



TỔNG QUAN VỀ PHÂN TÍCH & THIẾT KẾ  
HTTT

# NỘI DUNG

- ❖ Giới thiệu về hệ thống thông tin
  - Khái niệm chung về hệ thống thông tin
  - Một số hệ thống thông tin thường gặp
  - Vai trò của của phân tích & thiết kế hệ thống
- ❖ Vấn đề cơ bản về phát triển hệ thống thông tin
  - Các cách tiếp cận phát triển HTTT
  - Vòng đời phát triển HTTT
  - Các phương pháp phát triển HTTT



- ❖ **Giới thiệu về Hệ thống thông tin**
  - Khái niệm chung về hệ thống thông tin
  - Phân loại hệ thống thông tin thường gặp
  - Vai trò của của phân tích & thiết kế hệ thống

# Khái niệm về HTTT

- ❖ Là một tập hợp các thành phần được tổ chức để **thu thập, xử lý, lưu trữ, phân phối** và **biểu diễn thông tin**, trợ giúp việc ra quyết định và kiểm soát hoạt động trong mộ



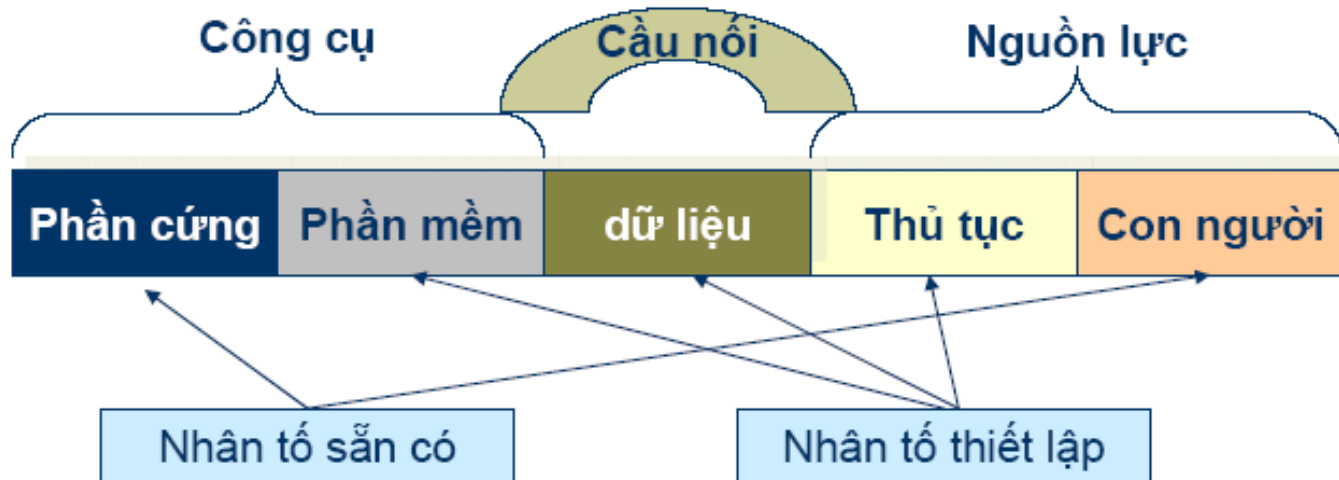
- Dữ liệu?
- Thông tin?
- Xử lý thông tin?
- Biểu diễn thông tin?

# Khái niệm về HTTT....

- ❖ Dữ liệu: mô tả sự kiện, sự vật, con người thế giới thực
- ❖ Thông tin: hiểu biết từ dữ liệu
- ❖ Xử lý thông tin:
  - Hoạt động tác động lên dữ liệu làm nó biến đổi
  - Số và phi số
  - Xử lý theo lô, tương tác, thời gian thực, phân tán...
- ❖ Biểu diễn thông tin: biểu diễn dữ liệu ở dạng mà con người nhận biết được (màn

# Hệ thống thông tin quản lý

- ❖ Là sự phát triển và sử dụng HTTT có hiệu quả trong một tổ chức
- ❖ Các yếu tố cấu thành một HTTT quản lý



# Phân loại HTTT

- ❖ Phân theo lĩnh vực nghiệp vụ
  - Hệ thống tự động văn phòng
  - Hệ thống truyền thông
  - Hệ thống thông tin thực hiện
  - Hệ thống xử lý giao dịch
  - Hệ thống thông tin quản lý
  - Hệ trợ giúp quyết
  - Hệ chuyên gia
  - Hệ trợ giúp hệ điều hành
  - Hệ trợ giúp làm việc nhóm

# Phân loại HTTT...

- ❖ Phân loại theo quy mô kĩ thuật
  - Hệ thống thông tin cá nhân
  - Hệ thống thông tin làm việc nhóm
  - Hệ thống thông tin doanh nghiệp
- ❖ Phân loại theo đặc tính kĩ thuật
  - Hệ thống thông thường
  - Hệ thống thời gian thực
  - Hệ thống nhúng
- ❖ Hệ thống thông tin tích hợp
  - Được hình thành trên cơ sở tích hợp các hệ thống khác nhau để được một hệ thống tổng thể



# Tại sao cần phải phát triển HTTT?

- ❖ **Khắc phục** hạn chế, khó khăn cản trở việc đạt mục tiêu hiện tại
- ❖ **Tạo ưu thế** để vượt qua thách thức và tận dụng cơ hội trong tương lai
- ❖ Để **hợp tác** với đối tác

# Vai trò và tầm quan trọng môn học

- ❖ Tại sao phải thiết kế sản phẩm?
  - Hình dung sản phẩm một cách đầy đủ
  - Đưa vào sản phẩm những yêu cầu, ý tưởng hay
  - Sử dụng tài nguyên thích hợp, hiệu quả
  - Chuẩn bị một quy trình sản xuất hữu hiệu
  - Tạo kiến trúc tốt cho phần mềm
- ❖ Thiết kế phần mềm là gì?
  - Chuyển yêu cầu bài toán thành đặc tả để người lập trình chuyển thành chương trình vận hành được, đáp ứng yêu cầu đặt ra

# Vai trò và tầm quan trọng...

- ❖ Thiết kế một HTTT bao gồm:
  - Thiết kế hệ thống thiết bị phần cứng
  - Thiết kế hệ thống phần mềm
  - Thiết kế cơ sở dữ liệu
  - Phù hợp môi trường cài đặt, đáp ứng yêu cầu đặt ra
- ❖ Nhận xét:
  - Các mô hình, thiết bị phần cứng không nhiều, dễ nắm
  - Thiết kế CSDL có nhiều kinh nghiệm, tự động



## ❖ Vấn đề cơ bản về phát triển hệ HTTT

- Các cách tiếp cận phát triển HTTT
- Vòng đời phát triển HTTT
- Các phương pháp phát triển HTTT

# Các cách tiếp cận phát triển HTTT

- ❖ Tiếp cận hướng tiến trình
- ❖ Tiếp cận hướng dữ liệu
- ❖ Tiếp cận hướng cấu trúc
- ❖ Tiếp cận hướng đối tượng

# Tiếp cận hướng tiến trình

- ❖ Xuất hiện vào thời điểm máy tính mới ra đời
- ❖ Tập trung vào hiệu quả xử lý của chương trình
- ❖ Dữ liệu được tổ chức khi tiến trình cần
- ❖ Nhược điểm:

- **Hệ thống trả lương** tiến trình

- Dư thừa, tốn nhiều công sức tổ chức

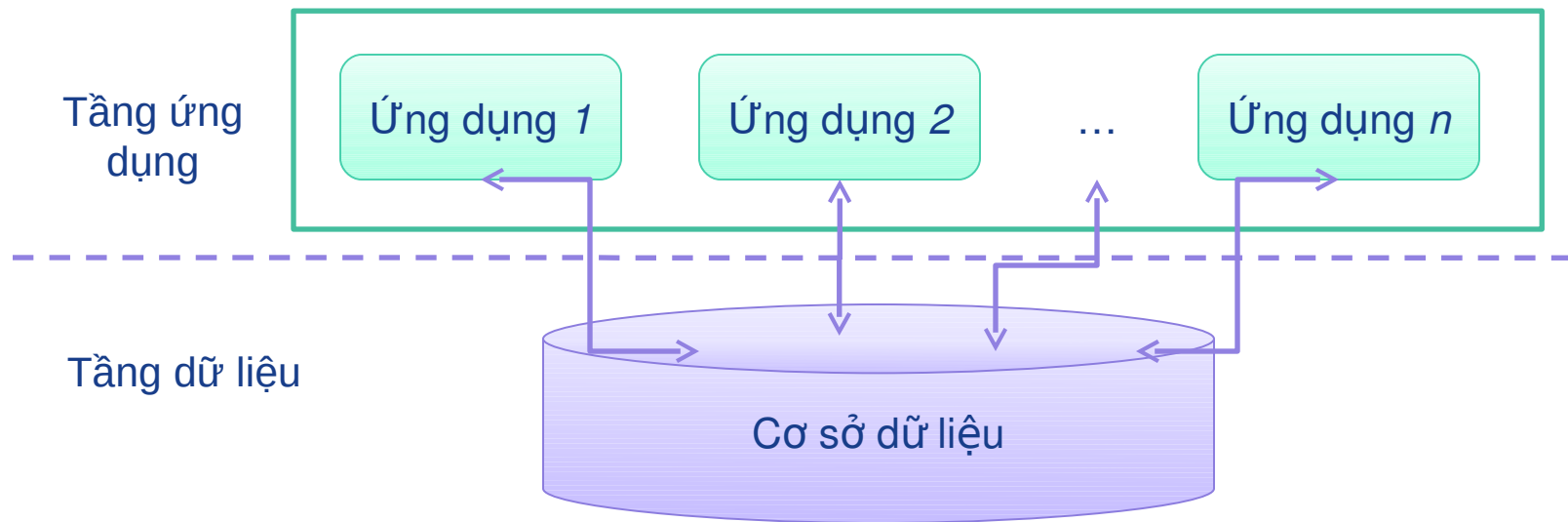


- **Hệ thống quản lý dự án**



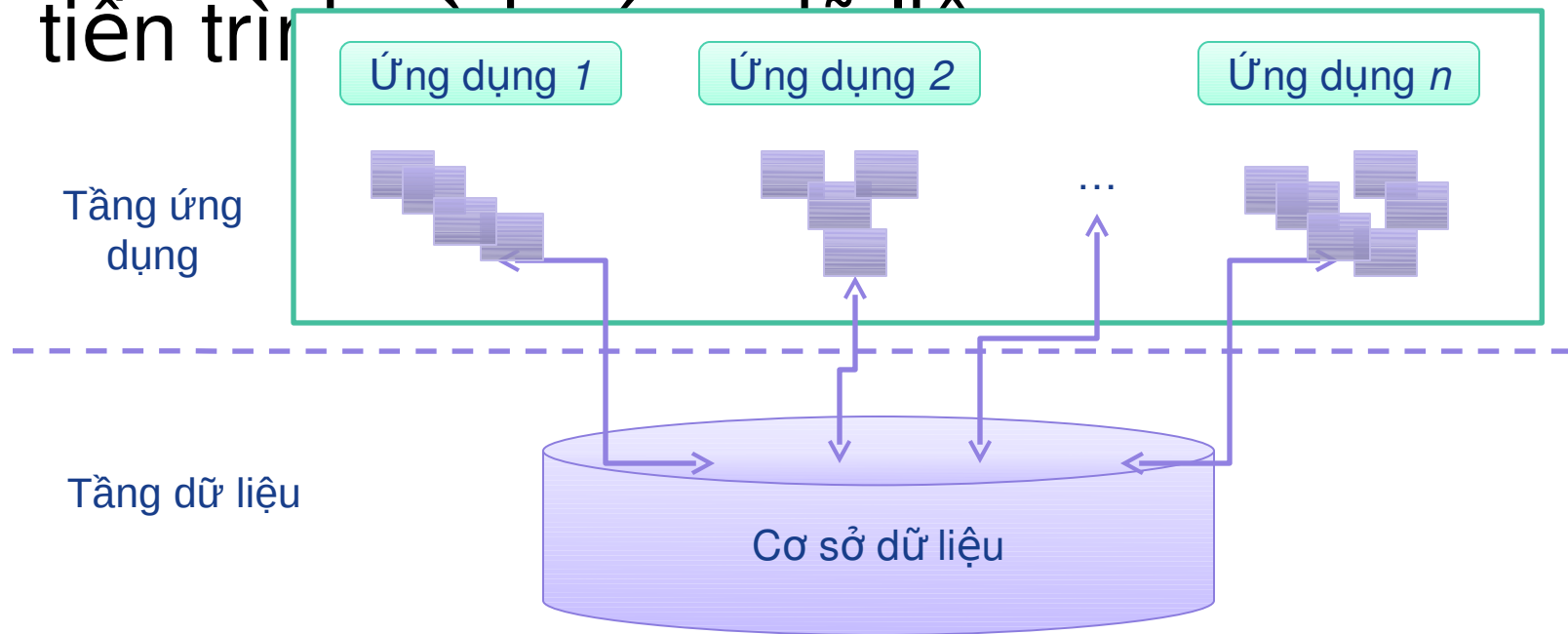
# Tiếp cận hướng dữ liệu

- ❖ Tập trung dữ liệu một cách lý tưởng:
  - Tách dữ liệu ra khỏi quá trình xử lý
  - Tổ chức dữ liệu dùng chung cho các ứng dụng



# Tiếp cận hướng cấu trúc

- ❖ Hệ thống phân thành các module chức năng
- ❖ Là sự kết hợp giữa cách tiếp cận hướng tiến trình và hướng cấu trúc





# Tiếp cận hướng cấu trúc...

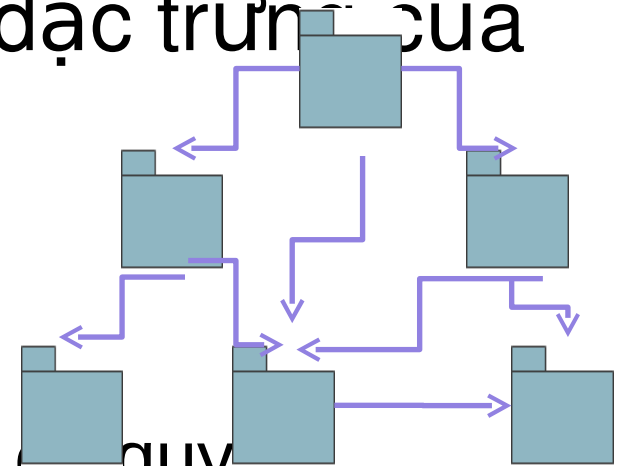
## ❖ Ưu điểm:

- Làm giảm sự phức tạp
- Tập trung vào ý tưởng
- Chuẩn mực hóa tiến trình
- Hướng về tương lai
- Giảm tính nghệ thuật trong thiết kế



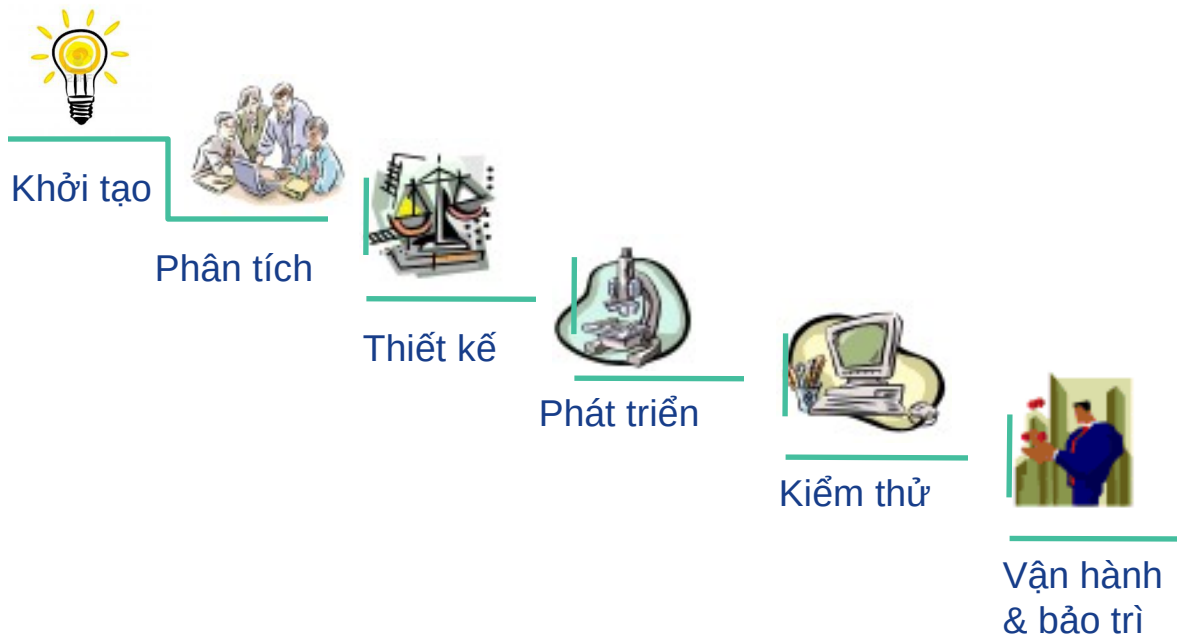
# Tiếp cận hướng đối tượng

- ❖ Hệ thống gồm các lớp đối tượng tương tác với nhau bằng các thông điệp
- ❖ Mỗi đối tượng được bao gói cả dữ liệu và các thao tác dữ liệu trên đó
- ❖ Một lớp có thể thừa kế các đặc trưng của lớp khác
- ❖ Ưu điểm:
  - Sử dụng lại
  - Cho phép phát triển hệ thống cơ quy mô tùy ý



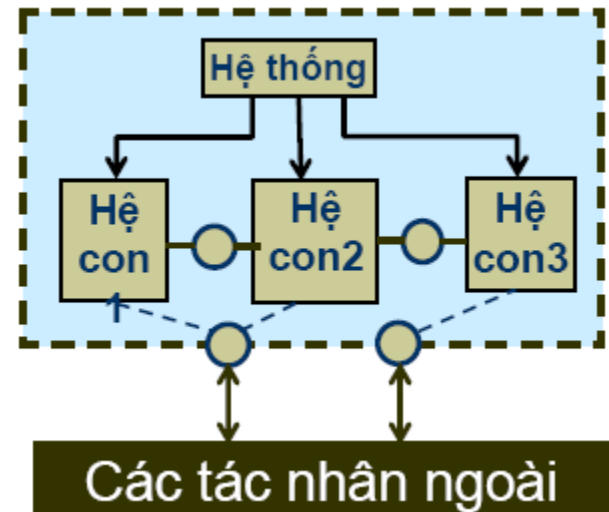
# Vòng đời phát triển HTTT

- ❖ Các pha phát triển HTTT từ lúc nó sinh ra đến lúc “hủy bỏ”



# Khởi tạo

- ❖ Nghiên cứu sơ bộ hệ thống
  - Các chức năng chính
  - Phạm vi
  - Các ràng buộc chung
- ❖ Nghiên cứu tính khả thi
  - Khả thi về mặt kỹ thuật
  - Khả thi về mặt kinh tế
  - Khả thi về mặt thời gian
  - Khả thi pháp lý và hoạt động
- ❖ Xây dựng kế hoạch tổng quát



# Phân tích

- ❖ Xác định yêu cầu thông tin của tổ chức:
  - Xác định yêu cầu (chức năng, phi chức năng)
  - Nghiên cứu và đặc tả
  - Tìm giải pháp cho thiết kế b  
pháp tổng thể: kiểm trúc và



# Thiết kế

- ❖ Tìm giải pháp công nghệ thông tin cụ thể để đáp ứng yêu cầu đặt ra
- ❖ Chuyển đặc tả yêu cầu thành bản thiết kế mô tả hệ thống
  - Thiết kế logic
  - Thiết kế vật lý



# Phát triển

- ❖ Chọn hệ thống nền (platform)
  - Cấu hình phần cứng
  - Phần mềm hệ thống, ngôn ngữ lập trình, hệ quản trị CSDL
- ❖ Chuyển thiết kế thành cơ sở dữ liệu
- ❖ Kiểm thử đơn vị
- ❖ Kiểm thử tích hợp
  - Các module con
  - Các hệ con



# Kiểm thử

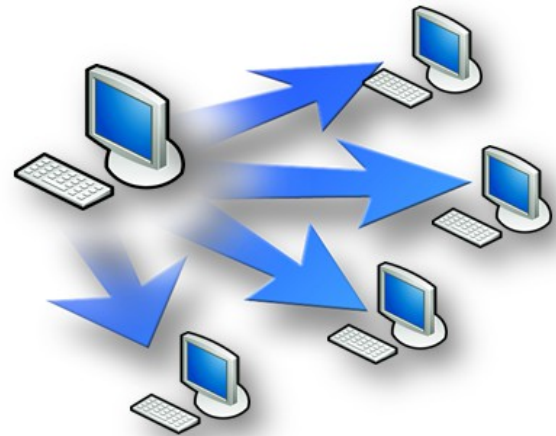
- ❖ Kiểm thử hệ thống
  - Kiểm thử chức năng, giao diện
  - Kiểm thử hiệu năng
  - Kiểm thử phục hồi
  - Kiểm thử chịu tải
  - Kiểm thử an toàn, bảo mật
- ❖ Kiểm thử chấp nhận
  - Kiểm thử Alpha
  - Kiểm thử Beta





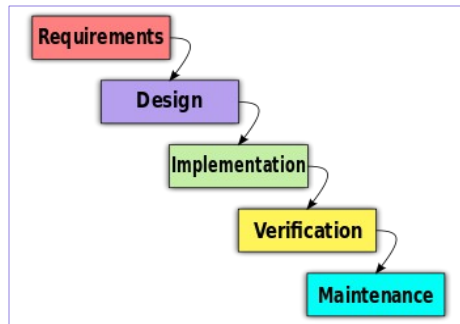
# Vận hành và bảo trì

- ❖ Lắp đặt hệ thống
  - Lắp đặt phần cứng
  - Cài đặt phần mềm
- ❖ Chuyển đổi hệ thống cũ sang hệ thống mới
  - Chuyển đổi dữ liệu
  - Bố trí cán bộ
  - Lập tài liệu hướng dẫn
  - Đào tạo
  - Vận hành

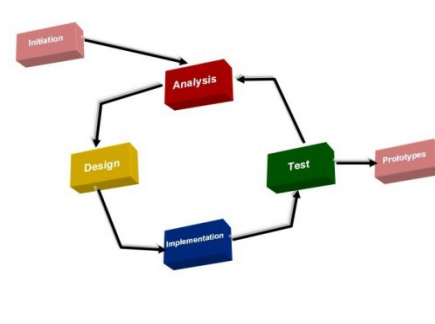


# Các cách phát triển HTTT

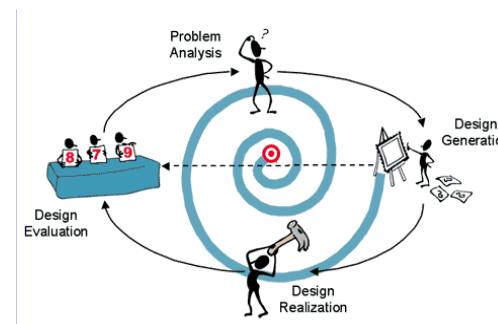
# Các phương pháp phát triển HTTT



Thác nước



Làm bản mẫu



Xoắn ốc

## ❖ **Nắm được:**

- Ý tưởng
- Nội dung
- Ưu & nhược điểm

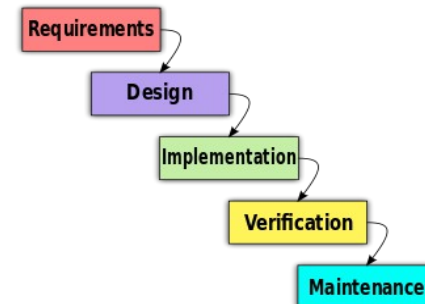


***Phương pháp nào ưu thế hơn cả? Tại sao?***

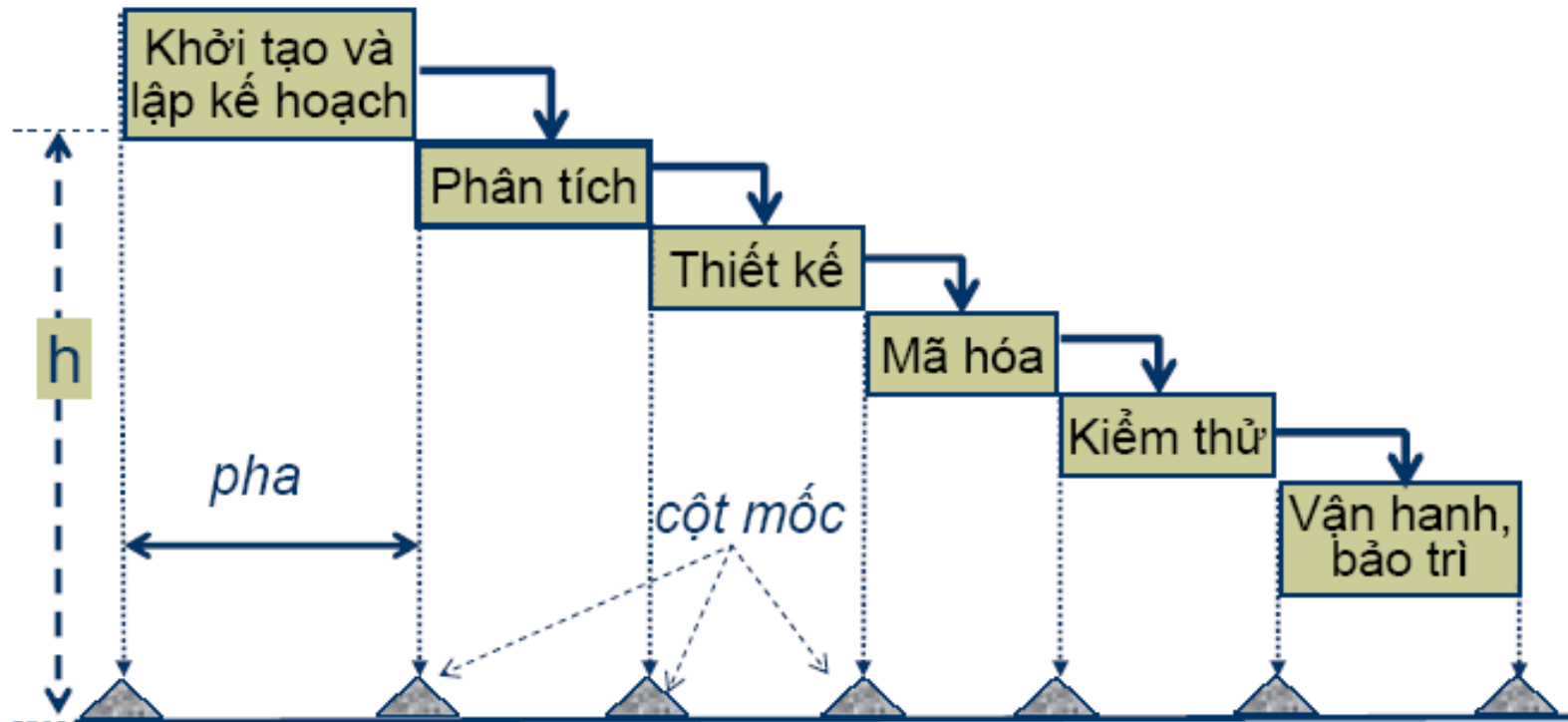
# Thác nước

## Ý tưởng

- Phân chia quá trình phát triển HTTT thành các giai đoạn (pha) liên tiếp
- Mỗi giai đoạn chỉ có thể bắt đầu khi giai đoạn trước đã hoàn tất



# Thác nước...



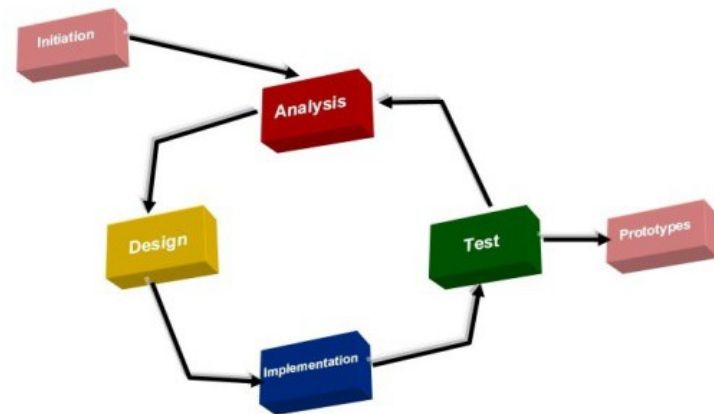
# Thác nước...

- ❖ Nhận xét và đánh giá
  - Dễ phân công công việc, phân bổ chi phí, giám sát công việc
  - Thời gian hoàn thành dự án thường chính xác hơn
  - Tài liệu xây dựng đầy đủ và hệ thống
  - Đòi hỏi có một bản yêu cầu chính xác
  - Khách hàng ít tin tưởng

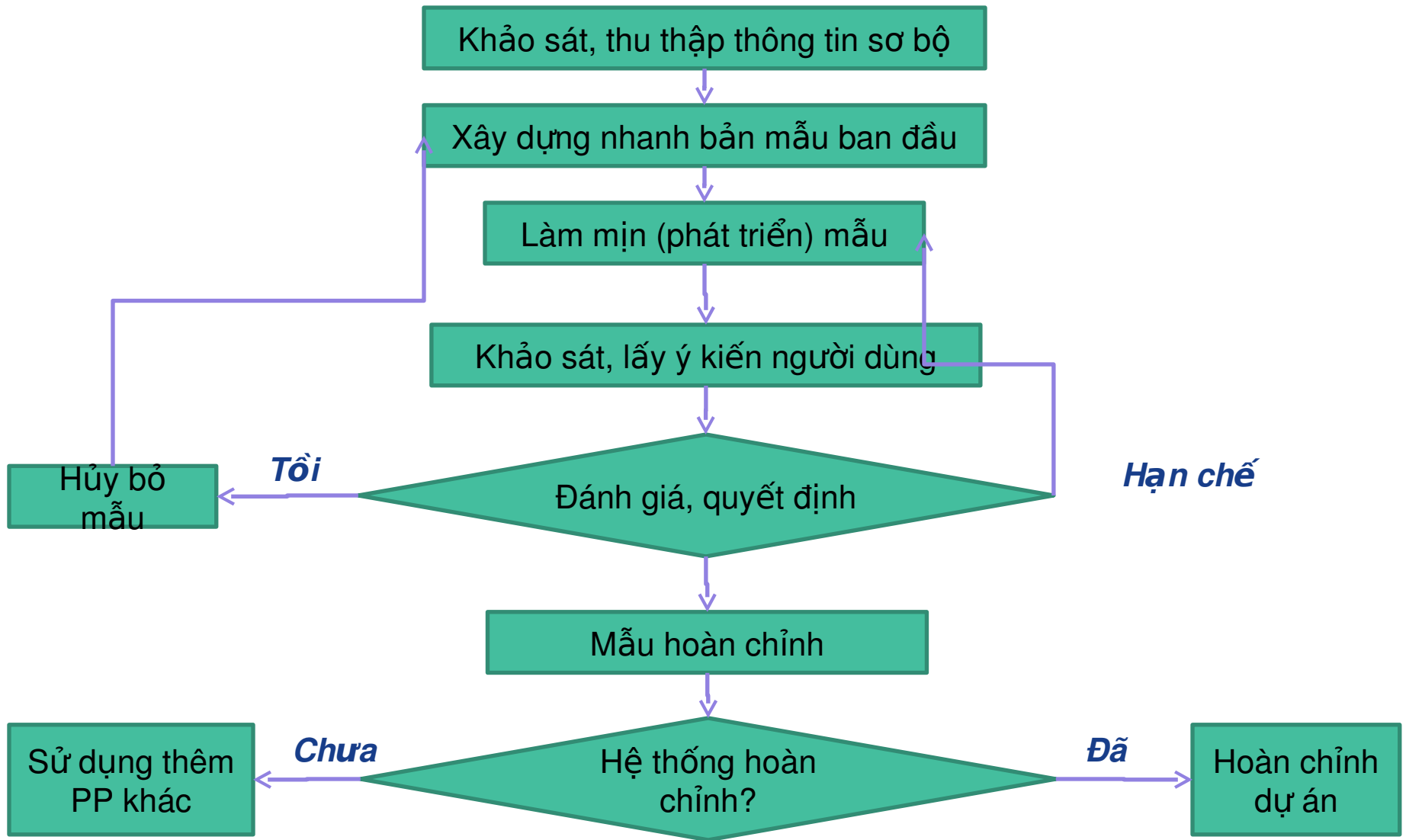
# Làm bản mẫu

## 💡 Ý tưởng

- Nhanh chóng tạo ra mô hình làm việc thực nghiệm để người dùng xem xét, đánh giá
- Bản mẫu được hoàn thiện và sử dụng cho các bước tiếp theo



# Làm bản mẫu...





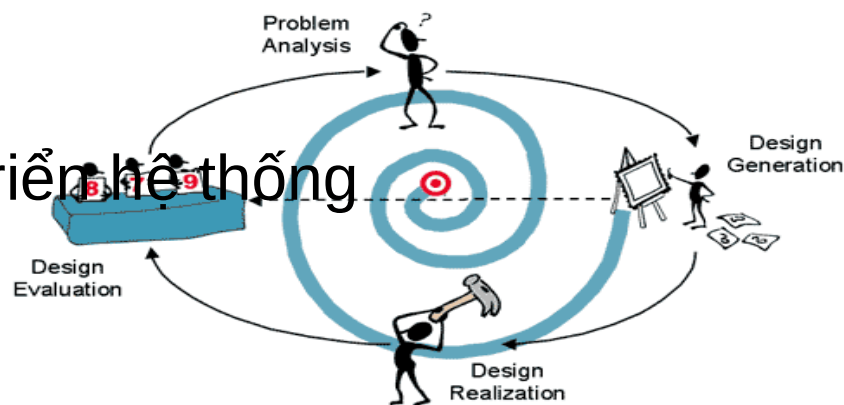
# Làm bản mẫu...

- ❖ Nhận xét và đánh giá
  - Thích hợp trong trường hợp nhu cầu thông tin hay giải pháp chưa xác định
  - Thích hợp với hệ thống nhỏ, hệ thống lớn có thể chia nhỏ và nhiều tương tác người dùng
  - Tạo cơ sở cho kí kết hợp đồng, tăng lòng tin của khách hàng, huấn luyện người dùng
  - Bảo trì khó khăn, sai sót chậm sửa đổi, tài liệu làm không kịp thời

# Xoắn ốc

## 💡 Ý tưởng

- Quá trình phát triển chia thành nhiều bước lặp, phát triển từ trung tâm
- Mỗi bước gồm các hoạt động
  - Lập kế hoạch
  - Phân tích rủi ro
  - Tạo bản mẫu
  - Hoàn thiện và phát triển hệ thống



# Xoắn ốc

Tập hợp yêu cầu ban đầu và kế hoạch dự án

Kế hoạch dựa trên ý kiến của khách hàng

Đánh giá của khách hàng

**Lập kế hoạch**

**Phân tích rủi ro**

**Lấy ý kiến đánh giá**

**Kỹ nghệ**

Phân tích rủi ro dựa trên yêu cầu ban đầu

Phân tích rủi ro dựa trên phản ứng của KH

Quyết định tiếp tục? (cao điểm của việc phân tích rủi ro)

Hướng tới hệ thống hoàn chỉnh

Bản mẫu ban đầu  
Bản mẫu tiếp theo

# Mô hình xoắn ốc...

- ❖ **Đánh giá và nhận xét**
  - Thích hợp với các hệ thống quy mô lớn
  - Giảm rủi ro
  - Thỏa thuận lại với khách hàng nếu cần thiết  
→ có sự thay đổi linh hoạt trong phát triển

