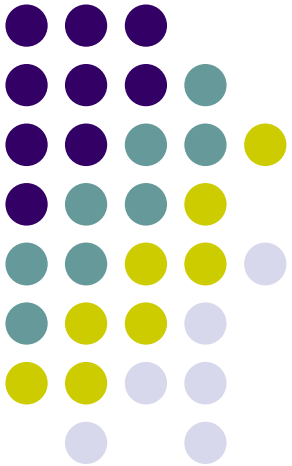
NHẬP MÔN CÔNG NGHỆ PHẦN MỀM

PHẦN I – TỔNG QUAN VỀ CÔNG NGHỆ PHẦN MỀM

Bộ môn Công nghệ phần mềm,
Khoa CNTT&TT, Đại học Cần Thơ

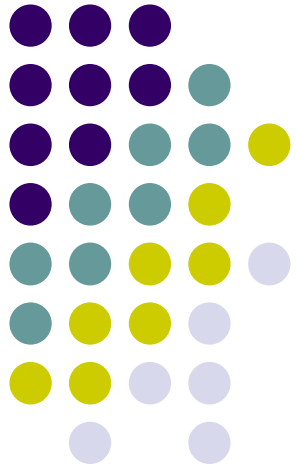


Nội dung

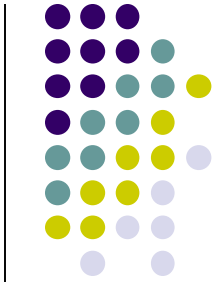
- Giới thiệu về Công nghệ phần mềm
- Các mô hình về tiến trình phần mềm
- Quản lý phần mềm

NHẬP MÔN CÔNG NGHỆ PHẦN MỀM

I.2 – CÁC MÔ HÌNH VỀ TIẾN TRÌNH PHẦN MỀM

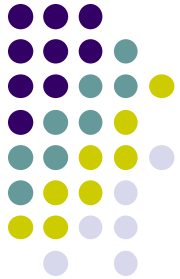


Nội dung



- Tiến trình
- Một số mô hình về tiến trình phần mềm
  - Mô hình xây dựng và hiệu chỉnh
  - Mô hình thác nước
  - Mô hình thác nước với bản mẫu
  - Mô hình chữ V
  - Mô hình bản mẫu
  - Mô hình tăng trưởng
  - Mô hình lặp và tăng trưởng
  - Mô hình phát triển ứng dụng nhanh
  - Mô hình xoắn ốc
  - Phương pháp nhanh
    - Lập trình cực đoan (XP)
    - SCRUM
  - Mô hình RUP

Tiến trình/Quy trình (Process)

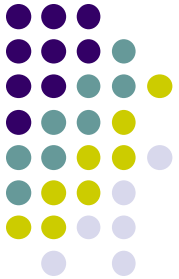


- **Định nghĩa**
  - **Tiến trình**: một chuỗi các bước bao gồm các hoạt động, các ràng buộc và các tài nguyên mà chúng tạo ra kết quả được mong đợi.
- **Tiến trình** bao gồm một bộ các công cụ và các kỹ thuật.

Tiến trình

- **Các đặc trưng của tiến trình**
  - Quy định tất cả các hoạt động của tiến trình chính.
  - Sử dụng các nguồn tài nguyên, phụ thuộc vào tập các ràng buộc (chẳng hạn như kế hoạch làm việc).
  - Tạo ra các sản phẩm cuối cùng hoặc trung gian.
  - Có thể được tạo thành từ các tiến trình con bằng hệ thống phân cấp hay các liên kết.

Tiến trình

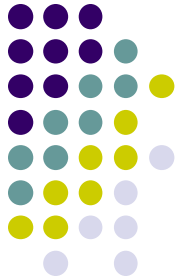


- **Các đặc trưng của tiến trình**
  - Mỗi hoạt động của tiến trình có tiêu chuẩn vào và ra.
  - Các hoạt động được tổ chức theo trình tự vì thế sự tính toán về thời gian là rõ ràng.
  - Mỗi tiến trình có các nguyên tắc hướng dẫn, bao gồm các mục tiêu của từng hoạt động.
  - Các ràng buộc có thể áp dụng vào một hoạt động, tài nguyên hay sản phẩm.

Tiến trình

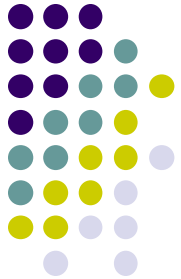
- **Tầm quan trọng của tiến trình**
  - Áp đặt cấu trúc và tính bền vững lên một tập các hoạt động.
  - Hướng dẫn ta hiểu, điều khiển, kiểm tra và cải thiện các hoạt động.
  - Cho phép ta có được các kinh nghiệm.

Tiến trình



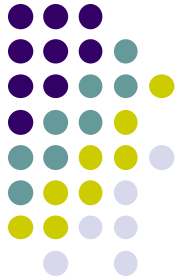
- **Mô hình hóa (modelling) tiến trình**
  - Mô tả cách tạo ra phần mềm theo cách được thực hiện trong thực tế.
  - Các quy định của tiến trình phát triển phần mềm phải tiến bộ.
- **Lý do để mô hình hóa tiến trình**
  - Hình thành một cách hiểu chung.
  - Tìm ra sự không nhất quán, sự dư thừa hay thiếu.
  - Tìm ra và đánh giá các hoạt động phù hợp để đạt được các mục tiêu của tiến trình.
  - Cụ thể hóa một tiến trình chung cho một hoàn cảnh cụ thể.

Tiến trình



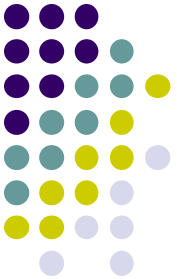
- **Chu kỳ sống của phần mềm (Software life cycle)**
  - Khi một tiến trình liên quan tới việc xây dựng một phần mềm, tiến trình có thể được xem như chu kỳ sống của phần mềm.

Một số mô hình về tiến trình phần mềm



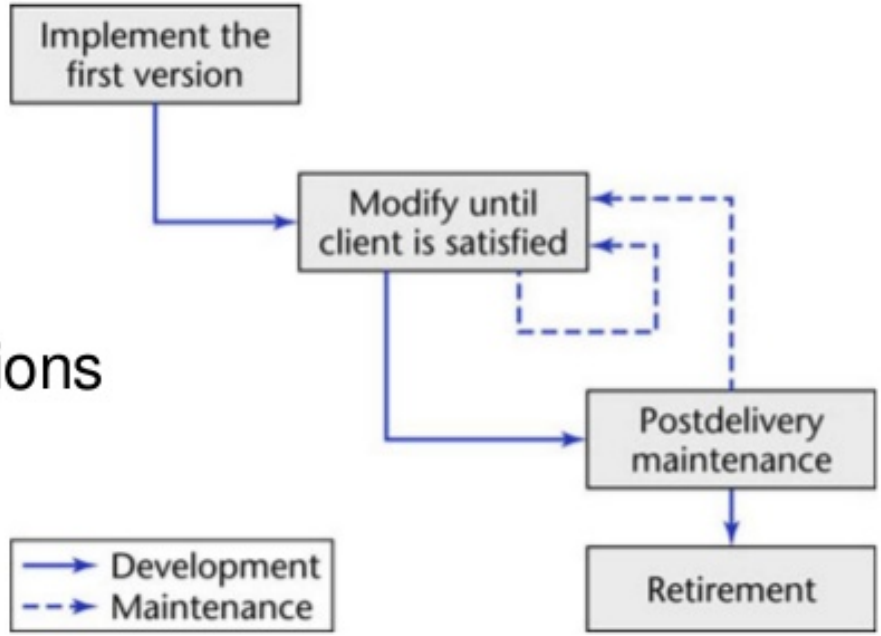
- Mô hình viết mã và hiệu chỉnh
- Mô hình thác nước
- Mô hình chữ V
- Mô hình lập bản mẫu
- Mô hình tăng trưởng
- Mô hình lặp và tăng trưởng
- Mô hình ứng dụng nhanh
- Mô hình xoắn ốc
- Phương pháp nhanh
  - Lập trình cực đoan (XP)
  - SCRUM
- Mô hình RUP

Mô hình viết mã và hiệu chỉnh (Code-and-Fix Model)



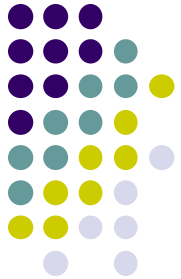
➢ Code-and-Fix Life-Cycle Model

- No design
- No specifications



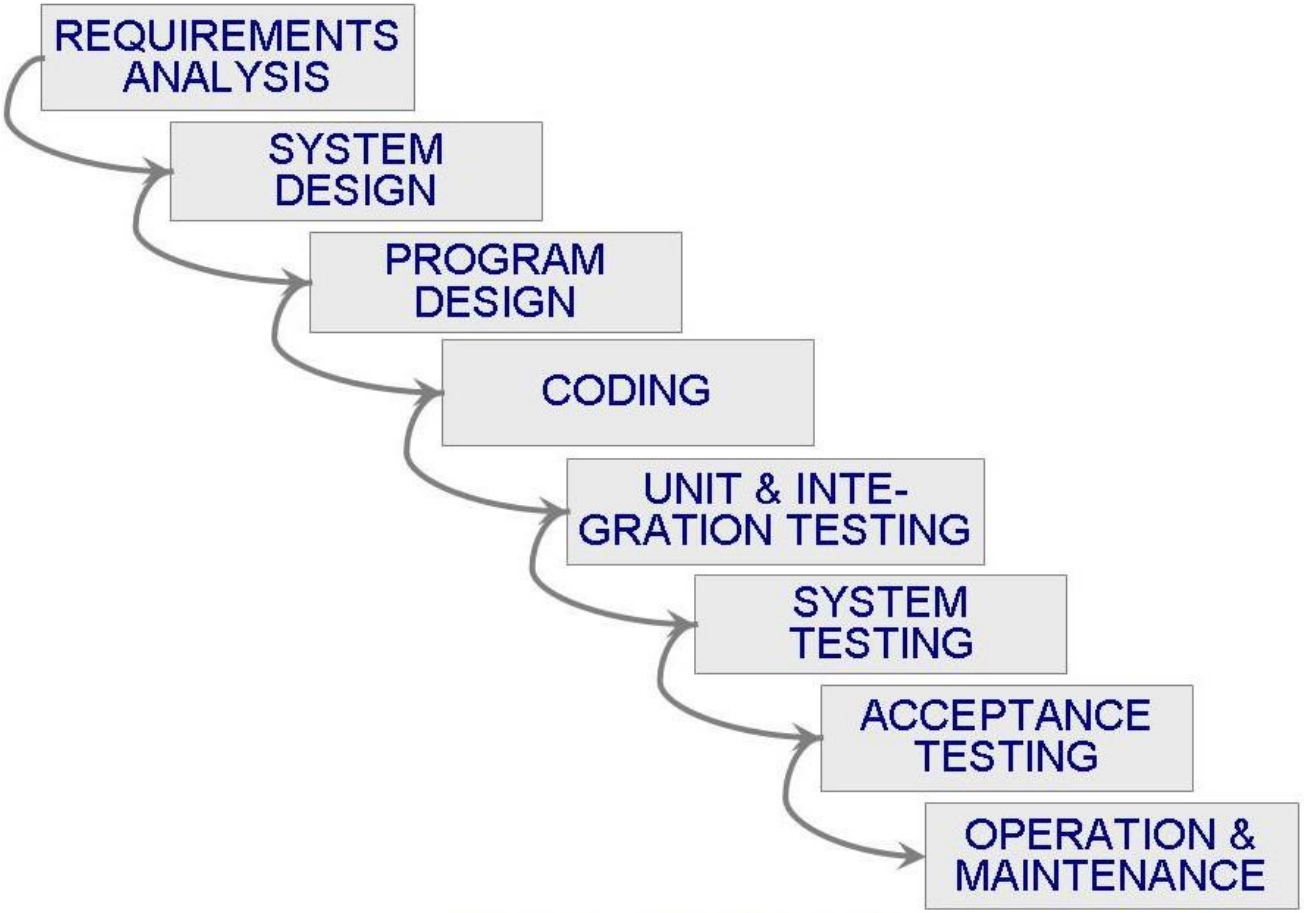
The easiest way to develop software
The most expensive way for maintenance
(i.e., maintenance nightmare)

Mô hình thác nước (Waterfall Model)



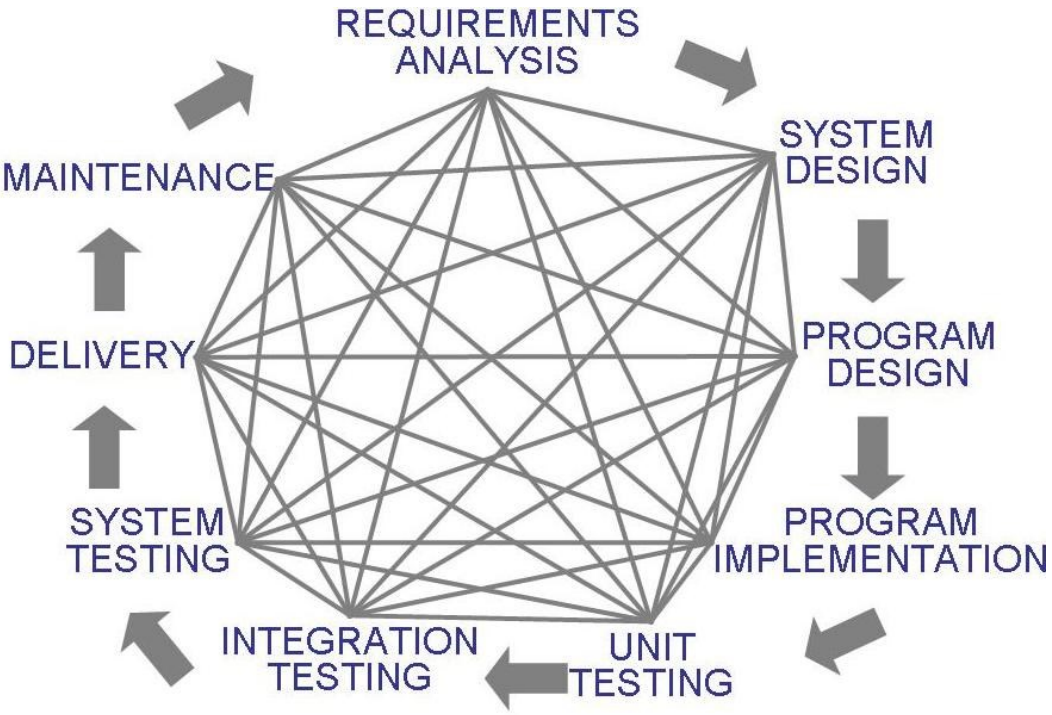
- Là một trong các mô hình đầu tiên về tiến trình phần mềm.
- Phù hợp với những bài toán được hiểu kỹ, có ít hay không có các thay đổi về yêu cầu.
- Đơn giản và dễ giải thích với khách hàng.
- Biểu diễn:
  - Một tổng quan mức rất cao của tiến trình phát triển.
  - Một chuỗi tuần tự các hoạt động của tiến trình.
- Mỗi giai đoạn chính được đánh dấu bởi các cột mốc và các sản phẩm (deliverables/artifacts)

Mô hình thác nước



Mô hình thác nước

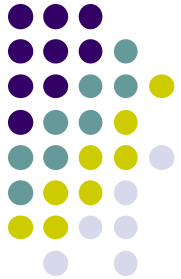
- Không có sự lặp lại trong mô hình thác nước
- Thực tế các dự án ít khi tuân theo dòng tuần tự của mô hình, mà thường có lặp lại



Mô hình thác nước

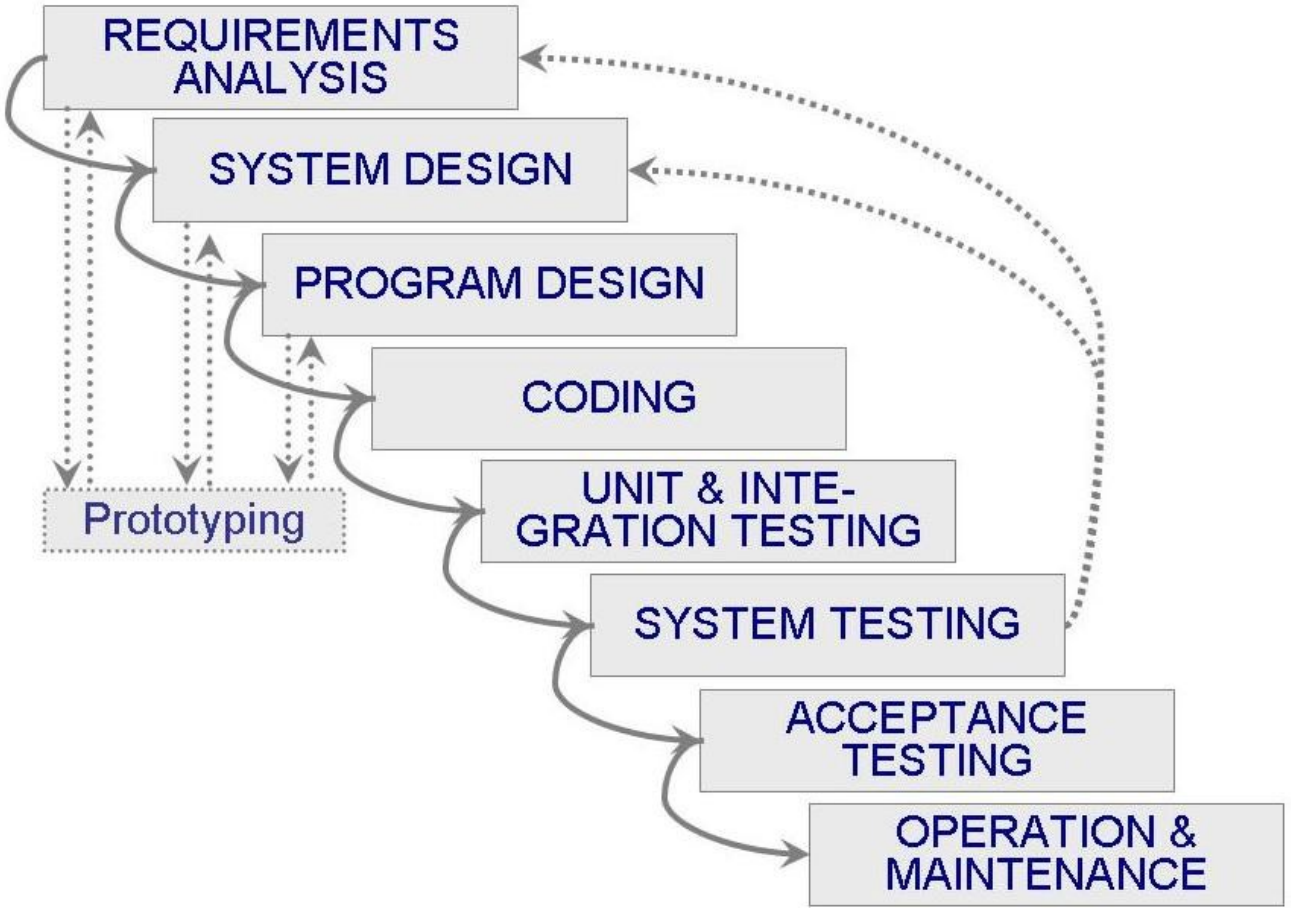
- Hạn chế của mô hình thác nước
  - Không cung cấp các hướng dẫn về cách thức xử lý những thay đổi cho sản phẩm và hoạt động trong suốt sự phát triển (giả thiết các yêu cầu bị đóng băng).
  - Xem sự phát triển phần mềm như một tiến trình sản xuất hơn là tiến trình sáng tạo.
  - Không có các hoạt động lặp mà chúng đưa đến việc tạo ra sản phẩm cuối cùng.
  - Phải chờ đợi lâu trước khi có sản phẩm cuối.

Mô hình thác nước với bản mẫu

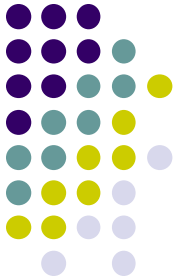


- Bản mẫu (prototype) là sản phẩm được phát triển một phần.
- Lập bản mẫu (prototyping) giúp:
  - Các nhà phát triển đánh giá các chiến lược thiết kế thay thế (bản mẫu thiết kế).
  - Người dùng hiểu hệ thống sẽ như thế nào (bản mẫu giao diện người dùng).
- Lập bản mẫu là hữu ích để kiểm tra và xác nhận.

Mô hình thác nước với bản mẫu

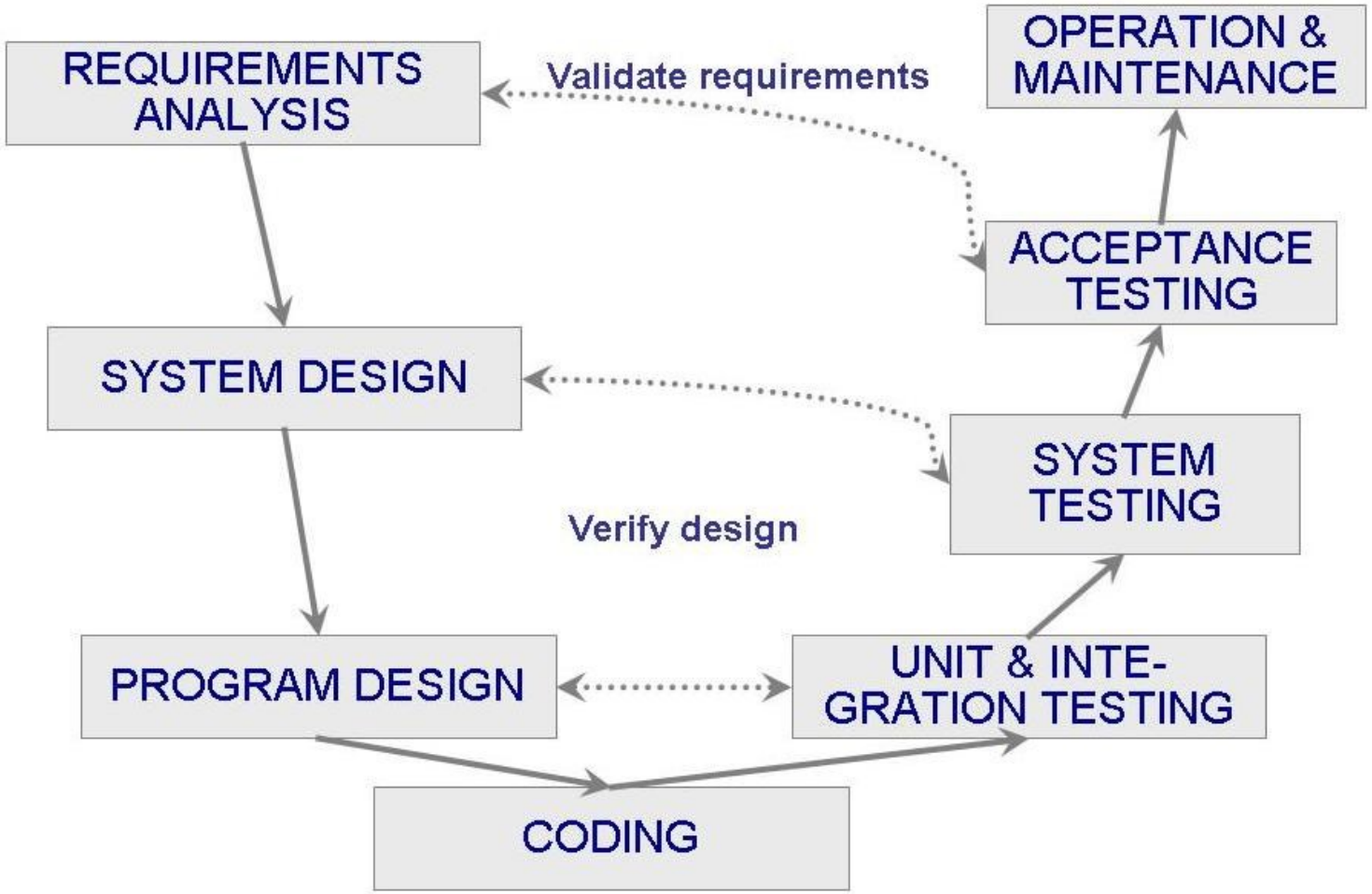


Mô hình chữ V (V Model)

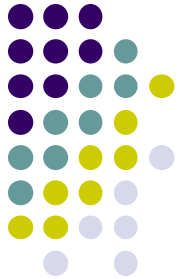


- Là một biến thể của mô hình thác nước.
- Sử dụng kiểm thử đơn vị để kiểm tra/thẩm tra (verify) thiết kế thủ tục.
- Sử dụng kiểm thử tích hợp để kiểm tra thiết kế hệ thống
- Sử dụng kiểm thử chấp nhận để xác nhận/công nhận hợp lệ (valiadate) các yêu cầu.
- Nếu các vấn đề được tìm thấy trong suốt sự kiểm tra và xác nhận, phần bên trái của mô hình chữ V có thể được tái thực hiện trước khi việc kiểm thử phần bên phải được tái thực hiện.

Mô hình chữ V

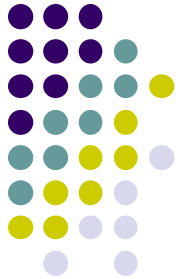

REQUIREMENTS ANALYSIS
Validate requirements
OPERATION & MAINTENANCE
ACCEPTANCE TESTING
SYSTEM DESIGN
SYSTEM TESTING
Verify design
PROGRAM DESIGN
UNIT & INTE-GRATION TESTING
CODING

Mô hình chữ V



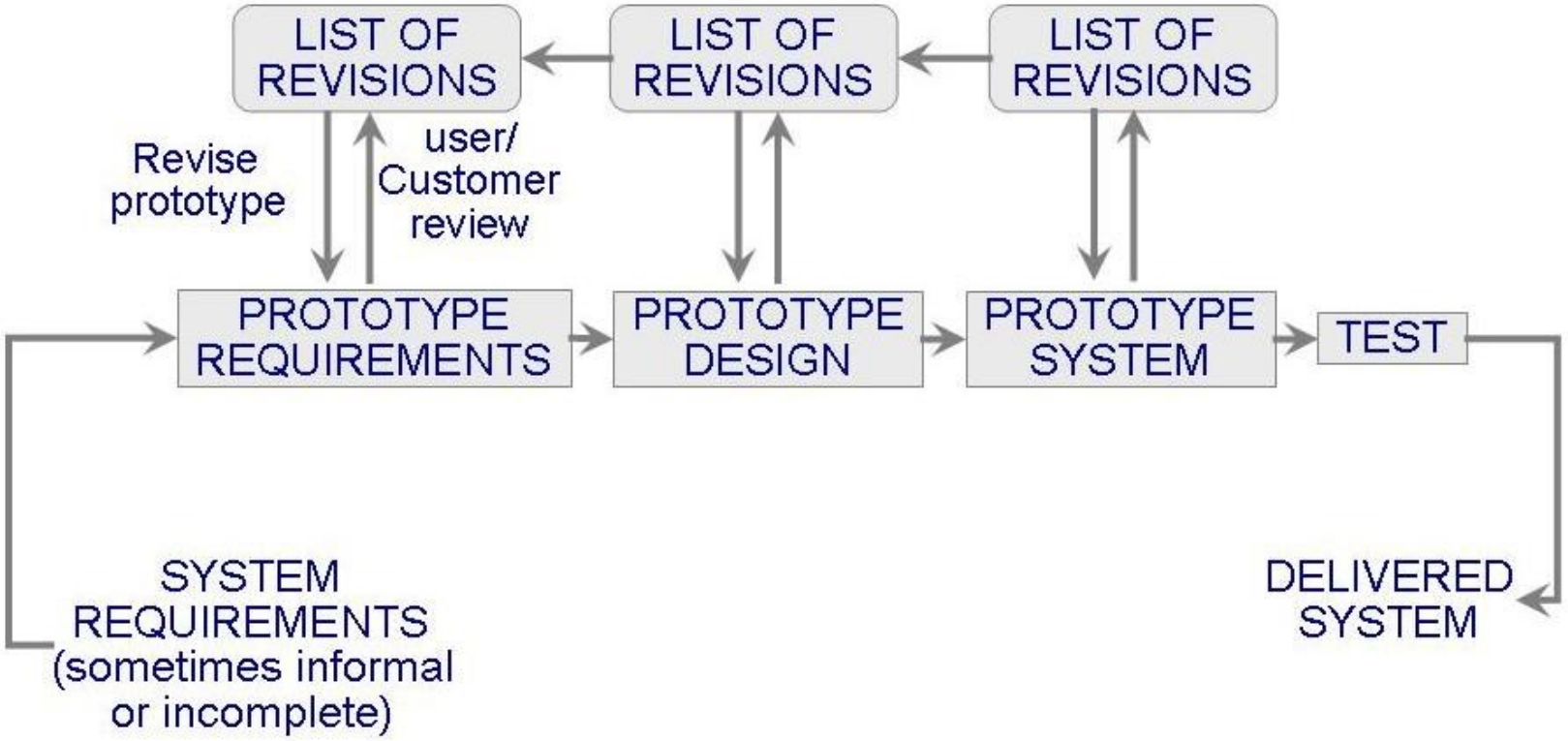
- Boehm succinctly expressed the difference between them:
  - Validation: Are we building the right product?
  - Verification: Are we building the product right?
- According to the Capability Maturity Model (CMMI-SW v1.1),
  - Validation: The process of evaluating software during or at the end of the development process to determine whether it satisfies specified requirements. [IEEE-STD-610]
  - Verification: The process of evaluating software to determine whether the products of a given development phase satisfy the conditions imposed at the start of that phase. [IEEE-STD-610]

Mô hình lập bản mẫu (Prototyping Model)



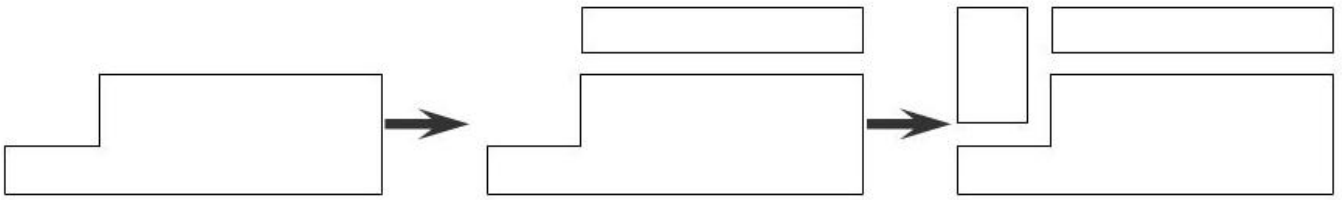
- Cho phép sự nghiên cứu về các yêu cầu và thiết kế được lặp lại.
- Giảm sự rủi ro và sự không chắc chắn trong phát triển.
- Sử dụng mô hình bản mẫu khi các yêu cầu không rõ ràng và cần minh họa cao về giao diện người dùng.

Mô hình lập bản mẫu

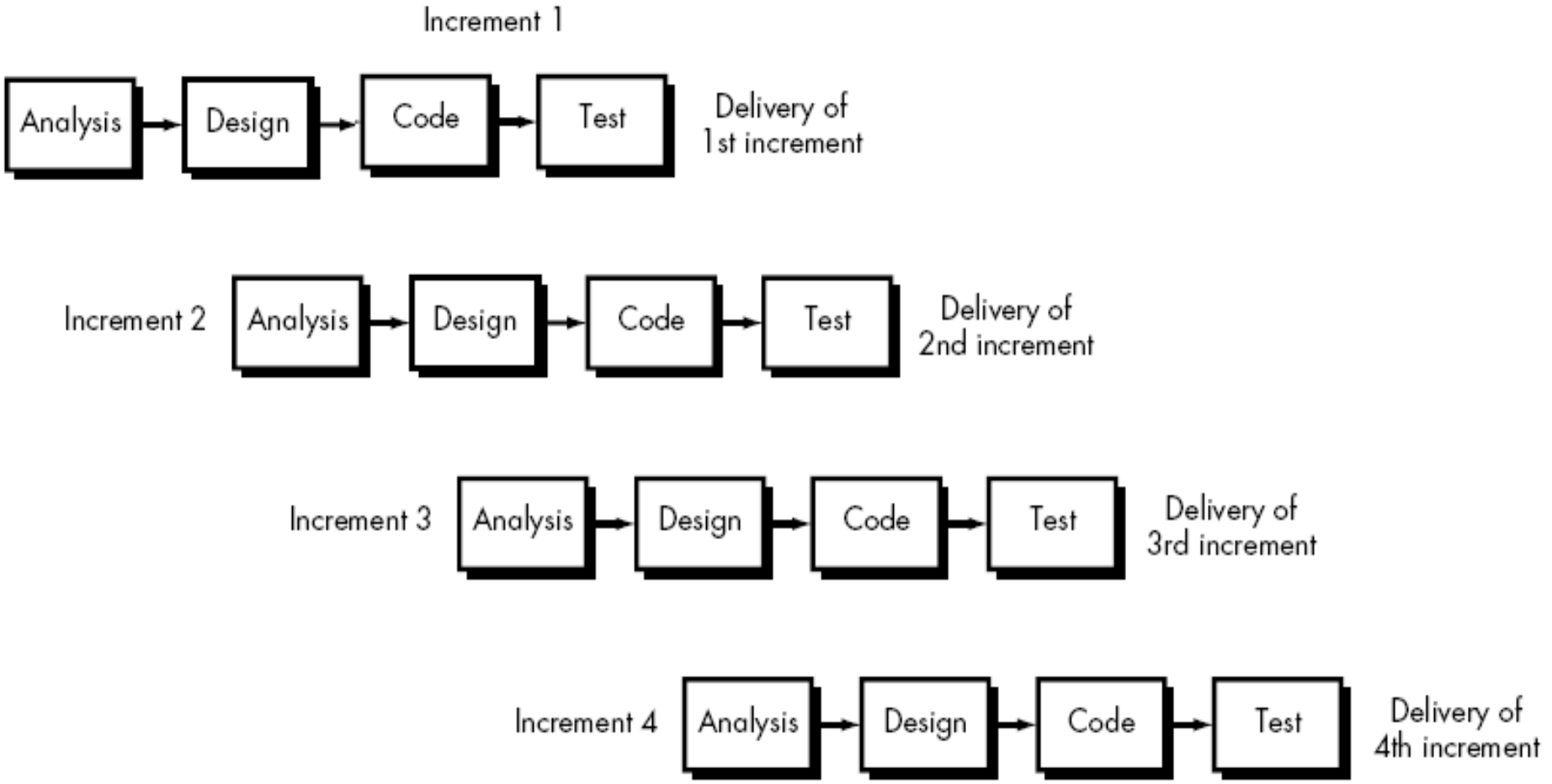


Mô hình tăng trưởng (Incremental Model)

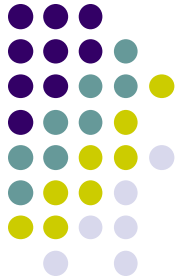
- Sản phẩm lõi cho những yêu cầu cơ bản nhất của hệ thống được phát triển.
- Các chức năng cho những yêu cầu khác được phát triển thêm sau (tăng trưởng, gia tăng).
- Lặp lại quy trình để hoàn thiện dần.



Mô hình tăng trưởng



Mô hình lặp và tăng trưởng (Increments and Iterations Model)

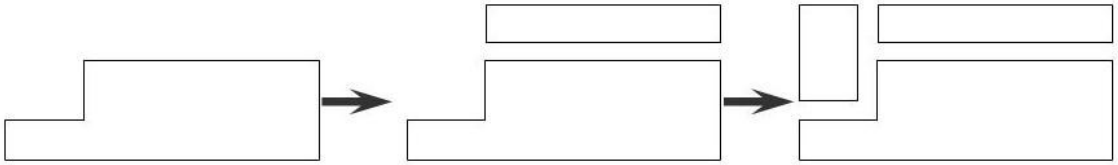


- Chu kỳ có thời gian ngắn hơn.
- Hệ thống cung cấp từng phần
  - Cho phép khách hàng có một số chức năng trong khi phần còn lại đang được phát triển
- Cho phép hai hệ thống hoạt động song song
  - Hệ thống (phát hành n): hiện đang được sử dụng
  - Hệ thống (phát hành n + 1): phiên bản tiếp theo hiện đang được phát triển

Mô hình lặp và tăng trưởng (Increments and Iterations Model)

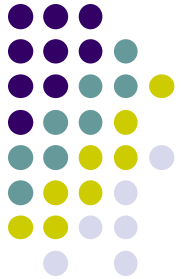
- Phát triển tăng trưởng: bắt đầu với hệ thống con chức năng nhỏ và thêm chức năng ở mỗi bản phát hành mới.
- Phát triển lặp: bắt đầu với hệ thống đầy đủ, sau đó thay đổi chức năng của từng hệ thống con ở mỗi bản phát hành mới.

INCREMENTAL DEVELOPMENT



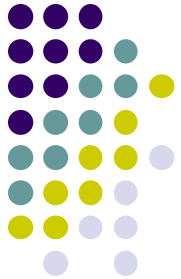
ITERATIVE DEVELOPMENT

Mô hình lặp và tăng trưởng (Increments and Iterations Model)



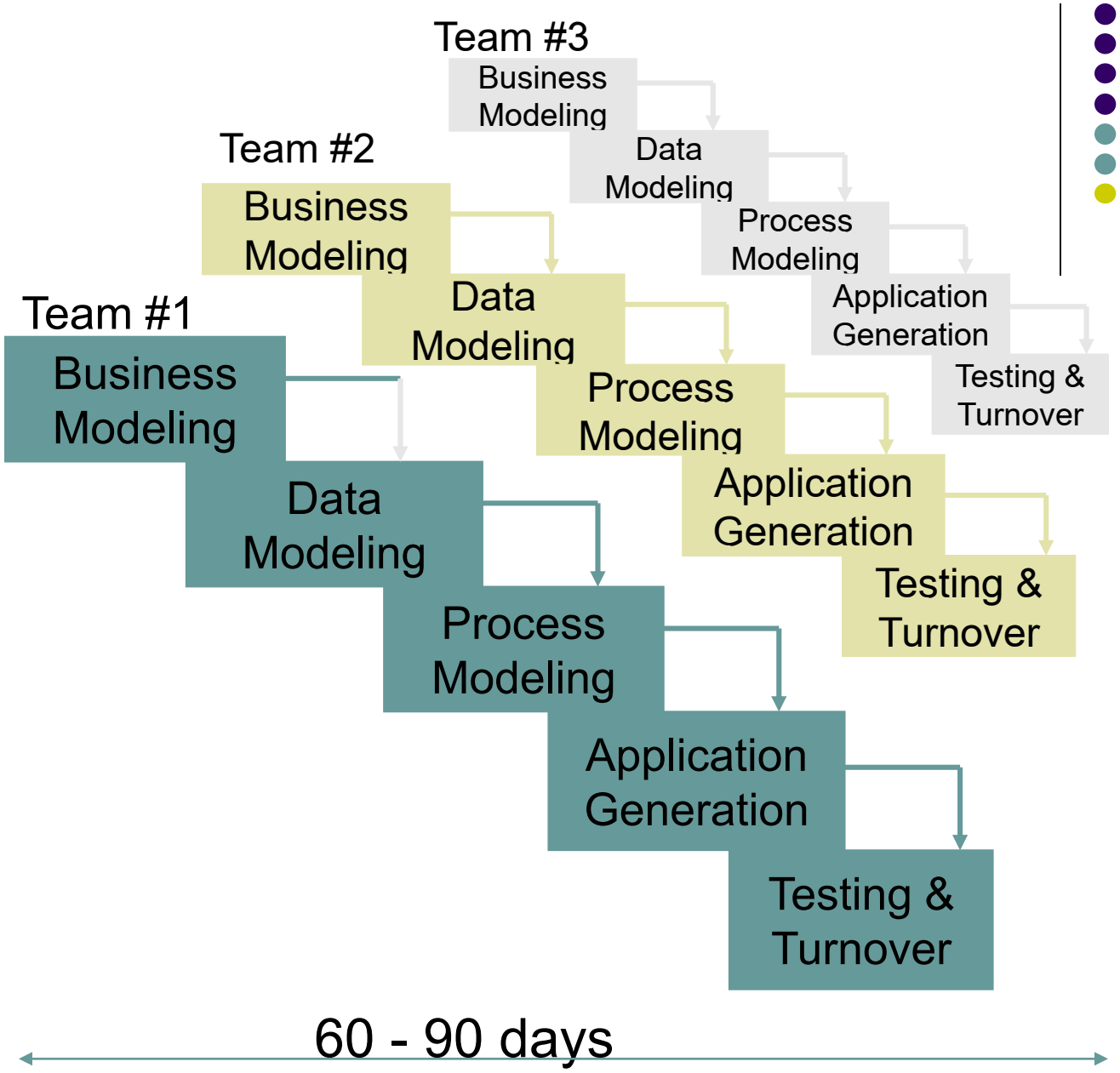
- Phát triển tăng trưởng và lặp là mong muốn vì một số lý do:
  - Việc đào tạo có thể bắt đầu sớm, mặc dù thiếu một số chức năng.
  - Thị trường có thể được tạo sớm cho chức năng chưa từng được cung cấp trước đây.
  - Các bản phát hành thường xuyên cho phép các nhà phát triển khắc phục các sự cố không mong muốn tổng quát và nhanh chóng.
  - Nhóm phát triển có thể tập trung vào các lĩnh vực chuyên môn khác nhau với các bản phát hành khác nhau.

Mô hình phát triển ứng dụng nhanh (Rapid Application Development: RAD)

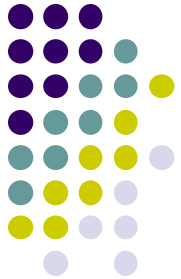


- Là tiến trình phát triển phần mềm dạng tăng trưởng, tăng dần từng bước với mỗi chu kỳ phát triển rất ngắn (60-90 ngày).
- Xây dựng dựa trên hướng thành phần với khả năng tái sử dụng.
- Gồm một số nhóm, mỗi nhóm làm 1 RAD theo các pha: Mô hình hóa nghiệp vụ, Mô hình hóa dữ liệu, Mô hình hóa xử lý, Tạo ứng dụng, Kiểm thử và luân chuyển (Business, Data, Process, Appl. Generation, Testing&Turnover).

Mô hình phát triển ứng dụng nhanh

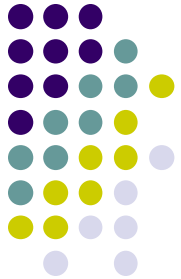


Mô hình phát triển ứng dụng nhanh



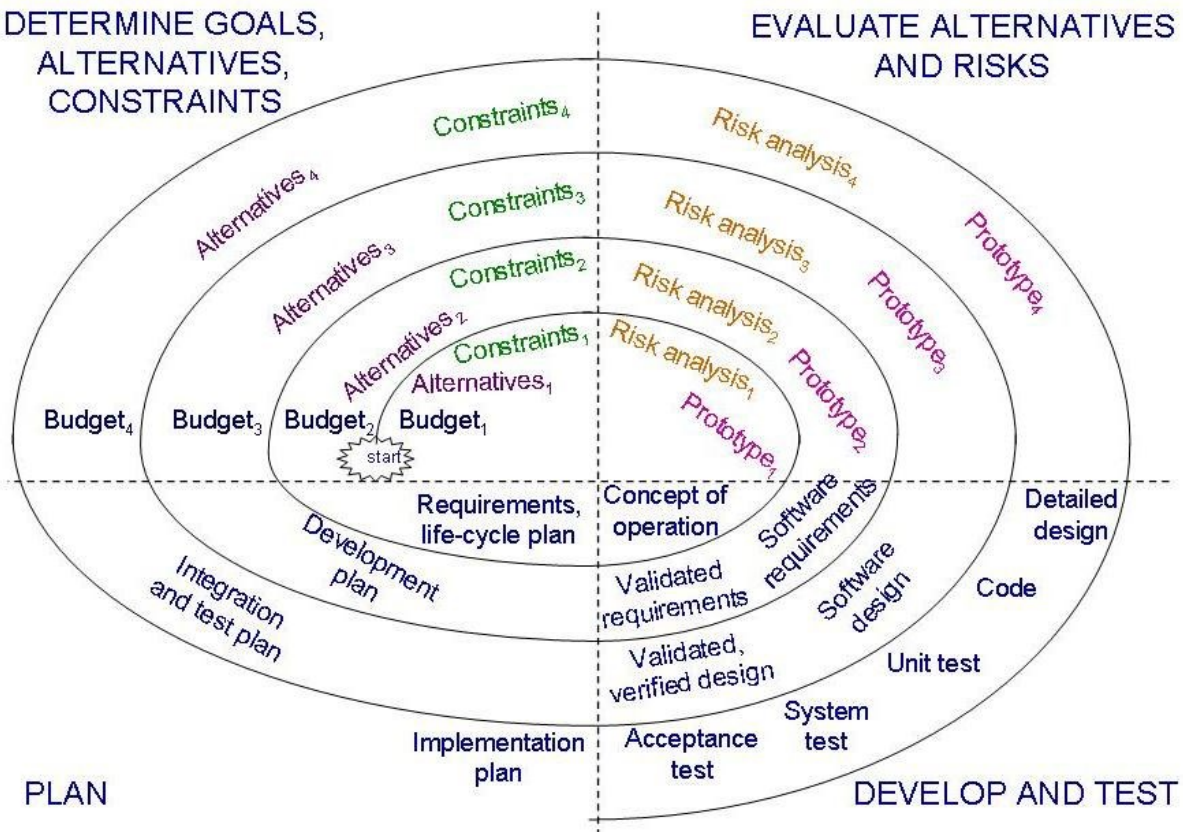
- Cần nguồn nhân lực dồi dào để tạo các nhóm cho các chức năng chính.
- Yêu cầu hai bên cam kết trong thời gian ngắn phải có phần mềm hoàn chỉnh, thiếu trách nhiệm của một bên dễ làm dự án đổ vỡ.
- Phụ thuộc cao vào kỹ năng mô hình hóa.
- Không tốt cho mọi ứng dụng, nhất là với ứng dụng không thể module hóa.
- Không thể áp dụng cho các dự án rẻ vì chi phí mô hình hóa và sinh mã tự động rất cao.

Mô hình xoắn ốc (Spiral Model)

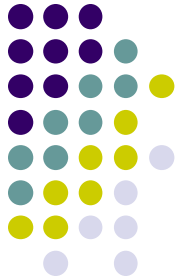


- Được đề nghị bởi Boehm (1988).
- Kết hợp các hoạt động phát triển với quản lý rủi ro để giảm đến mức tối thiểu và kiểm soát các rủi ro.
- Mô hình được trình bày ở dạng xoắn ốc trong đó mỗi lần lặp được biểu diễn bởi một đường vòng quanh bốn hoạt động chính:
  - Lập kế hoạch
  - Xác định các mục tiêu, các lựa chọn và các ràng buộc
  - Đánh giá các lựa chọn và các rủi ro
  - Phát triển và kiểm thử

Mô hình xoắn ốc

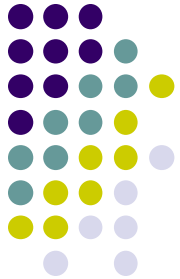


Phương pháp nhanh (Agile Methods)



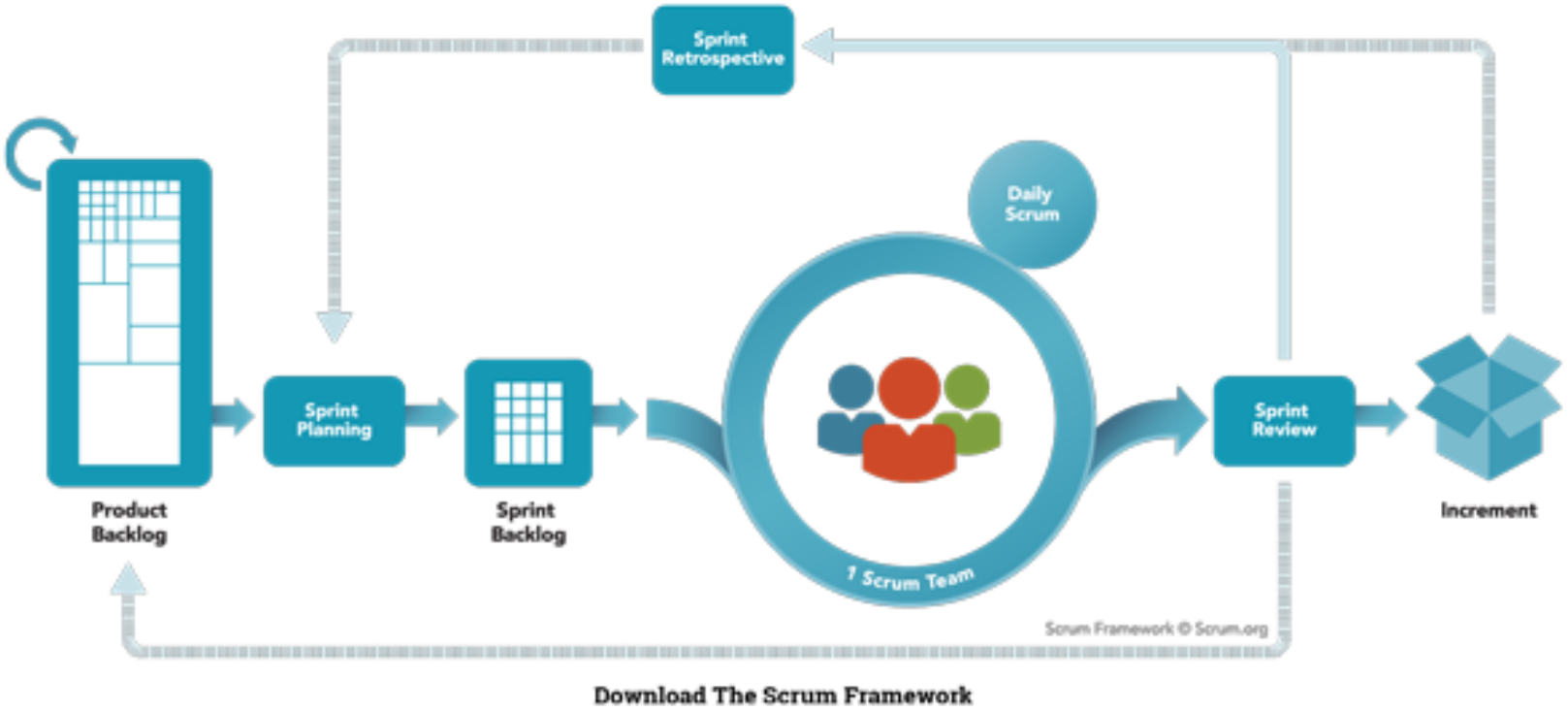
- Nhấn mạnh vào tính linh hoạt trong việc phát triển phần mềm một cách nhanh chóng và hiệu quả.
- Tuyên ngôn
  - Đánh giá cao các cá nhân và tương tác qua quy trình và công cụ.
  - Thích đầu tư thời gian vào việc tạo ra phần mềm hoạt động hơn là tạo ra tài liệu toàn diện.
  - Tập trung vào sự hợp tác của khách hàng hơn là đàm phán hợp đồng.
  - Tập trung vào phản ứng với sự thay đổi hơn là tạo ra một kế hoạch và sau đó làm theo nó.

Phương pháp nhanh - Lập trình cực đoan (eXtreme Programming/XP)



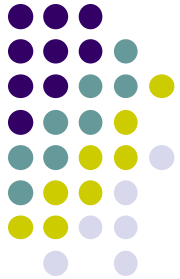
- Nhấn mạnh vào bốn đặc trưng của sự nhanh chóng:
  - Giao tiếp: trao đổi liên tục giữa khách hàng và nhà phát triển.
  - Đơn giản: chọn thiết kế hoặc lập trình đơn giản nhất.
  - Dũng cảm: cam kết cung cấp chức năng sớm và thường xuyên.
  - Phản hồi: các vòng lặp được tích hợp vào các hoạt động khác nhau trong quá trình phát triển.

Phương pháp nhanh - SCRUM



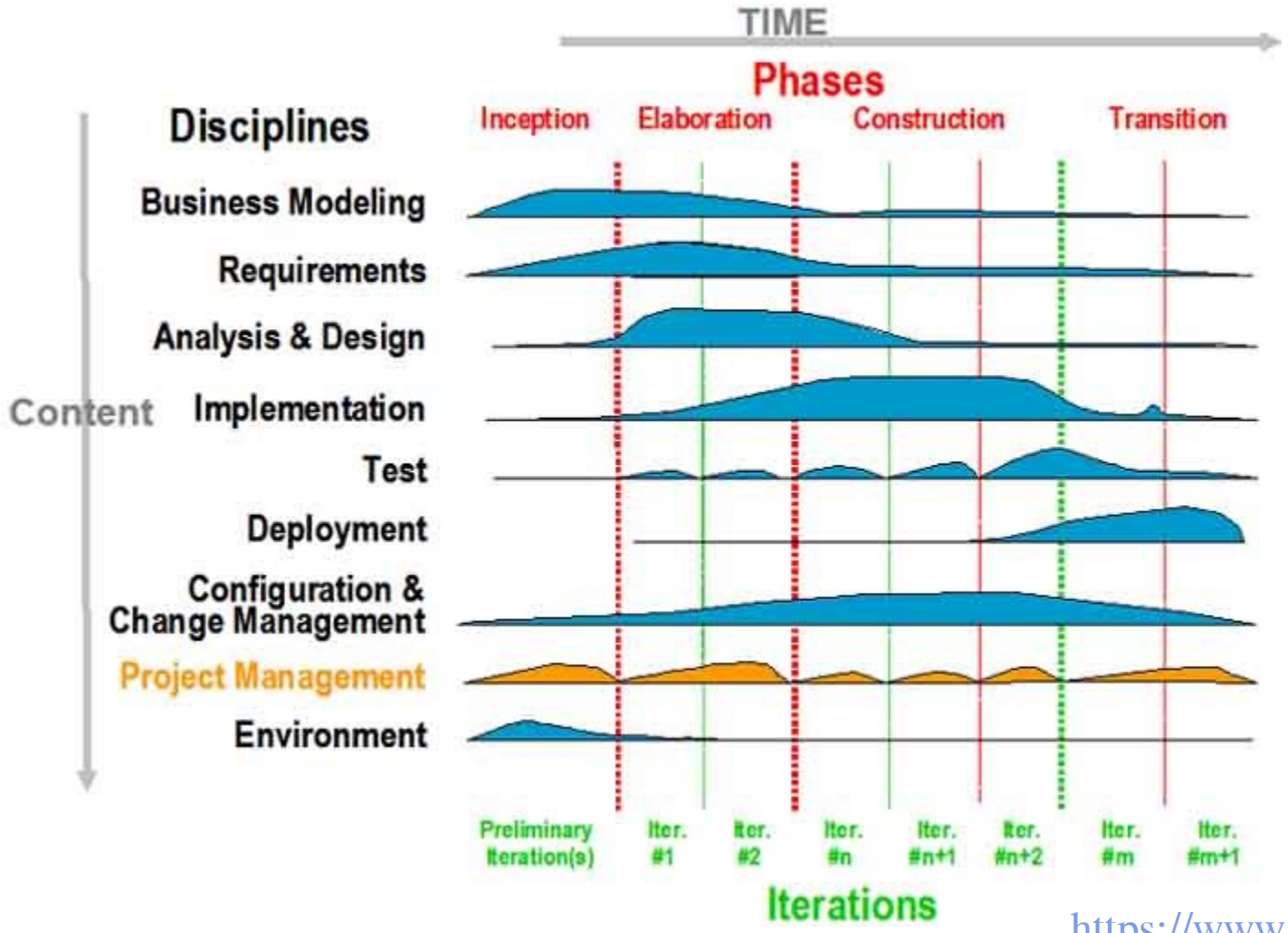
https://www.scrum.org

Phương pháp nhanh - SCRUM



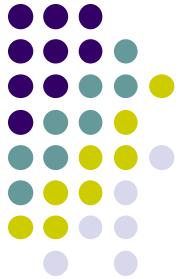
- Scrum là một khung quy trình đã được sử dụng để quản lý công việc trên các sản phẩm phức tạp từ đầu những năm 1990.
- Khung Scrum bao gồm các nhóm Scrum (Scrum Teams) và các vai trò (roles) liên quan của họ, sự kiện (events), hiện vật (artifacts) và quy tắc (rules). Mỗi thành phần trong khung Scrum phục vụ một mục đích cụ thể và là yếu tố cần thiết cho sự thành công và sử dụng của Scrum.

Mô hình RUP (Rational Unified Process)



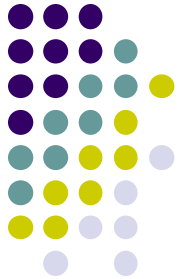
https://www.ibm.com

Mô hình RUP (Rational Unified Process)



- Chiều động - Các giai đoạn
  - *Khởi đầu (Inception)*: thiết lập phạm vi, giới hạn, các use case quan trọng, các kiến trúc ứng viên, các dự đoán về chi phí và kế hoạch làm việc.
  - *Phác thảo/Triển khai (Elaboration)*: phân tích vấn đề, thiết lập nền tảng kiến trúc, thiết lập sự hỗ trợ công cụ.
  - *Xây dựng (Construction)*: phát triển các thành phần, tích hợp chúng và kiểm tra kỹ lưỡng.
  - *Chuyển giao (Transition)*: phát hành ra cộng đồng người sử dụng.

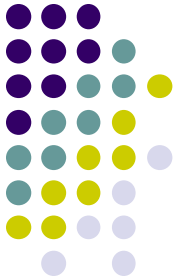
Mô hình RUP (Rational Unified Process)



- Chiều tĩnh - Các luồng công việc
  - Một quy trình mô tả ‘ai’ đang làm ‘cái gì’, ‘như thế nào’ và ‘khi nào’. RUP được biểu diễn bằng bốn phần tử chính:
    - Nhân viên, ‘ai’
    - Các hoạt động, ‘như thế nào’
    - Đồ tạo tác, ‘cái gì’
    - Luồng công việc, ‘khi nào’. Có chín luồng công việc cốt lõi trong RUP, đại diện cho sự phân chia của tất cả nhân viên và hoạt động thành các nhóm hợp lý.

Các mô hình khác

- Tự đọc trong giáo trình



HẾT PHẦN I.2