các hàm chức năng khác
- Giữa các đối tượng có mối liên kết (link) với nhau
- Ví dụ:

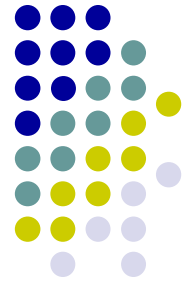


# Lớp



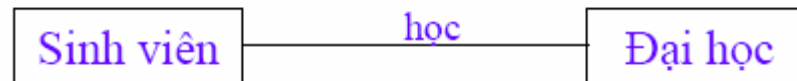
- Lớp là khái niệm dùng để mô tả một tập hợp các đối tượng có cùng một cấu trúc, cùng hành vi và có cùng những mối quan hệ với các đối tượng khác
- Lớp = các thuộc tính + các phương thức
- Lớp là một bước trừu tượng hóa
  - Tìm kiếm các điểm giống, bỏ qua các điểm khác nhau của đối tượng
  - Trừu tượng hóa làm giảm độ phức tạp

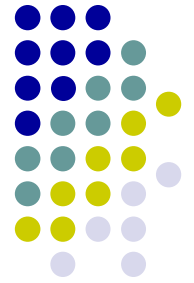




# Lớp..

- Quan hệ giữa các lớp: kết hợp
- Một kết hợp là một tập hợp các mối liên kết giữa các đối tượng

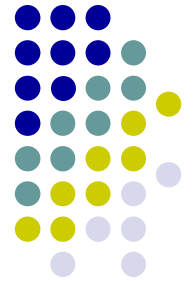




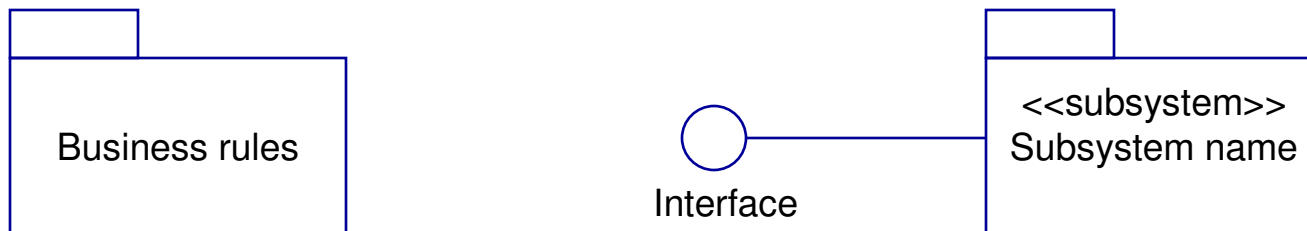
# Lớp & đối tượng

- Đối tượng là thể hiện (instance) của lớp
- Giá trị là thể hiện của thuộc tính
- Liên kết là thể hiện của kết hợp
  - Lớp  $\rightarrow$  đối tượng
  - Thuộc tính  $\rightarrow$  giá trị
  - Kết hợp  $\rightarrow$  liên kết

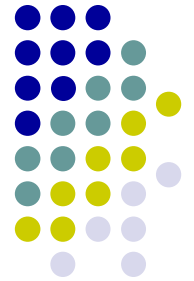
# Gói (package)



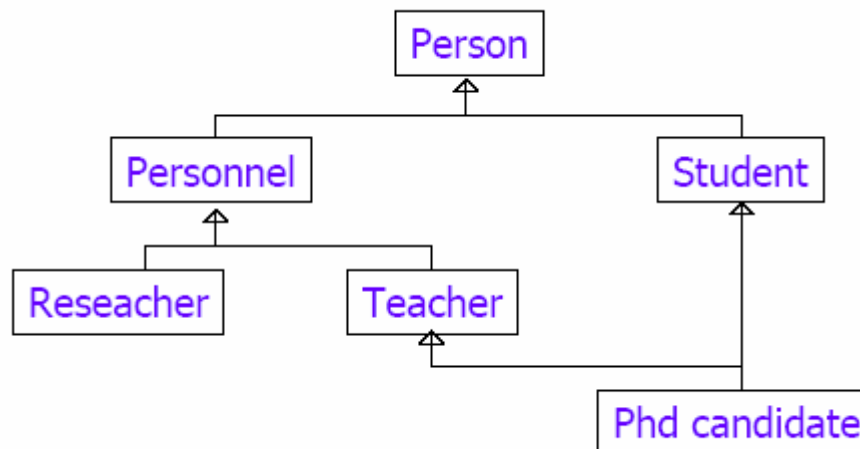
Là một cách tổ chức các thành phần, phần tử trong hệ thống thành các nhóm.  
Nhiều gói có thể được kết hợp với nhau để trở thành một hệ thống con (subsystem).



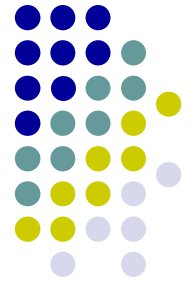
# Kế thừa



Trong phương pháp hướng đối tượng, một lớp có thể có sử dụng lại các thuộc tính và phương thức của một hoặc nhiều lớp khác. Kiểu quan hệ này gọi là quan hệ kế thừa, được xây dựng dựa trên mối quan hệ kế thừa trong bài toán thực tế.



# Các nguyên tắc cơ bản của phương pháp hướng đối tượng



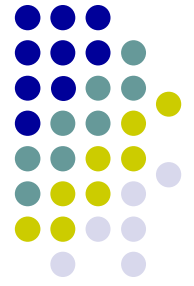
- *Trừu tượng hóa (abstraction)*

- Các thực thể phần mềm được mô hình hóa dưới dạng các đối tượng.
- Các đối tượng được trừu tượng hóa ở mức cao hơn dựa trên thuộc tính và phương thức mô tả đối tượng để tạo thành các lớp.
- Các lớp được trừu tượng hóa ở mức cao hơn nữa để tạo thành một sơ đồ các lớp được kế thừa lẫn nhau. Trong phương pháp hướng đối tượng có thể tồn tại những lớp không có đối tượng tương ứng, gọi là *lớp trừu tượng*.

Như vậy, nguyên tắc cơ bản để xây dựng các khái niệm trong hướng đối tượng là sự trừu tượng hóa theo các mức độ khác nhau.

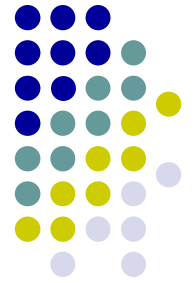


# Các nguyên tắc cơ bản của phương pháp hướng đối tượng..

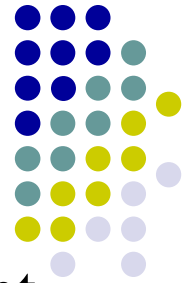


- **Tính bao đóng(encapsulation):** che dấu mọi chi tiết hiện thực của đối tượng không cho bên ngoài thấy và truy xuất => tính độc lập cao giữa các đối tượng
  - Che dấu các thuộc tính dữ liệu: nếu cần cho phép truy xuất 1 thuộc tính dữ liệu, ta tạo 2 phương thức get/set tương ứng để giám sát việc truy xuất và che dấu chi tiết hiện thực bên trong ( thuộc tính private)
  - Che dấu chi tiết hiện thực các phương thức.
  - Che dấu các hàm và sự hiện thực của chúng.

# Các nguyên tắc cơ bản của phương pháp hướng đối tượng..

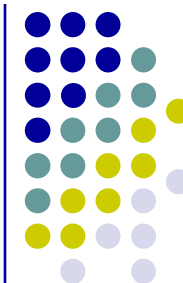


- ***Tính modul hóa (modularity)***: các bài toán sẽ được phân chia thành những vấn đề nhỏ hơn, đơn giản và quản lý được.
- ***Tính phân cấp (hierarchy)***: cấu trúc chung của một hệ thống hướng đối tượng là dạng phân cấp theo các mức độ trừu tượng từ cao đến thấp.



# Quy trình phát triển phần mềm

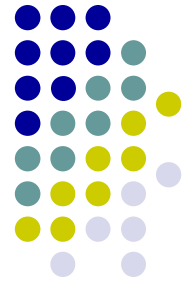
- Chu trình phát triển phần mềm (Software Development Life Cycle - SDLC): là một chuỗi các hoạt động của nhà phân tích (Analyst), nhà thiết kế (Designer), người phát triển (Developer) và người dùng (User) để phát triển và thực hiện một hệ thống thông tin. Những hoạt động này được thực hiện trong nhiều giai đoạn khác nhau.
- **Nhà phân tích (Analyst):** nghiên cứu yêu cầu của khách hàng/người dùng để định nghĩa phạm vi bài toán, nhận dạng nhu cầu của tổ chức, xác định xem nhân lực, phương pháp và công nghệ để cải thiện một cách tốt nhất công tác của tổ chức này



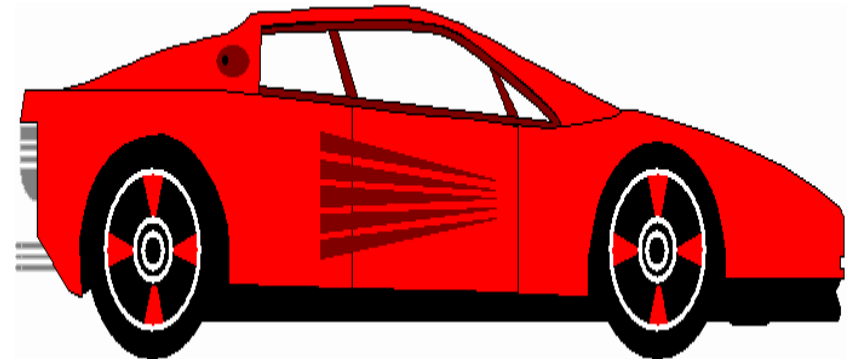
# Quy trình phát triển phần mềm

- **Nhà thiết kế (Designer):** thiết kế database, screens, forms và reports – quyết định các yêu cầu về phần cứng và phần mềm cho hệ thống cần được phát triển.
- **Chuyên gia lĩnh vực (Domain Experts):** là những người hiểu thực chất vấn đề cùng sự phức tạp của hệ thống cần tin học hoá. Họ không nhất thiết phải là nhà lập trình, nhưng họ có thể giúp nhà lập trình hiểu yêu cầu đặt ra đối với hệ thống cần phát triển.
- **Lập trình viên (Programmer):** là những người dựa trên các phân tích và thiết kế để viết chương trình (coding) cho hệ thống bằng ngôn ngữ lập trình đã được thống nhất.
- **Người dùng (User):** là đối tượng phục vụ của hệ thống cần được phát triển.

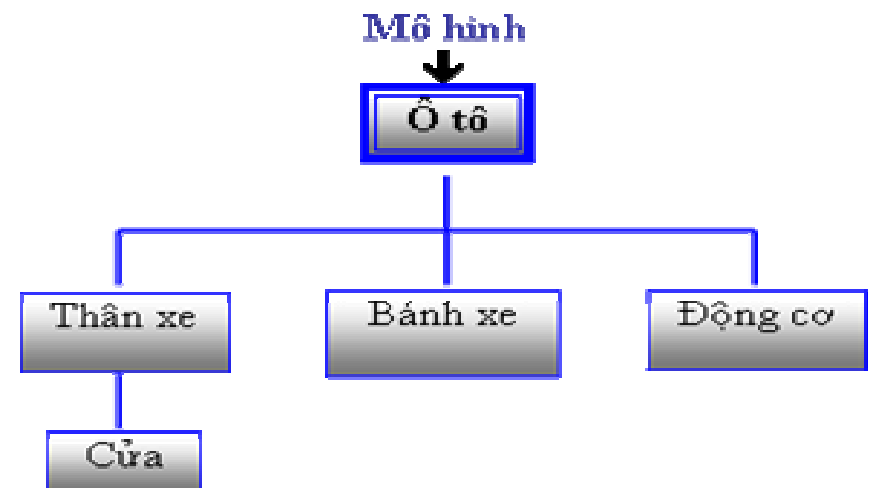
# Quy trình phát triển phần mềm

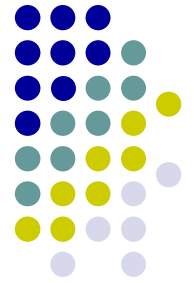


Người bình thường khi nhìn một chiếc xe ô tô thường sẽ có một bức tranh từ bên ngoài như sau:



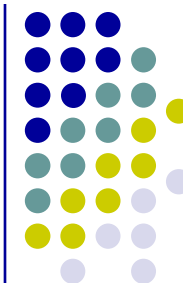
Chuyên gia lĩnh vực sẽ giúp nhà phân tích "trình bày lại" vấn đề như sau:





# Quy trình phát triển phần mềm..

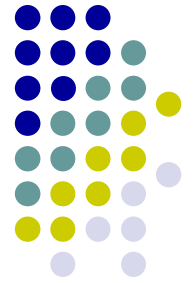
- **Các giai đoạn của Quy trình phát triển phần mềm**
- Quy trình phát triển của một phần mềm có thể được chia thành các giai đoạn như sau:
  - Nghiên cứu sơ bộ (Preliminary Investigation hay còn gọi là Feasibility Study)
  - Phân tích yêu cầu (Analysis)
  - Thiết kế hệ thống (Design of the System)
  - Xây dựng phần mềm (Software Construction)
  - Thử nghiệm hệ thống (System Testing)
  - Thực hiện, triển khai (System Implementation)
  - Bảo trì, nâng cấp (System Maintenance)



# Quy trình phát triển phần mềm



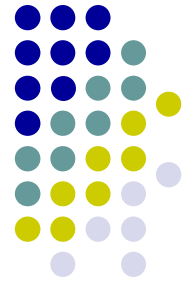
## Nghiên cứu sơ bộ



Một giai đoạn nghiên cứu sơ bộ thích đáng sẽ lập nên tập hợp các yêu cầu (dù ở mức độ khái quát cao) đối với một hệ thống khả thi và được mong muốn, kể cả về phương diện kỹ thuật lẫn xã hội.

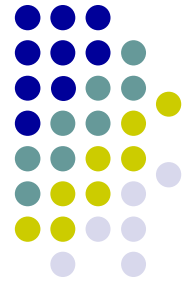
Kết quả của giai đoạn nghiên cứu sơ bộ là Báo cáo kết quả nghiên cứu tính khả thi. Khi hệ thống tương lai được chấp nhận dựa trên bản báo cáo này cũng là lúc giai đoạn Phân tích bắt đầu.





# Phân tích yêu cầu

- Quá trình phân tích nhìn chung là hệ quả của việc trả lời câu hỏi "Hệ thống cần phải làm gì?".
  - Những mục tiêu cụ thể của giai đoạn phân tích là:
    - Xác định hệ thống cần phải làm gì.
    - Nghiên cứu thấu đáo tất cả các chức năng cần cung cấp và những yếu tố liên quan
    - Xây dựng một mô hình nêu bật bản chất vấn đề từ một hướng nhìn có thực (trong đời sống thực).
    - Trao định nghĩa vấn đề cho chuyên gia lĩnh vực để nhận sự đánh giá, góp ý.
  - Kết quả của giai đoạn phân tích là bản Đặc tả yêu cầu (Requirements Specifications).



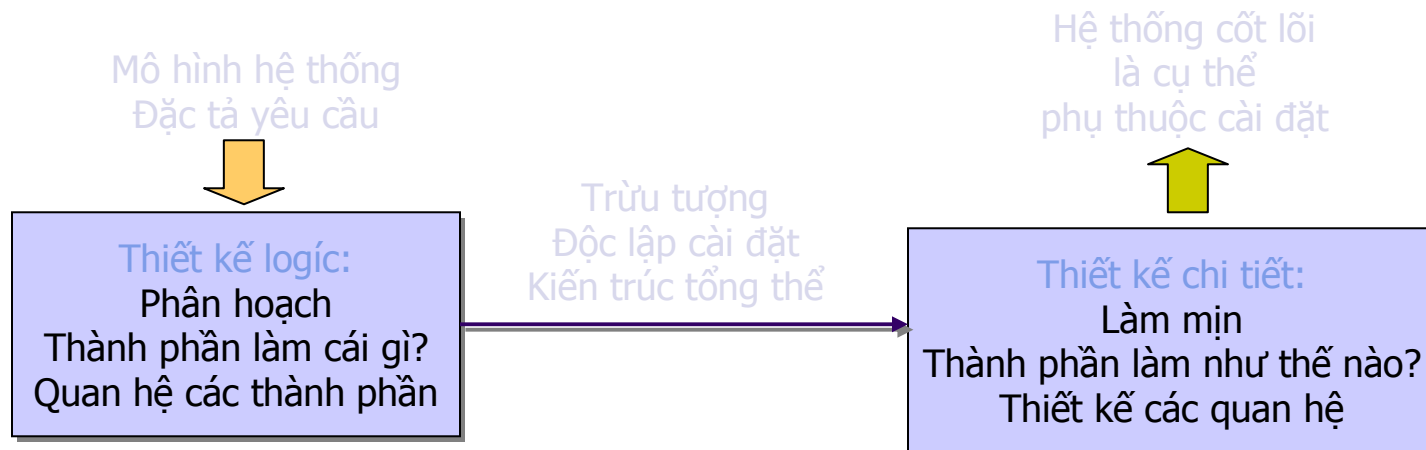
# Phân tích yêu cầu

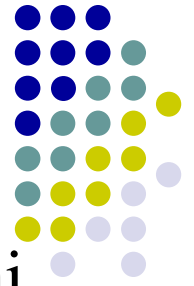
- Đặc tả yêu cầu
  - là thông báo chính thức cái đòi hỏi hệ thống phải được phát triển
  - Nó không phải là tài liệu thiết kế
- Mô tả đặc tả yêu cầu
  - Ngôn ngữ đặc tả
  - Ký pháp đồ họa

Pha thu thập và phân tích yêu cầu rất quan trọng.  
Nếu không phát hiện ra lỗi tại pha này thì rất khó  
và tốn kém để phát hiện ra nó ở pha tiếp theo

# Thiết kế hệ thống

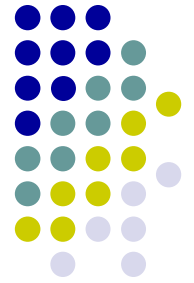
- Sau giai đoạn phân tích, khi các yêu cầu cụ thể đối với hệ thống đã được xác định, giai đoạn tiếp theo là thiết kế cho các yêu cầu mới. Công tác thiết kế xoay quanh câu hỏi chính: Hệ thống làm cách nào để thỏa mãn các yêu cầu đã được nêu trong Đặc tả yêu cầu?
- Các hoạt động của thiết kế





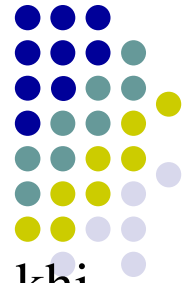
# Thiết kế hệ thống

- Một số các công việc thường được thực hiện trong giai đoạn thiết kế:
  - Nhận biết form nhập liệu tùy theo các thành phần dữ liệu cần nhập.
  - Nhận biết reports và những output mà hệ thống mới phải sản sinh
  - Thiết kế forms (vẽ trên giấy hay máy tính, sử dụng công cụ thiết kế)
  - Nhận biết các thành phần dữ liệu và bảng để tạo database
  - Ước tính các thủ tục giải thích quá trình xử lý từ input đến output.



# Lập trình và kiểm thử

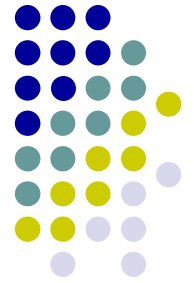
- Mỗi thành phần trong pha thiết kế được hiện thực thành một modul chương trình
- Kiểm chứng hay kiểm thử mỗi modul chương trình theo đặc tả có từ pha thiết kế
- Tổ hợp các modul chương trình thành hệ thống
- Kiểm thử hệ thống chương trình để đảm bảo đáp ứng đầy đủ yêu cầu
- Khi người phát triển thỏa mãn với sản phẩm
  - khách hàng kiểm thử hệ thống
- Pha này kết thúc khi khách hàng chấp nhận sản phẩm



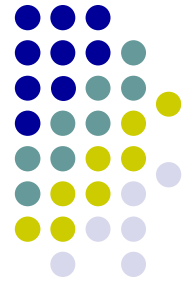
# Bảo trì hệ thống

- Pha này bắt đầu khi hệ thống được cài đặt sử dụng thực tế, sau khi đã cấp phát sản phẩm cho khách hàng
- Bảo trì bao gồm mọi thay đổi sản phẩm để khách hàng đồng ý rằng họ đã thỏa mãn với sản phẩm.
- Bảo trì bao gồm
  - sửa phần mềm
    - loại bỏ các lỗi mà không phát hiện trong các pha trước đó
  - nâng cấp phần mềm
    - Hiệu năng: Bổ sung chức năng, tăng tốc độ thực hiện chương trình
    - Thích nghi: Các thay đổi cho phù hợp với môi trường phần mềm hoạt động thay đổi, thí dụ yêu cầu mới của chính phủ

# Một số mô hình phát triển hệ thống

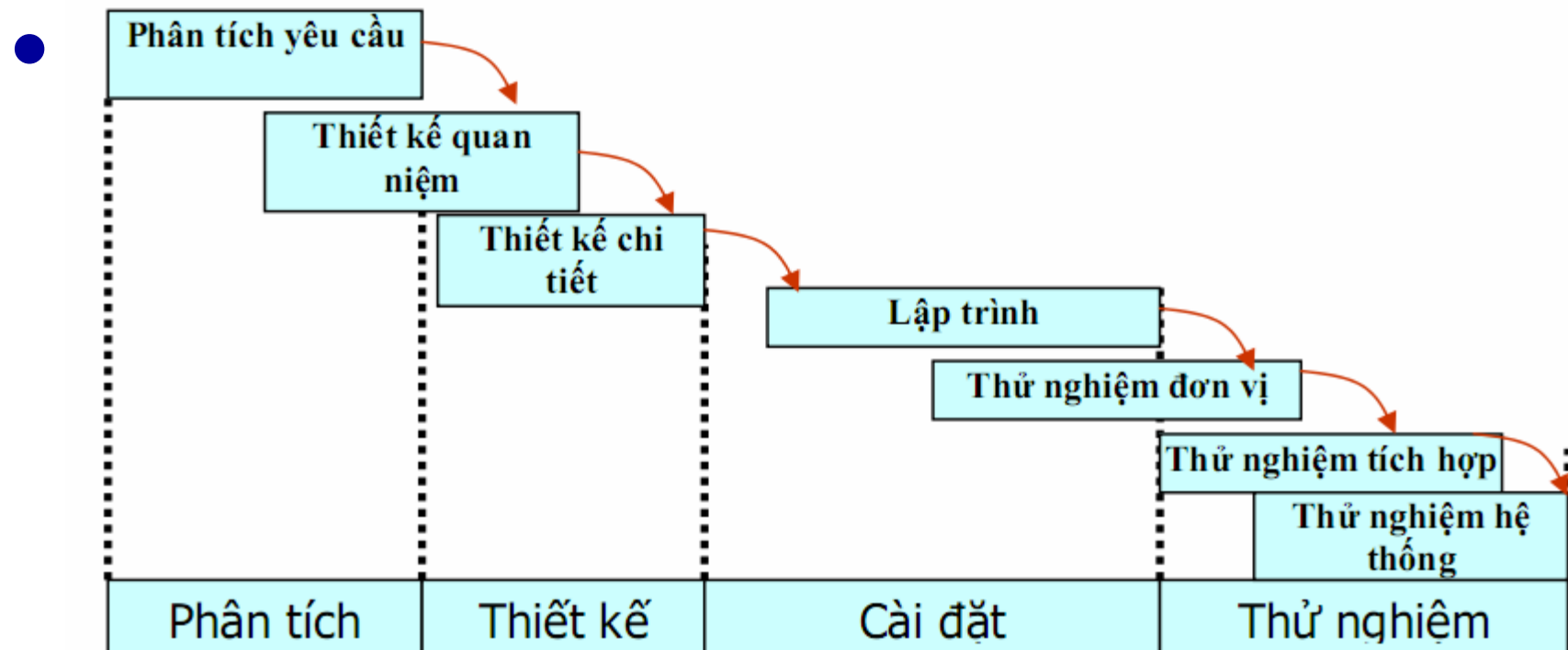


- Mô hình thác nước
- Mô hình tăng trưởng
- Tiến trình RUP

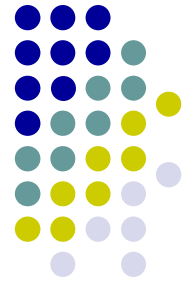


# Mô hình thác nước

- Các hoạt động phát triển phần mềm có thể biểu diễn bằng mô hình thác nước



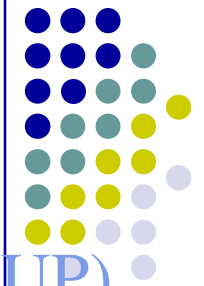




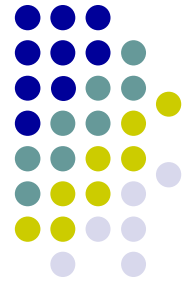
# Mô hình tăng trưởng



# Tiến trình lặp và tăng dần



- Tiến trình thống nhất (Rational Unified Process - RUP)
  - Là Software Engineering process
  - Là sản phẩm tiến trình (process product) do Rational Software phát triển và bảo trì
  - RUP nâng cao team productivity
  - Các hoạt động RUP tạo lập và quản lý models
  - Là hướng dẫn cách sử dụng hiệu quả UML



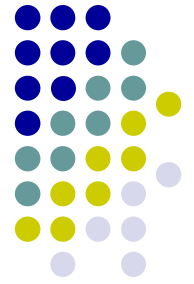
# Các nguyên tắc cơ bản của RUP

- Lặp và tăng trưởng

Dự án được cắt thành những vòng lặp hoặc giai đoạn ngắn. Cuối mỗi vòng lặp thì một phần thi hành được của hệ thống được sản sinh theo cách tăng trưởng (thêm vào) dần dần.

- Tập trung vào kiến trúc

Toàn bộ hệ thống phức tạp phải được chia thành từng phần (các modun) để có thể dễ dàng triển khai và bảo trì, tạo nên một kiến trúc. (Theo 5 góc nhìn)



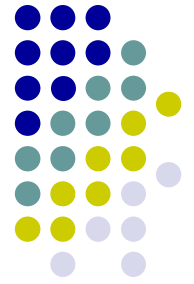
# Các nguyên tắc cơ bản của RUP..

- Dẫn dắt các ca sử dụng

RUP nhấn mạnh sự đáp ứng nhu cầu người dùng, thể hiện bởi các ca sử dụng. Các ca sử dụng ảnh hưởng và dẫn đường cho mọi giai đoạn phát triển của hệ thống.

- Không chế bởi các nguy cơ

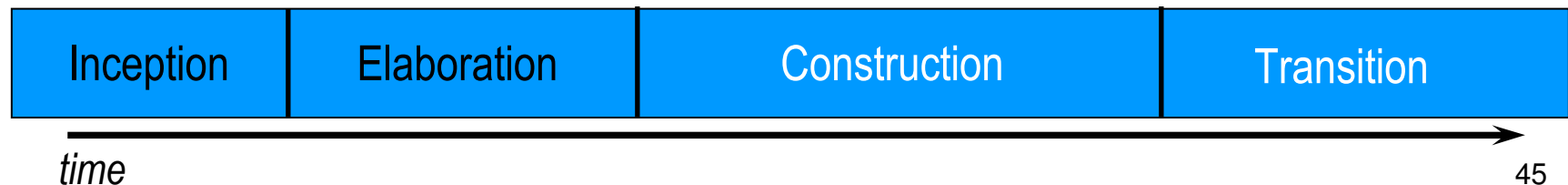
Các nguy cơ chính đối với dự án phải phát hiện sớm và loại bỏ càng sớm càng tốt. Yêu cầu này cũng là căn cứ để xác định thứ tự trước sau của các vòng lặp.



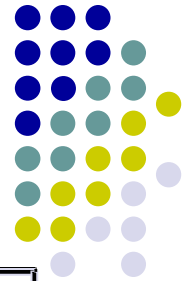
# Các pha và công đoạn của tiến trình RUP

Có 4 pha

- Khởi đầu (inception)  
Cho một cái nhìn tổng quát về hệ thống sẽ xây dựng và về dự án sẽ triển khai.
- Phác thảo (elaboration)  
Bao gồm sự phân tích chi tiết hơn về hệ thống, cả về chức năng lẫn cấu trúc tĩnh. Đồng thời một kiến trúc hệ thống cũng được đề xuất. Kiến trúc này có thể dựng thành nguyên mẫu, trên đó thể hiện nhiều ý đồ đối với hệ thống
- Xây dựng (construction)  
Tập trung vào việc thiết kế và thực thi hệ thống.
- Chuyển giao (transition)  
Nhằm chuyển hệ thống đã xây dựng cho người dùng cuối



# Các lặp và luồng công việc



## Core Workflows

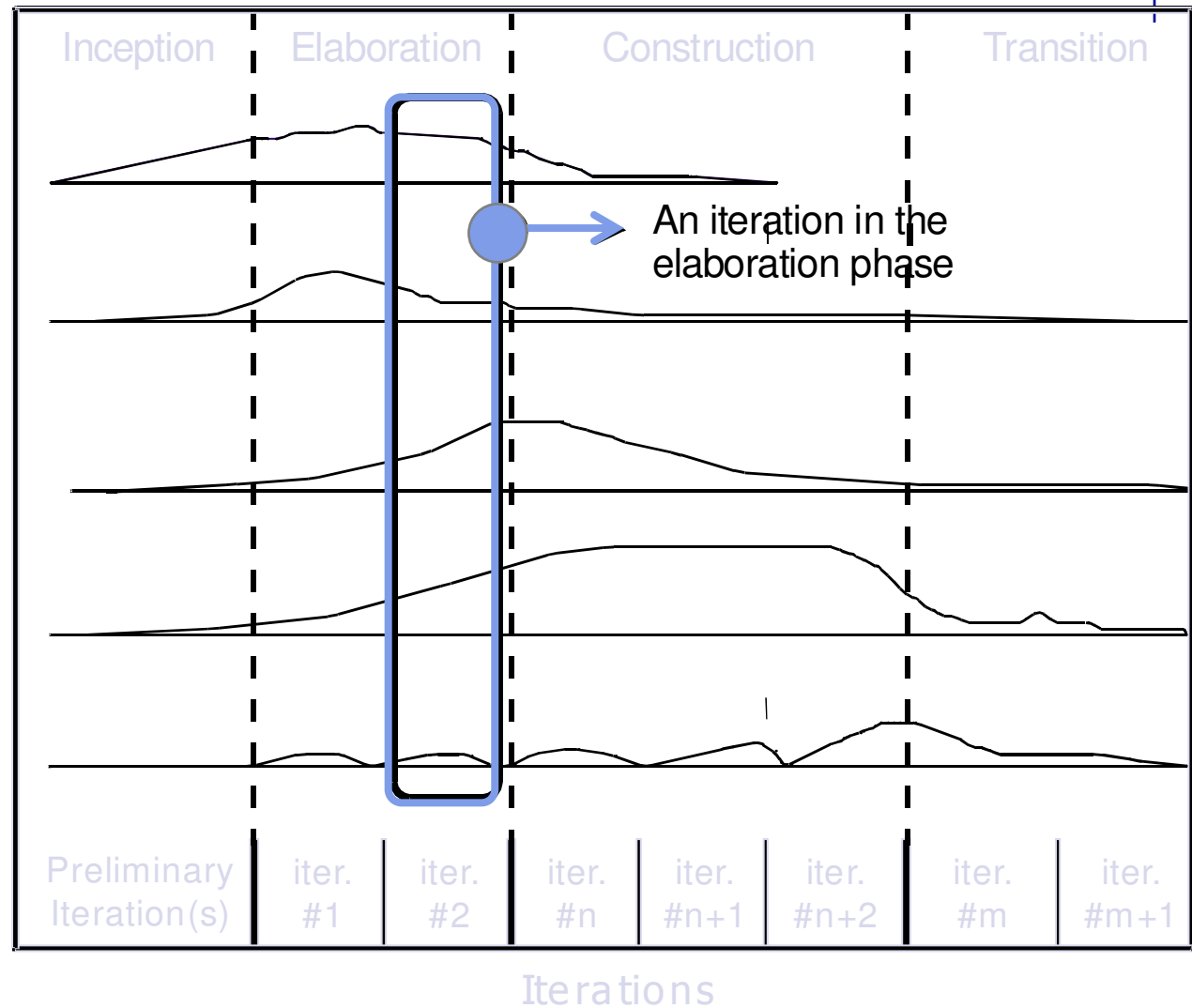
Requirements

Analysis

Design

Implementation

Test



# Tiến trình 10 bước

