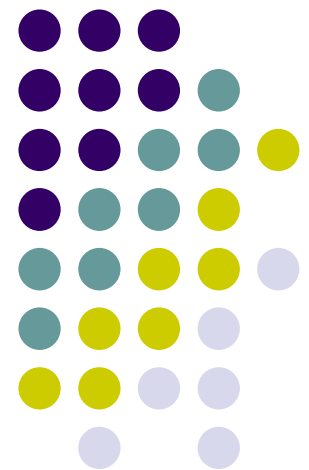
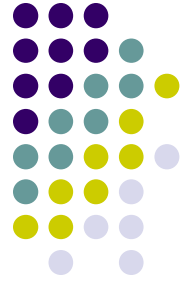


NHẬP MÔN CÔNG NGHỆ PHẦN MỀM

CHƯƠNG 2 – CÁC MÔ HÌNH VỀ TIẾN TRÌNH PHẦN MỀM

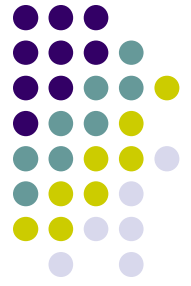




Nội dung

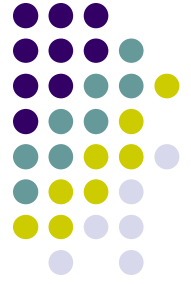
- Tiến trình
- Các mô hình về tiến trình phần mềm
 - Mô hình thác nước
 - Mô hình chữ V
 - Mô hình bản mẫu
 - Mô hình định khung nhanh
 - Mô hình xoắn ốc
 - Mô hình RUP

Tiến trình (Process)



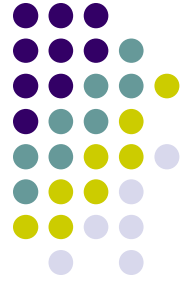
- ❖ Tiến trình phần mềm là cách thức tạo ra phần mềm, mỗi công ty có tiến trình phần mềm riêng
- ❖ Khách hàng (client): cá nhân hay công ty đặt hàng sản phẩm
- ❖ Nhà phát triển (developer): các thành viên của công ty có trách nhiệm phát triển phần mềm đã được đặt hàng
 - ❖ có thể quán xuyến toàn bộ các công việc của sản phẩm
 - ❖ có trách nhiệm một phần như thiết kế, cài đặt,...
- ❖ Người sử dụng (user): một hay nhiều cá nhân thay mặt khách hàng để sử dụng sản phẩm
- ❖ Phát triển phần mềm (software development): bao gồm tất cả các công việc tạo ra sản phẩm trước khi nó được chuyển sang giai đoạn bảo trì

Tiến trình



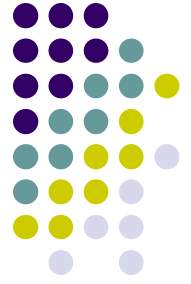
- **Các đặc trưng của tiến trình**
 - Quy định tất cả các hoạt động của tiến trình chính
 - Sử dụng các nguồn tài nguyên, phụ thuộc vào tập các ràng buộc (chẳng hạn như kế hoạch làm việc)
 - Tạo ra các sản phẩm cuối cùng hoặc trung gian
 - Có thể được tạo thành từ các tiến trình con bằng hệ thống phân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

Tiến trình



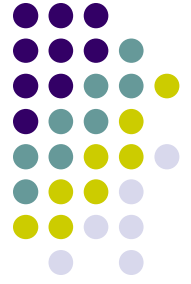
- **Các đặc trưng của tiến trình**
 - Mỗi hoạt động của tiến trình có tiêu chuẩn vào và ra
 - Các hoạt động được tổ chức theo trình tự vì thể sự tính toán về thời gian là rõ ràng
 - Mỗi tiến trình có các nguyên tắc hướng dẫn, bao gồm các mục tiêu của từng hoạt động
 - Các ràng buộc có thể áp dụng vào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



- **Chu kỳ sống của phần mềm**
 - 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.

Các mô hình về tiến trình phần mềm



- Mô hình xây dựng và hiệu chỉnh
- Mô hình thác nước
- Mô hình chữ V
- Mô hình bản mẫu
- Mô hình định khung nhanh
- Mô hình xoắn ốc
- Mô hình hướng đối tượng
- Mô hình RUP

Mô hình xây dựng và hiệu chỉnh (Build-and-fix model)



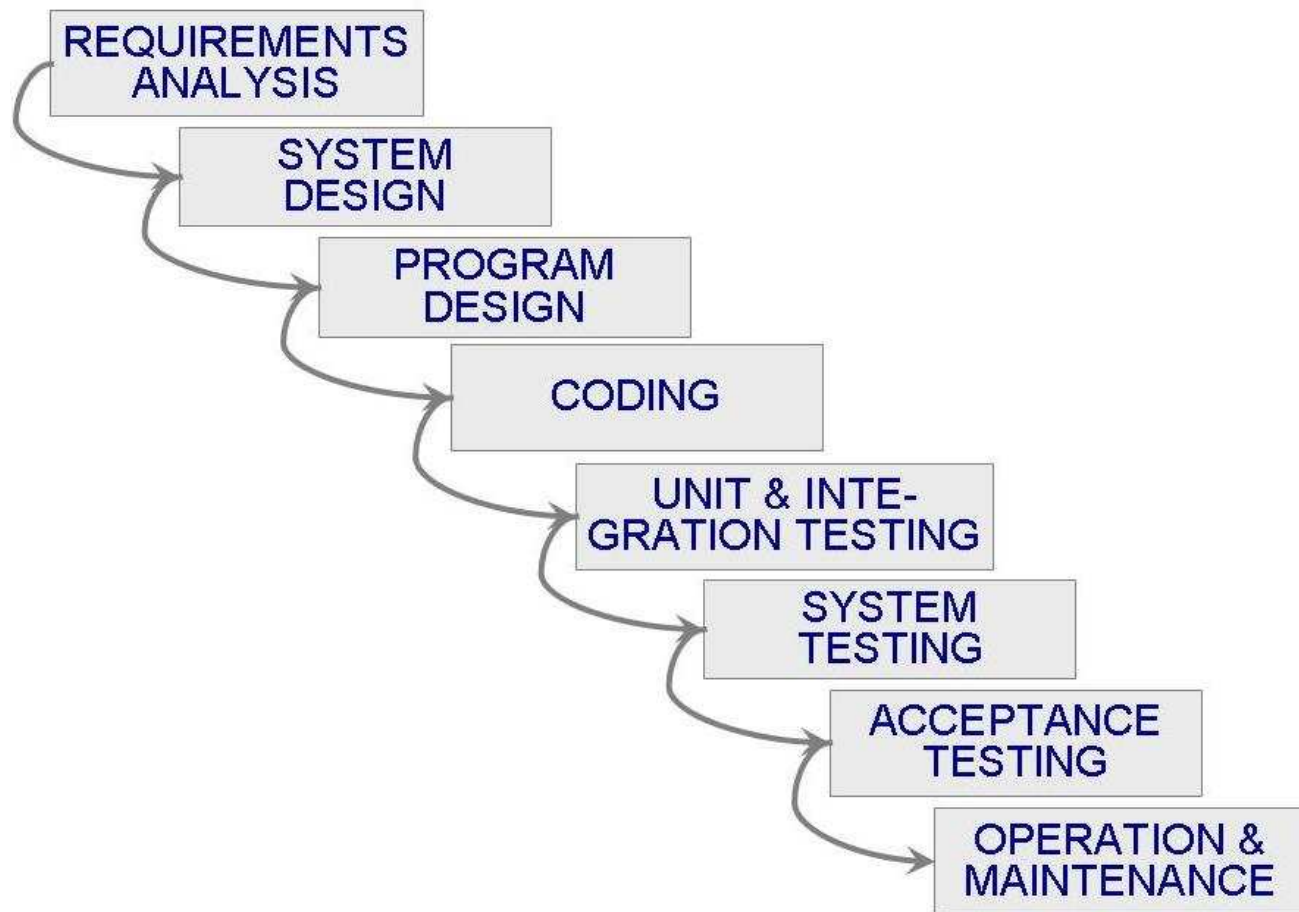
- Không dự tính trước
- Không có đặc tả hay thiết kế
- Xây dựng 1 phiên bản, chỉnh sửa theo yêu cầu của khách hàng cho đến khi nào đáp ứng được yêu cầu của khách hàng
- Sử dụng trong các hệ thống rất nhỏ (100-200 dòng lệnh)

Mô hình thác nước (Waterfall Model)

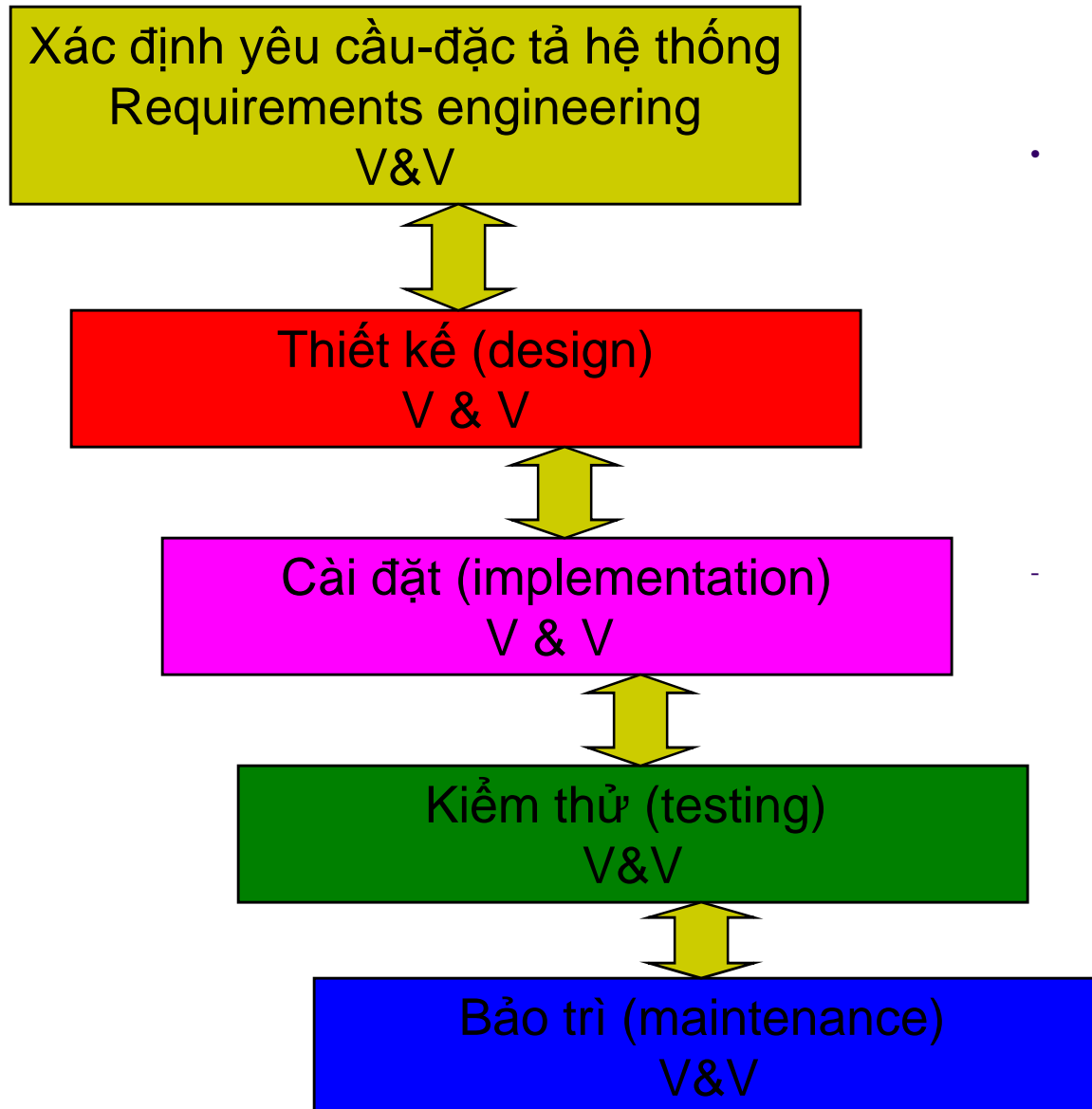
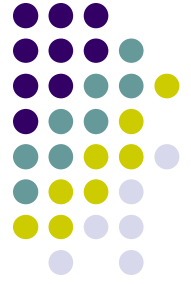


- Royce, 1970
- Phù hợp với những bài toán được hiểu kỹ có ít hay không có các thay đổi về yêu cầu
- Đơn giản và dễ giải thích với khách hàng
- Nó biểu diễn
 - Một tổng quan mức rất cao của tiến trình phát triển
 - Một chuỗi tuần tự các hoạt động của tiến trình

Mô hình thác nước

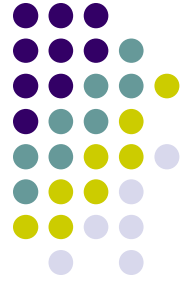


Mô hình thác nước



- V & V:
 - - **Verification (kiểm tra)**: hệ thống thỏa mãn đặc tả (Build the system right)
 - **Validation (kiểm tra-xác nhận)**: hệ thống thỏa mãn yêu cầu người dùng (Build the right system)
 - Đặc điểm:
 - Hướng tài liệu
 - Phân tích kỹ trước khi xây dựng hệ thống
 - kiểm tra từng bước
 - Kiểm tra chuyển tiếp giữa các bước

Mô hình thác nước



- Đặc tính:
 - Các lỗi ở một số giai đoạn trước được phản hồi bởi các giai đoạn sau
 - Mỗi giai đoạn chỉ được xem là hoàn thành sau khi đã có đầy đủ tài liệu cho giai đoạn đó và được nhóm SQA chấp thuận
- Các bước tiến hành chính:
 - Các yêu cầu được xác định và kiểm chứng bởi khách hàng và nhóm SQA
 - Các đặc tả được kiểm chứng bởi nhóm SQA và gửi cho khách hàng
 - Giai đoạn thiết kế bắt đầu sau khi khách hàng đồng ý về giá thành và thời gian thực hiện; thực hiện cài đặt và tích hợp
 - Khách hàng cho hoạt động thử
 - Chấp nhận sản phẩm
 - Chuyển sang giai đoạn bảo trì
- Ưu điểm:

Kỷ luật cao; quy định tốt về tài liệu cho mỗi giai đoạn; kiểm chứng cẩn thận bởi nhóm SQA; được ứng dụng rộng rãi
- Khuyết điểm:
 - Quá nhiều kiểm thử, kiểm tra-xác nhận và tài liệu
 - Hướng tài liệu: khó hình dung và khó hiểu đối với khách hàng



Mô hình thác nước

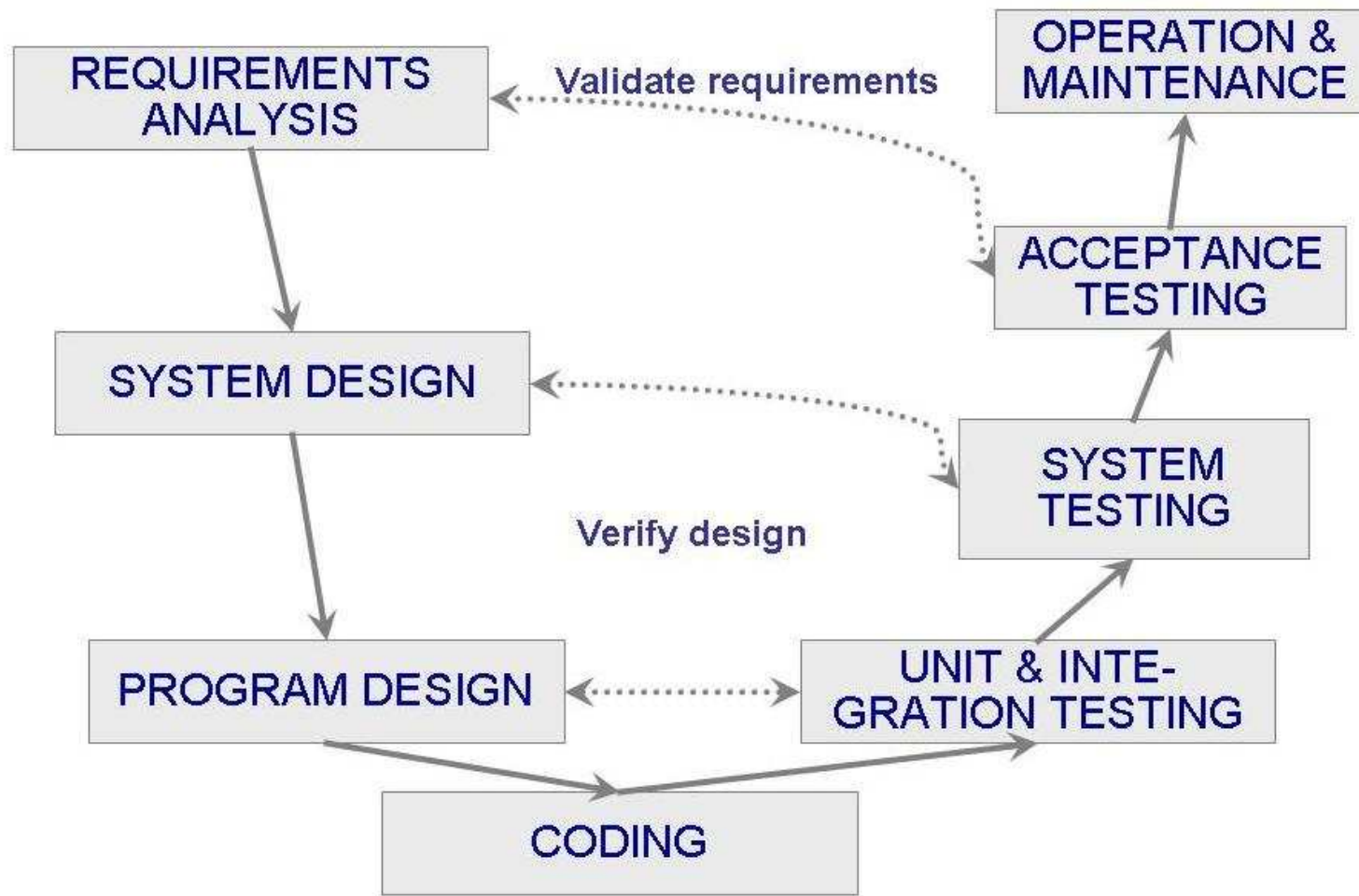
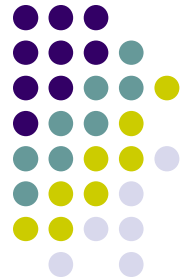
- Hạn chế của mô hình thác nước
 - Không cung cấp các hướng dẫn về cách thức xử lý những thay đổi đối về sản phẩm và hoạt động trong suốt sự phát triển
 - Xem sự phát triển phần mềm như một tiến trình sản xuất hơn là tiến trình sáng tạo
 - Không có các hoạt động lặp mà chúng đưa đến việc tạo ra sản phẩm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



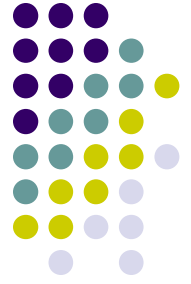
Mô hình chữ V (V Model)

- Một sự biến đổi của mô hình thác nước
- Sử dụng kiểm thử đơn vị để kiểm chứng thiết kế thủ tục
- Sử dụng kiểm thử tích hợp để kiểm chứng thiết kế hệ thống
- Sử dụng kiểm thử chấp nhận để xác nhận tính hợp lệ các yêu cầu
- Nếu các vấn đề được tìm thấy trong suốt sự kiểm chứng và sự xác nhận tính hợp lệ, 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

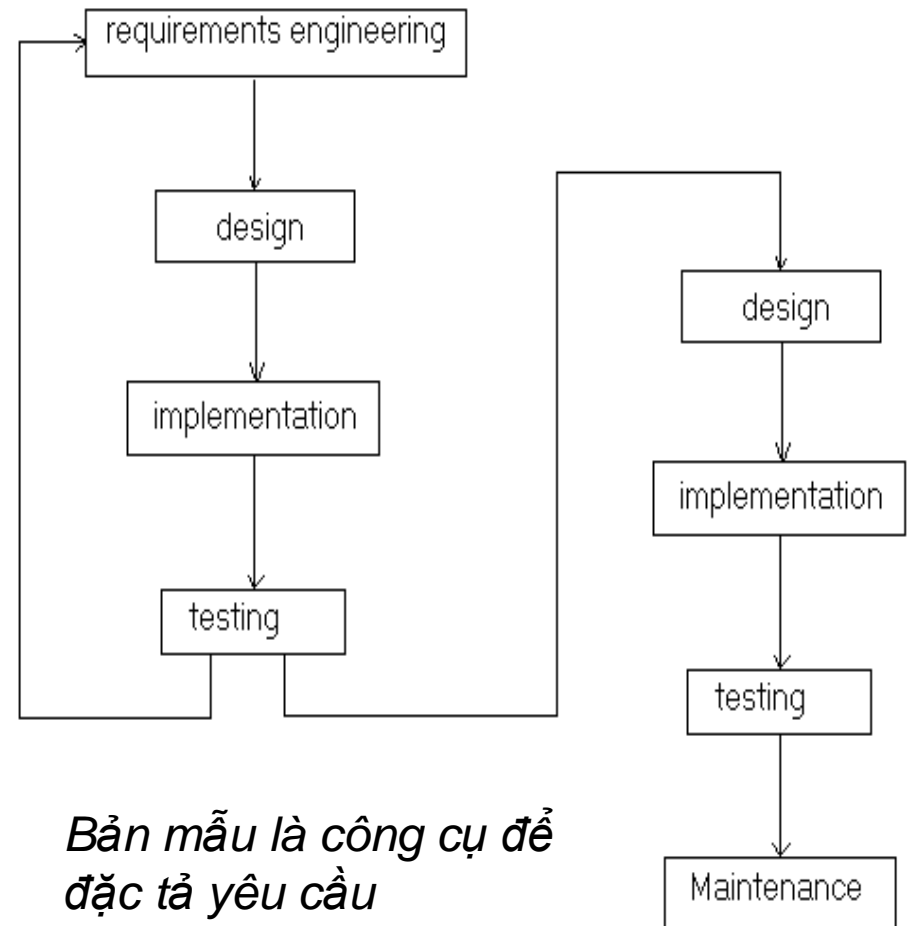
Mô hình chữ V



Mô hình bản mẫu (Prototyping Model)



- Khó khăn để có 1 nhận thức đầy đủ về hệ thống → làm bản mẫu.
 - Một mô hình về một phần của hệ thống
 - Nhấn mạnh vào một vài khía cạnh nào đó
 - Tạo ra một bản mẫu # làm “bản thật”
 - Làm nhanh
 - Rẻ
 - Thể hiện được ý tưởng trước khi đầu tư lớn
- Dùng ngôn ngữ cấp cao
- Phát triển một hệ thống với ít chức năng



Bản mẫu là công cụ để đặc tả yêu cầu

Mô hình bản mẫu



- 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.
- Mô hình có thể lấy một trong 3 dạng:
 - Bản mẫu trên giấy hay PC.
 - Bản mẫu là việc cài đặt tập con các chức năng.
 - Bản mẫu là chương trình đã có.

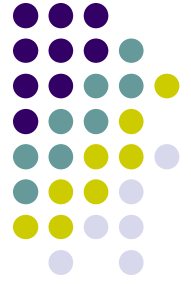
Mô hình bản mẫu

- Ưu điểm

- Hệ thống thật là dễ dùng hơn
- Dễ thỏa mãn yêu cầu người dùng
- Các vấn đề dễ được phát hiện sớm
- Thiết kế có chất lượng cao hơn
- Hệ thống thật dễ bảo trì hơn
- Tiết kiệm công sức phát triển hệ thống.

- Nhược điểm

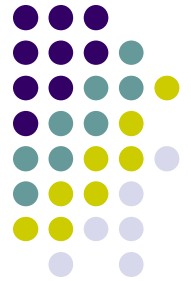
- Hệ thống thật có nhiều chức năng hơn
- Đòi hỏi đội ngũ phát triển nhiều kinh nghiệm hơn.



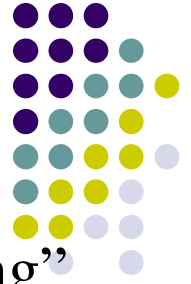
Mô hình bản mẫu

Khuyến cáo cho việc dùng mô hình bảng mẫu

- Yêu cầu của người dùng không rõ ràng
- Cần minh họa cao về giao diện người dùng
- Người dùng phải ý thức về sự thay đổi yêu cầu là rất khó khăn.
Bản mẫu không làm nâng cao chất lượng hệ thống.
- Việc làm bản mẫu phải có kế hoạch và được kiểm soát tiến độ



Mô hình tăng trưởng (Incremental Development model)



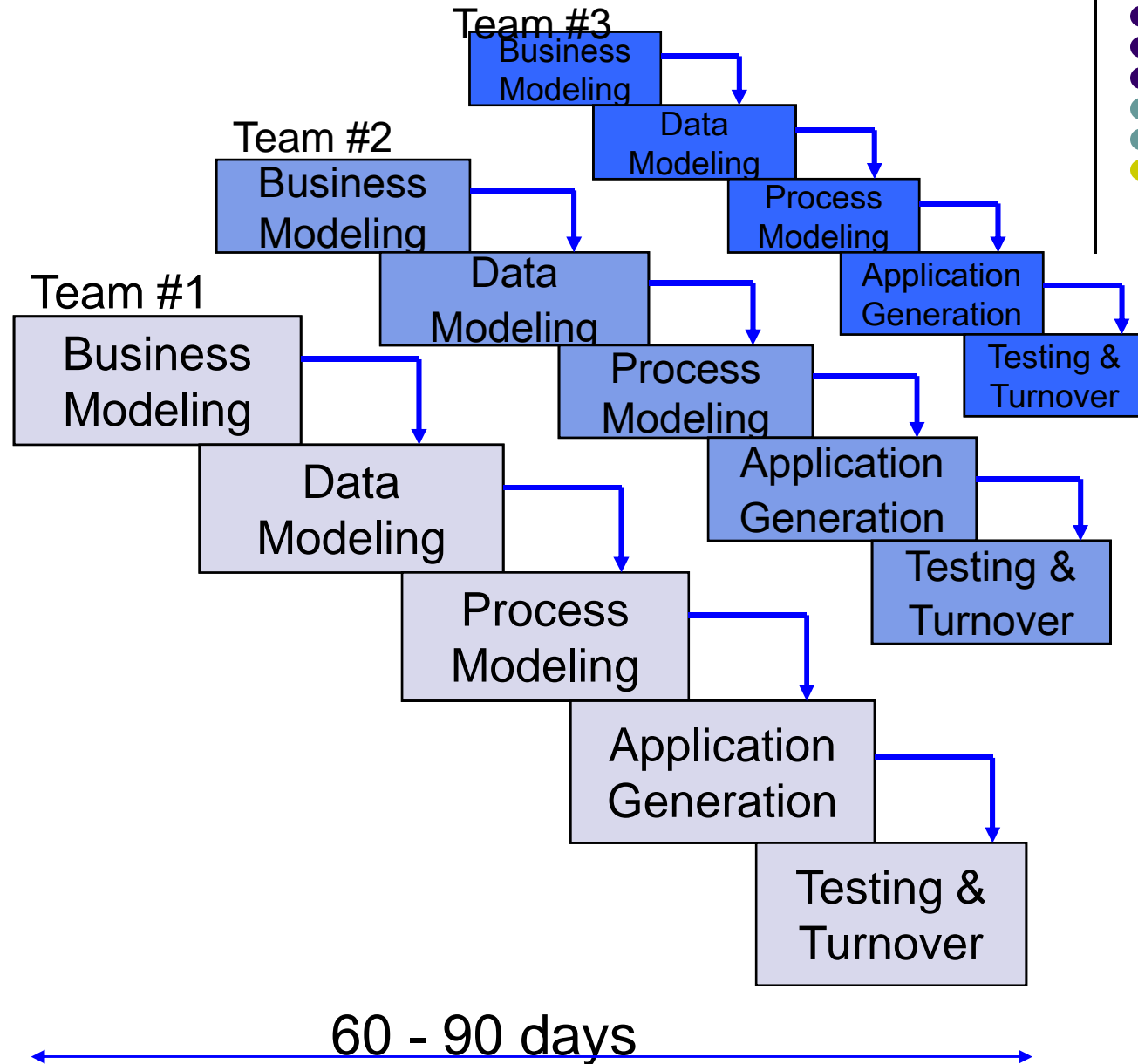
- Chức năng của hệ thống được xây dựng và chuyển giao dần dần cho người dùng. Bắt đầu từ trạng thái hiện tại dần đến trạng thái mong muốn: từng bước nhỏ.
- Mô hình tăng trưởng
 - Tránh bị “big bang”: một thời gian dài chẳng có gì, dùng một cái, cả một hệ thống mới.
 - Người dùng tham gia tích cực vào việc lập kế hoạch cho bước tiếp theo
- Tránh “dư thừa chức năng” (overfunctionality)
 - Yêu cầu quá nhiều
 - Quá dư thừa chức năng sẽ làm hệ thống phức tạp và khó sử dụng
- Người phân tích dễ dàng ước lượng thời gian, công sức xây dựng một chức năng, đặc tính nào đó của phần mềm.
- Cách tiếp cận tăng trưởng giúp tập trung vào các điểm cốt lõi và các chức năng cần thiết đáp ứng yêu cầu thực tiễn.

Mô hình định khung nhanh (Rapid Application Development: RAD)



- Là tiến trình phát triển phần mềm gia tăng, 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 đánh giá (Business, Data, Process, Appl. Generation, Testing)

Mô hình định khung nhanh





Mô hình định khung nhanh

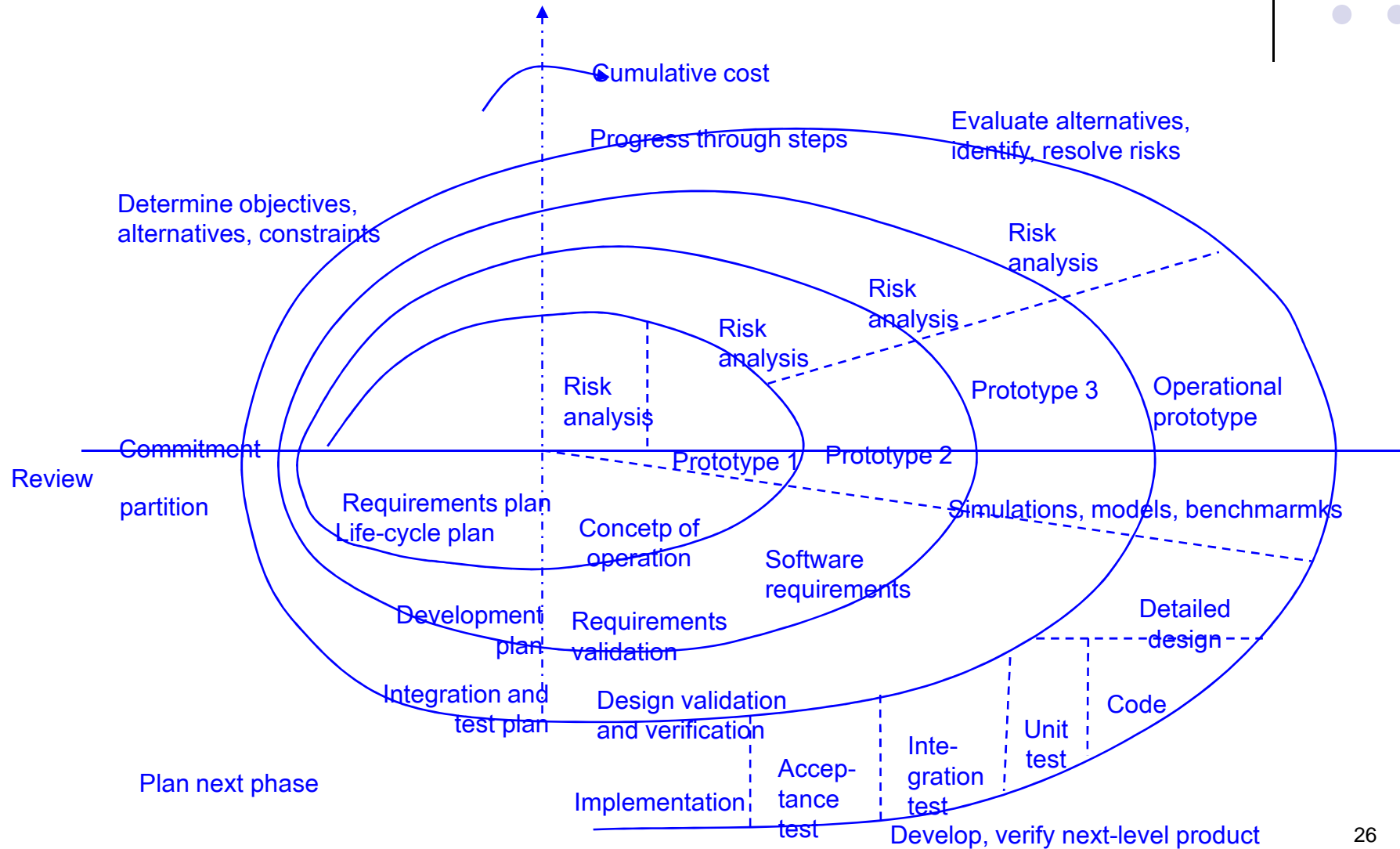
- Cần nguồn nhân lực dồi dào để tạo các nhóm cho các chức năng chính
- Yêu cầu hai bên cam kết trong thời gian ngắn phải có phần mềm hoàn chỉnh, thiếu trách nhiệm của một bên dễ làm dự án đổ vỡ
- RAD không phải tốt cho mọi ứng dụng, nhất là với ứng dụng không thể module hóa hoặc đòi hỏi tính năng cao

Mô hình xoắn ốc (Spiral Model)

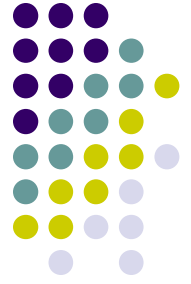


- Được đề nghị bởi Boehm (1988)
- Kết hợp các hoạt động phát triển với sự quản lý rủi ro để giảm đến mức tối thiểu và kiểm soát các rủi ro
- Mô hình được trình bày ở dạng xoắn ốc trong đó mỗi lần lặp được biểu diễn bởi một đường vòng quanh.
- Bốn hoạt động chính:
 - Lập kế hoạch(xác định các mục tiêu, các lựa chọn và các ràng buộc)
 - Phân tích rủi ro (phân tích các phương án và xác định/giải quyết rủi ro)
 - Kỹ Nghệ (phát triển sản phẩm)
 - Đánh giá của khách hàng(khẳng định kết quả của kỹ nghệ)

Mô hình xoắn ốc

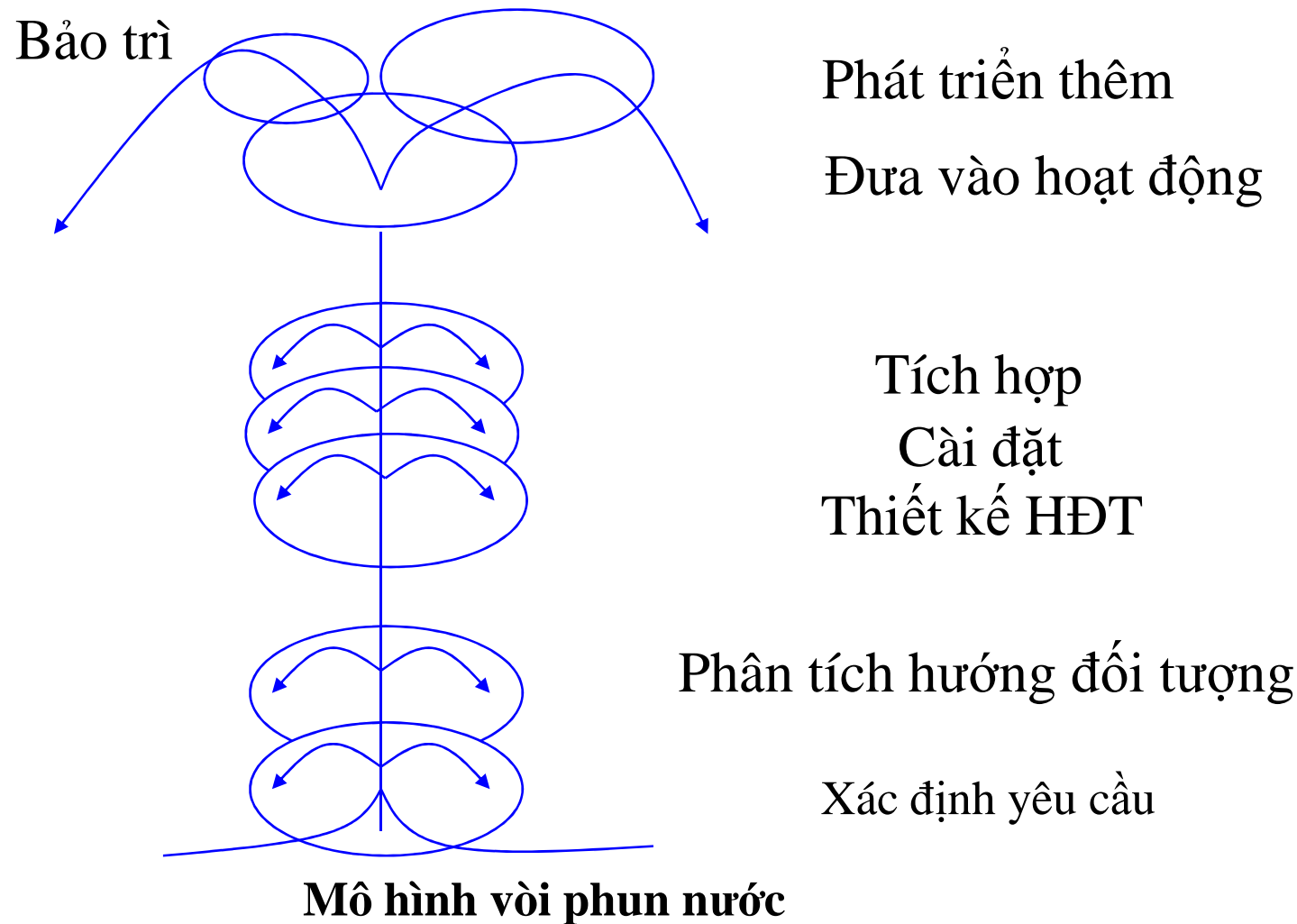
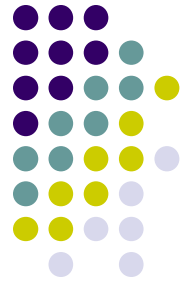


Mô hình xoắn ốc



- Hướng rủi ro (risk-driven); giảm thiểu rủi ro
 - Các công việc luân phiên và chịu các ràng buộc đã hỗ trợ cho việc tái sử dụng phần mềm hiện có
 - Đánh giá mức độ rủi ro
 - Mục tiêu quan trọng luôn là chất lượng phần mềm
 - Giảm nhẹ kiểm thử và nhanh chóng sửa chữa những lỗi xảy ra
 - Bảo trì đơn giản chỉ là một vòng tròn trong xoắn ốc, như vậy không có sự phân biệt giữa phát triển và bảo trì
- Dành riêng cho các phần mềm nội bộ có kích thước lớn
- Có thể chấm dứt do các đánh giá về rủi ro, do đó sẽ rất không hay khi đã ký kết các hợp đồng, rắc rối về mặt luật pháp
- Kích thước sản phẩm ảnh hưởng đến giá thành việc phân tích rủi ro

Mô hình hướng đối tượng (object-oriented life-cycle model)



Mô hình hướng đối tượng (object-oriented life-cycle model)



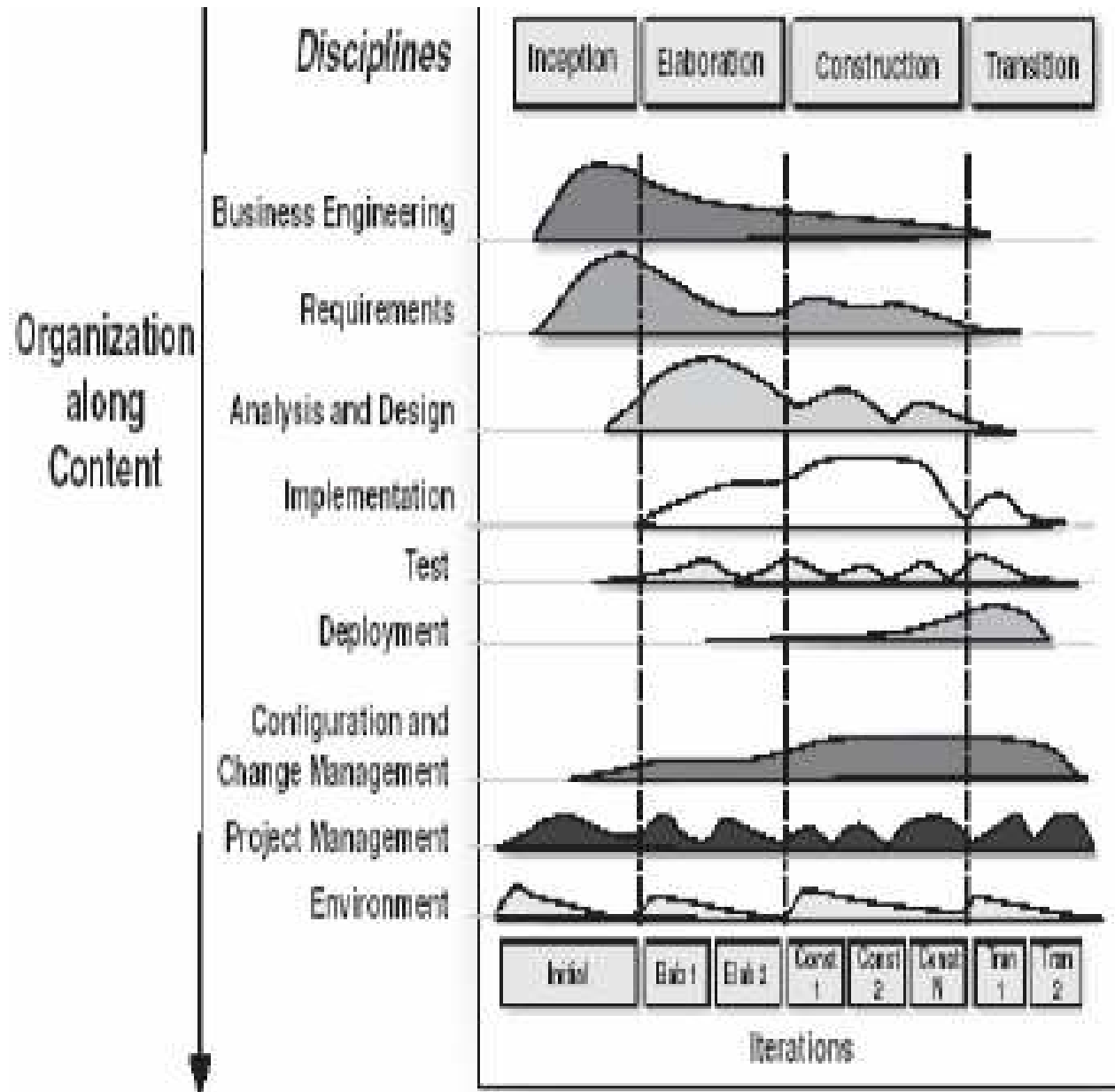
- Đặc tính quan trọng nhất là lập:
 - giữa các giai đoạn
 - một phần trong giai đoạn
- Mô hình vòi phun nước thể hiện các giai đoạn gói lên nhau
- Giảm bớt nhân lực cho công tác bảo trì

RUP – Rational Unified Process

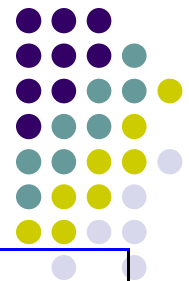


- Bổ sung cho UML
- Cách tiếp cận lặp cho các hệ thống hướng đối tượng, bao gồm các use case để mô hình hóa các yêu cầu
- Các giai đoạn của RUP
 - *Bắt đầu*: thiết lập phạm vi, giới hạn, các use case quan trọng, các kiến trúc ứng viên, các dự đoán về chi phí và kế hoạch làm việc
 - *Sửa soạn*: cơ sở của kiến trúc, thiết lập sự hỗ trợ công cụ
 - *Xây dựng*: sản xuất tiến trình, một hay nhiều sự phát hành
 - *Chuyển tiếp*: phát hành ra cộng đồng người dùng, thường là một số phát hành

RUP

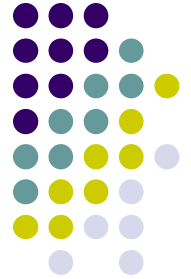


So sánh các mô hình



Mô hình	Điểm mạnh	Điểm yếu
Mô hình xây dựng và hiệu chỉnh	Tốt đối với các chương trình ngắn không yêu cầu về bảo trì	Không đáp ứng được các chương trình tương đối lớn trở đi
Mô hình thác nước	Tiếp cận có kỷ luật Hướng tài liệu	Sản phẩm chuyển giao có thể không theo những gì khách hàng cần
Mô hình định khung nhanh	Đảm bảo sản phẩm được chuyển giao có được những gì khách hàng cần	Xem nhẹ tài liệu, khó bảo trì
Mô hình xoắn ốc	Kết hợp nhiều đặc điểm của tất cả các mô hình phía trên	Chỉ có thể sử dụng cho các sản phẩm có kích thước lớn hay cho các tổ chức Các nhà phát triển phải có khả năng phân tích rủi ro và giải quyết rủi ro
Các mô hình hướng đối tượng	Hỗ trợ việc lặp lại bên trong các giai đoạn, song song hóa giữa các giai đoạn	Có thể suy thoái thành CABTAB (thuật ngữ về sự thiếu kỷ luật trong công việc: trình tự thực hiện các công việc lung tung, bừa bãi)
So sánh giữa các mô hình tiến trình phần mềm		

Bài tập



- Tìm hiểu về các phương pháp ước lượng phần mềm.