

# Chapter 2

---

## ■ Mô hình tiến trình

*Slide được soạn cùng quyển sách:*

*Software Engineering: A Practitioner's Approach, 7/e*

**bởi Roger S. Pressman**

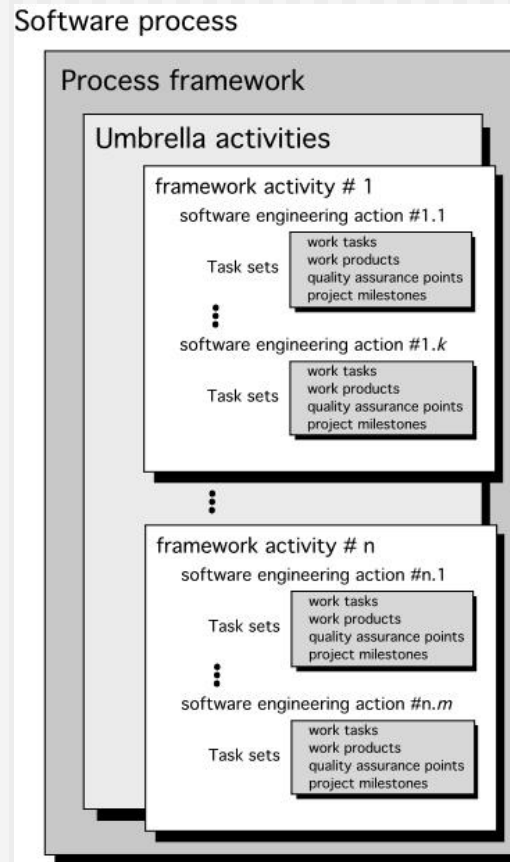
Slides copyright © 1996, 2001, 2005, 2009 by Roger S. Pressman

***Chỉ được dùng với mục đích giáo dục phi lợi nhuận***

May be reproduced ONLY for student use at the university level when used in conjunction with *Software Engineering: A Practitioner's Approach, 7/e*. Any other reproduction or use is prohibited without the express written permission of the author.

All copyright information MUST appear if these slides are posted on a website for student use.

# Ví dụ về mô hình sinh tiến trình

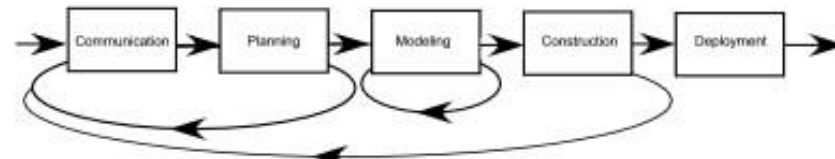


These slides are designed to accompany Software Engineering: A Practitioner's Approach, 7/e (McGraw-Hill, 2009). Slides copyright 2009 by Roger Pressman.

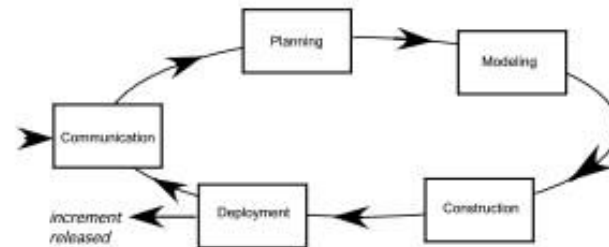
# Luồng tiến trình



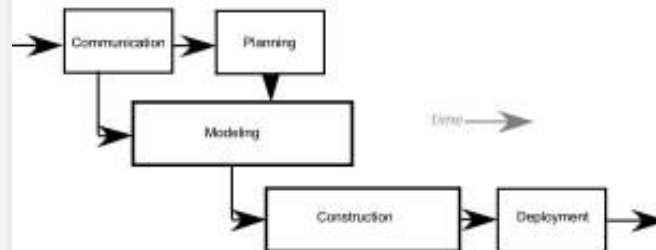
(a) linear process flow



(b) iterative process flow



(c) evolutionary process flow



(d) parallel process flow

These slides are designed to accompany Software Engineering: A Practitioner's Approach, 7/e (McGraw-Hill, 2009). Slides copyright 2009 by Roger Pressman.

# Xác định tập các tác vụ

---

- Một công việc thực tế được định nghĩa bởi một tổ hợp các tác vụ, nhằm hoàn thành mục tiêu, đối tượng đề ra của một hành động thiết kế phần mềm nào đó.
  - Một tập hợp các tác vụ phải hoàn thành.
  - Một tập hợp các sản phẩm cần được sản xuất
  - Một tập hợp các bộ lọc đảm bảo chất lượng cần được áp dụng

# Mẫu tiến trình

---

## ■ Một mẫu tiến trình:

- Xác định một vấn đề liên quan đến tiến trình làm việc mà đã xảy ra trong quá trình thiết kế phần mềm,
  - xác định môi trường mà vấn đề đã xảy ra,
  - gợi ý một hoặc nhiều lời giải đã được chứng minh cho vấn đề.
- Được biểu diễn bởi các thành phần mang tính tổng quát, một mẫu tiến trình cung cấp cho bạn một template [Amb98]—một phương pháp nhất quán để mô tả các lời giải của vấn đề từ bên trong nội dung của quá trình thiết kế phần mềm.

# Các kiểu mẫu tiến trình

---

- *Mẫu trạng thái*—Xác định một vấn đề cùng với một hoạt động nền tảng cho tiến trình.
- *Mẫu tác vụ*—Xác định một vấn đề cùng với một hành động thiết kế phần mềm hoặc một tác vụ công việc thích hợp với thực tế.
- *Mẫu pha*—Xác định một chuỗi các hoạt động nền tảng xảy ra cùng với tiến trình, thậm chí khi các luồng hoạt động tổng thể được lặp đi lặp lại trong thực tế.

# Các phương pháp đánh giá và cải tiến cho tiến trình.

- **Standard CMMI Assessment Method for Process Improvement (SCAMPI)** — cung cấp một mô hình đánh giá theo 5 bước chặt chẽ: chuẩn bị, chuẩn đoán, thiết lập, hành động và học hỏi.
- **CMM-Based Appraisal for Internal Process Improvement (CBA IPI)**—cung cấp một kĩ thuật chuẩn đoán để đánh giá chất lượng của một tổ chức phần mềm; sử dụng SEI CMM là nền tảng cho sự đánh giá [Dun01]
- **SPICE—Chuẩn SPICE (ISO/IEC15504)** định nghĩa một tập các yêu cầu cơ việc đánh giá tiến trình phần mềm. Mục đích của chuẩn này là giúp đỡ các tổ chức trong việc phát triển một đối tượng ước lượng cho sự hiệu quả của bất kì tiến trình định nghĩa phần mềm. [ISO08]
- **ISO 9001:2000 cho phần mềm**—Một tiêu chuẩn tổng quát áp dụng cho bất kỳ tổ chức nào muốn cải thiện chất lượng tổng thể của sản phẩm, hệ thống hoặc dịch vụ nào đó mà họ cung cấp. Vì vậy, chuẩn này có thể dùng được trực tiếp cho các tổ chức và công ty phần mềm.[Ant06]

# Mô hình ra lệnh

---

- Mô hình ra lệnh ủng hộ cách tiếp cận theo thứ tự trong việc thiết kế phần mềm.

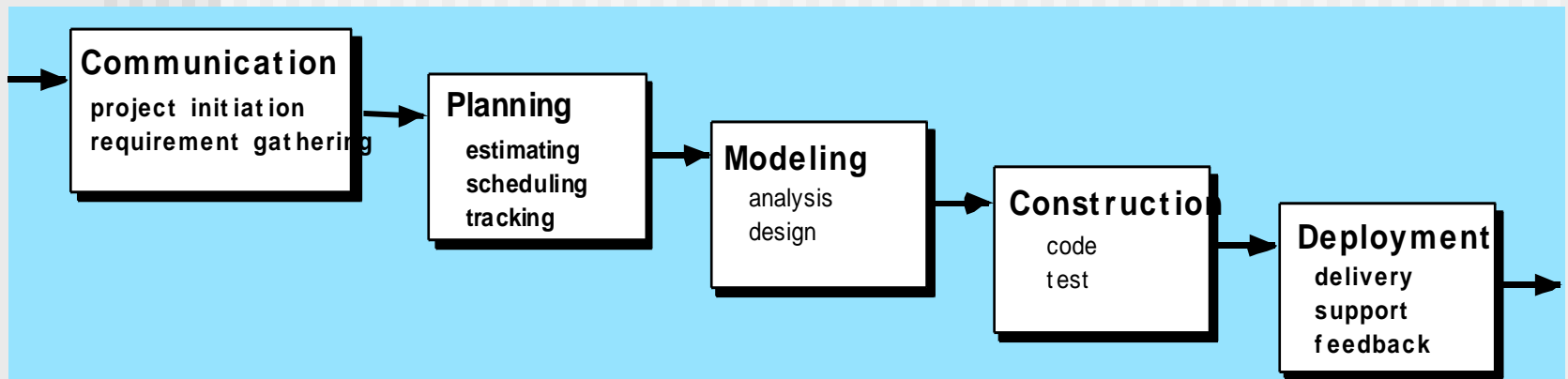
*Việc này dẫn đến một số câu hỏi:*

- Nếu mô hình ra lệnh đấu tranh cho cấu trúc và thứ tự, liệu có phải nó không thích hợp cho một thế giới mà phần mềm phát triển nhanh chóng ?
- Hơn nữa, nếu chúng ta loại bỏ mô hình tiến trình cổ điển, và thay thế chúng bởi cái gì đó ít tính cấu trúc hơn, có phải chúng ta đang không thể đạt được sự phối kết hợp trong công việc phần mềm ?

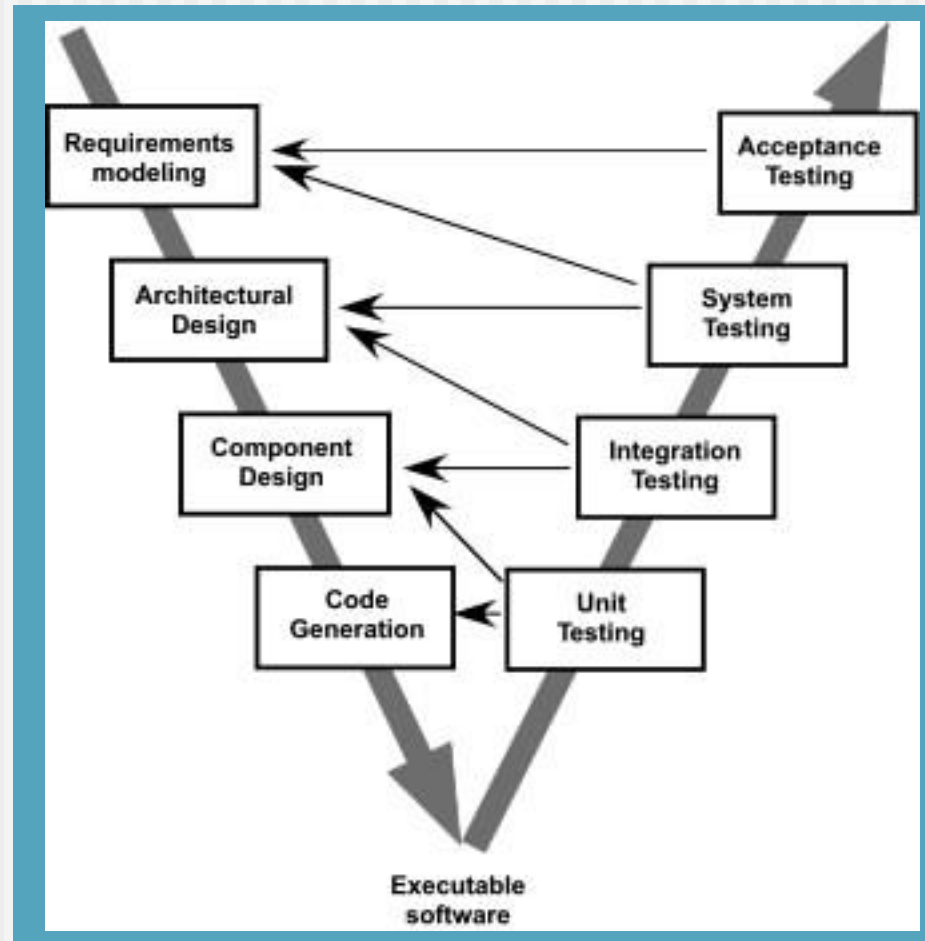


# Mô hình thác nước

---

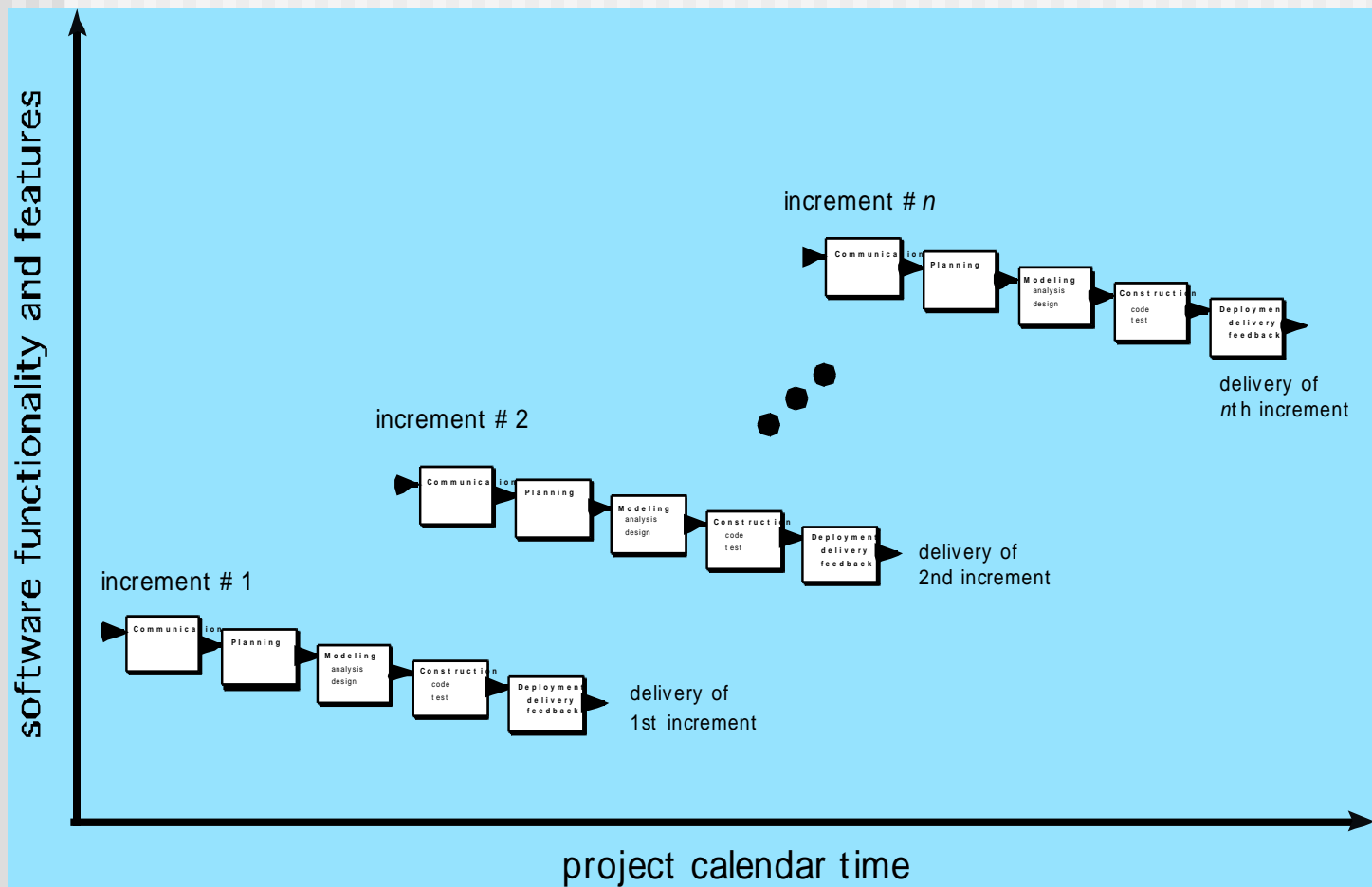


# Mô hình chữ V



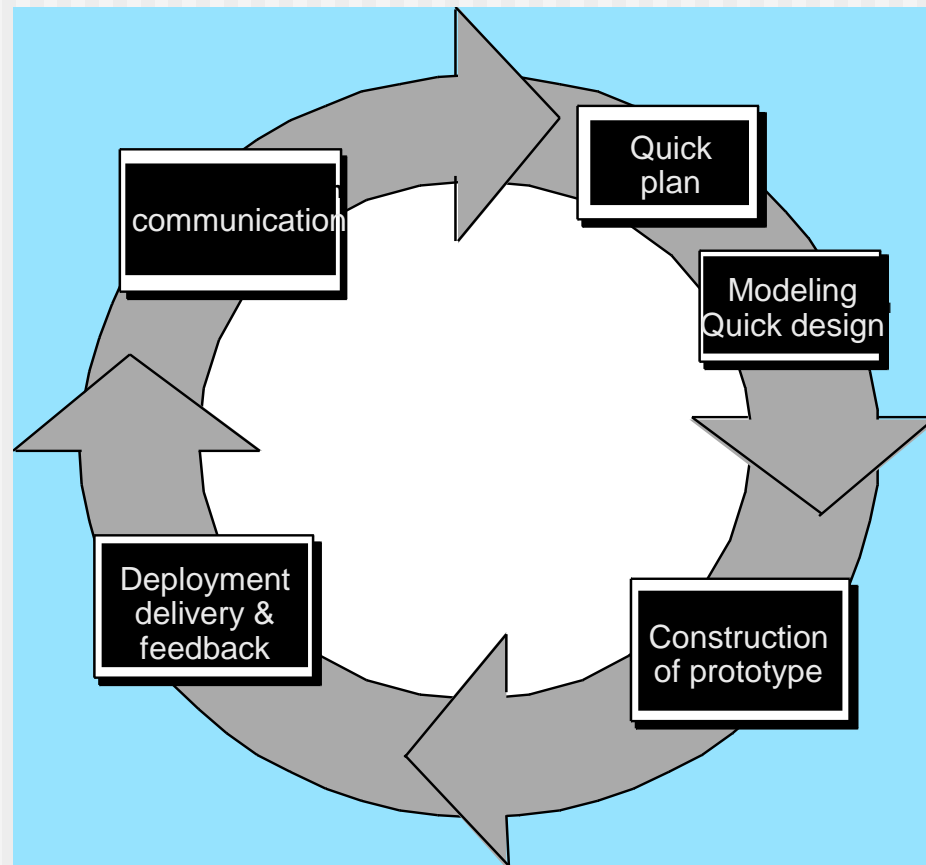
These slides are designed to accompany Software Engineering: A Practitioner's Approach, 7/e (McGraw-Hill, 2009). Slides copyright 2009 by Roger Pressman.

# Mô hình tăng dần

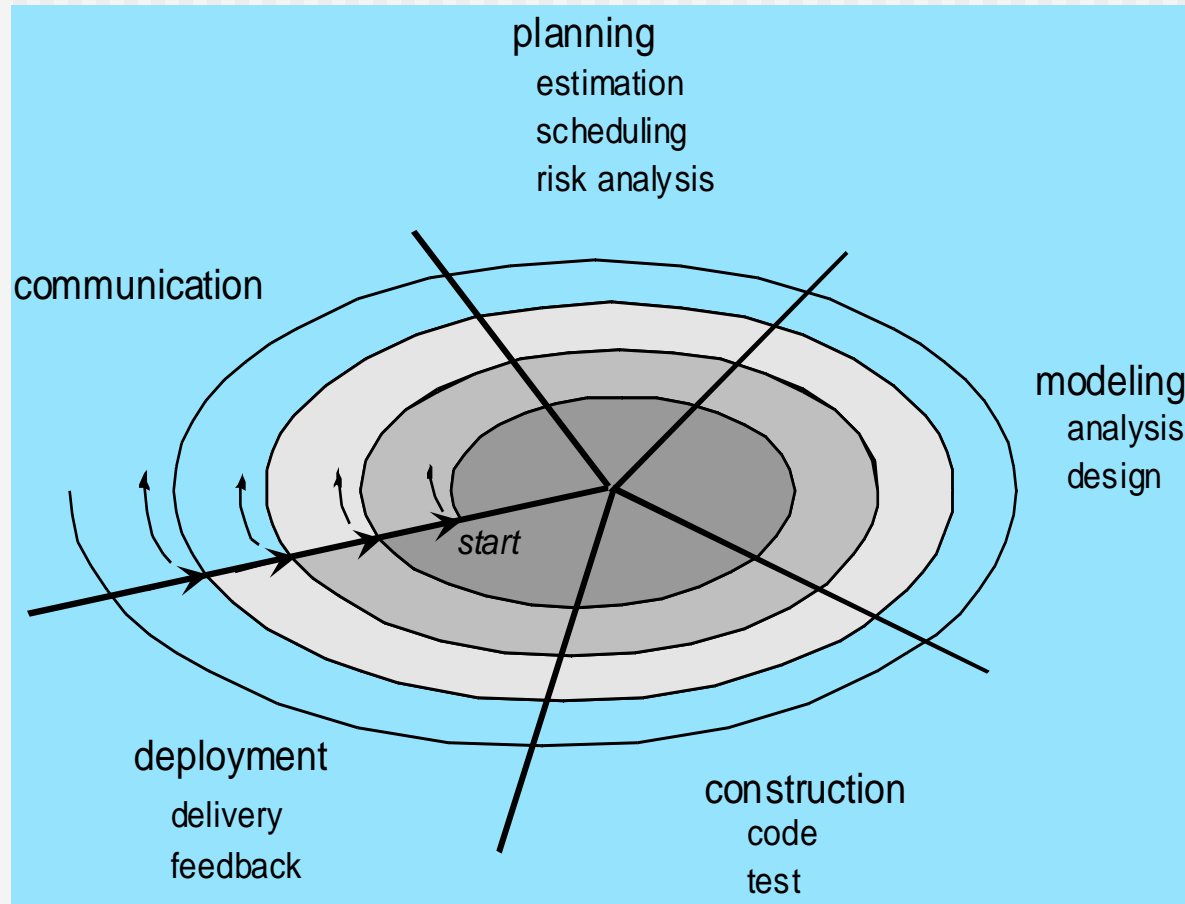


# Mô hình mạng tính cách mạng: Prototyping

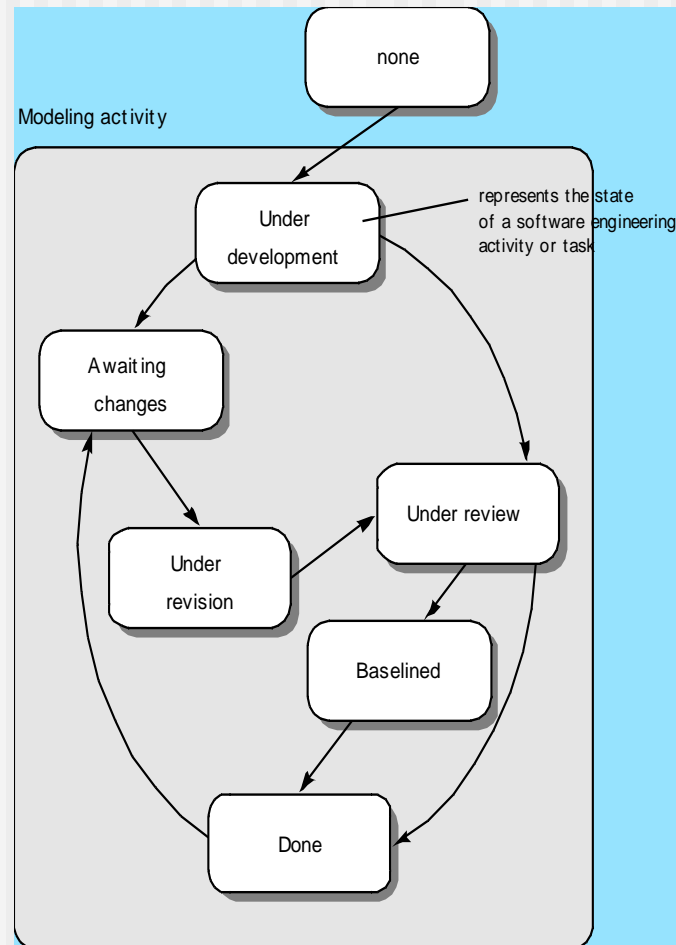
---



# Mô hình mạng tính cách mạng: Mô hình xoắn ốc



# Mô hình mạng tính cách mạng: Mô hình đồng thời



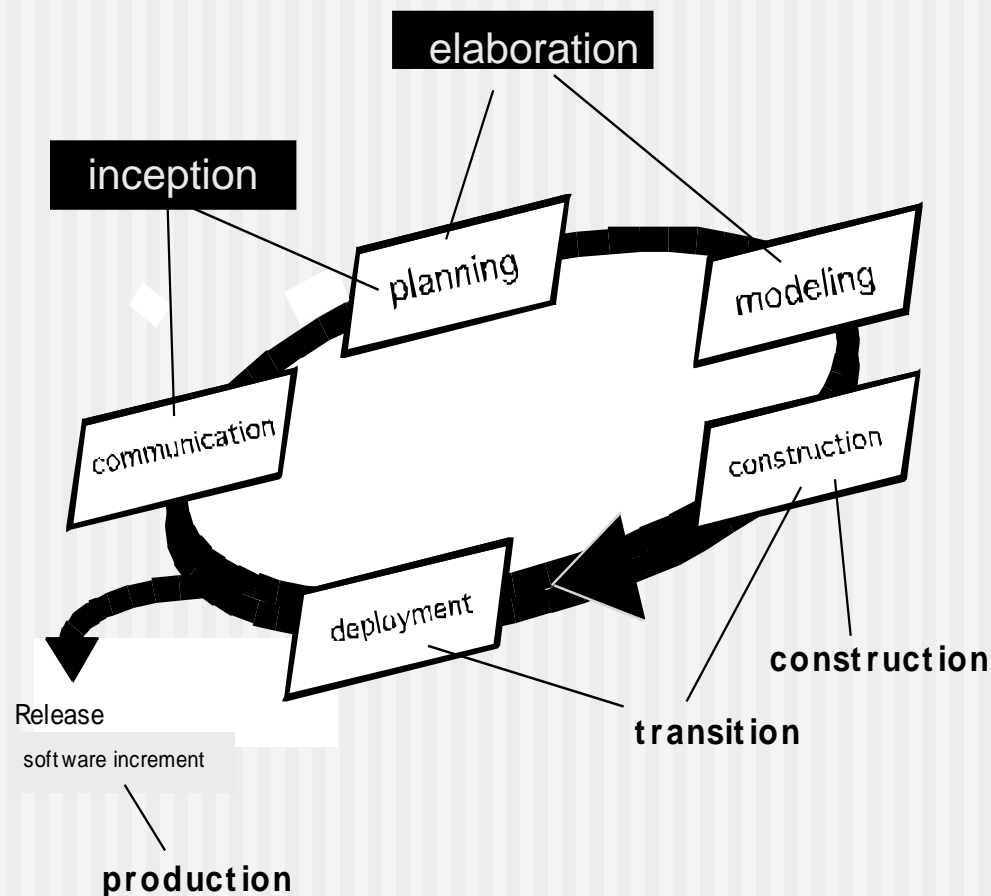
These slides are designed to accompany Software Engineering: A Practitioner's Approach, 7/e (McGraw-Hill, 2009). Slides copyright 2009 by Roger Pressman.

# Một số mô hình khác

---

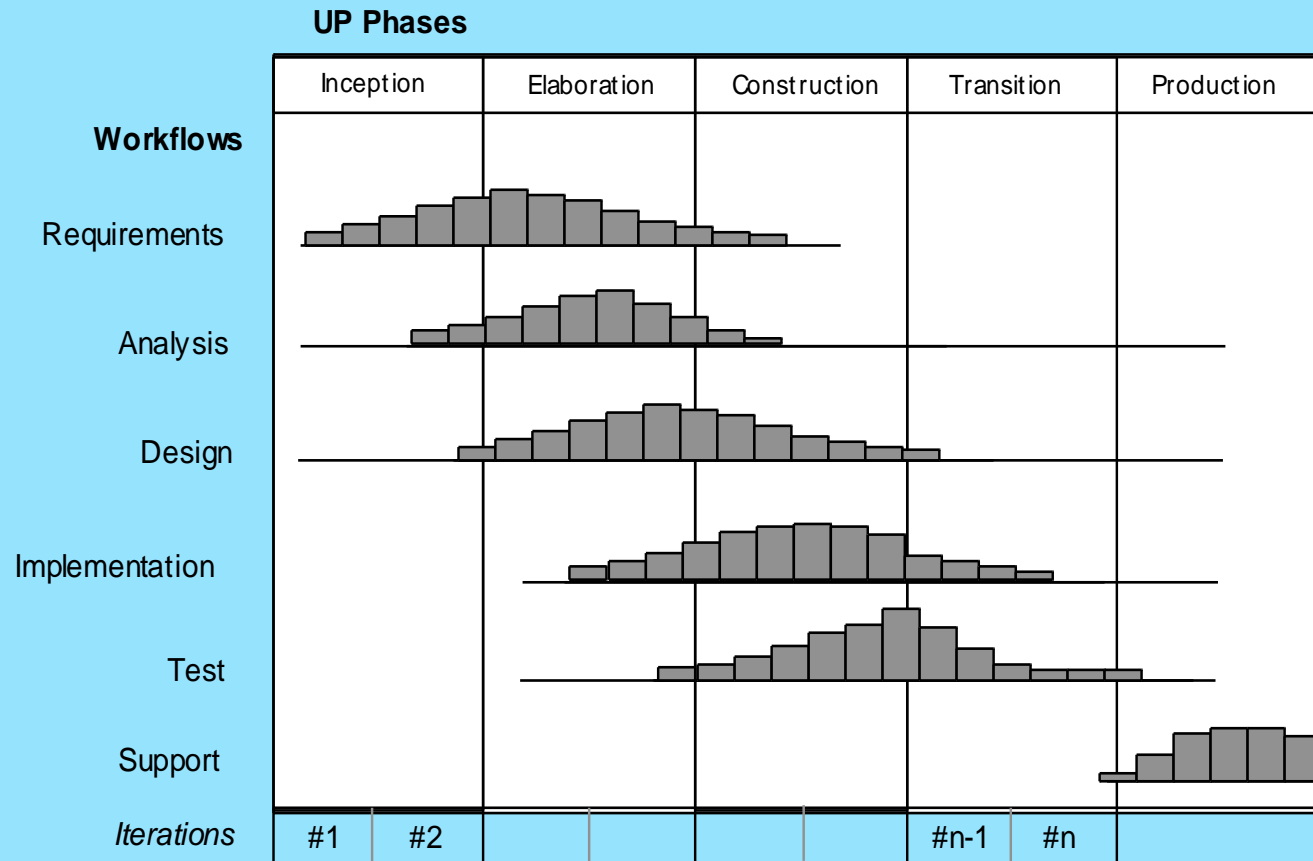
- **Component based development**—tiến trình được áp dụng khi sử dụng lại là một đối tượng phát triển.
- **Formal methods**—nhấn mạnh đặc điểm toán học của yêu cầu.
- **AOSD**—cung cấp một tiến trình và phương pháp luận để tiếp cận việc định nghĩa, đặc biệt hóa, thiết kế và xây dựng các khía cạnh.
- **Unified Process**—Một tiến trình phần mềm có đặc tính “hướng hành động (use-case), kiến trúc trung tâm, lặp và tăng dần” gần với ngôn ngữ UML (Unified Modeling Language)

# The Unified Process (UP)





# Pha UP



# UP Work Products

## Inception phase

Vision document  
Initial use-case model  
Initial project glossary  
Initial business case  
Initial risk assessment.  
Project plan,  
phases and iterations.  
Business model,  
if necessary.  
One or more prototypes

## Elaboration phase

Use-case model  
Supplementary requirements  
including non-functional  
Analysis model  
Software architecture  
Description.  
Executable architectural  
prototype.  
Preliminary design model  
Revised risk list  
Project plan including  
iteration plan  
adapted workflows  
milestones  
technical work products  
Preliminary user manual

## Construction phase

Design model  
Software components  
Integrated software  
increment  
Test plan and procedure  
Test cases  
Support documentation  
user manuals  
installation manuals  
description of current  
increment

## Transition phase

Delivered software increment  
Beta test reports  
General user feedback

# Tiến trình phần mềm cá nhân (PSP)

- **Lập kế hoạch.** Hành động này song song với yêu cầu và phát triển cả kích cỡ và dự đoán tài nguyên. Mặt khác, một dự đoán sai sót được tạo ra. Tất cả các metrics được ghi lại. Các tác vụ phát triển được xác định và lịch công việc được tạo ra.
- **Thiết kế bậc cao.** Các đặc điểm bên ngoài của mỗi thành phần được xây dựng và phát triển, thiết kế cho mỗi thành phần cũng được tạo ra. Các bản mẫu được xây dựng khi có thể và các vấn đề gặp phải được theo dõi và ghi lại.
- **Review thiết kế bậc cao.** Các phương pháp xác minh hình thức (Chapter 21) được áp dụng để phát hiện các lỗi trong thiết kế. Các metrics được bảo trì cho các tác vụ quan trọng và kết quả công việc.
- **Phát triển.** Thiết kế mức thành phần được tinh lọc và kiểm tra. Code được sinh, kiểm tra, biên dịch và test. Các metrics cho tất cả các tác vụ quan trọng và kết quả công việc.
- **Mổ xẻ phân tích.** Sử dụng các định lượng và metrics đã thu thập được (đây là lượng dữ liệu thực tế đã được phân tích thống kê), hiệu quả của tiến trình được xác định. Các định lượng và metrics cần được cung cấp và điều khiển để sửa đổi tiến trình sao cho hiệu quả hơn.

# Tiến trình phần mềm theo nhóm (TSP)

---

- Xây dựng nhóm tự định hướng kế hoạch và theo dõi công việc của họ, thiết lập mục tiêu, tiến trình và kế hoạch của họ. Nhóm ở đây có thể là một nhóm hoặc một nhóm tích hợp sản phẩm từ 3 đến 20 kỹ sư.
- Cần phải cho các quản lý biết làm thế nào để huấn luyện và thúc đẩy động lực cho nhóm của họ, và làm thế nào để giúp họ giữ vững được đỉnh cao hiệu suất.
- Tăng tốc quá trình cải tiến phần mềm bằng việc sử dụng mô hình CMM mức 5 cho hành vi bình thường và kì vọng.
  - Mô hình năng lực trưởng thành (CMM) là thang đo sự hiệu quả của tiến trình phần mềm, được thảo luận ở chương 30.
- Cung cấp sự hướng dẫn để trở thành tổ chức có tính trưởng thành cao.
- Tạo điều kiện cho các trường đại học giảng dạy về kỹ năng làm việc theo nhóm.