

# Kỹ thuật phần mềm ứng dụng

*Bộ môn: Điện tử & kỹ thuật máy tính  
Viện: Điện tử Viễn Thông  
Trường: ĐH Bách khoa Hà nội*

9/5/2017

# Giới thiệu

---

- *2 tín chỉ = 45 Tiết*
- *Học trên lớp*
- *Điểm = Bài tập lớn + ý thức học trên lớp*
- *Mục đích môn học:*
  - *Các khái niệm cơ bản trong kỹ thuật phần mềm (software engineering)*
  - *Các bước cơ bản xây dựng phần mềm, từ lập kế hoạch, phân tích, thiết kế cho đến bảo trì phần mềm*
  - *Có kỹ năng thực hành làm phần mềm theo nhóm*

# Nội dung

---

1. *Giới thiệu tổng quan*
2. *Các pha trong phát triển phần mềm*
3. *Cơ sở dữ liệu*
4. *Ngôn ngữ SQL*

# Tài liệu tham khảo

---

1. *Software Engineering \_a practitioner's approach*  
(5th edition, Roger Pressman)

# Chương 1. Giới thiệu tổng quan

---

1.1 Khái niệm và các giai đoạn tiến hóa phần mềm

1.2 Các ứng dụng của phần mềm

1.3 Khái niệm kỹ thuật phần mềm

1.4 Các lớp của kỹ thuật phần mềm

1.5 Tiến trình phần mềm

# 1.1 Khái niệm phần mềm

---

Máy tính:

Thực hiện các nhiệm vụ rất khác nhau bằng cách sử dụng các phần mềm khác nhau

*=> Phần mềm tạo ra sự khác biệt giữa các máy tính và cũng quyết định năng lực của máy tính.*

- Phần mềm là một cấu phần quan trọng của một hệ thống thông tin, nó là cách gọi khác của chương trình. Nó bao gồm tập lệnh để từng bước hướng dẫn máy tính làm việc nhằm chuyển đổi dữ liệu thành thông tin.

# Các giai đoạn tiến hóa phần mềm

- *1950 đến 1960: những năm đầu*
  - ✓ Phần cứng thay đổi liên tục
  - ✓ Phương thức chính là xử lý theo lô
  - ✓ Lập trình “theo bản năng”
  - ✓ Môi trường lập trình có tính chất cá nhân
- *1960 đến giữa những năm 1970*
  - ✓ Xuất hiện hệ thống đa nhiệm, đa người sd
  - ✓ Hệ thống thời gian thực
  - ✓ Xuất hiện thể hệ đầu tiên của hệ quản trị CSDL
  - ✓ Việc bảo trì phần mềm tiêu tốn công sức và tài nguyên

# Các giai đoạn tiến hóa phần mềm (tiếp)

---

- **1970 đến đầu những năm 1990**
  - ✓ **Hệ thống phân tán**
  - ✓ **Mạng toàn cục và cục bộ, liên lạc số giải thông cao**
  - ✓ **Công nghệ chế tạo các bộ vi xử lý tiến bộ nhanh**
  - ✓ **Thị trường phần cứng đi vào ổn định**
- **Thời kỳ sau 1990**
  - ✓ **Kỹ nghệ hướng đối tượng**
  - ✓ **Quy mô và độ phức tạp của những hệ thống phần mềm mới cũng tăng đáng kể**
  - ✓ **Ra ứng dụng thực tế Phần mềm trí tuệ nhân tạo**



# 1.2 Các ứng dụng của phần mềm

---

- Phần mềm hệ thống
- Phần mềm thời gian thực
- Phần mềm nghiệp vụ
- Phần mềm khoa học và kỹ thuật
- Phần mềm nhúng
- Phần mềm cho máy tính cá nhân
- Phần mềm trí tuệ nhân tạo
- Phần mềm phục vụ KTPM

# Phần mềm hệ thống

---

- Phục vụ chương trình khác,
- Giao tiếp với các thiết bị phần cứng
- Sử dụng nhiều bởi các người dùng
- Lập lịch các hoạt động đồng thời
- Chia sẻ tài nguyên,
- Giao tiếp với nhiều thiết bị ngoại vi

# Phần mềm thời gian thực

---

- Phần mềm thời gian thực là các chương trình giám sát/phân tích/điều khiển các sự kiện trong thế giới thực ngay khi nó xảy ra.
- Điển hình của phần mềm thời gian thực là các phần mềm điều khiển các thiết bị tự động

# Phần mềm thời gian thực

---

- Phần mềm thời gian thực bao gồm các thành tố:
  - Thành phần thu thập dữ liệu
  - Thành phần phân tích để biến đổi thông tin
  - Thành phần kiểm soát hoặc đưa ra đáp ứng
  - Thành phần điều phối

# Phần mềm nghiệp vụ

---

- Phần mềm nghiệp vụ có đặc trưng:
  - Phục vụ các hoạt động kinh doanh hay các nghiệp vụ của tổ chức, doanh nghiệp
  - Là lĩnh vực ứng dụng phần mềm lớn nhất
  - Là các hệ thống thông tin quản lý gắn chặt với CSDL, các ứng dụng tương tác như xử lý giao tác cho các điểm bán hàng.

# Phần mềm khoa học và kỹ thuật

- *Đặc trưng bởi số lượng thuật toán rất lớn*
- *Thường đòi hỏi phần cứng có năng lực tính toán cao*
- *Ứng dụng trong các lĩnh vực khoa học khác nhau*
- *Thiết kế có sự hỗ trợ của máy tính, các hệ thống mô phỏng, và các ứng dụng tương tác khác bắt đầu tính đến thời gian thực và các đặc điểm của phần mềm hệ thống*

# Phần mềm nhúng

---

- *Trong các thiết bị thông minh*
- *Nằm trong bộ nhớ và được sử dụng để điều khiển thiết bị*
- *Thực hiện các chức năng một cách hạn chế và đóng kín*
- *Đặc trưng của phần mềm nhúng:*
  - *Nằm trong bộ nhớ chỉ đọc*
  - *Có các đặc trưng của phần mềm thời gian thực và phần mềm hệ thống*

# Phần mềm cho máy tính cá nhân

- *Phát triển mạnh trong khoảng hơn 1 thập kỷ trở lại đây*
- *Yếu tố giao diện người-máy rất được chú trọng*
- *Một số ứng dụng:*
  - *phần mềm xử lý văn bản*
  - *phần mềm xử lý bảng tính*
  - *phần mềm xử lý đồ họa*
  - *phần mềm giải trí đa phương tiện*



# Phần mềm trí tuệ nhân tạo

---

- *Sử dụng các thuật toán phi số để giải quyết các vấn đề phức tạp*
- *Còn gọi là hệ chuyên gia, hay còn gọi là hệ thống dựa trên tri thức.*
- *Một số ứng dụng: các hệ thống nhận dạng (tiếng nói và hình ảnh), chứng minh định lý, mô phỏng và các trò chơi*

# Phần mềm phục vụ KTPM

---

- Ví dụ: chương trình dịch, phần mềm gỡ rối, các công cụ hỗ trợ phân tích thiết kế (CASE)...
- Có thể xuất hiện dưới dạng phần mềm máy tính cá nhân, phần mềm hệ thống hoặc là phần mềm nghiệp vụ.

# 1.3 Khái niệm kỹ thuật phần mềm

---

- Kỹ thuật phần mềm (Software Engineering): Là một chuyên ngành kỹ thuật mà quan tâm đến tất cả các khía cạnh của việc sản xuất phần mềm, với mục tiêu sản xuất ra các sản phẩm phần mềm đa dạng, chất lượng cao, một cách hiệu quả nhất.

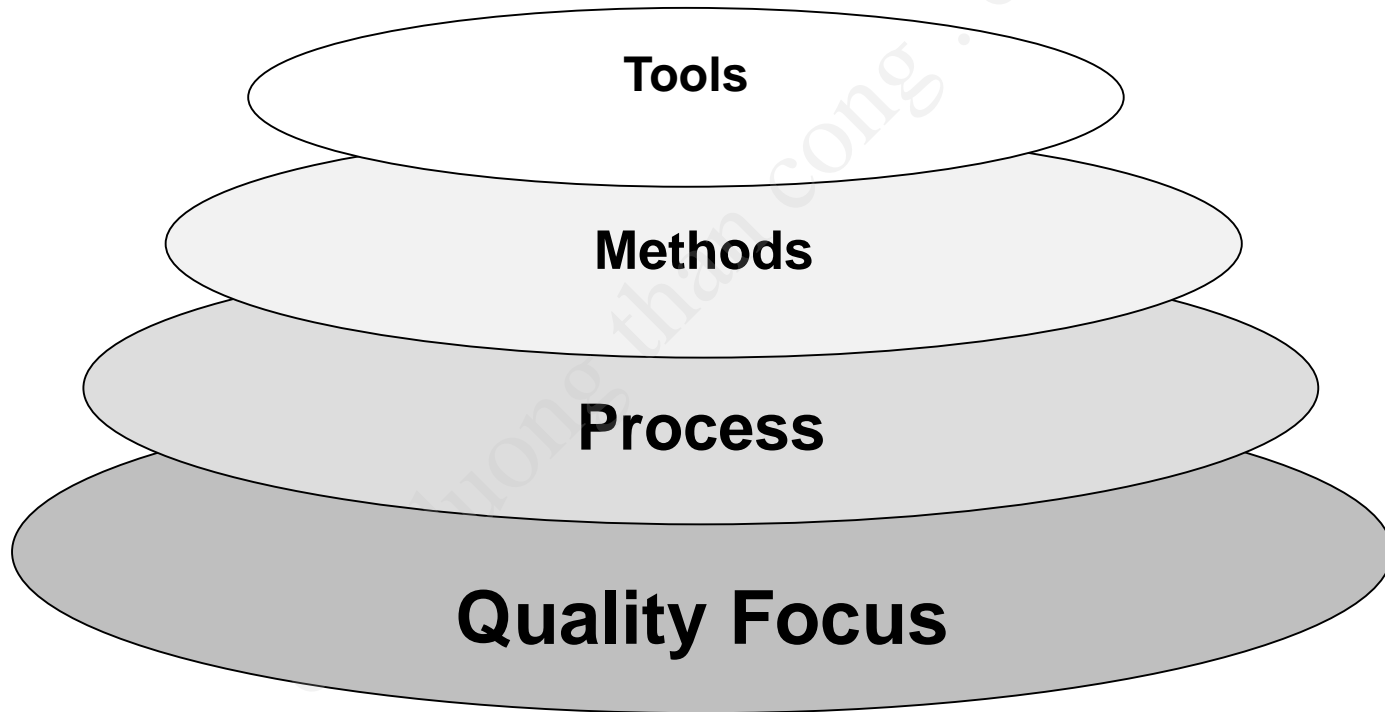
# Thành phần của phần mềm

---

- *Phần mềm (sản phẩm phần mềm), bao gồm:*
  - **Chương trình (Program):** là phần được thi hành trên máy tính
  - **Dữ liệu (Data):** gồm các cấu trúc dữ liệu, cơ sở dữ liệu lưu giữ các dữ liệu vào và ra của chương trình
  - **Tài liệu (Documentation):** tài liệu hệ thống, tài liệu người dùng

# 1.4 Các lớp của kỹ thuật phần mềm

---



# Quality focus

---

- *Đảm bảo chất lượng sản phẩm hay dịch vụ luôn là một nhiệm vụ sống còn của các công ty hay tổ chức. Do đó, mọi nền tảng công nghệ và kỹ thuật đều phải lấy việc đảm bảo chất lượng là mục tiêu hướng tới, và kỹ thuật phần mềm cũng không thể nằm ngoài mục tiêu này*

# Process

---

- *Lớp Tiến trình (process) có nhiệm vụ định nghĩa một khung các giai đoạn và các hoạt động cần thực hiện, cũng như các kết quả kèm theo chúng. Tầng này đóng vai trò nền tảng để kết nối các phương pháp, công cụ trong các bước thực hiện cụ thể, để có thể tạo ra các phần mềm có chất lượng và đúng thời hạn*

# Methods

---

- Chỉ ra cách làm về mặt kỹ thuật để xây dựng phần mềm, được sử dụng trong các bước: lập kế hoạch, ước lượng dự án, phân tích yêu cầu hệ thống và phần mềm, thiết kế cấu trúc dữ liệu, kiến trúc chương trình và thủ tục thuật toán, mã hóa kiểm thử và bảo trì



# Tools

---

- Cung cấp sự hỗ trợ tự động hay bán tự động để phát triển phần mềm theo từng phương pháp khác nhau
- Các hệ thống phần mềm hỗ trợ trong công nghệ phần mềm được gọi là CASE (computer-aided software engineering)

# 1.5 Tiến trình phần mềm

---

- *Là một dãy các giai đoạn và các hoạt động trong đó, cũng như các kết quả kèm theo. Kết quả cuối cùng chính là phần mềm cần phải xây dựng, đáp ứng được các yêu cầu của người dùng, và hoàn thành theo đúng kế hoạch về thời gian và ngân sách*
- *Có ba giai đoạn chính trong tiến trình phần mềm:*
  - *Giai đoạn định nghĩa (definition phase)*
  - *Giai đoạn phát triển (development phase)*
  - *Giai đoạn hỗ trợ (support phase)*

# Tiến trình phần mềm

---

- **Giai đoạn định nghĩa:** tập trung vào làm rõ **Cái gì**, bao gồm:
  - Thông tin gì cần xử lý, bao gồm thông tin đầu vào và đầu ra.
  - Các chức năng gì cần thực hiện.
  - Hành vi nào của hệ thống sẽ được mong đợi.
  - Các tiêu chuẩn hợp lệ nào để đánh giá được sự đúng đắn và thành công của hệ thống.

# Tiến trình phần mềm

---

- **Giai đoạn phát triển:** tập trung vào **Làm thế nào**, bao gồm:
  - Kiến trúc hệ thống (system architecture) được tổ chức thế nào.
  - Các chức năng được cài đặt và liên kết với nhau thế nào.
  - Tổ chức các cấu trúc dữ liệu, cơ sở dữ liệu thế nào.
  - Chuyển từ thiết kế sang cài đặt thế nào?
  - Việc kiểm thử sẽ được thực hiện thế nào?

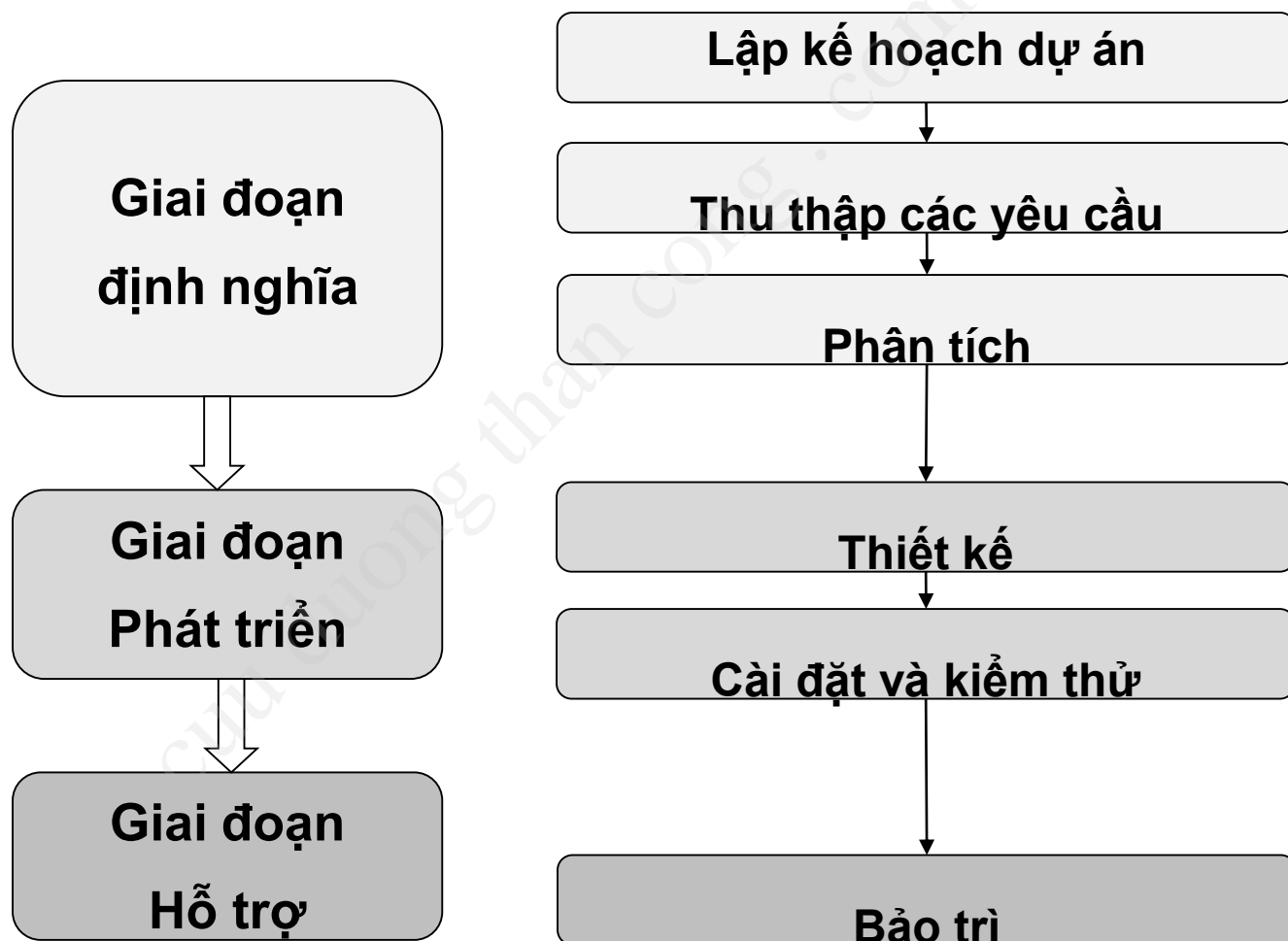
# Tiến trình phần mềm

---

- **Giai đoạn hỗ trợ:** còn gọi là giai đoạn bảo trì, tập trung vào việc ứng phó với các thay đổi của hệ thống phần mềm, bao gồm:
  - Sửa lỗi (Correction)
  - Làm thích ứng (Adaptation)
  - Nâng cấp (Upgrade)
  - Phòng ngừa, còn gọi là tái kỹ thuật phần mềm (software reengineering)

# Tiến trình phần mềm

---



# Mô hình tiến trình phần mềm

---

- **Mô hình tiến trình** (*process model*) Là một chiến lược phát triển phần mềm , bao gồm các cách thức kết hợp, sử dụng tiến trình phần mềm, cách vận dụng các phương pháp và các công cụ trong mỗi giai đoạn phát triển.
- Mô hình tiến trình cũng còn được gọi là **mẫu tiến trình** (*process paradigm*), hay **mô hình phát triển phần mềm**.

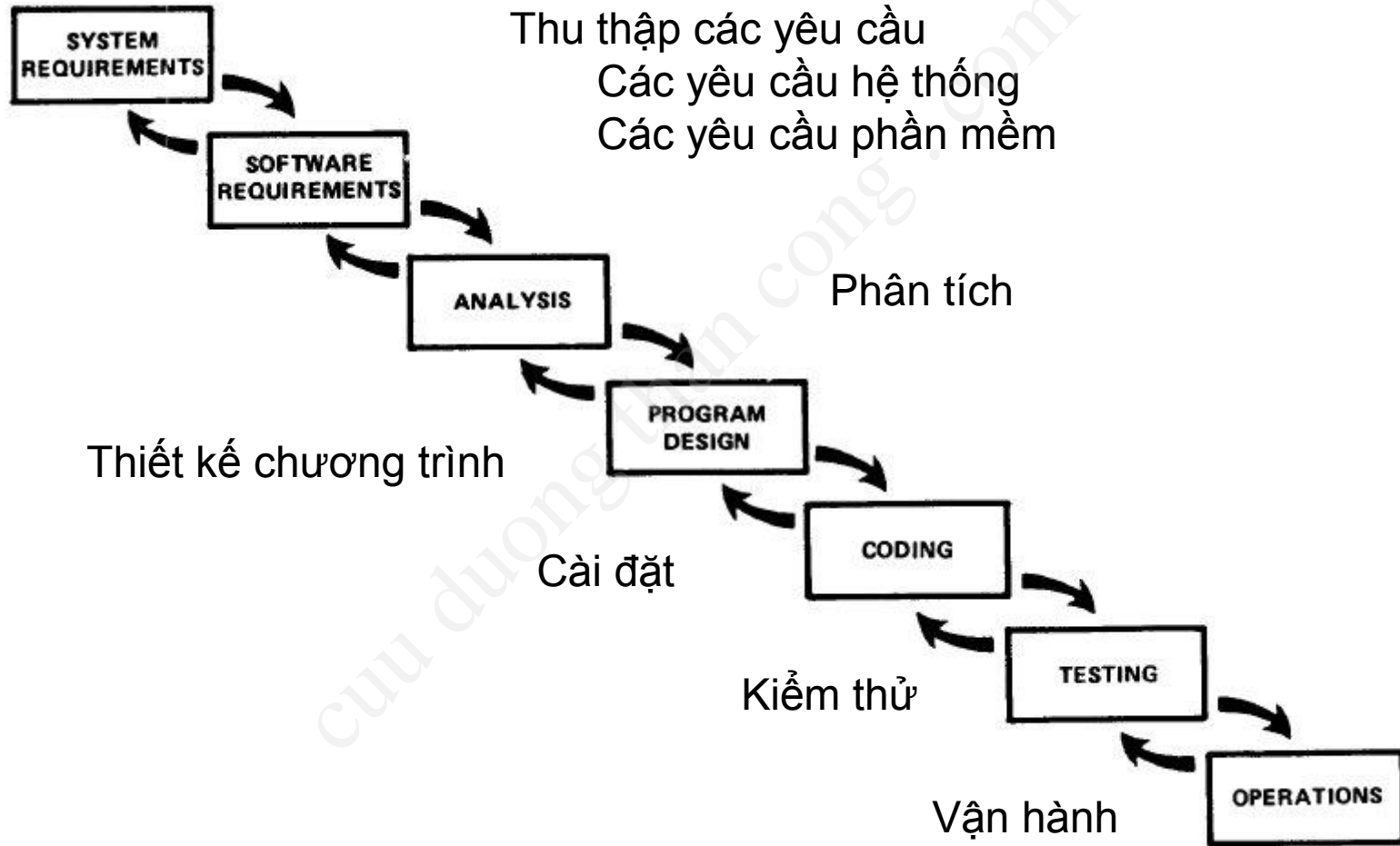
# Các mô hình tiến trình

---

- *Mô hình tuyến tính cổ điển (mô hình thác nước – Waterfall model)*
- *Mô hình bản mẫu (Prototyping model)*
- *Mô hình RAD (Rapid Application Development model)*
- *Mô hình tăng trưởng (Incremental model)*
- *Mô hình xoáy ốc (Spiral model)*



# Mô hình tuyến tính cổ điển\*



# Mô hình tuyến tính cổ điển

---

- *Mô hình này có một số đặc điểm như sau:*
  - *Các bước được tiến hành tuần tự, kết thúc bước trước thì mới thực hiện đến bước sau*
  - *Thời gian thực hiện mỗi bước thường kéo dài do phải làm thật hoàn chỉnh*
  - *Thường chỉ tiếp xúc với người dùng vào giai đoạn đầu và giai đoạn cuối. Người dùng thường không tham gia vào các bước ở giữa, như từ thiết kế, cài đặt và đến tích hợp*

# Mô hình tuyến tính cổ điển

- **Ưu điểm:**

- Đơn giản và rõ ràng
- Đóng vai trò như một mẫu tham chiếu cho các mô hình khác
- Vẫn còn được sử dụng rộng rãi cho đến nay

- **Nhược điểm:**

- Không dễ dàng cho việc thu thập đầy đủ và tường minh tất cả các yêu cầu hệ thống ngay từ ban đầu
- Người dùng phải chờ đến cuối cùng mới có được hệ thống để dùng, nên thời gian chờ đợi là khá lâu, có khi đến hàng năm. Khi đó có thể có các yêu cầu mới đã phát sinh, dễ dẫn khả năng hệ thống không còn đáp ứng được kỳ vọng của người dùng.
- Dễ dẫn đến tình trạng “**blocking states**”, tức là khi có một nhóm bị chậm tiến độ, thì các nhóm khác phải chờ, và thời gian chờ đợi thậm chí vượt quá thời gian làm việc.

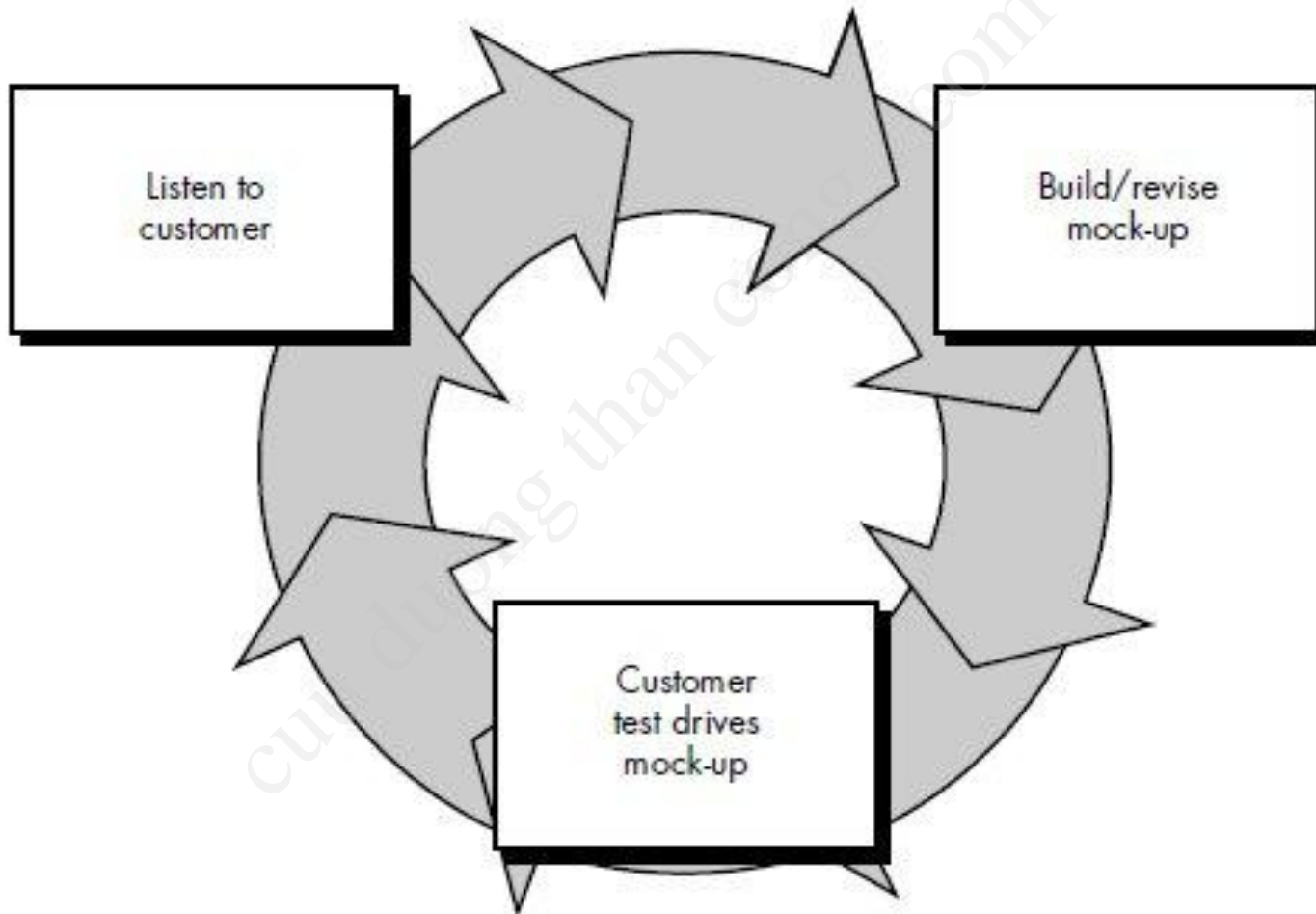
# Mô hình bản mẫu

---

- *Thông thường trong thực tế, các yêu cầu của hệ thống khó có thể xác định rõ ràng và chi tiết ngay trong giai đoạn đầu của dự án phần mềm:*
    - ***Người dùng** cũng chỉ đưa ra các mục tiêu tổng quát của phần mềm, chứ cũng chưa định rõ được một cách chi tiết các chức năng cụ thể, hay các thông tin chi tiết đầu vào, đầu ra như thế nào.*
    - ***Nhà phát triển** cũng chưa xác định rõ ràng ngay các yêu cầu, cũng như chắc chắn về chất lượng phần mềm, cũng như khả năng thỏa mãn của khách hàng*
- ***mô hình bản mẫu***

# Mô hình bản mẫu

---



# Mô hình bản mẫu

---

*Gồm các giai đoạn:*

- **Thu thập các yêu cầu** (*requirements gathering*): khách hàng và nhà phát triển sẽ gặp nhau để xác định ra các mục tiêu tổng thể của phần mềm. Sau đó họ sẽ định ra phần nào đã rõ, phần nào cần phải định nghĩa thêm.
- **Thiết kế nhanh** (*quick design*): thiết kế này tập trung vào những phần mà khách hàng có thể nhìn thấy được (giao diện, các dữ liệu vào, ra). Sau đó, từ thiết kế này, một bản mẫu sẽ được xây dựng.
- **Kiểm tra và đánh giá bản mẫu**: Bản mẫu này sẽ được dùng để cho phép người dùng đánh giá, nhằm làm rõ hơn các yêu cầu của họ. Đồng thời, thông qua bản mẫu, người phát triển hệ thống cũng hình dung cụ thể hơn về những yêu cầu của khách hàng, cũng như khả năng cài đặt và hiệu quả hoạt động của hệ thống.

# Mô hình bản mẫu

---

- **Ưu điểm:**

- Cho phép người dùng xác định yêu cầu của mình rõ ràng và cụ thể hơn, đồng thời nhà phát triển cũng nắm được chính xác hơn các yêu cầu đó.
- Cả người dùng và nhà phát triển thường đều thích mô hình này, do người dùng luôn cảm nhận được hệ thống thực sẽ như thế nào, và nhà phát triển cũng luôn có cái để xây dựng và dần hoàn thiện.

- **Nhược điểm:**

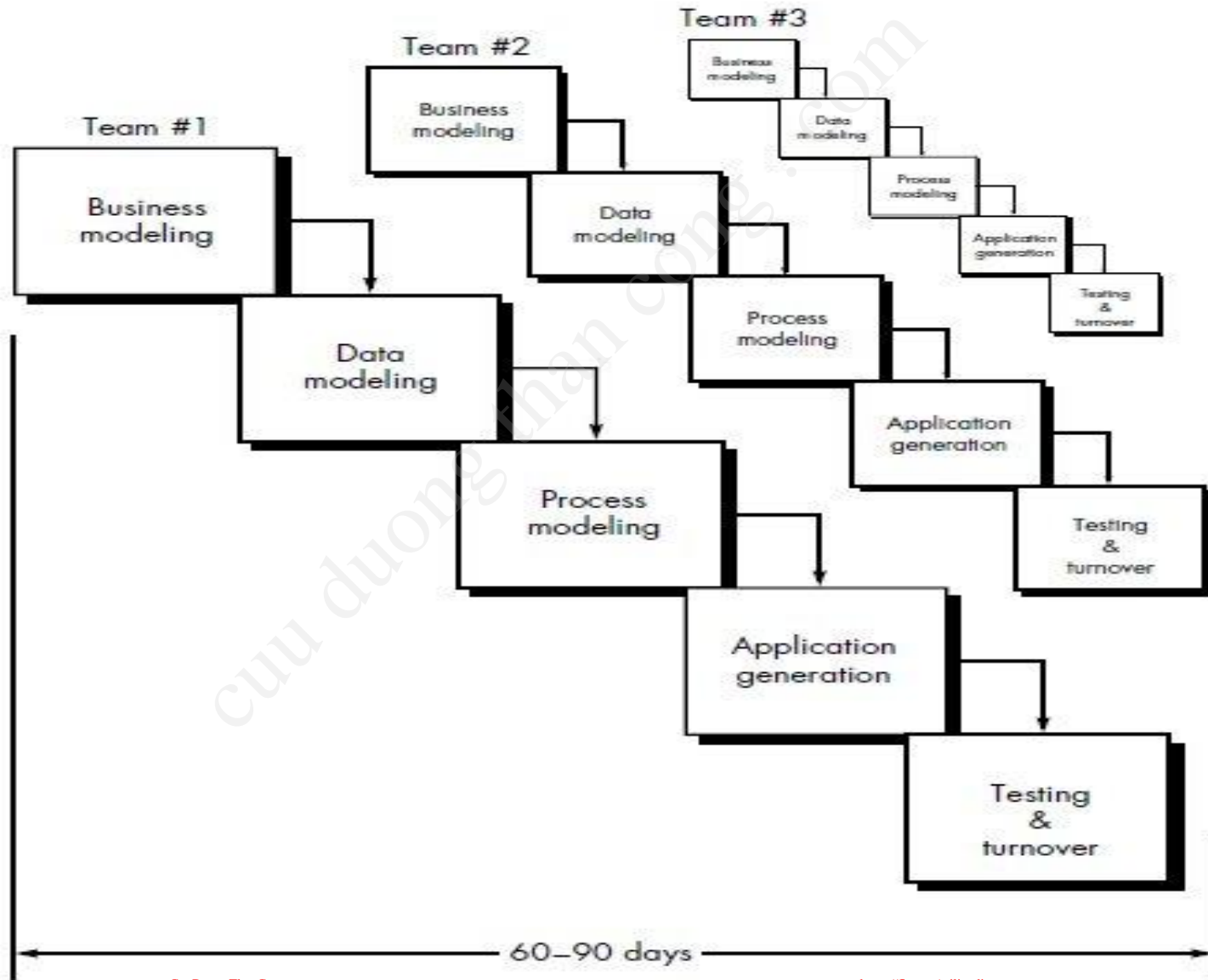
- Để có được bản mẫu nhanh, việc thiết kế cũng được làm nhanh, nên thường được làm không cẩn thận. Điều này dễ dẫn đến các thiết kế có tính chấp vá, không có cái nhìn tổng thể và dài hạn.
- Việc làm bản mẫu nhanh cũng thường kéo theo việc lựa chọn các công cụ cài đặt vội vàng, không cẩn thận, (như ngôn ngữ lập trình, hệ quản trị cơ sở dữ liệu, v.v). Điều này sẽ ảnh hưởng đến các giai đoạn phát triển sau khi quy mô và yêu cầu của hệ thống ngày càng lớn lên

# Mô hình RAD

- Là mô hình tiến trình phát triển phần mềm tăng trưởng, nhưng nhấn mạnh vào chu trình phát triển phần mềm có thời gian rất ngắn. Mô hình này gồm các giai đoạn:
  - **Mô hình hóa nghiệp vụ (Business modeling)**: mô hình hóa các luồng thông tin nghiệp vụ giữa các chức năng nghiệp vụ
  - **Mô hình hóa dữ liệu (Data modeling)**: từ các thông tin nghiệp vụ, các thực thể dữ liệu, các thuộc tính của chúng, và các liên kết giữa các thực thể này sẽ được xác định và được mô hình hóa.
  - **Mô hình hóa xử lý (Process modeling)**: Mô tả các chức năng xử lý trên các đối tượng dữ liệu đã được xác định ở giai đoạn trên.
  - **Sản sinh ứng dụng (Application generation)**: RAD sử dụng các kỹ thuật công nghệ phần mềm thế hệ thứ 4, cho phép dễ dàng sản sinh mã chương trình từ các đặc tả và thiết kế trừu tượng. Các kỹ thuật này cũng cho phép tái sử dụng các thành phần chương trình có sẵn (kết hợp mô hình Component-based development).
  - **Kiểm thử và bàn giao (Testing and turnover)**: phần ứng dụng đã xây dựng sẽ được kiểm tra và bàn giao cho bên tích hợp hệ thống.



# Mô hình RAD



# Mô hình RAD

---

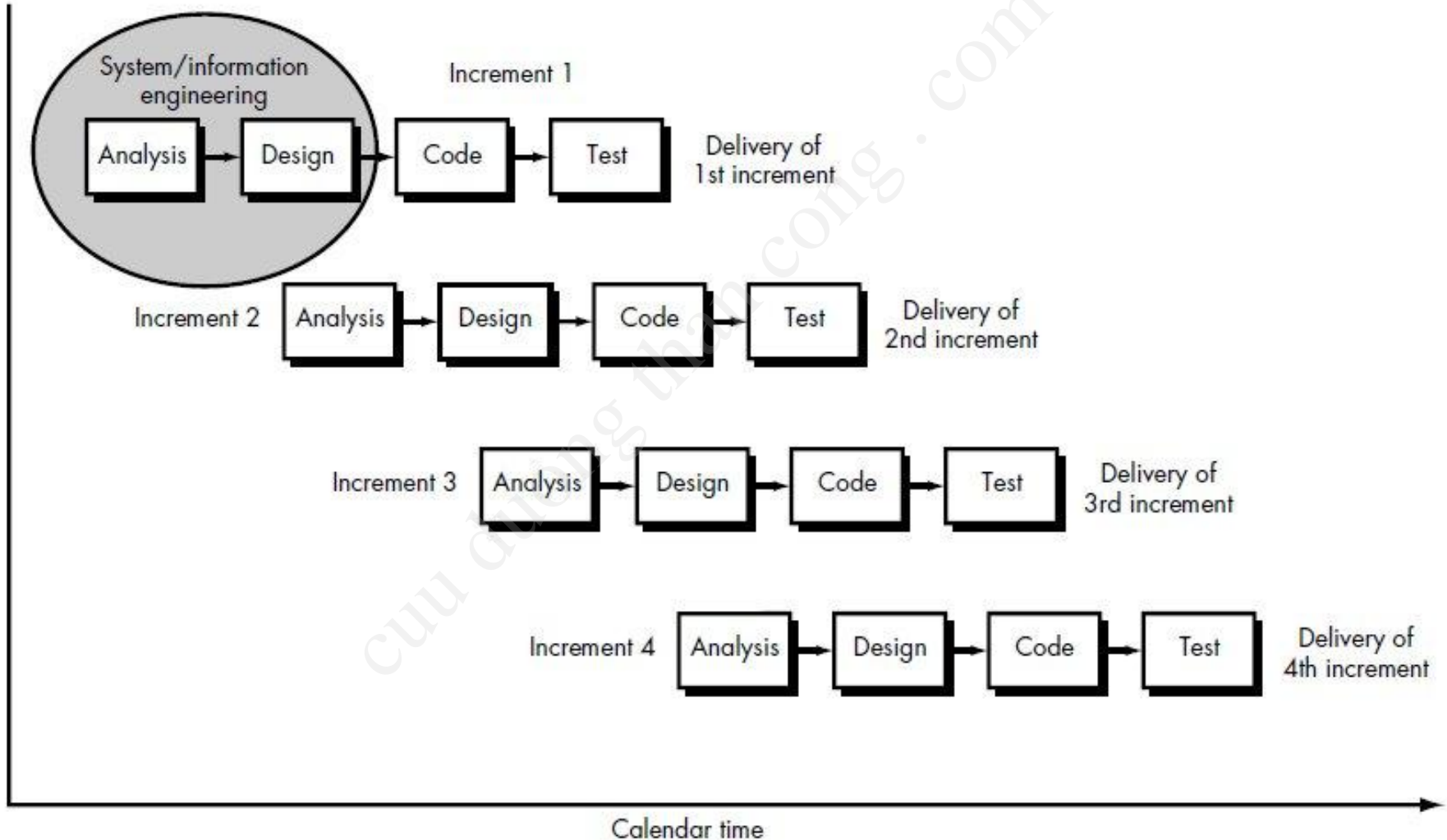
- *Ưu điểm:*
  - *Tận dụng các công nghệ mới trong phát triển hệ thống, cho phép hoàn thành hệ thống trong thời gian ngắn hơn đáng kể.*
  - *Khuyến khích việc tái sử dụng các thành phần của chương trình*
- *Nhược điểm*
  - *Không phù hợp với các phần mềm mà không có sự phân chia modul rõ ràng,*
  - *Đòi hỏi tài nguyên và chi phí phát triển cao như số lượng nhân lực nhiều, công cụ CASE thế hệ 4 đắt tiền*

# Mô hình tăng trưởng

---

- Là sự kết hợp của mô hình tuyến tính và triết lý lặp lại của mô hình bản mẫu
- Phần mềm được chia thành các **phần tăng trưởng** (*increment*), trong đó mỗi phần là một sản phẩm hoàn chỉnh (đã chạy được và có thể bàn giao cho người dùng). Đồng thời phần tăng trưởng sau sẽ bổ sung thêm tính năng còn thiếu trong những phần trước

# Mô hình tăng trưởng



# Mô hình tăng trưởng

---

## ■ **Ưu điểm**

- *Kết hợp được các ưu điểm của các mô hình tuyến tính và làm bản mẫu*
- *Rất phù hợp khi số lượng nhân viên hạn chế, và người dùng có đòi hỏi phải sớm có hệ thống thử nghiệm*

## ■ **Nhược điểm**

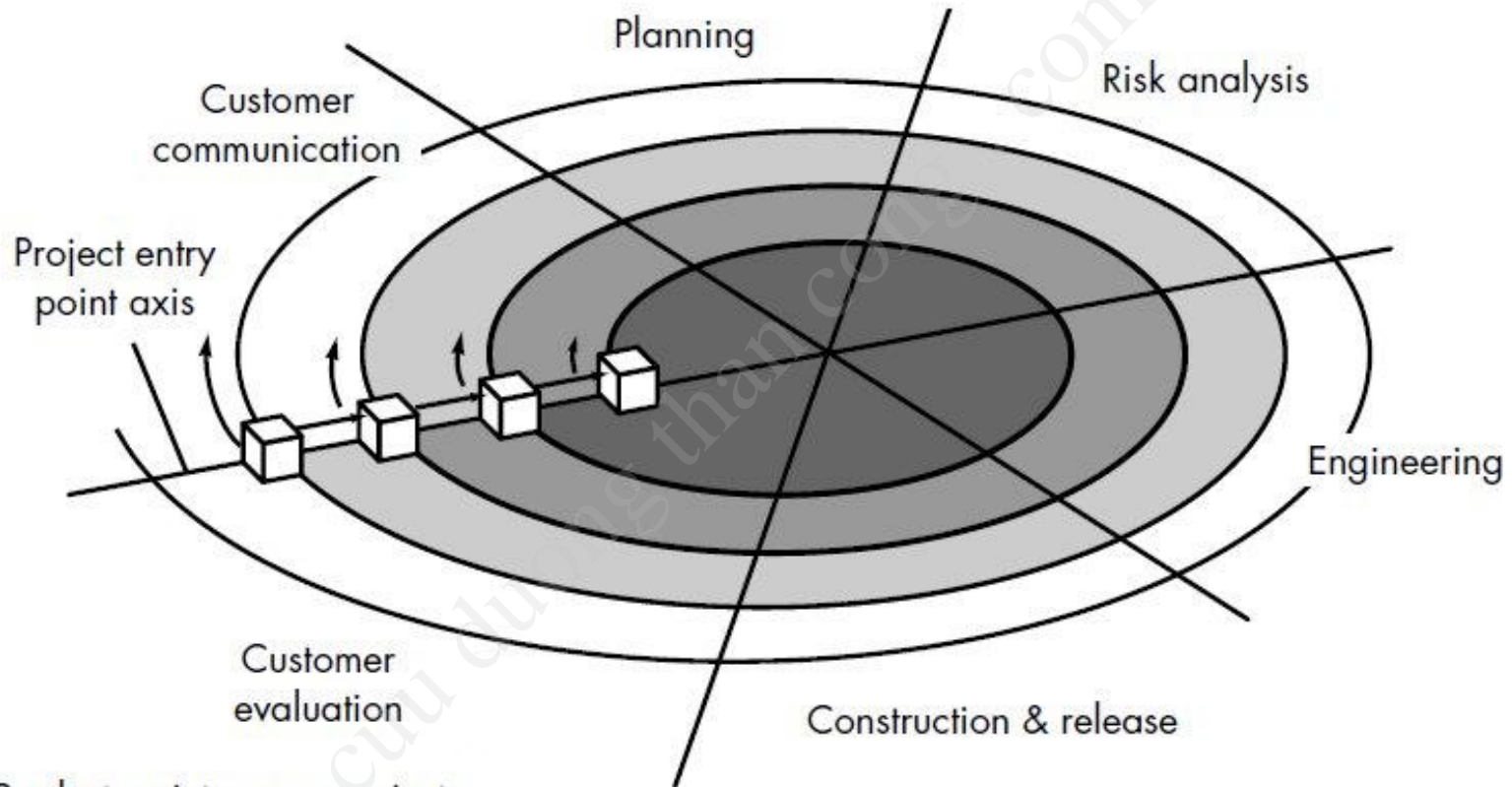
- *Việc gấp gáp đưa ra các thành phần tăng trưởng cũng có thể gây ra sự manh mún trong phân tích và thiết kế*
- *Khó khăn trong việc đảm bảo tính tương thích (compatibility) giữa các thành phần tăng trưởng.*

# Mô hình xoáy ốc

---

- *Cũng là một mô hình tiến hóa kết hợp đặc tính lặp lại của mô hình bản mẫu và tính hệ thống của mô hình thác nước cổ điển*
- *Mô hình này cũng cho phép tạo ra một dãy các phiên bản tăng trưởng (incremental release). Tuy nhiên khác với mô hình tăng trưởng, các phiên bản đầu tiên của mô hình xoáy ốc thường chỉ là các mô hình trên giấy hoặc bản mẫu (prototype). Đến các phiên bản sau thì mới là các bản chạy được và càng ngày càng hoàn chỉnh.*

# Mô hình xoáy ốc



- Product maintenance projects
- Product enhancement projects
- New product development projects
- Concept development projects

# Mô hình xoáy ốc

---

- *Mô hình này phân chia thành các giai đoạn, được gọi là các **vùng nhiệm vụ** (task regions).*
- *Số lượng vùng nhiệm vụ có thể thay đổi, và thường có từ 3 cho đến 6 vùng.*
- *Mỗi vùng lại bao gồm **một tập các nhiệm vụ** (set of tasks), và số lượng cũng thay đổi tùy theo tính chất của dự án.*



# Mô hình xoáy ốc

---

- *Ưu điểm:*

- *Linh hoạt, dễ thích ứng với các loại phần mềm và các nhu cầu sử dụng khác nhau, nhất là các phần mềm quy mô lớn*
- *Có khá đầy đủ các bước trong tiến trình phát triển, nhất là việc chú trọng phân tích tính rủi ro (risk) của phần mềm cả về mặt kỹ thuật và quản lý*

- *Hạn chế:*

- *Phức tạp, cần khá nhiều thời gian để hiểu và vận dụng được một cách hiệu quả*
- *Khó khăn trong việc quản lý nhiều chu trình phát triển*