



Bài giảng:

# CÔNG NGHỆ PHẦN MỀM

Giảng viên: Nguyễn Quang Vũ  
Khoa Khoa học máy tính



## Nội dung bài giảng:



Chương 1:

# TỔNG QUAN VỀ CÔNG NGHỆ PHẦN MỀM



## 1.1 Các khái niệm cơ bản

- **Phần mềm** (software): *là một tập hợp các câu lệnh được viết bằng một hoặc nhiều ngôn ngữ lập trình (được gọi là các chương trình), nhằm tự động thực hiện một số các chức năng giải quyết một bài toán.*



## CHƯƠNG 1. TỔNG QUAN VỀ CNPM

### 1.1 Các khái niệm cơ bản (tt)

- **Công nghệ** (engineering): *là cách sử dụng các công cụ, các kỹ thuật trong cách giải quyết một vấn đề.*



## 1.1 Các khái niệm cơ bản (tt)

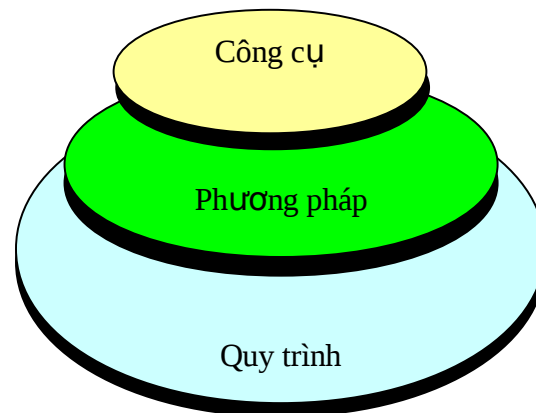
- **Công nghệ phần mềm** (software engineering):  
*là việc áp dụng các công nghệ một cách hệ thống trong việc phát triển các ứng dụng dựa trên máy tính.*



## CHƯƠNG 1. TỔNG QUAN VỀ CNPM

### 1.1 Các khái niệm cơ bản (tt)

#### - Mô hình 3 tầng của CNPM





## 1.1 Các khái niệm cơ bản (tt)

- Nói một cách khác, công nghệ phần mềm bao trùm kiến thức, các công cụ, và các phương pháp để:
  - . định nghĩa yêu cầu phần mềm
  - . thiết kế phần mềm
  - . xây dựng phần mềm
  - . kiểm thử phần mềm
  - . bảo trì phần mềm





## 1.1 Các khái niệm cơ bản (tt)

- Công nghệ phần mềm còn sử dụng kiến thức của các lĩnh vực khác:
  - . kỹ thuật máy tính
  - . khoa học máy tính
  - . quản lý
  - . toán học
  - . quản lý dự án
  - . quản lý chất lượng
  - . công nghệ hệ thống (systems engineering).



## 1.1 Các khái niệm cơ bản (tt)

*“Khi máy tính chưa xuất hiện, thì việc lập trình chưa có khó khăn gì cả. Khi mới xuất hiện một vài chiếc máy tính chức năng kém thì việc lập trình bắt đầu gặp một vài khó khăn nho nhỏ. Giờ đây khi chúng ta có những chiếc máy tính khổng lồ thì những khó khăn ấy trở nên vô cùng lớn. Như vậy ngành công nghiệp điện tử không giải quyết khó khăn nào cả mà họ chỉ tạo thêm ra những khó khăn mới. Khó khăn mà họ tạo nên chính là việc sử dụng sản phẩm của họ.”*

**(Edsger Dijkstra)**



## CHƯƠNG 1. TỔNG QUAN VỀ CNPM

### 1.1 Các khái niệm cơ bản (tt)

- Và nhiều khái niệm khác ....

### 1.2 Lịch sử công nghệ phần mềm



## 1.3 Tiêu chuẩn của một sản phẩm phần mềm

- Tính đúng
- Tính khoa học
- Tính tin cậy
- Tính kiểm thử được
- Tính hữu hiệu
- Tính sáng tạo
- Tính an toàn
- Tính toàn vẹn



### 1.3 Tiêu chuẩn của một sản phẩm phần mềm (tt)

- Tính đối xứng và đầy đủ chức năng
- Tính tiêu chuẩn và tính chuẩn
- Tính độc lập
- Tính dễ phát triển, hoàn thiện
- Ngoài ra: phổ dụng, đơn giản, liên tác, súc tích, thứ lỗi, modul hóa, đầy đủ hồ sơ, theo dõi được, vận hành dễ,...



## 1.4 Hồ sơ của một sản phẩm phần mềm

- Đặc tả hệ thống.
- Kế hoạch dự án phần mềm.
  - . Đặc tả yêu cầu phần mềm.
  - . Bản mẫu thực hiện được hay "trên giấy".
- Tài liệu người dùng sơ bộ



## 1.4 Hồ sơ của một sản phẩm phần mềm (tt)

### • Đặc tả thiết kế.

- . Mô tả thiết kế dữ liệu.
- . Mô tả thiết kế kiến trúc.
- . Mô tả thiết kế module.
- . Mô tả thiết kế giao diện.
- . Mô tả sự vật (nếu kỹ thuật hướng sự vật được dùng).



## 1.4 Hồ sơ của một sản phẩm phần mềm (tt)

### ● Chương trình gốc

- . Chương trình nguồn.
- . Bản in chương trình nguồn (listing).
- . Bản mô tả thuật toán tương ứng với chương trình nguồn.
- . Kế hoạch và thủ tục kiểm thử.
- . Các trường hợp kiểm thử và kết quả ghi lại.





## 1.4 Hồ sơ của một sản phẩm phần mềm (tt)

- Tài liệu vận hành và cài đặt.
  - . Bản liệt kê các lỗi và cách xử lý.
  - . Bản liệt kê các thông số đặc trưng của hệ thống.
- Mô tả cơ sở dữ liệu.
  - . Diagram và tự diễn dữ liệu.
  - . Dữ liệu ban đầu



## CHƯƠNG 1. TỔNG QUAN VỀ CNPM

### 1.4 Hồ sơ của một sản phẩm phần mềm (tt)

- Tài liệu người sử dụng đã xây dựng.
  - . Bản hướng dẫn sử dụng chi tiết.
  - . Bản tóm tắt hướng dẫn sử dụng.
  - . Các chương trình trợ giúp có liên quan.
- Tài liệu bảo trì.
  - . Báo cáo vấn đề còn tồn tại.
  - . Yêu cầu bảo trì.
  - . Trình tự thay đổi công nghệ.
- Các chuẩn và thủ tục cho kỹ thuật phần mềm .
- Các tư liệu khác: hợp đồng, phiên bản, tài liệu pháp lý,...



Chương 2:

# CÁC HOẠT ĐỘNG TRONG TIẾN TRÌNH PHẦN MỀM



## 2.1 Tiến trình phần mềm

- Là một tập hợp các hành động mà mục đích của nó là xây dựng và phát triển phần mềm
- Bao gồm các hoạt động:
  - . Đặc tả
  - . Phát triển: Thiết kế và cài đặt
  - . Kiểm thử
  - . Mở rộng: Bảo trì, cải tiến



## 2.2 Đặc tả

- Còn gọi là kỹ thuật xác định yêu cầu
- Là quy trình tìm hiểu và định nghĩa những dịch vụ nào được yêu cầu và các ràng buộc trong quá trình vận hành và xây dựng hệ thống.
- Gồm 4 pha chính
  - . Nghiên cứu khả thi
  - . Phân tích và rút ra các yêu cầu
  - . Đặc tả yêu cầu
  - . Đánh giá yêu cầu



## 2.3 Thiết kế

- Là quá trình thiết kế cấu trúc phần mềm dựa trên những tài liệu đặc tả
- Gồm các công việc chính
  - . Thiết kế kiến trúc
  - . Đặc tả trừu tượng
  - . Thiết kế giao diện
  - . Thiết kế thành phần
  - . Thiết kế cấu trúc dữ liệu
  - . Thiết kế thuật toán



## 2.4 Cài đặt

- Là quá trình chuyển đổi từ tài liệu đặc tả hệ thống thành một hệ thống thực, có thể vận hành được và phải loại bỏ các lỗi của chương trình
- Hoạt động cá nhân
- Không có quy trình chung



## 2.5 Kiểm thử

### 2.5.1 Xác minh và thẩm định

- V&V – Verification and Validation
- Là từ chung cho các quá trình kiểm thử để đảm bảo rằng phần mềm thỏa mãn các yêu cầu của chúng và các yêu cầu đó thỏa mãn các nhu cầu của người sử dụng
- Có hai mục tiêu:
  - . Phát hiện các khuyết tật trong hệ thống.
  - . Đánh giá xem hệ thống liệu có dùng được hay không?





## 2.5 Kiểm thử

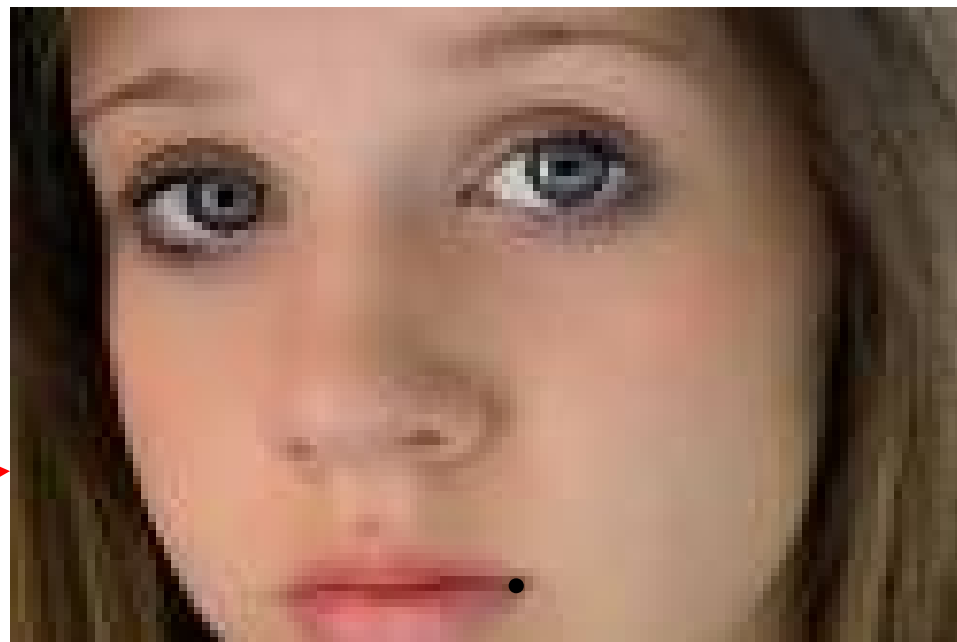
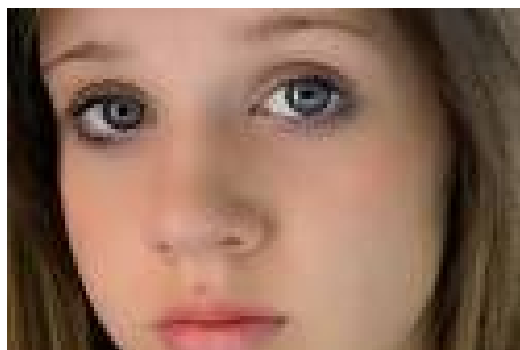
### 2.5.1 Xác minh và thẩm định (tt)

- Verification: Are we building the product right?
- Validation: Are we buiding the right product ?



## 2.5.2 Kiểm thử phần mềm (KTPM)

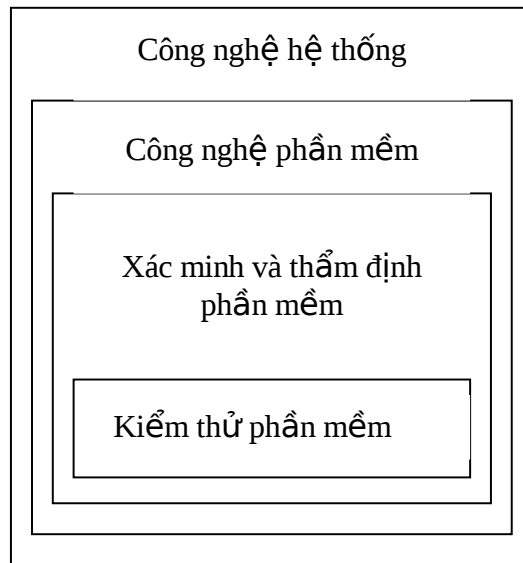
KTPM là quá trình thực hiện một chương trình phần mềm với mục đích là tìm ra **LỖI**, nếu có.



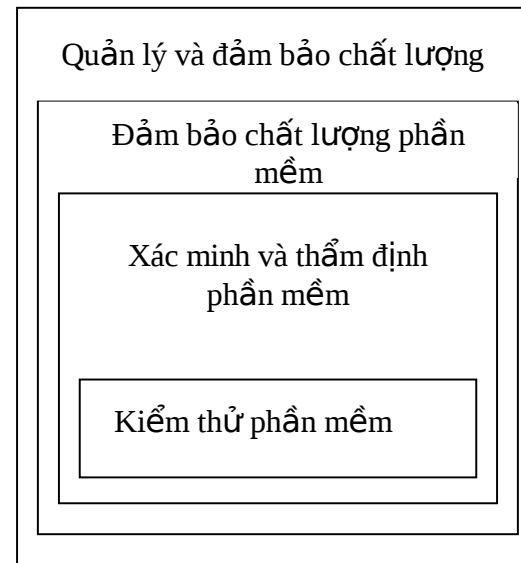
**KIỂM THỬ PHẦN MỀM**



## 2.5.2 Kiểm thử phần mềm (KTPM)



(a) Ngữ cảnh quy trình



(b) Ngữ cảnh chất lượng

**Kiểm thử phần mềm trong một số ngữ cảnh**



## 2.5.2 Kiểm thử phần mềm (KTPM)

- Việc kiểm thử của đội dự án được gọi là kiểm thử phát triển (Development test).
- Các kiểm thử bởi các cơ quan bên ngoài được gọi là đảm bảo chất lượng (Quality assurance-QA) và kiểm thử chấp nhận (Acceptance test).



## 2.5.2 Kiểm thử phần mềm (KTPM)

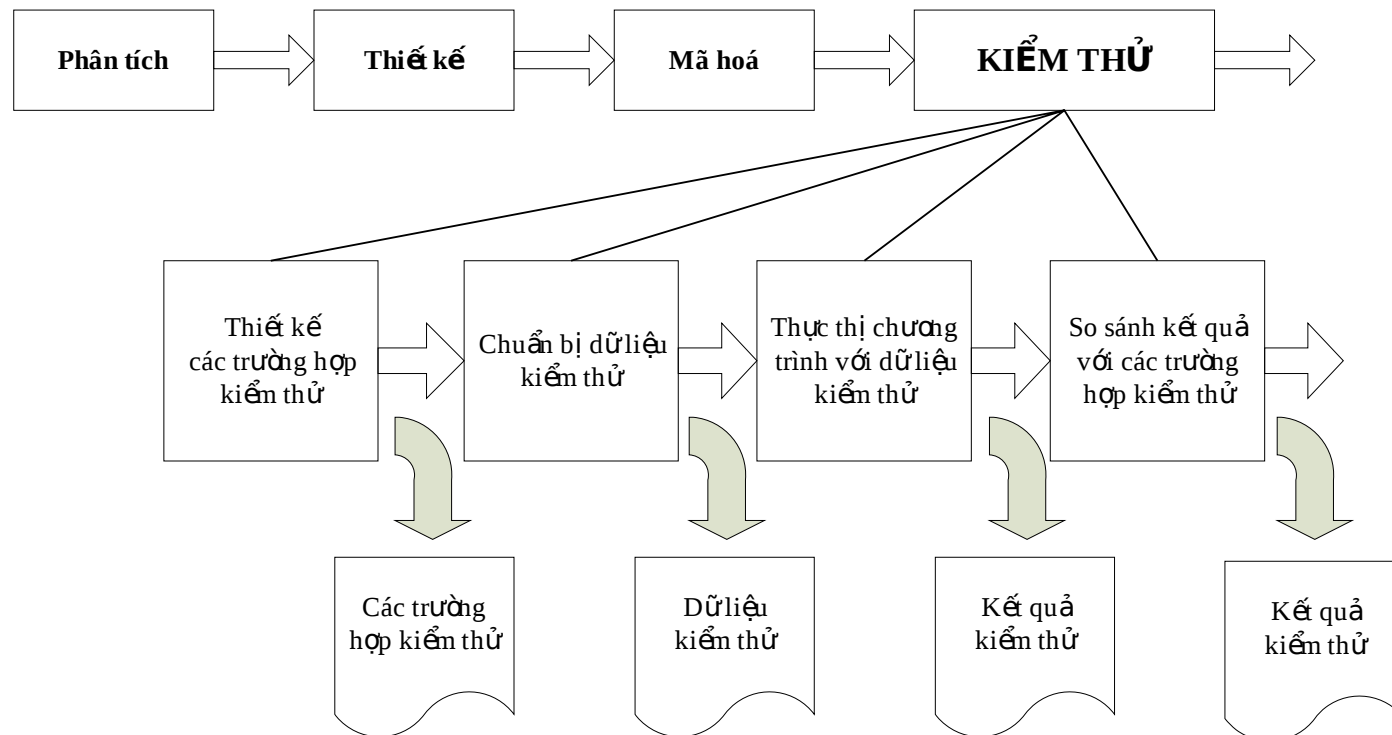
Quy trình kiểm thử:

- Lập kế hoạch kiểm thử
- Bố trí nhân viên kiểm thử.
- Thiết kế các trường hợp kiểm thử.
- Xử lý đo lường kiểm thử bằng cách thu thập dữ liệu.
- Đánh giá sản phẩm phần mềm.



## 2.5.2 Kiểm thử phần mềm (KTPM)

### Mô hình chung





## 2.5.2 KTPM (tt)

Dù **LỖI** nhỏ  
hay lớn, tớ  
vẫn sẽ tìm ra  
!

**LỖI !!!**  
 $1 + 1 =$   
**3 ???**

**LỖI !!!** Không  
dùng được !

Alo, **LỖI !!**  
Tôi có yêu  
cầu thế  
đâu !





## 2.5.2 KTPM (tt)

- **Theo quan điểm của người dùng:** Đảm bảo phần mềm đủ khả năng làm việc trong môi trường thực.
- **Sản phẩm của KTPM:** Bảng đánh giá về quá trình xây dựng phần mềm.
- **Vai trò của KTPM:** Công cụ tối quan trọng, quyết định đến việc đánh giá chất lượng phần mềm.





## - Kiểm thử viên:

- . **Giỏi** chuyên môn nghiệp vụ
- . **Sáng tạo**
- . **Tâm**



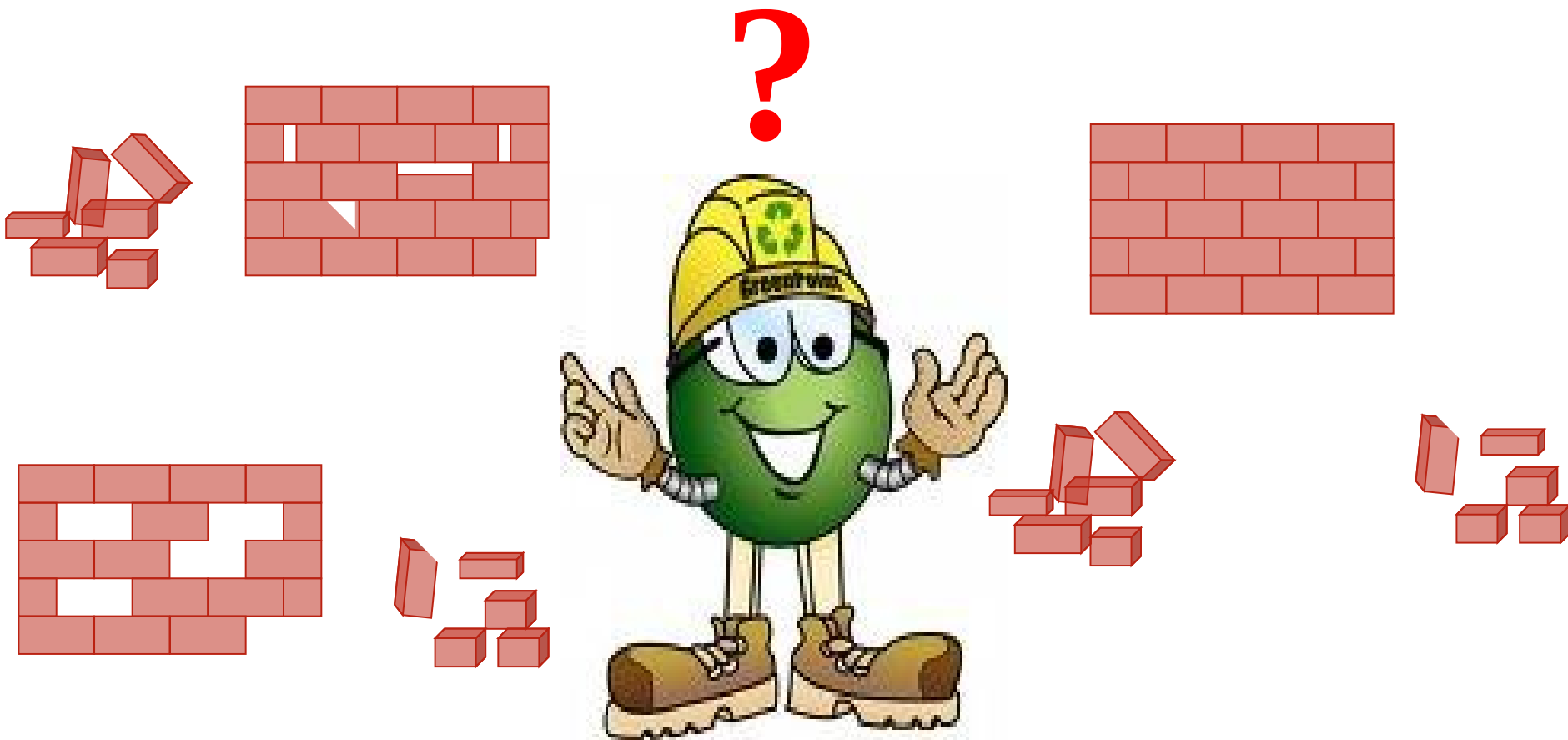


## Các nguyên tắc cơ bản của KTPM

- Nguyên tắc **khách quan**
- Nguyên tắc **ngẫu nhiên**
- Nguyên tắc “**Người sử dụng kém**”
- Nguyên tắc “**Kẻ phá hoại**”



## Các mức kiểm thử



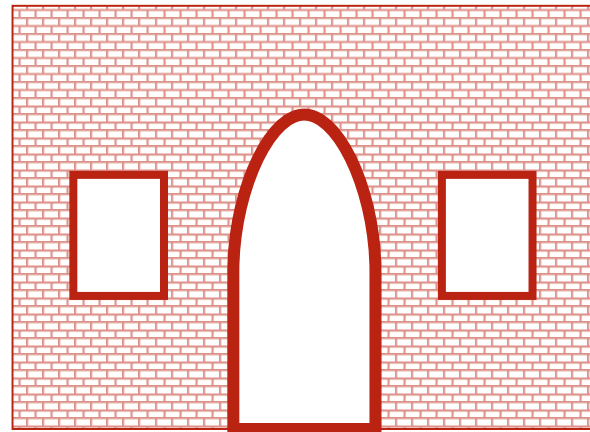


## Kiểm thử đơn vị ( Unit test)

- Thế nào là một đơn vị phần mềm ?
- Kiểm thử đơn vị:
  - . Riêng biệt từng đơn vị phần mềm
  - . Số lượng nhiều nhưng đơn giản
  - . Xuyên suốt thời gian lập trình và cả chu kỳ phần mềm
- Lập kế hoạch ngay khi lập trình

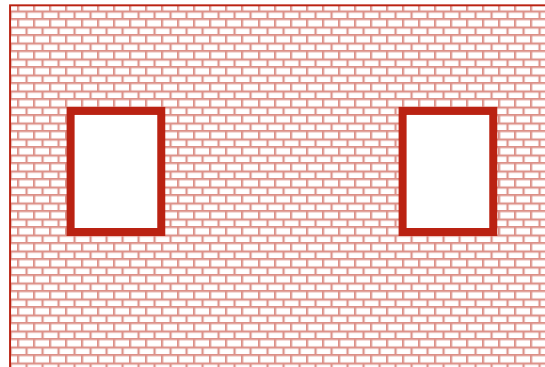


## - Thợ A: Bức tường phía trước



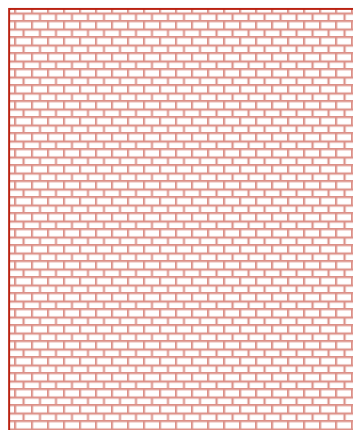


## - Thợ B: Bức tường phía sau



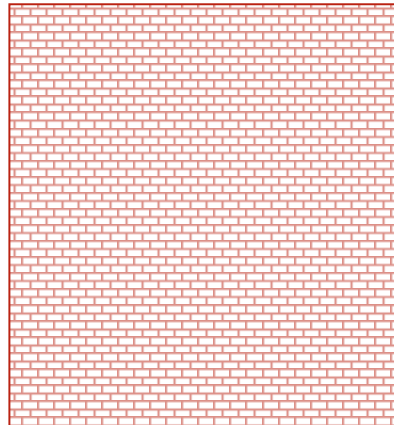


## - Thợ C: Bức tường bên trái

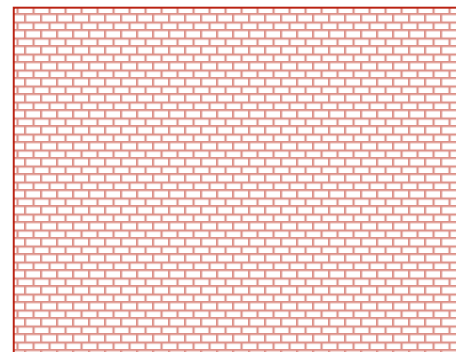
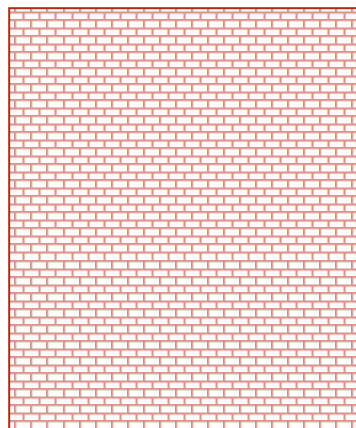
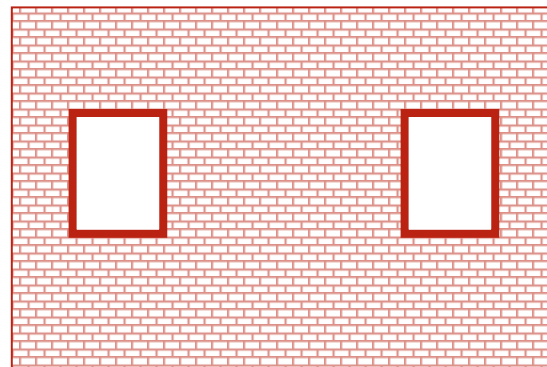
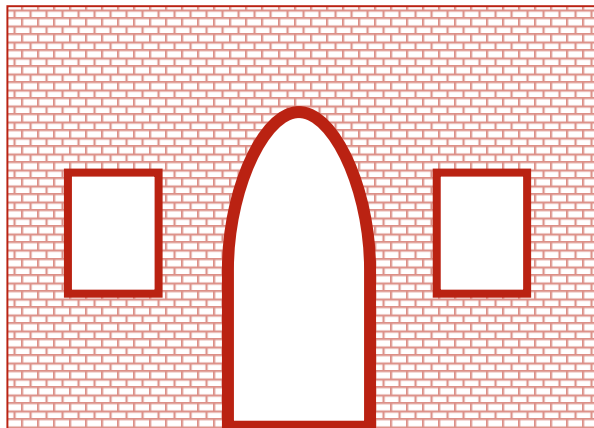




## - Thợ C: Bức tường bên phải







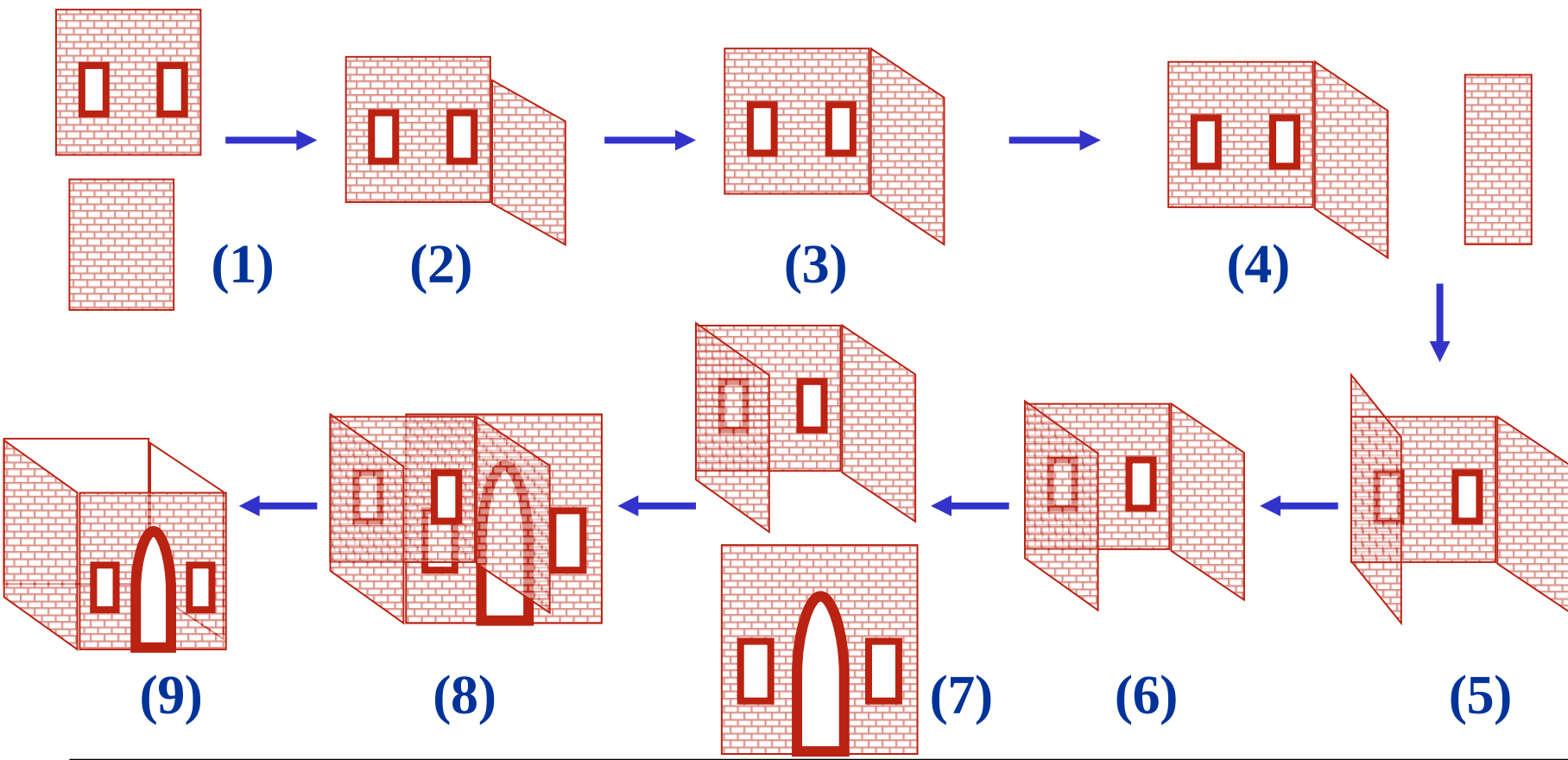


## **Kiểm thử tích hợp ( Intergration test)**

- Tích hợp dần các đơn vị phần mềm
- Phát hiện lỗi giao tiếp và sự không tương thích
- Tại mỗi thời điểm, chỉ tích hợp thêm 1 đơn vị phần mềm



## Kiểm thử tích hợp (tt)





## **Kiểm thử tích hợp ( tt)**

- Có 4 loại kiểm thử cơ bản trong kiểm thử tích hợp
  - . Kiểm thử cấu trúc
  - . Kiểm thử chức năng
  - . Kiểm thử hiệu năng
  - . Kiểm thử khả năng chịu tải
- Lập kế hoạch khi thiết kế chi tiết



- **Nắng nóng ?**
- **Mưa bão ?**
- **Thông gió ?**
- **Điện 220V ?**
- **PCCC ?**
- ....



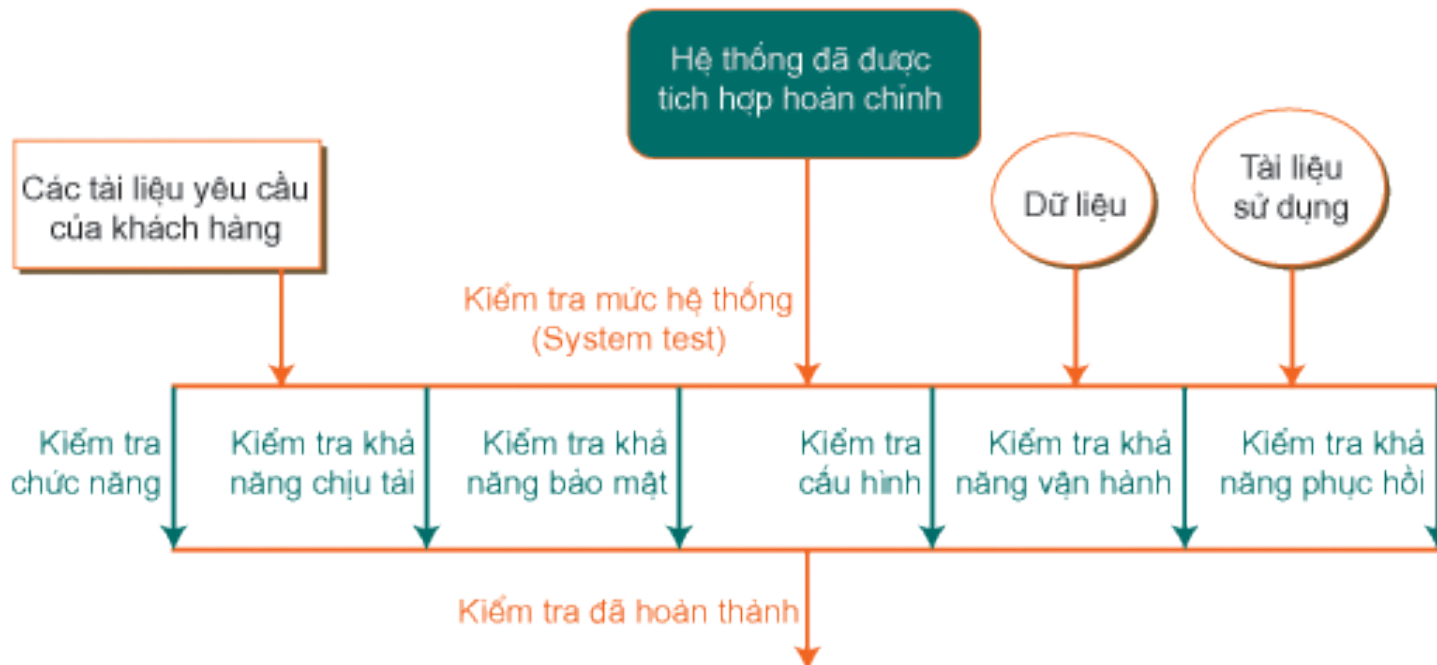


## Kiểm thử hệ thống (System test)

- Toàn bộ hệ thống
  - . Chức năng phần mềm
  - . Giao tiếp với phần mềm/phần cứng bên ngoài
  - . Các yêu cầu về chất lượng
- Hệ thống phải được tích hợp thành công trước đó
- Lập kế hoạch khi thiết kế kiến trúc (thiết kế cấp cao)



## Kiểm thử hệ thống ( tt)





Xe ô-tô để ở  
đâu nhỉ ?

Nhảy dây  
trên sàn  
có làm ồn  
không nhỉ ?



Nói  
chuyện  
với hàng  
xóm  
được  
không ?

Đá  
bóng ?





## Kiểm thử chấp nhận (Acceptance test)

- Khách hàng thực hiện
- Tương tự như System Test nhưng khác nhau về bản chất
- Có hai loại kiểm thử là Alpha Test và Beta Test
- Kèm theo một nhóm những dịch vụ và tài liệu đi kèm như hướng dẫn cài đặt, sử dụng,...
- Lập kế hoạch khi nhận yêu cầu khách hàng



**Tôi muốn sửa  
lại phòng khách  
lớn hơn ?**

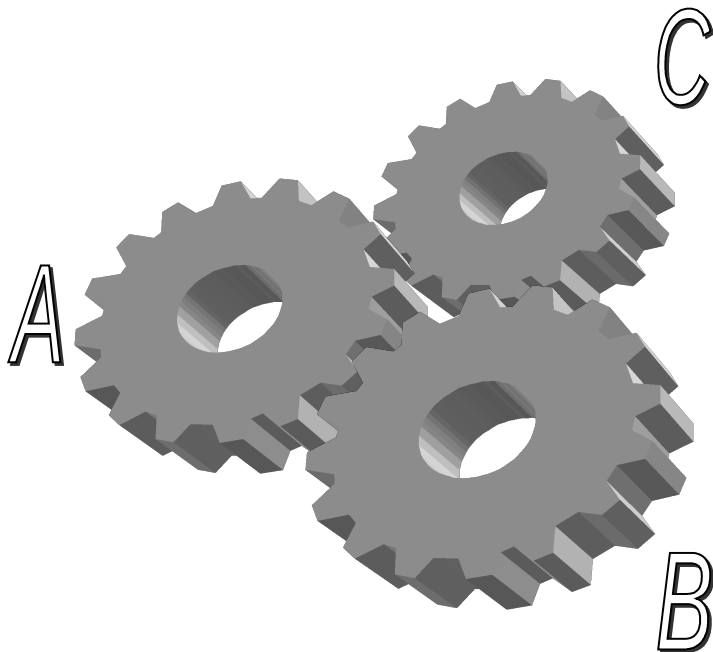


**Vấn đề  
này ...!!!  
?**





## Kiểm thử hồi quy (Regression test)



**Điều gì sẽ xảy ra khi C có sự thay đổi ?**



## Kiểm thử hồi quy (tt)

- Không phải là một mức kiểm thử !
- Kiểm thử lại phần mềm khi có sự thay đổi xảy ra
- Có thể thực hiện tại mọi mức kiểm thử
- Không được phép bỏ qua nếu có sự thay đổi !



## Phát triển phần mềm

## Kiểm thử phần mềm

Yêu cầu khách hàng  
(Requirements)

Lập kế hoạch cho

Chấp nhận sản phẩm  
(Acceptance)

Thiết kế cấp cao  
(High-level design)

Lập kế hoạch cho

Toàn bộ hệ thống  
(System)

Thiết kế chi tiết  
(Detailed design)

Lập kế hoạch cho

Tích hợp hệ thống  
(Integration)

Lập trình  
(Coding)

Lập kế hoạch cho

Đơn vị lập trình  
(Unit)



## 2.6 Bảo trì và cải tiến phần mềm

### 2.6.1 Bảo trì

- Hoạt động chỉnh sửa chương trình sau khi nó đã được đưa vào sử dụng
- Bảo trì là không thể tránh khỏi vì:
  - . Các yêu cầu hệ thống thường thay đổi
  - . Các hệ thống có gắn kết chặt chẽ với môi trường của nó.
  - . Các hệ thống phải được bảo trì nếu chúng muốn là những phần hữu ích trong môi trường nghiệp vụ.



## Phân loại bảo trì:

- Bảo trì sửa lỗi
- Bảo trì tích hợp hệ thống vào một môi trường khác
- Bảo trì để bổ sung hoặc chỉnh sửa các yêu cầu chức năng của hệ thống



## CHƯƠNG 2. CÁC HOẠT ĐỘNG ...

### Các nhân tố ảnh hưởng đến bảo trì:

- Sự ổn định của đội dự án
- Những trách nhiệm đã cam kết
- Kỹ năng của nhân viên
- Tuổi thọ và cấu trúc chương trình





## CHƯƠNG 2. CÁC HOẠT ĐỘNG ...

### Các nhân tố ảnh hưởng đến bảo trì:

- Sự ổn định của đội dự án
- Những trách nhiệm đã cam kết
- Kỹ năng của nhân viên
- Tuổi thọ và cấu trúc chương trình



## CHƯƠNG 2. CÁC HOẠT ĐỘNG ...

### Các công việc bảo trì:

- Thiết lập cơ cấu bảo trì
- Báo cáo
- Lưu giữ các hồ sơ
- Xác định giá bảo trì



## Các công việc bảo trì (tt)

Lưu giữ các hồ sơ gồm:

- Dấu hiệu nhận biết chương trình.
- Số lượng các câu lệnh trong chương trình nguồn.
- Số lượng các lệnh mã máy.
- Ngôn ngữ lập trình được sử dụng.
- Ngày cài đặt chương trình.
- Số các chương trình chạy từ khi cài đặt.
- Số các lỗi xử lý xảy ra.
- Mức và dấu hiệu thay đổi chương trình.
- Số các câu lệnh được thêm vào chương trình nguồn khi chương trình thay đổi.



## Các công việc bảo trì (tt)

Lưu giữ các hồ sơ gồm (tt):

- Số các câu lệnh được xóa khỏi chương trình nguồn khi chương trình thay đổi.
- Số giờ mỗi người sử dụng cho mỗi lần sửa đổi.
- Ngày thay đổi chương trình.
- Dấu hiệu của kỹ sư phần mềm.
- Dấu hiệu của đơn yêu cầu bảo trì.
- Kiểu bảo trì.
- Ngày bắt đầu và kết thúc bảo trì.
- Tổng số giờ của mỗi người dùng cho việc bảo trì.



## Các công việc bảo trì:

Xác định giá bảo trì bao gồm:

- Số lượng trung bình các lỗi xử lý cho một lần chạy chương trình.
- Tổng số giờ của mỗi người dùng cho mỗi loại bảo trì.
- Số lượng trung bình các thay đổi theo chương trình, theo ngôn ngữ lập trình, theo kiểu bảo trì.
- Số giờ trung bình của mỗi người cho một dòng lệnh được thêm vào và xóa đi.
- Số giờ trung bình của mỗi người cho một ngôn ngữ lập trình.
- Thời gian trung bình cho việc bảo trì một đơn yêu cầu bảo trì.
- Tỷ lệ phần trăm của mỗi kiểu bảo trì.



## 2.6 Bảo trì và cải tiến phần mềm

### 2.6.1 Cải tiến

Phải cải tiến vì thay đổi phần mềm là một điều không thể tránh khỏi vì những lí do sau:

- Những yêu cầu mới sẽ xuất hiện khi sử dụng phần mềm
- Môi trường nghiệp vụ thay đổi
- Các lỗi phần mềm cần phải sửa chữa
- Máy tính và các thiết bị mới được bổ sung vào hệ thống
- Hiệu năng hoặc độ tin cậy của hệ thống phải được cải thiện.



## 2.7 Quản lý thay đổi phần mềm

- Các ứng dụng thường xuyên phải thiết kế lại
- Các thay đổi có thể là các yêu cầu, thiết kế, chương trình, giao diện, phần cứng hoặc phần mềm phải mua.
- Việc quản lý thay đổi ứng dụng giúp cho nhóm triển khai bỏ qua những ý thích chợt nảy ra của người sử dụng trong khi vẫn cho phép thực hiện các yêu cầu hợp lý



## Chương 3

# MÔ HÌNH PHÁT TRIỂN PHẦN MỀM





CHƯƠNG 3. MÔ HÌNH PHÁT TRIỂN PM

- \* Qui trình phát triển/xây dựng phần mềm (Software Development/Engineering Process - SEP) có tính chất quyết định để tạo ra sản phẩm chất lượng tốt với chi phí thấp và năng suất cao



Thông thường một quy trình bao gồm các yếu tố cơ bản sau:

- Thủ tục (Procedures)
- Hướng dẫn công việc (Activity Guidelines)
- Biểu mẫu (Forms/templates)
- Danh sách kiểm định (Checklists)
- Công cụ hỗ trợ (Tools)



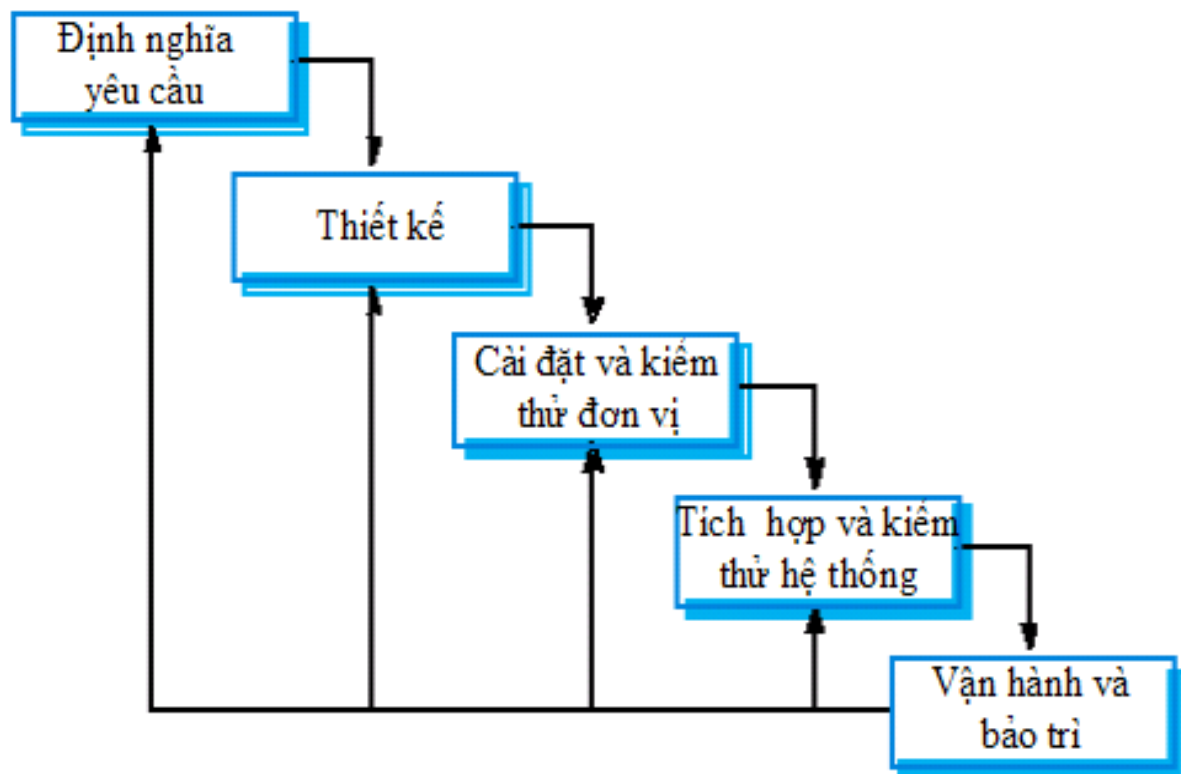
Và các nhóm công việc chính:

- Đặc tả yêu cầu (Requirements Specification)
- Phát triển phần mềm (Development)
- Kiểm thử phần mềm (Validation/Testing)
- Thay đổi phần mềm (Evolution)



## 3.1 Các mô hình một phiên bản

### Mô hình thác nước





### 3.1 Các mô hình một phiên bản

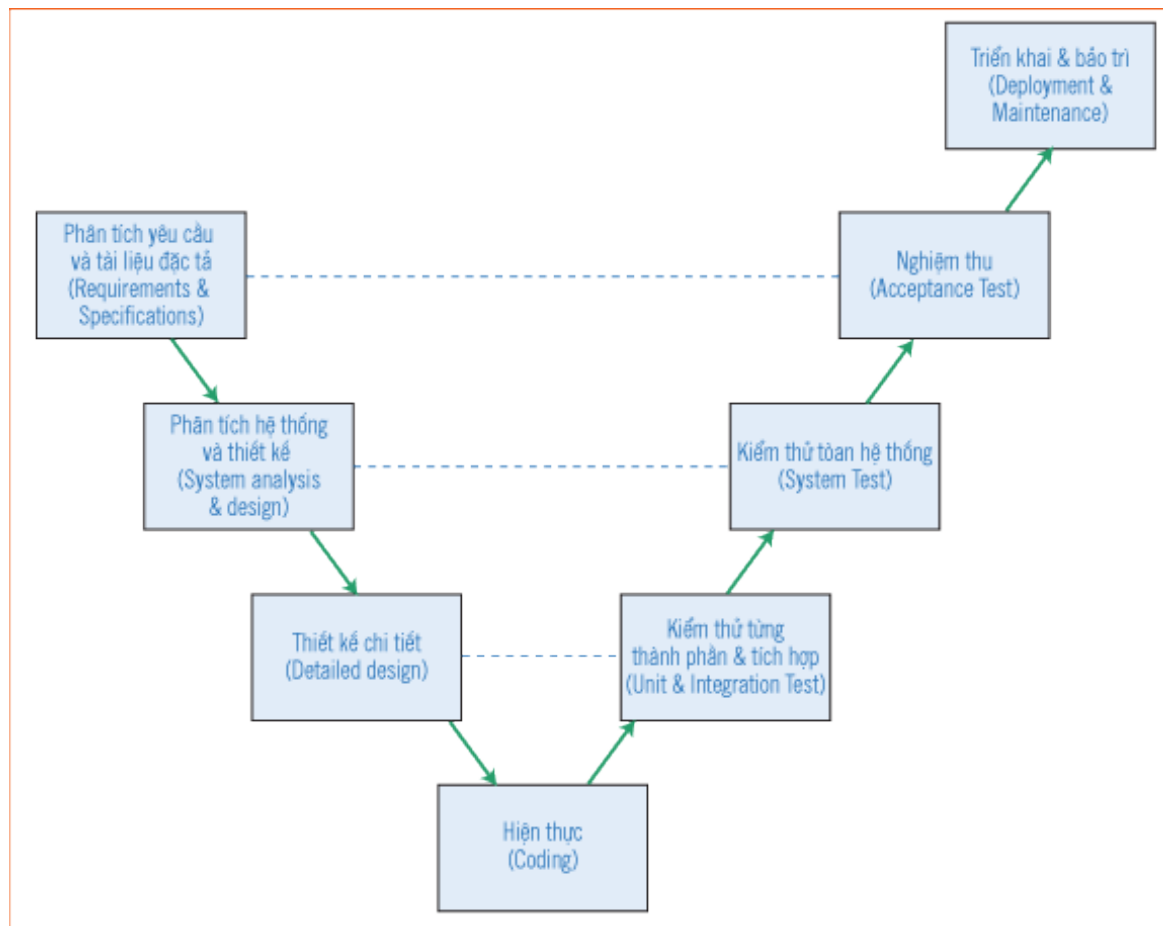
#### Nhược điểm của mô hình thác nước:

- Mô hình đòi hỏi một bản yêu cầu (requirement) đầy đủ và chính xác từ phía khách hàng
- Khách hàng chỉ được tham gia vào dự án ở giai đoạn phân tích yêu cầu và kiểm thử mà thôi
- nên được sử dụng khi đội dự án đã có kinh nghiệm, yêu cầu từ khách hàng được xác định rõ ngay từ đầu và ít có khả năng thay đổi



## 3.1 Các mô hình một phiên bản

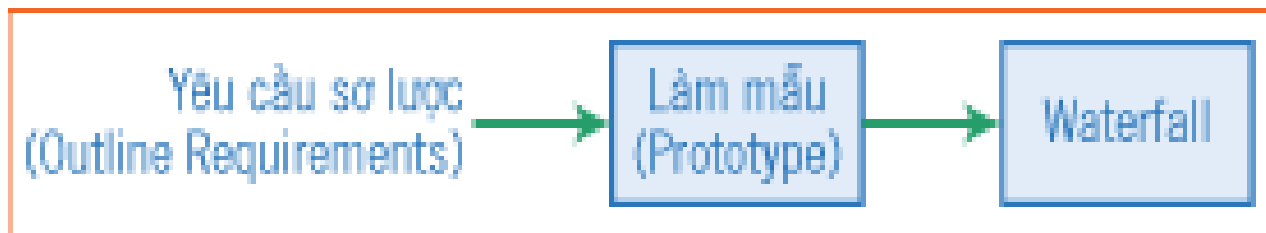
### Mô hình chữ V





## 3.1 Các mô hình nhiều phiên bản

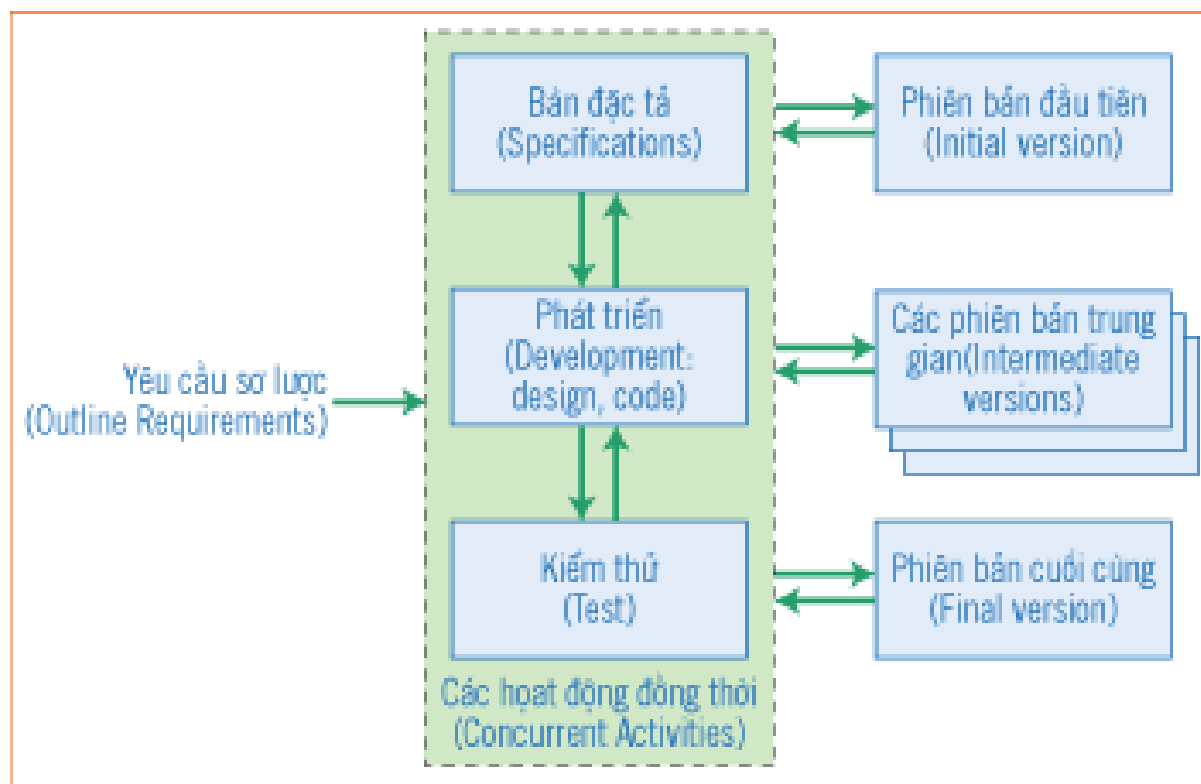
### Mô hình mẫu





## 3.1 Các mô hình nhiều phiên bản

### Mô hình tiến hóa







### 3.1 Các mô hình nhiều phiên bản

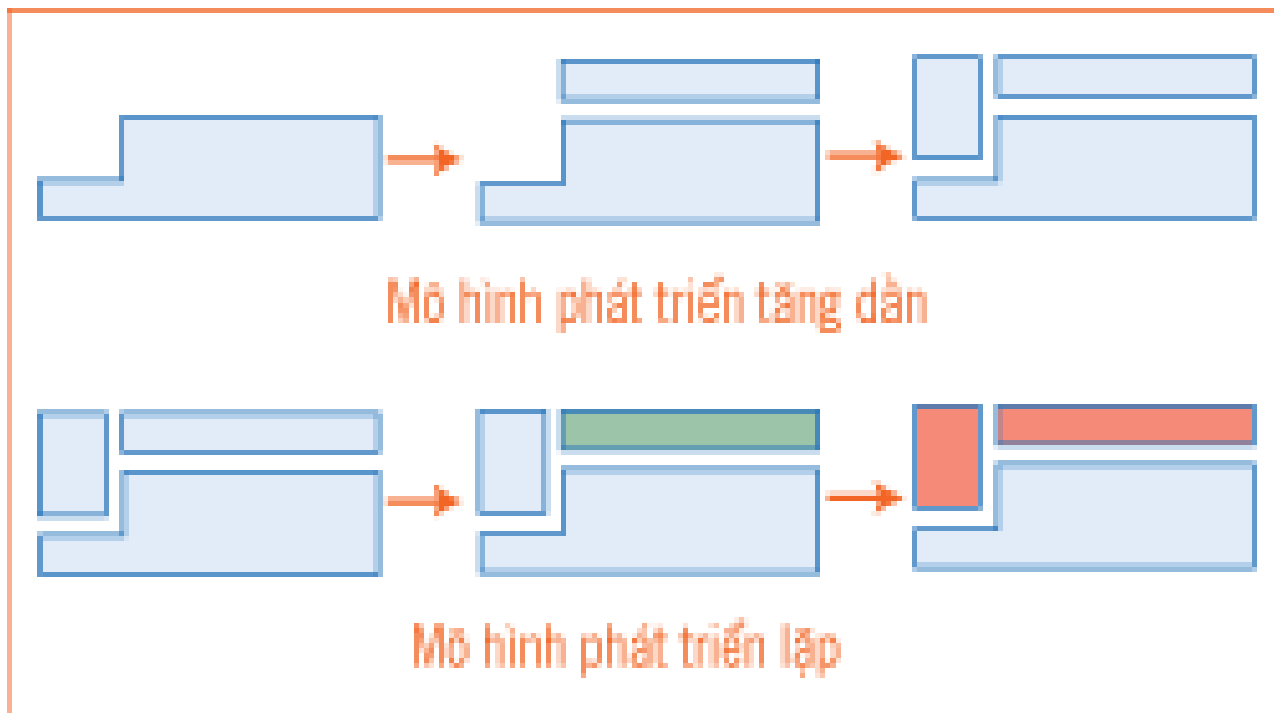
Mô hình tiến hóa thực sự cũng là một dạng dựa trên mô hình mẫu, tuy nhiên có sự khác biệt như sau:

- Mô hình tiến hóa xây dựng nhiều phiên bản mẫu liên tiếp nhau.
- Những phiên bản mẫu trước sẽ được xây dựng với mục tiêu có thể tái sử dụng trong những phiên bản sau.



### 3.1 Các mô hình nhiều phiên bản

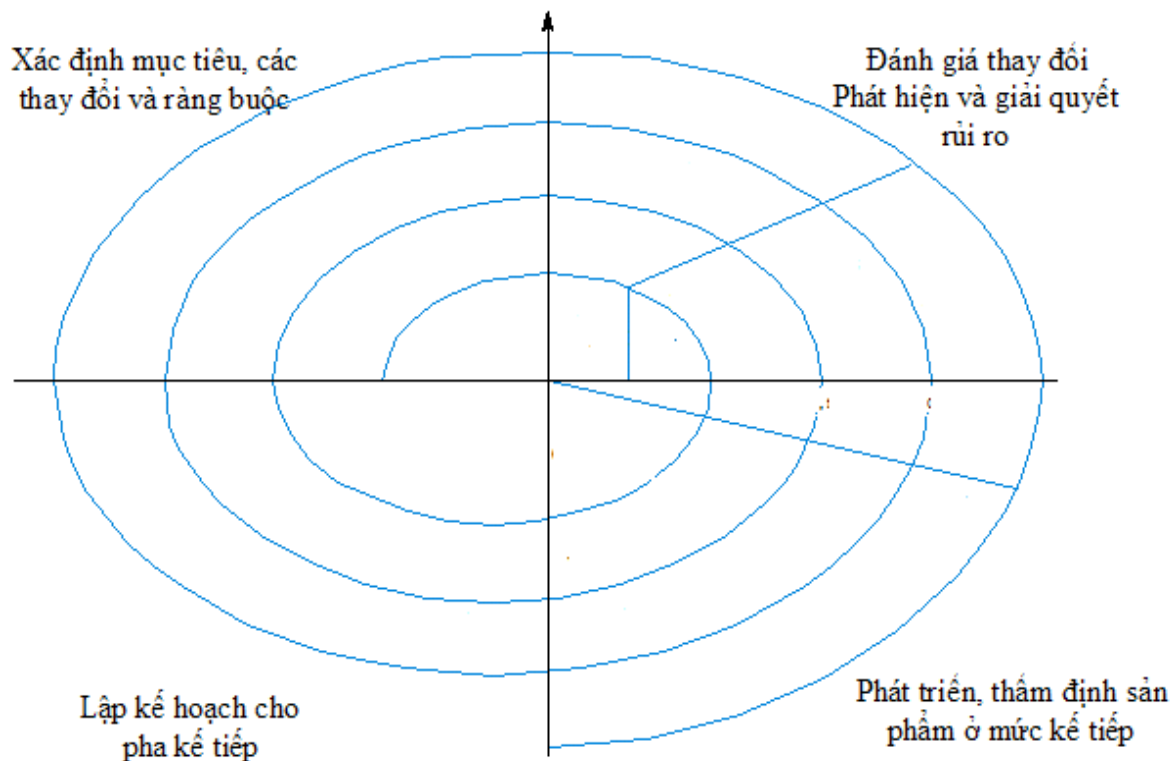
Mô hình phát triển lặp lại và tăng dần: Dựa trên mô hình tiến hóa





## 3.1 Các mô hình nhiều phiên bản

### Mô hình xoắn ốc





### 3.1 Các mô hình nhiều phiên bản

Ngoài ra còn có: Mô hình phát triển ứng dụng nhanh (RAD), Mô hình hướng đối tượng, Mô hình Công nghệ phần mềm dựa thành phần, ...