

CÔNG NGHỆ PHẦN MỀM

CHƯƠNG 2

Quy trình & Mô hình

Th.S Nguyễn Thị Thanh Trúc

NỘI DUNG

1. Quy trình phát triển phần mềm
2. Mô hình quy trình phần mềm

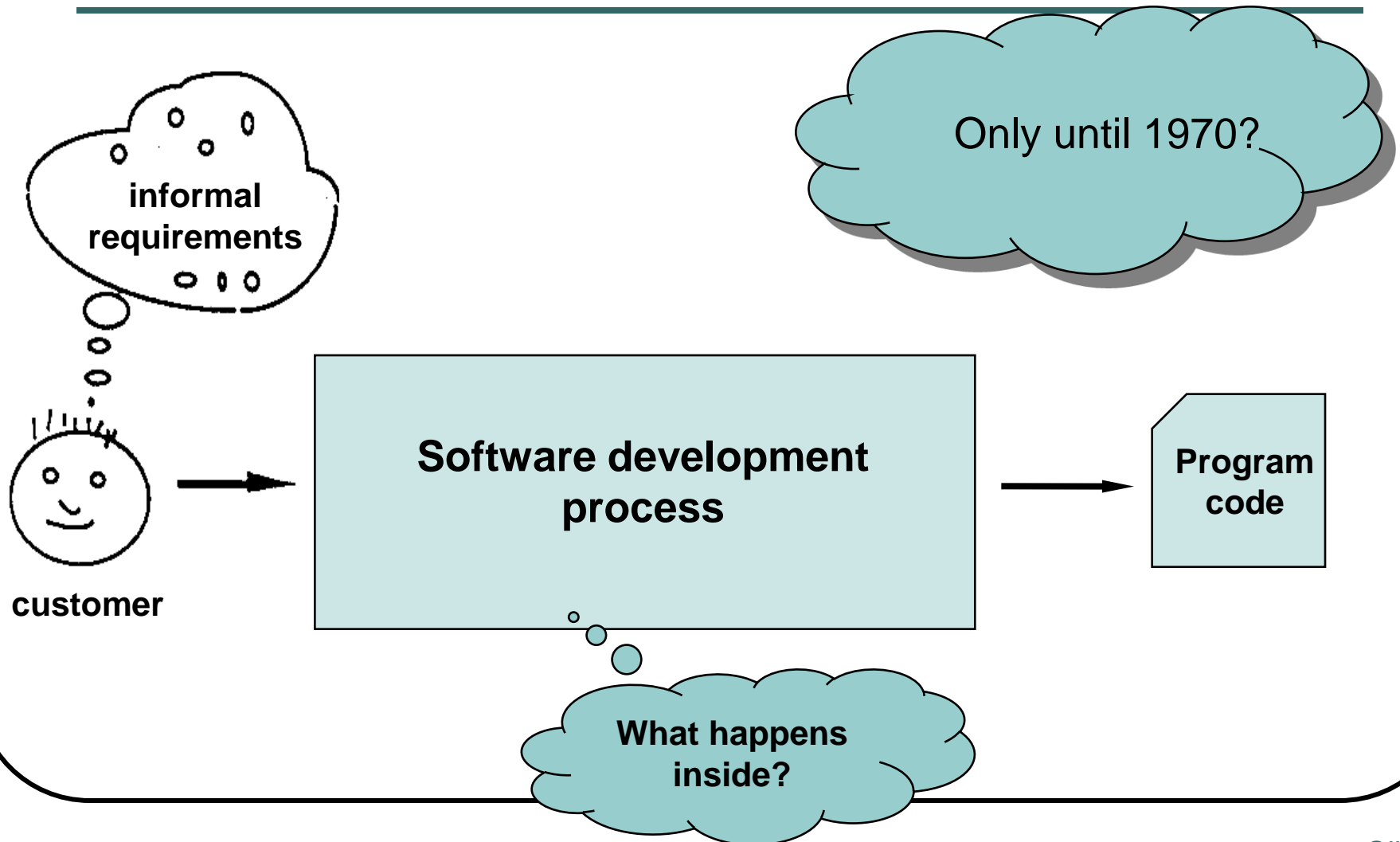
QUI TRÌNH PHÁT TRIỂN – Ý nghĩa

- ❑ Quyết định chất lượng sản phẩm phần mềm
- ❑ Đồng bộ hóa hoạt động trong dự án

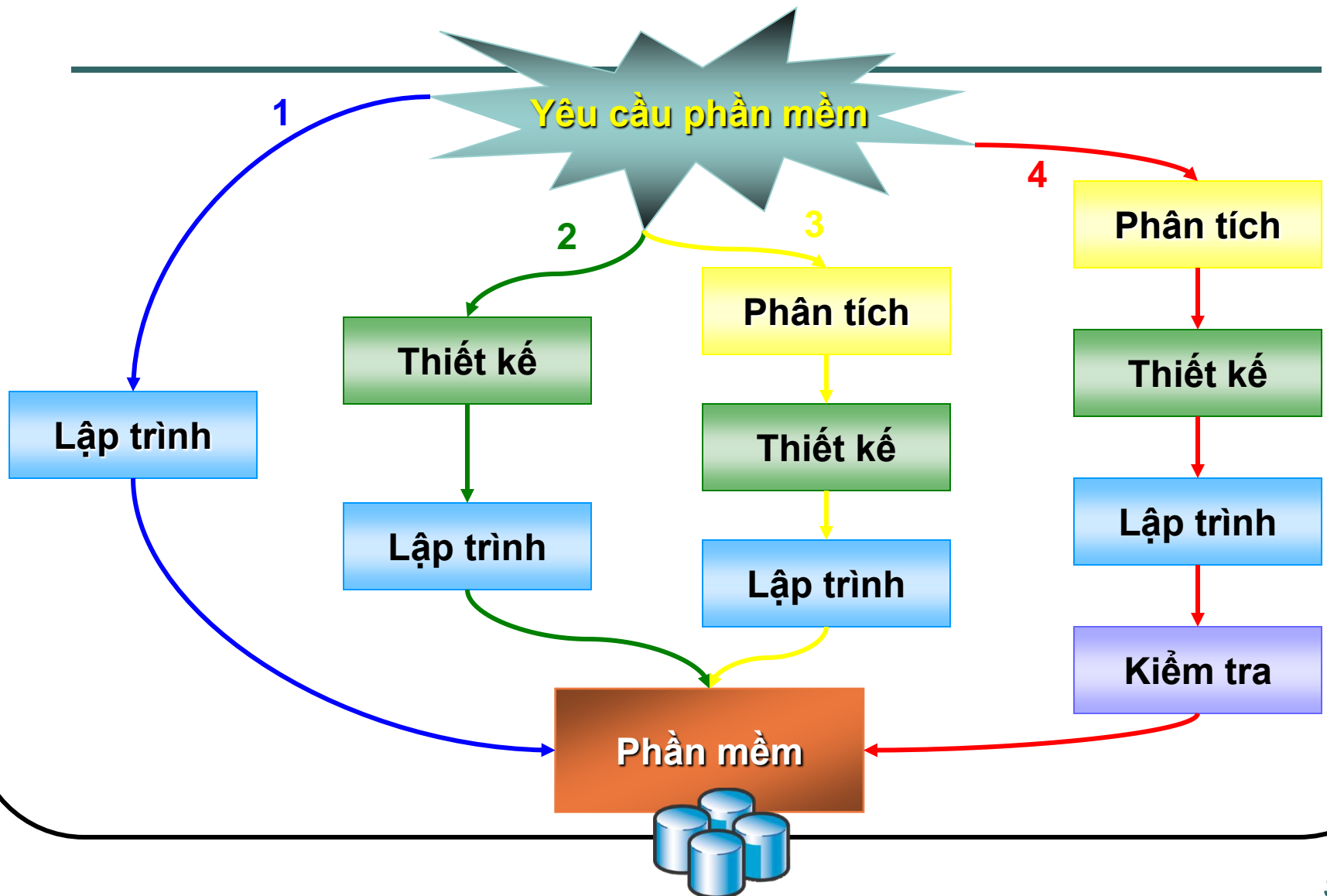
KHÔNG CÓ QUI TRÌNH VẠN NĂNG

CÓ CÁC HOẠT ĐỘNG CHUNG CỦA MỌI QUI TRÌNH

Software development process until 1970: considered as a black box

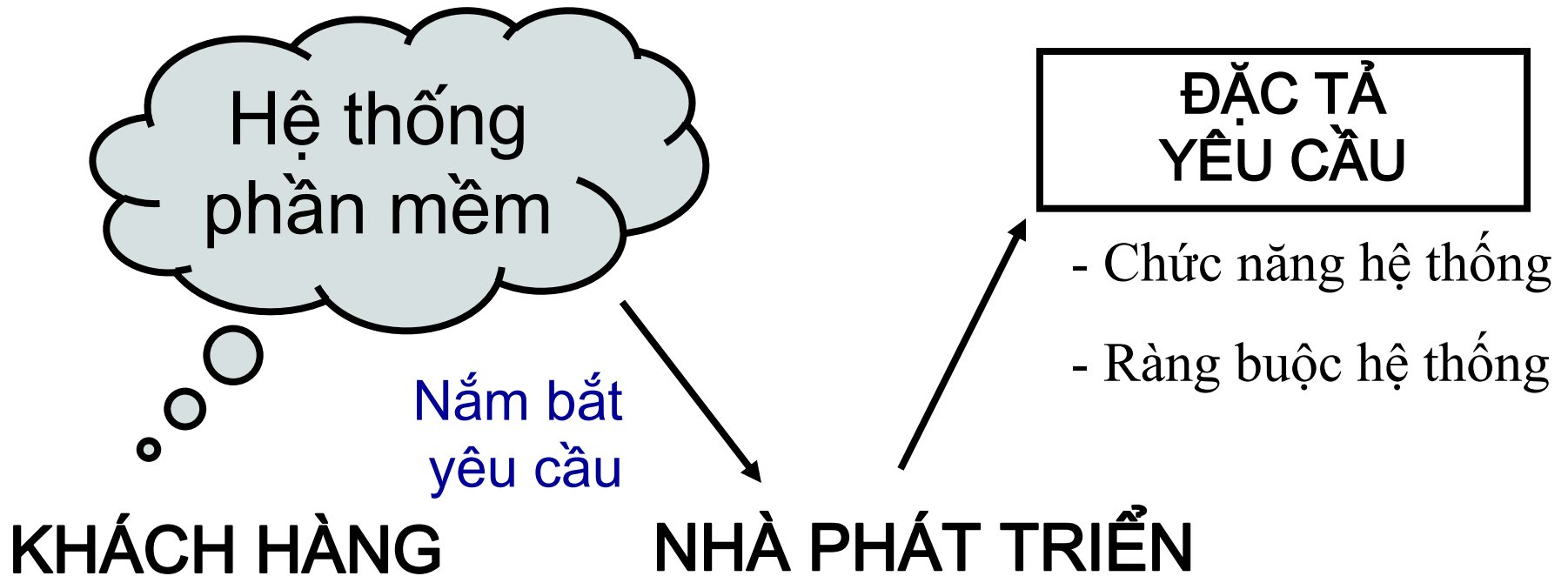


Quy trình Công nghệ phần mềm



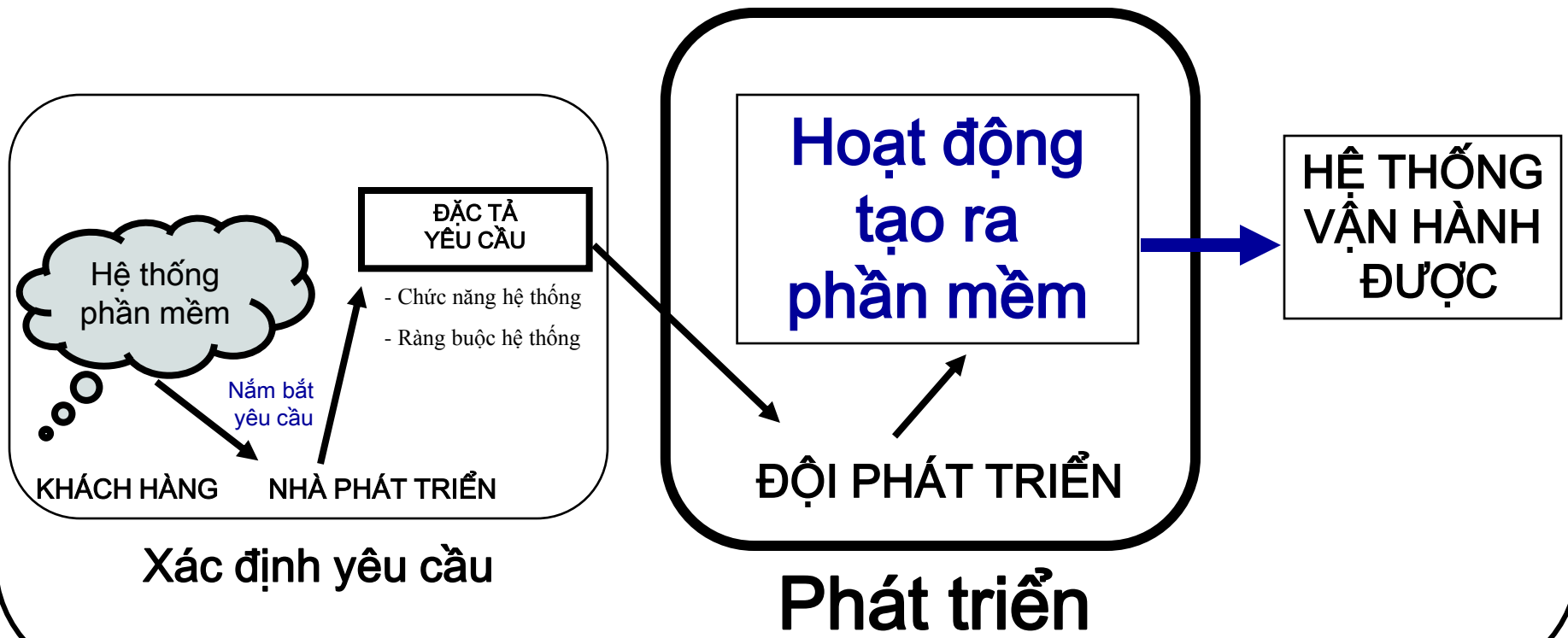
QUI TRÌNH PHÁT TRIỂN – Hoạt động chính

1. Xác định yêu cầu (Requirements)



QUI TRÌNH PHÁT TRIỂN – Hoạt động chính

2. Phát triển phần mềm (Development)



QUI TRÌNH PHÁT TRIỂN – Hoạt động chính

3. Kiểm thử phần mềm (Validation/Testing)

Xác định yêu cầu



Phát triển

Đảm bảo hệ thống
đáp ứng đòi hỏi
trong đặc tả yêu cầu

Kiểm thử phần mềm

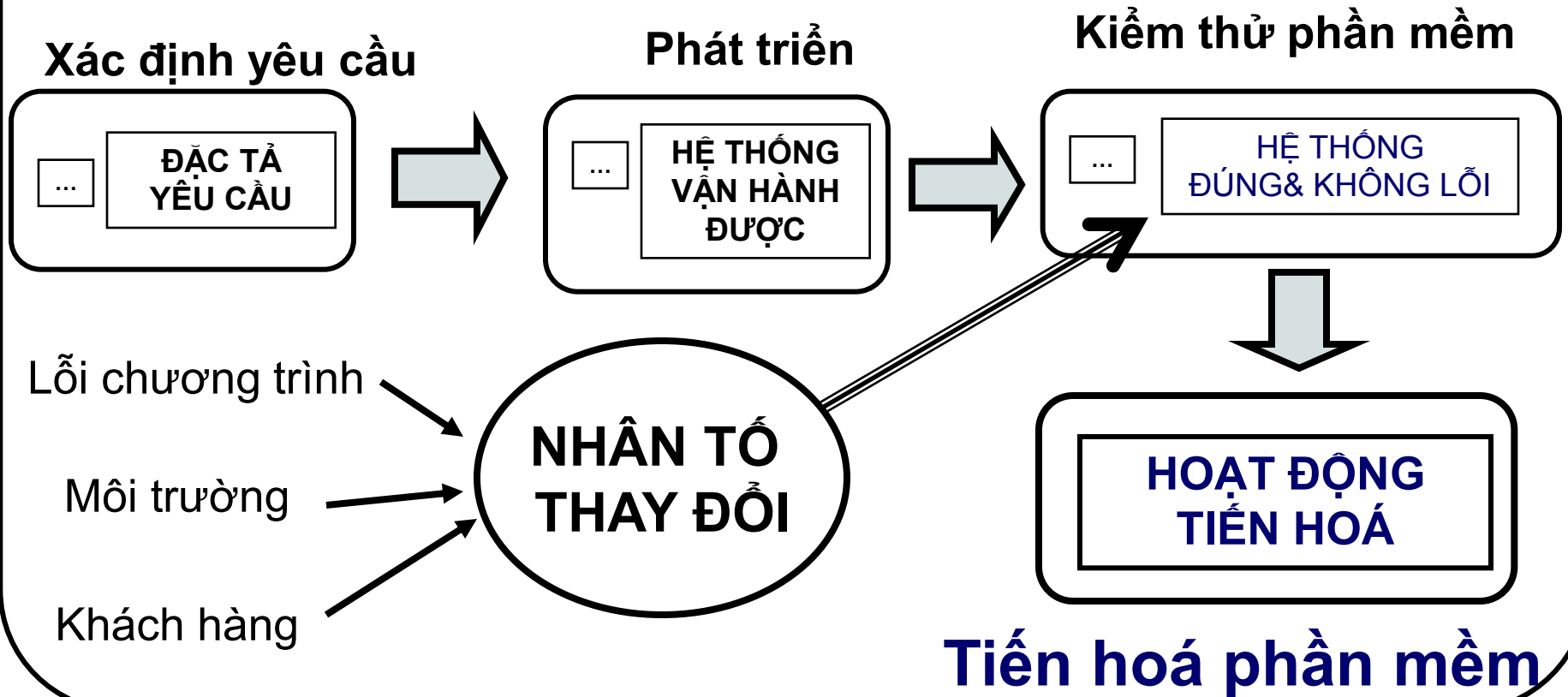
PRODUCT RIGHT

**HỆ THỐNG
ĐÚNG &
KHÔNG LỖI**

RIGHT PRODUCT

QUI TRÌNH PHÁT TRIỂN – Hoạt động chính

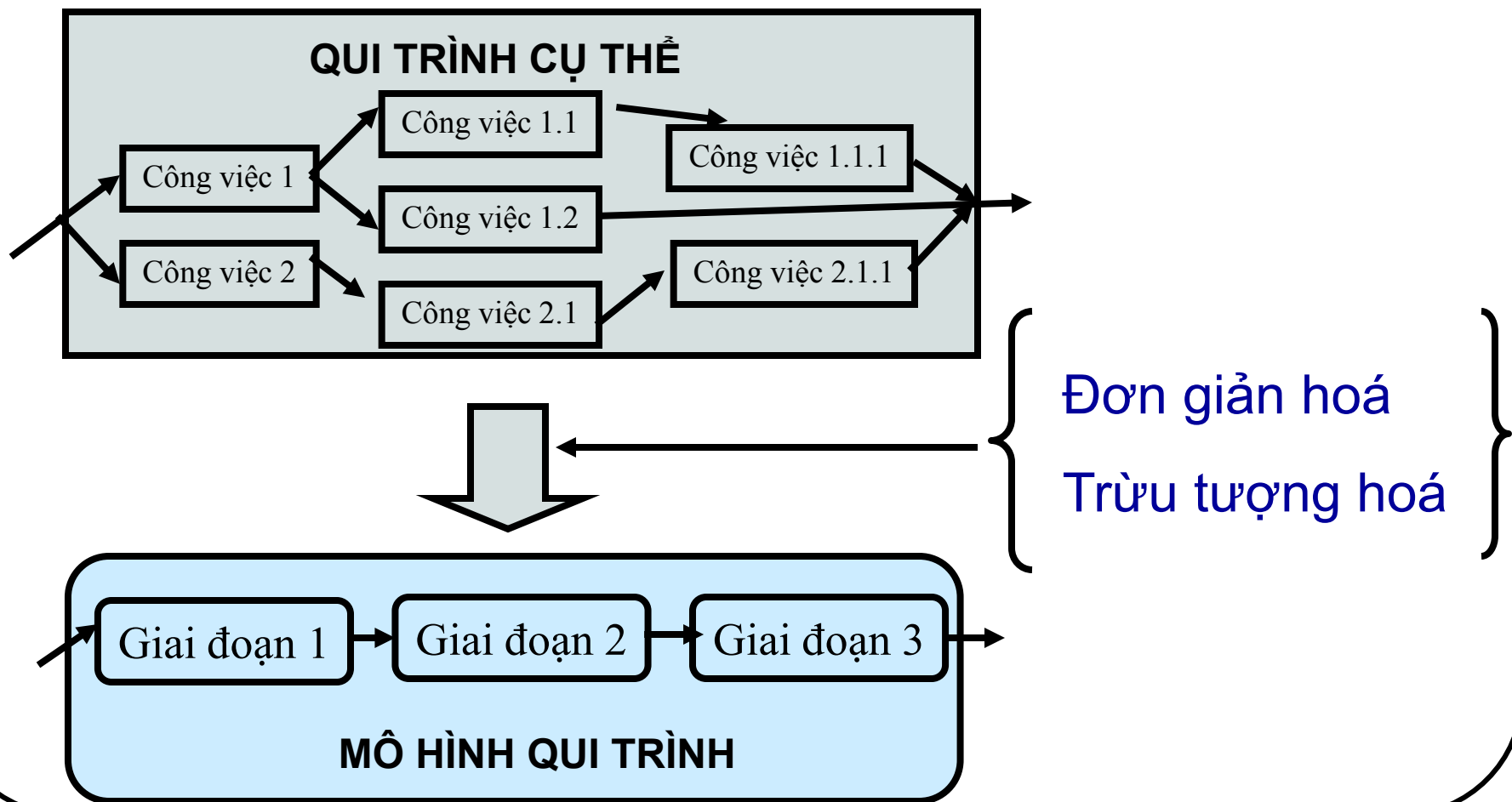
4. Tiến hóa phần mềm (Evolution)



QUI TRÌNH– Cải tiến nâng cao chất lượng

- ❑ Quy trình khung là cơ sở để cải tiến quy trình nâng cao chất lượng, năng suất
- ❑ Quy trình khung phổ biến (Các chuẩn)
 - ISO
 - CMM (Capability Maturity Model)
 - CMMI (Capability Maturity Model Integration)

MÔ HÌNH QUI TRÌNH – Khái niệm



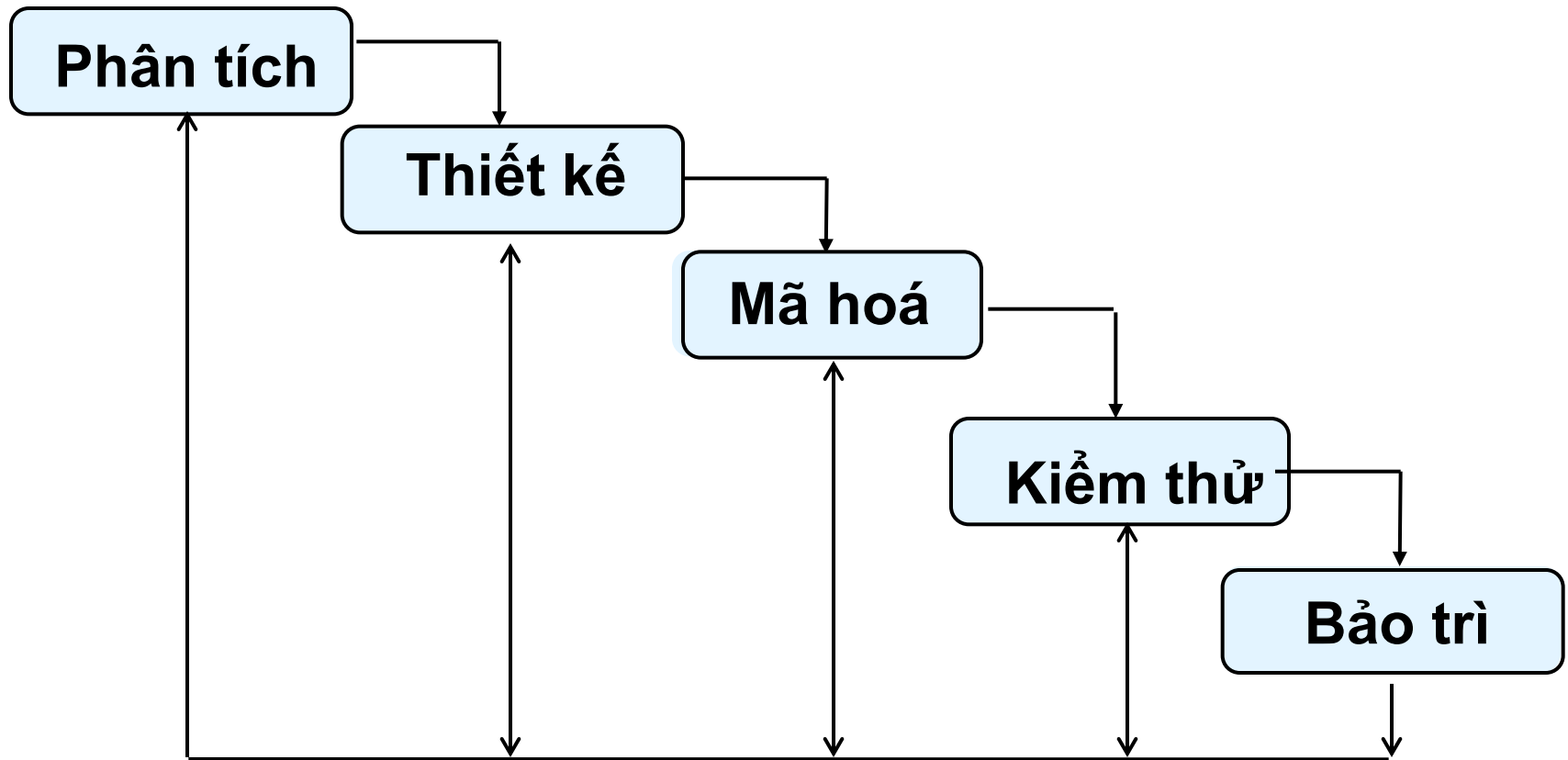
MÔ HÌNH QUI TRÌNH

Một số mô hình qui trình phổ biến

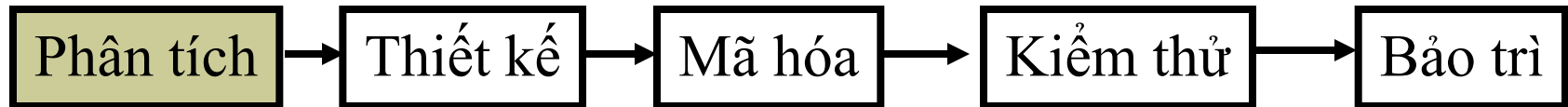
- ☐ Mô hình thác nước (*Water Fall Model*)
- ☐ Phát triển tiến hóa (*Evolutionary development*)
- ☐ Phát triển hệ thống hình thức (*Formal systems development*)
- ☐ Phát triển dựa trên sử dụng lại (*Reuse-based development*)
- ☐ Mô hình phát triển phần mềm hướng thành phần (Component)
- ☐ Mô hình công nghệ thứ tư
- ☐ Mô hình phát triển ứng dụng nhanh (Rapid prototyping Model)
- ☐ Mô hình phát triển lặp tăng cường (*Incremental development*)
- ☐ Mô hình lập trình cực đoan (XP-eXtreme Programming)/Agile

MÔ HÌNH THÁC NƯỚC

(Water Fall Model)



PHÂN TÍCH

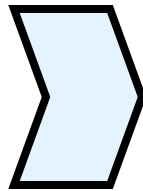


- Xác định và phân tích các yêu cầu cho hệ thống

- Chức năng
- Ràng buộc

Đối tác tham gia:

khách hàng
người dùng
người phân tích



TÀI LIỆU YÊU CẦU

- Danh sách yêu cầu
- Mô tả chi tiết yêu cầu

What to do?

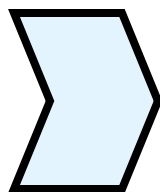
THIẾT KẾ



- Xây dựng giải pháp thiết kế cho yêu cầu phần mềm

Đầu vào:

tài liệu yêu cầu
môi trường triển khai



Đầu ra

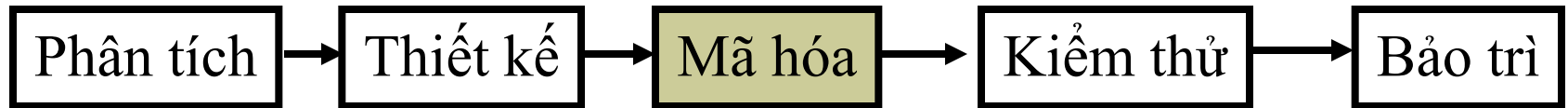
tài liệu thiết kế

TÀI LIỆU THIẾT KẾ

- Mô hình kiến trúc hệ thống
- Mô hình thiết kế chi tiết
 - Dữ liệu
 - Thuật toán
 - Giao diện

How to do?

MÃ HÓA



- Viết chương trình
- Kiểm tra, giám sát mã lệnh
- Gỡ lỗi (*Debugging*)

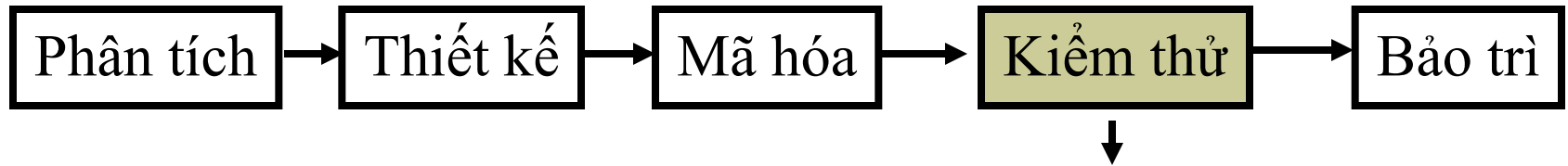
Đầu vào

tài liệu thiết kế

Đầu ra

chương trình thực hiện được
tài liệu chương trình

KIỂM THỬ



- Phát hiện và sửa lỗi phần mềm
- Đảm bảo phần mềm thỏa mãn yêu cầu khách hàng

Đầu vào:

tài liệu yêu cầu

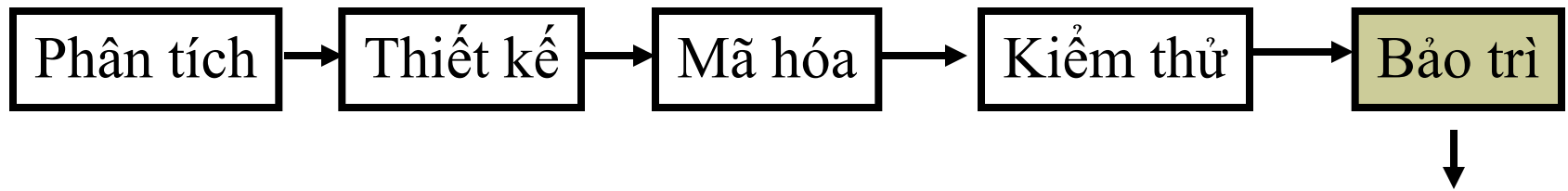
tài liệu thiết kế

chương trình

tài liệu chương trình

Đầu ra: tài liệu kết quả kiểm thử

BẢO TRÌ



- Đưa hệ thống vào vận hành (triển khai)
- Sửa lỗi phần mềm
- Làm thích nghi phần mềm với môi trường mới
- Thay đổi phần mềm đáp ứng yêu cầu mới

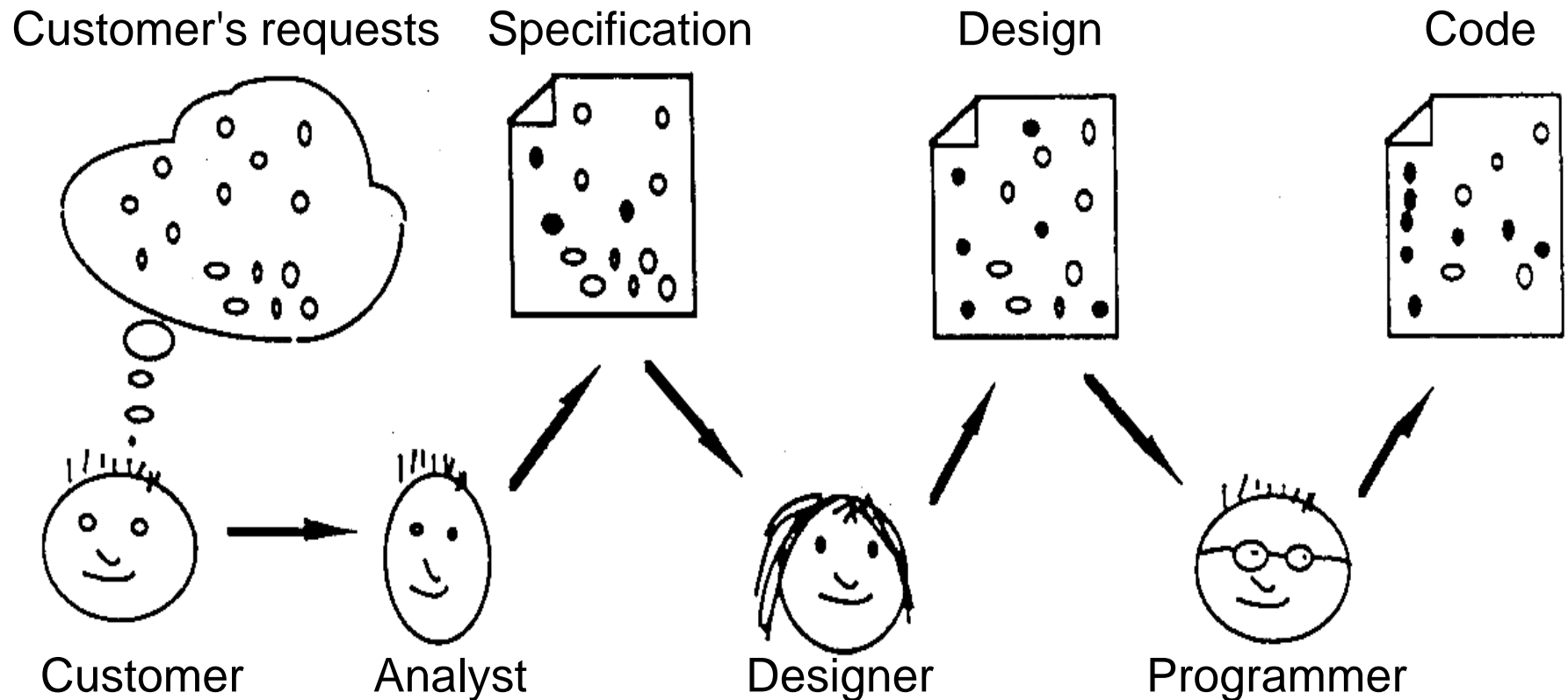
ƯU ĐIỂM MÔ HÌNH THÁC NƯỚC

- Xuất hiện sớm nhất
- Các pha được xác định rõ ràng (đầu vào/ra)
- Thấy được trình tự công nghệ từ đầu đến sản phẩm cuối
- Bảo trì thuận lợi
- Thích hợp khi yêu cầu hiệu tốt

NHƯỢC ĐIỂM MÔ HÌNH THÁC NƯỚC

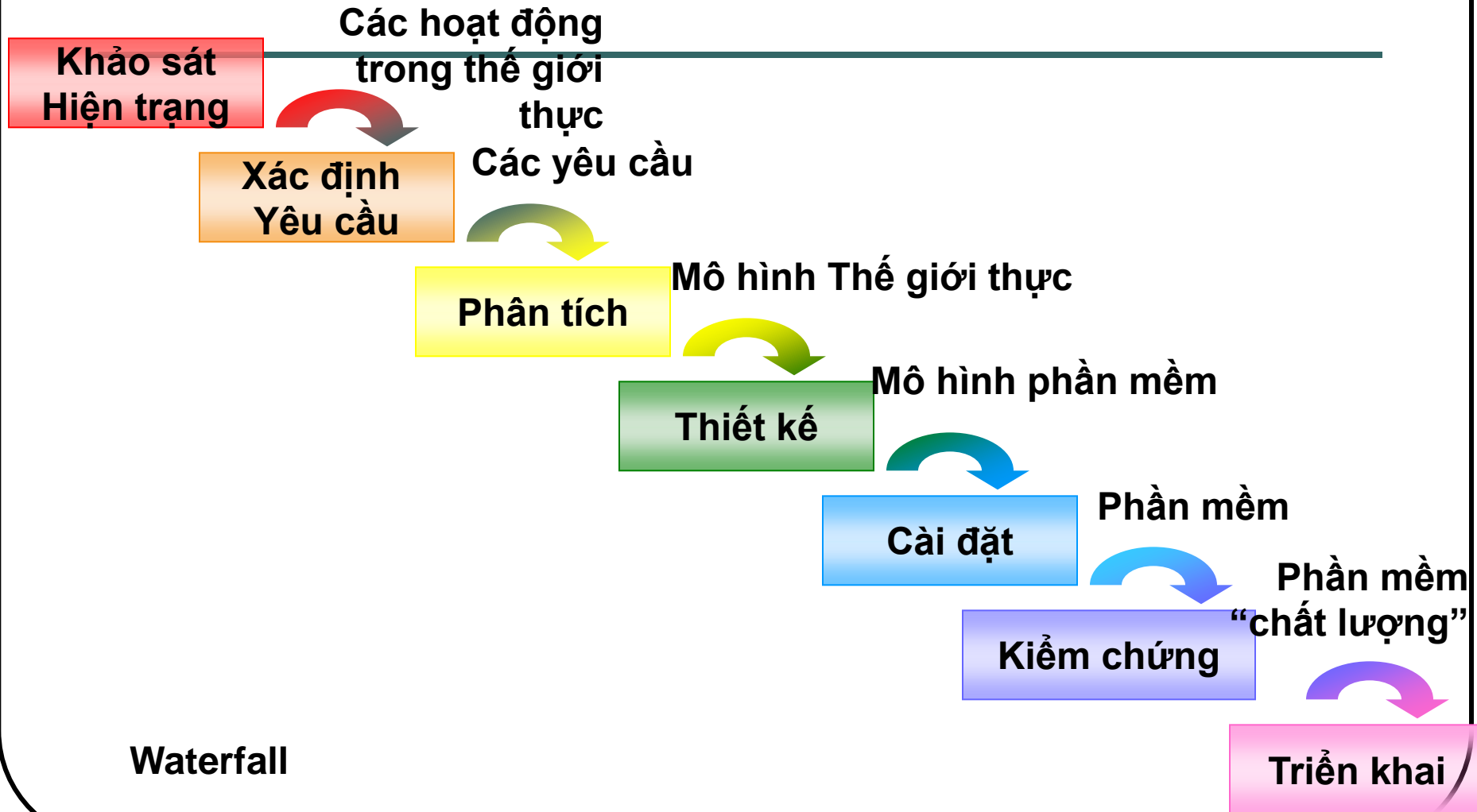
- Tách biệt giữa các pha, tiến hành tuần tự
 - Khó tuân thủ tuần tự: dự án lớn thường phải lặp lại
 - Khó đáp ứng yêu cầu thay đổi của khách hàng
- Sai sót phát hiện muộn có thể là thảm họa
- Chậm có phiên bản thực hiện được
 - Đòi hỏi khách hàng phải kiên nhẫn

Documents of SW development

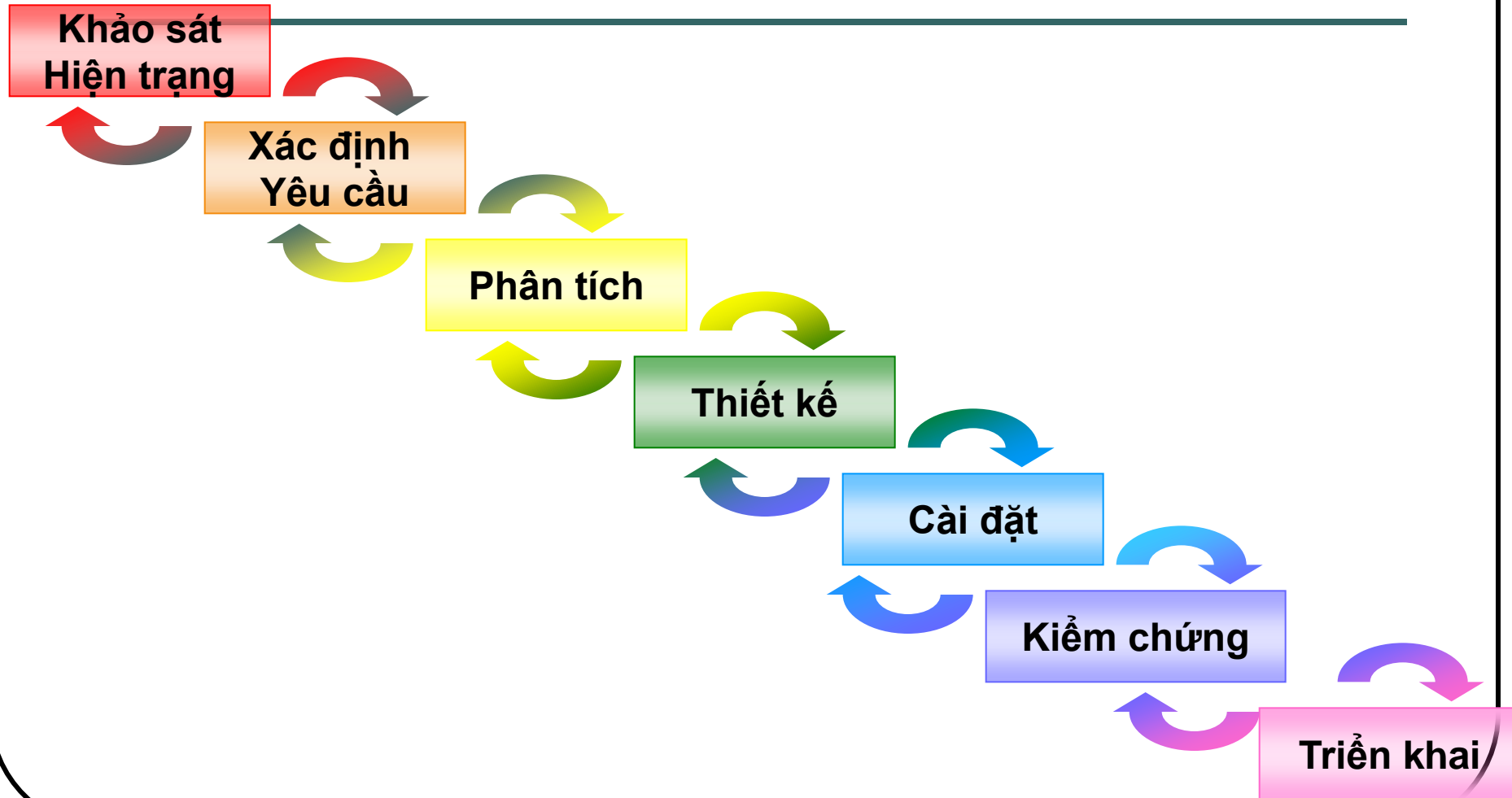


Required features (empty circles) got lost,
unnecessary (filled circles) are added

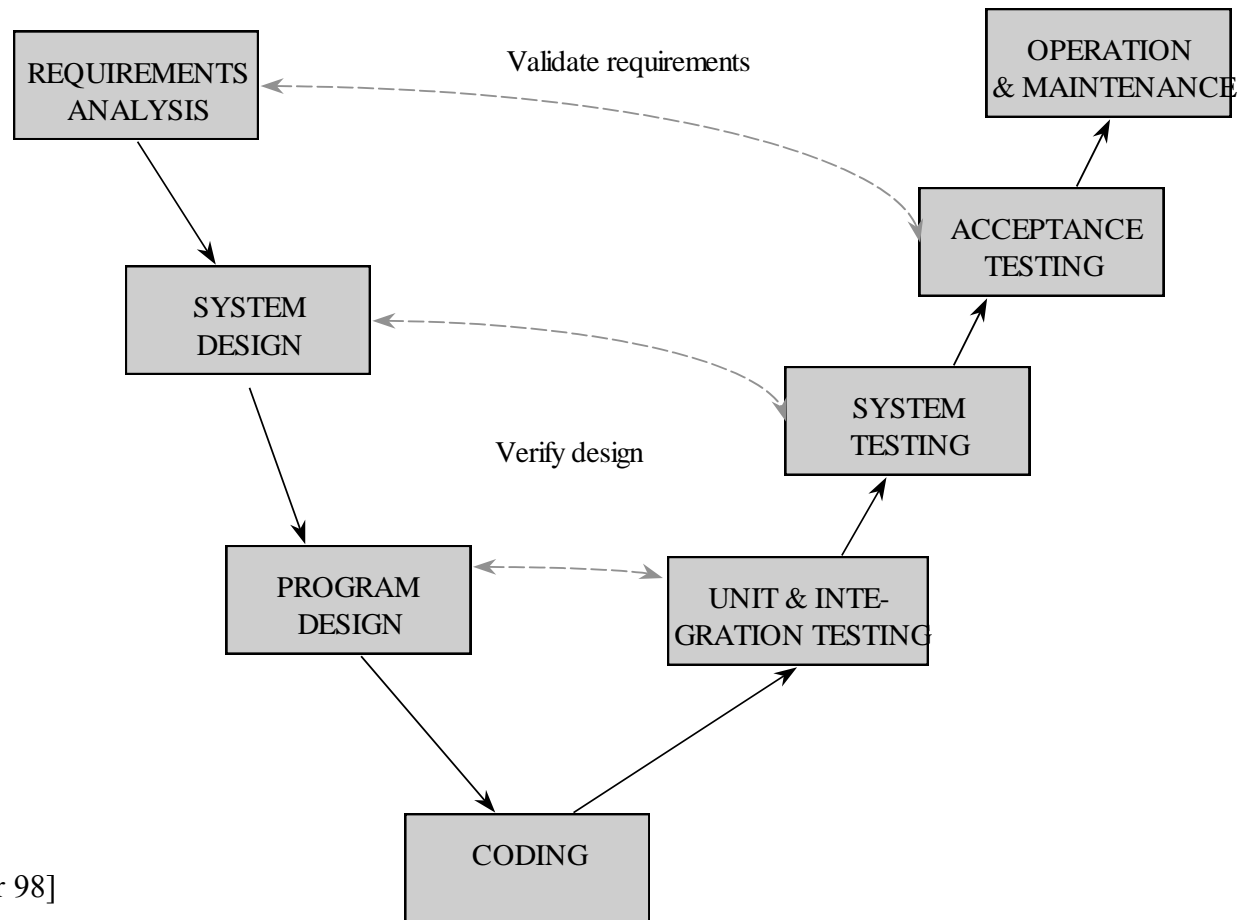
Mô hình thác nước



Mô hình thác nước cải tiến

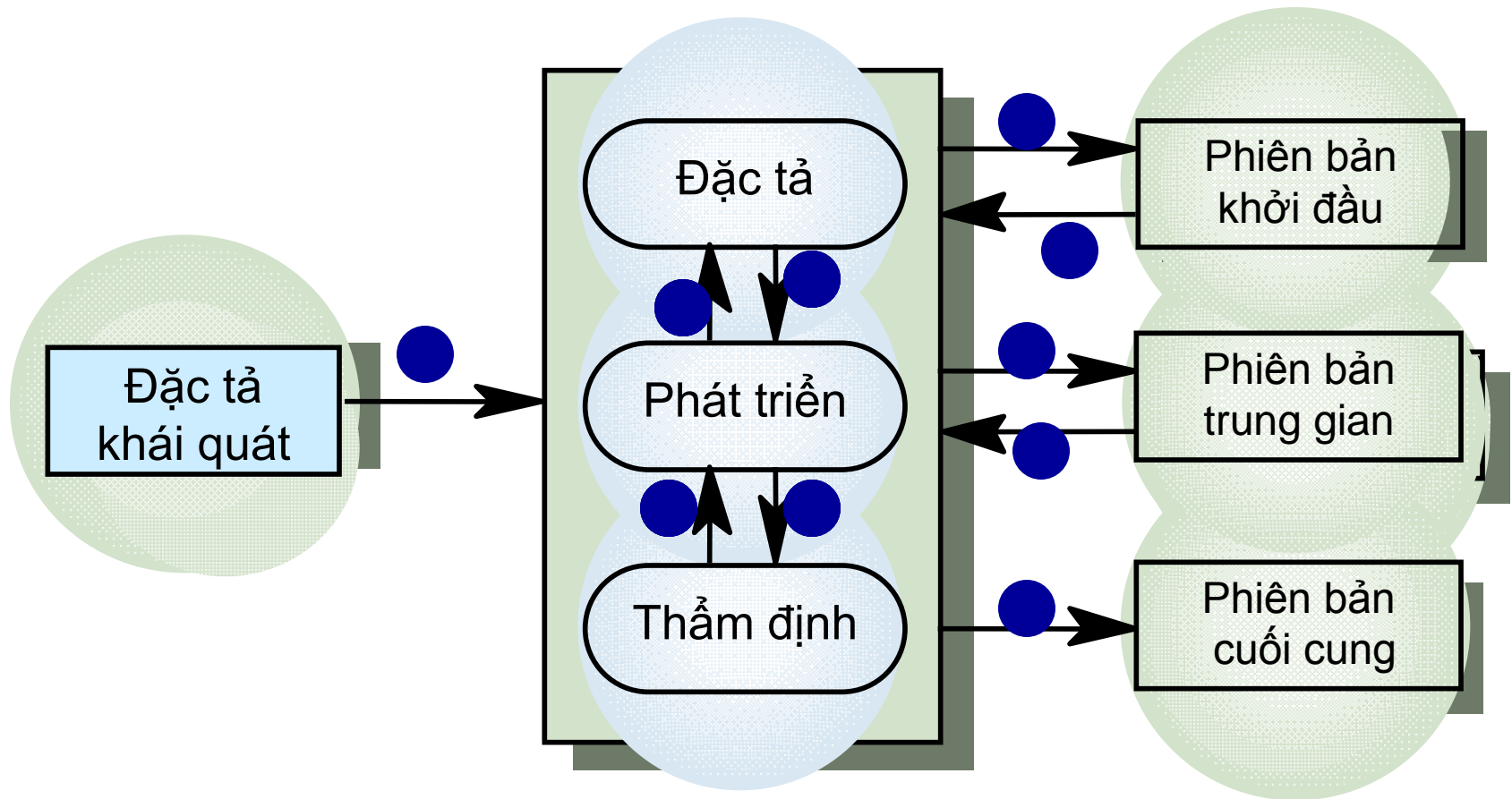


V Model



[Pfleeger 98]

LƯỢC ĐỒ MÔ HÌNH TIẾN HÓA



MÔ HÌNH PHÁT TRIỂN TIẾN HOÁ

1. Mô hình bản mẫu
2. Mô hình xoắn ốc
3. Mô hình tăng trưởng

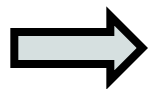
MÔ HÌNH LÀM BẢN MẪU

(*Prototyping model*)

NHU CẦU LÀM BẢN MẪU



- Khách hàng thường khó nói rõ được điều họ mong đợi
- Người phát triển thường hiểu sai yêu cầu khách hàng
- Khách hàng thường phát hiện sai sót khi dùng sản phẩm

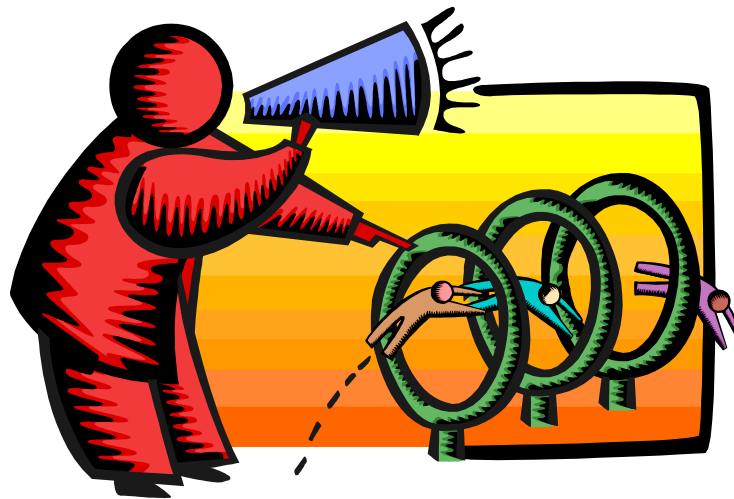


TẠO MÔI TRƯỜNG ĐỂ THÂU TÓM YÊU CẦU

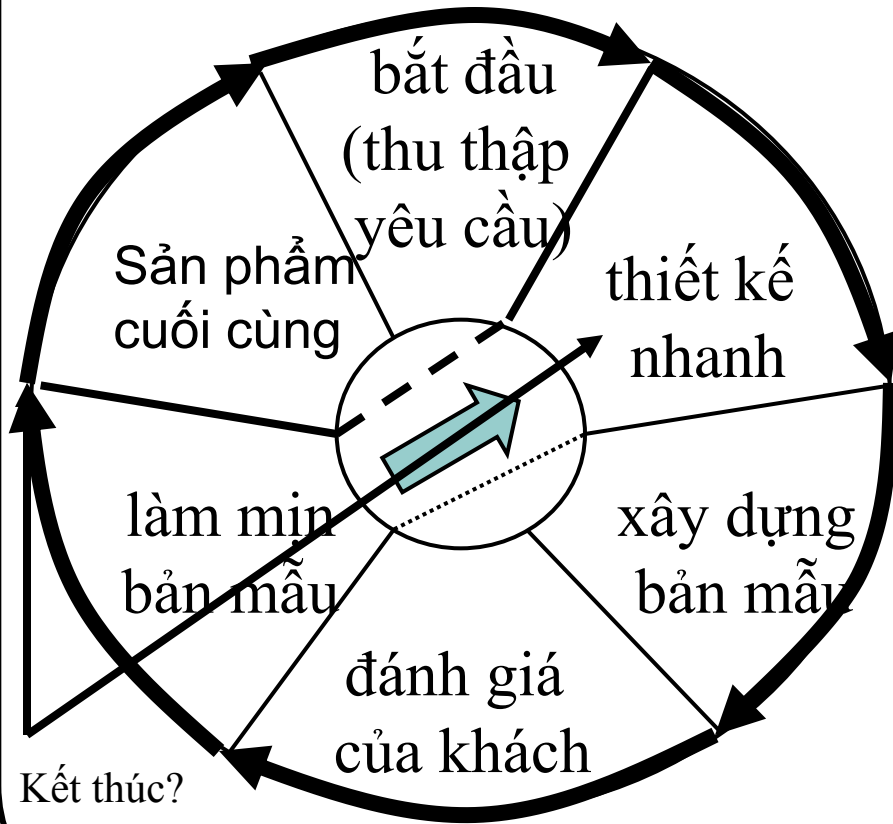
MÔ HÌNH LÀM BẢN MẪU

MỤC TIÊU LÀM MẪU

- Thu tóm yêu cầu người dùng, giảm thiểu sai sót
- Đảm bảo các yêu cầu là những gì khách hàng mong đợi



HOẠT ĐỘNG LÀM BẢN MẪU



- Khởi đầu bằng pha thu thập yêu cầu
- Tiến hành thiết kế nhanh
- Xây dựng bản mẫu
- Đánh giá khách hàng
- Làm mịn bản mẫu
- Nếu chưa được sản phẩm thì chuyển sang thiết kế nhanh và lặp lại

MÔ HÌNH BẢN MẪU – Ví dụ

? Hệ thống Quản lý giao dịch Ngân hàng

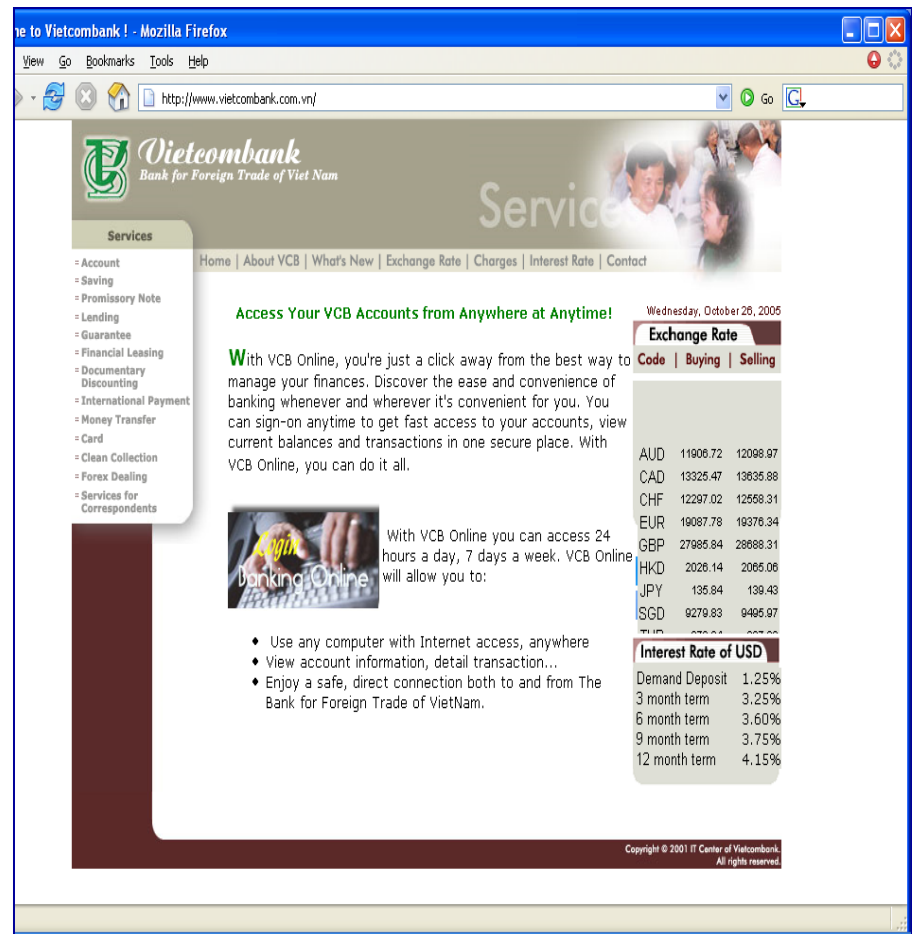
Khung cảnh vấn đề

- Nhân viên ngân hàng nắm rõ nghiệp vụ song chỉ diễn đạt được khi có tình huống
- Người phát triển gặp khó khăn khi thu thập các thông tin nghiệp vụ giao dịch ngân hàng
- Người phát triển không hình dung được nghiệp vụ tổng thể

MÔ HÌNH BẢN MẪU – Ví dụ

Giải pháp tạo mẫu

- Thông tin cho mỗi thao tác nghiệp vụ biểu diễn bằng trang Web tĩnh
- Trình tự nghiệp vụ được biểu diễn thông qua các link



THIẾT KẾ BẢN MẪU

- Là hoạt động quan trọng của quá trình làm bản mẫu
- Trình diễn khía cạnh phần mềm mà khách hàng nhìn thấy (nhập liệu, kết xuất dữ liệu, thao tác)
- Phải nhanh và ít công sức (sử dụng 4GT)
- Phụ thuộc vào các yếu tố
 - Loại bản mẫu
 - Định hướng làm bản mẫu

CÁC LOẠI BẢN MẪU

1. Bản mẫu thực hiện một phần chức năng
2. Bản mẫu giao diện
3. Bản mẫu hướng tới sản phẩm

ĐỊNH HƯỚNG LÀM BẢN MẪU

- Bản mẫu dùng xong bỏ đi (*throw-away approach*)
- Bản mẫu dùng tiếp cho bước sau (*evolutionary approach*)
- Bản mẫu là phần hệ thống vận hành được (*dựa trên thành phần*)

KHI NÀO SỬ DỤNG BẢN MẪU ?

KHUNG CẢNH

Xác định miền yêu cầu chưa rõ ràng để phân tích

Dữ liệu vào/ra phức tạp từ nhiều nguồn khác nhau

Đề xuất thuật toán, giải pháp mới

Hệ thống thông tin quản lý

VẤN ĐỀ

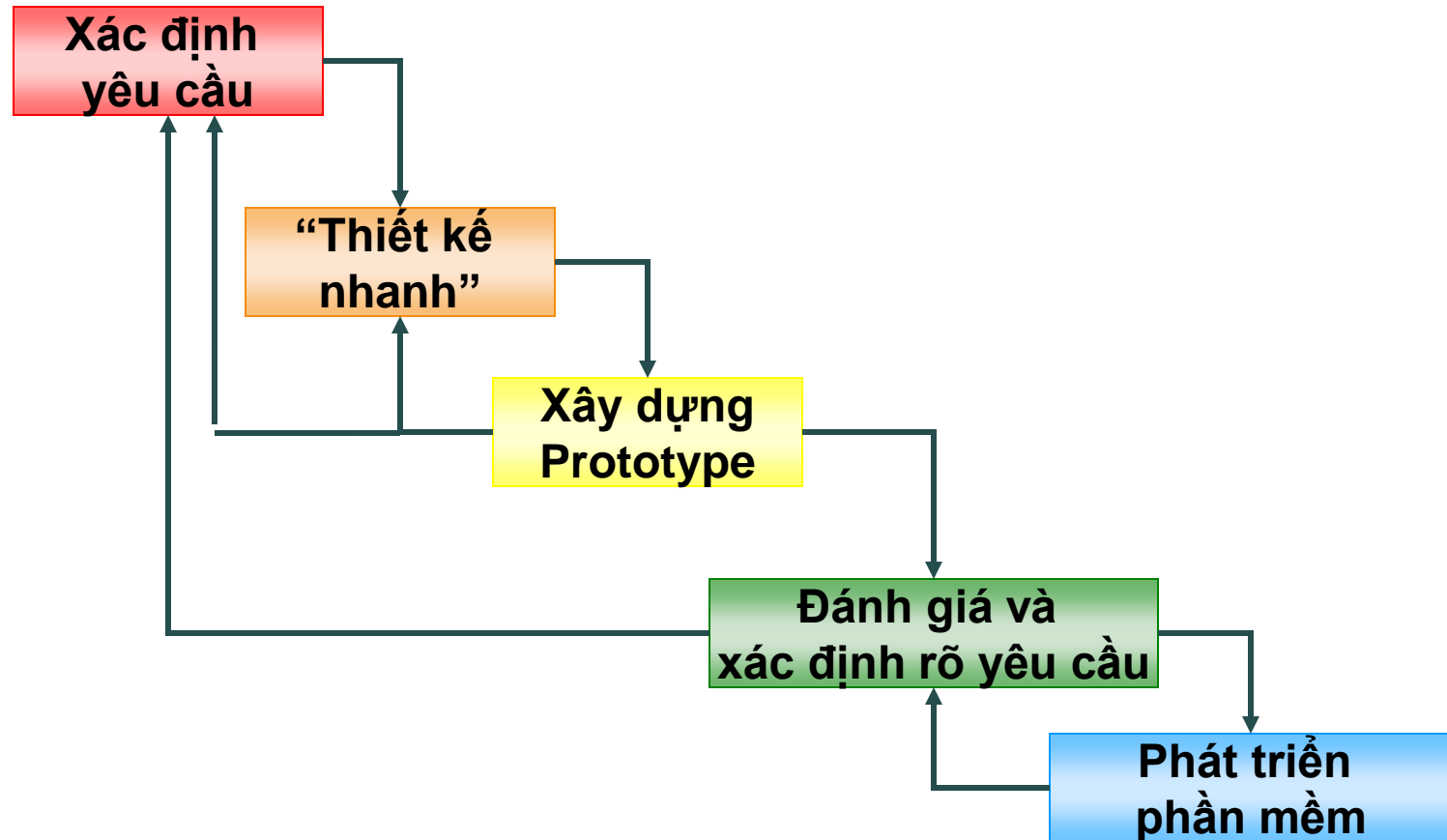
↔ Các yêu cầu chưa rõ ràng

↔ Input/output chưa rõ ràng

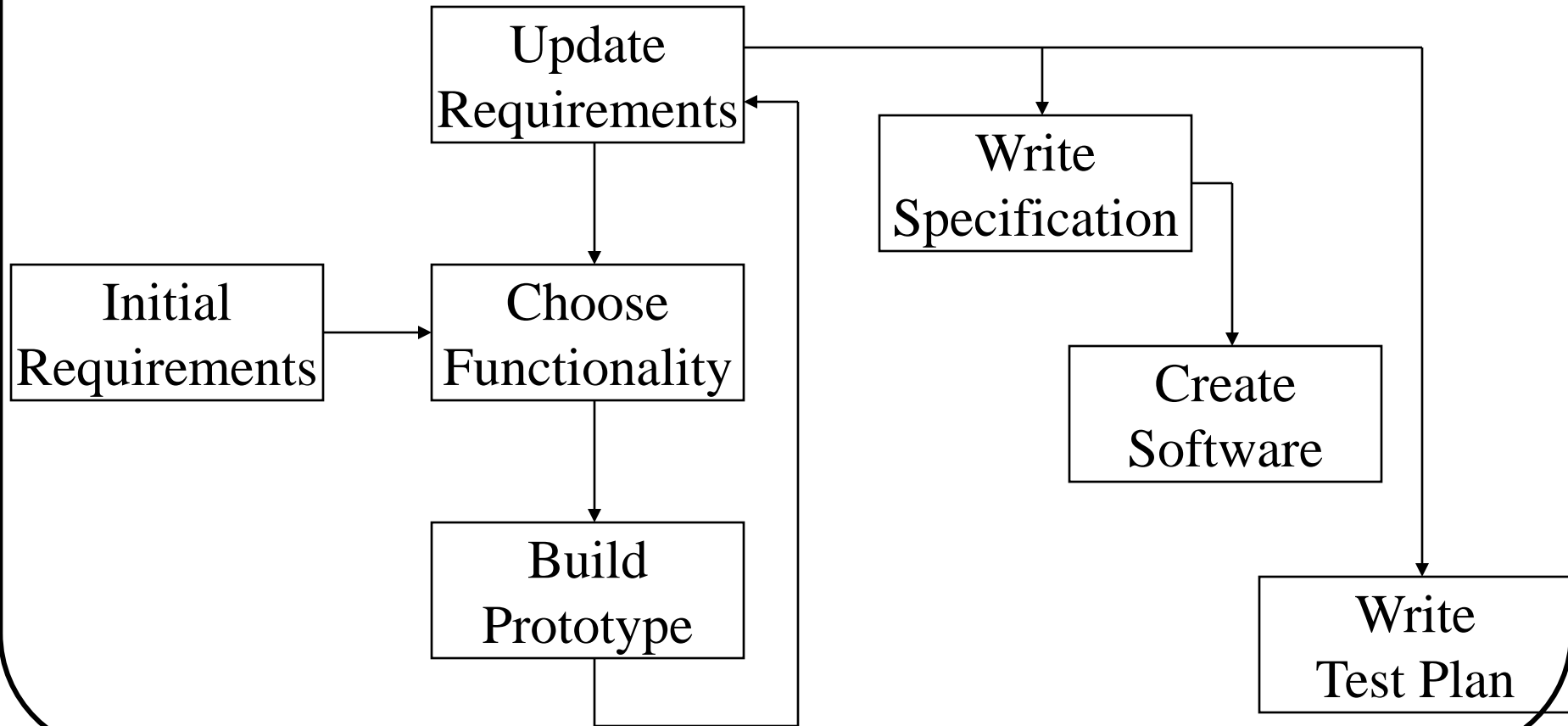
↔ Khó đánh giá hiệu quả thuật toán

↔ Có tương tác người máy nhiều

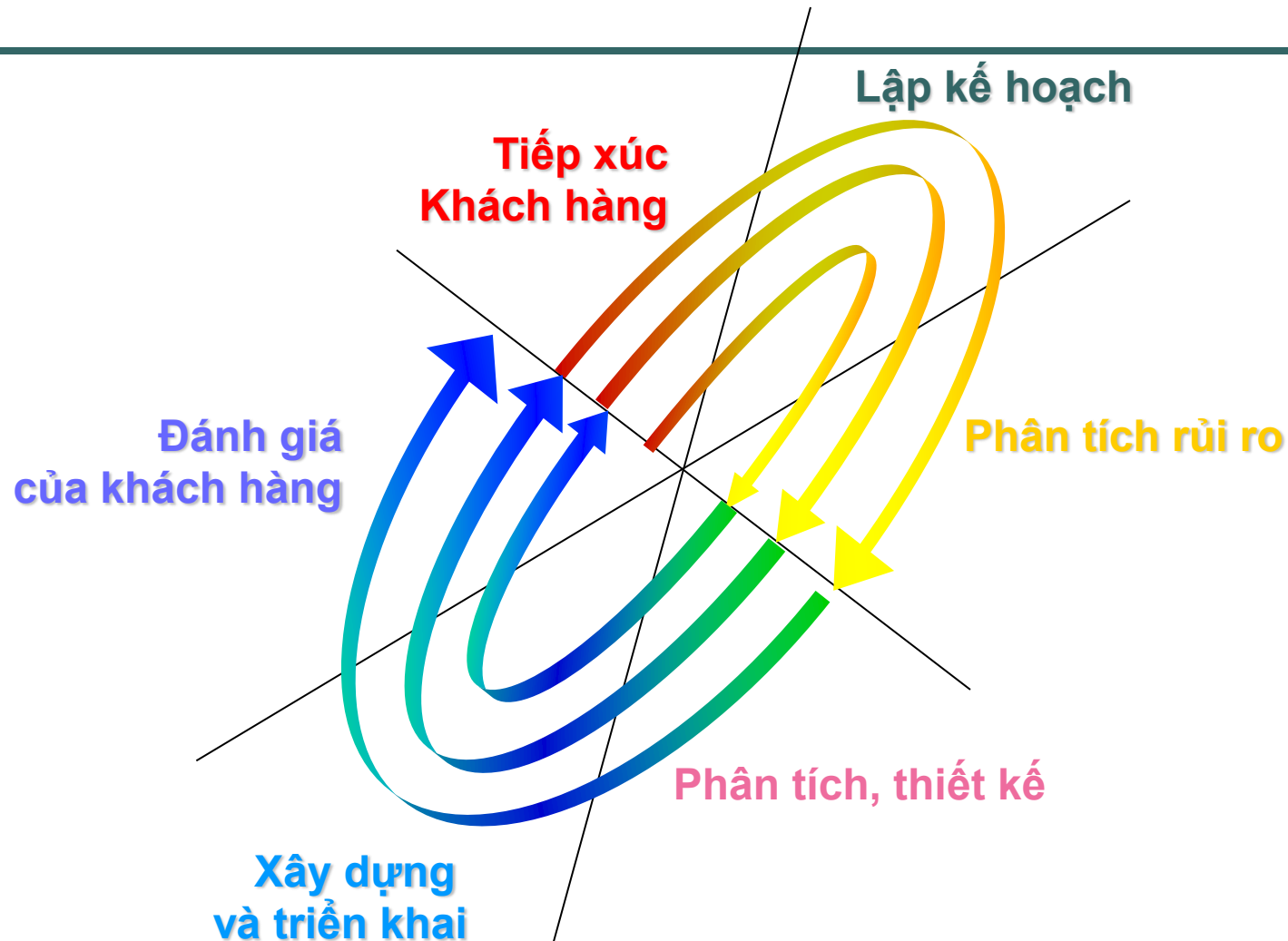
Mô hình Prototype



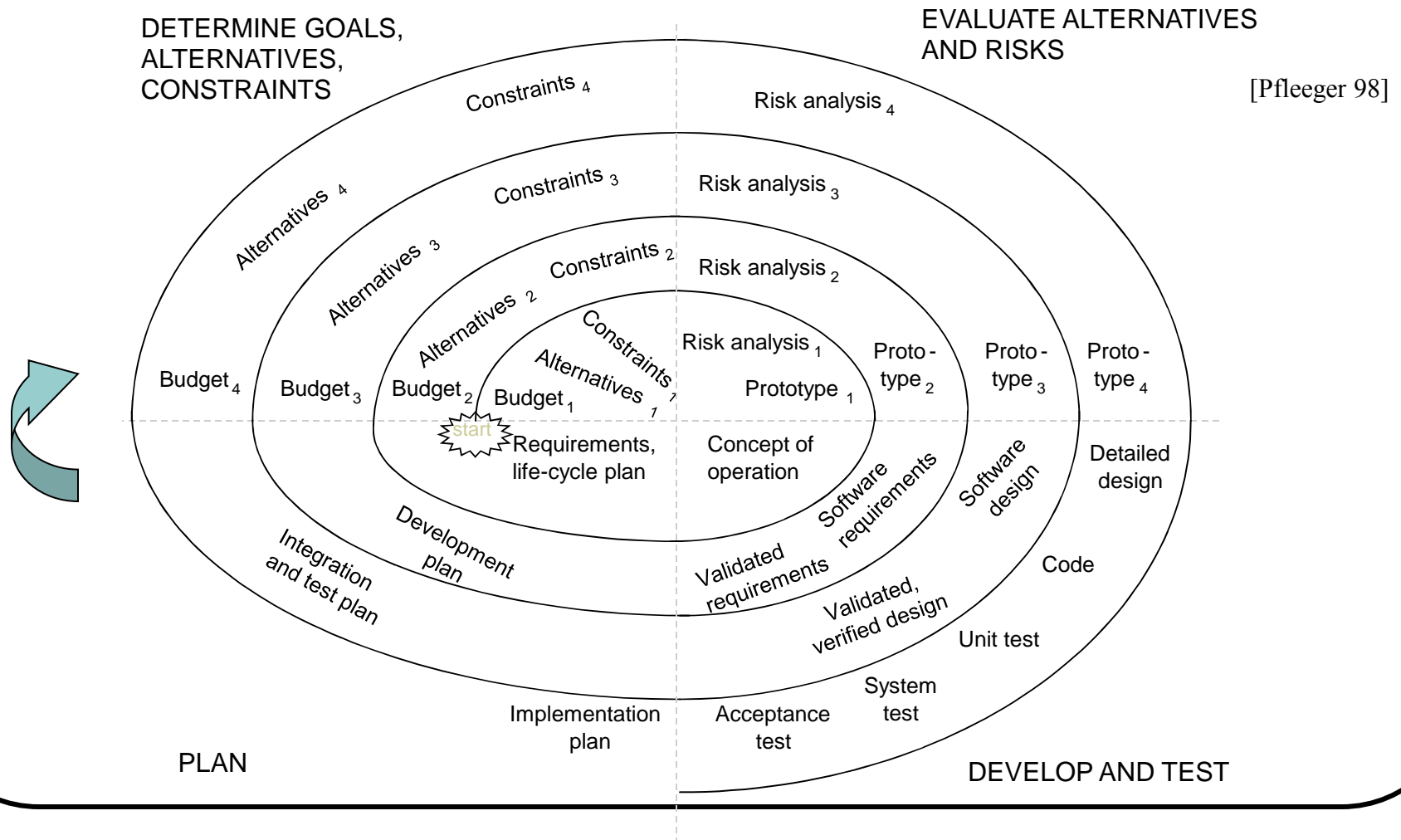
Rapid Prototyping + Waterfall



Mô hình xoắn ốc



Boehm's Spiral Model



MÔ HÌNH XOẮN ỐC (Spiral Model)

Phân tích

Thiết kế

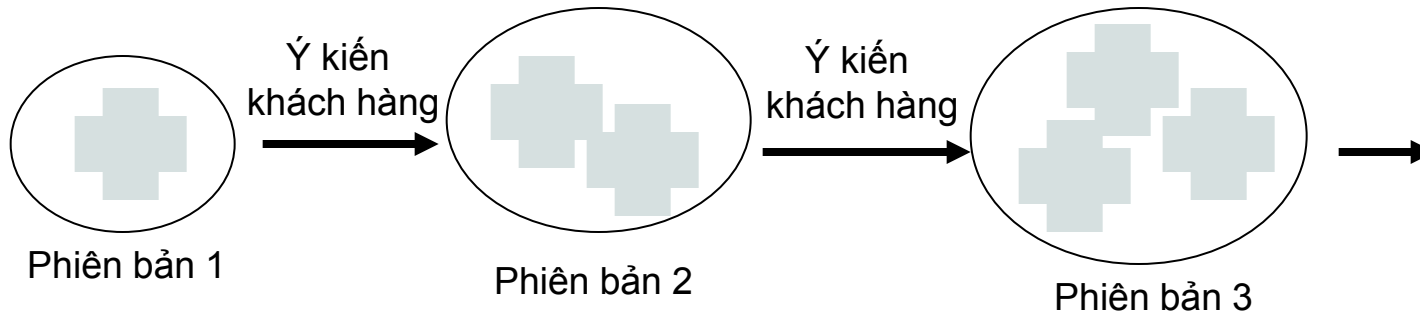
Mã hóa

Kiểm thử

Vận hành

Mô hình thác nước

- Trình tự tốt
- Rủi ro cao



Mô hình bản mẫu

- Loại trừ rủi ro
- Trình tự không rõ ràng



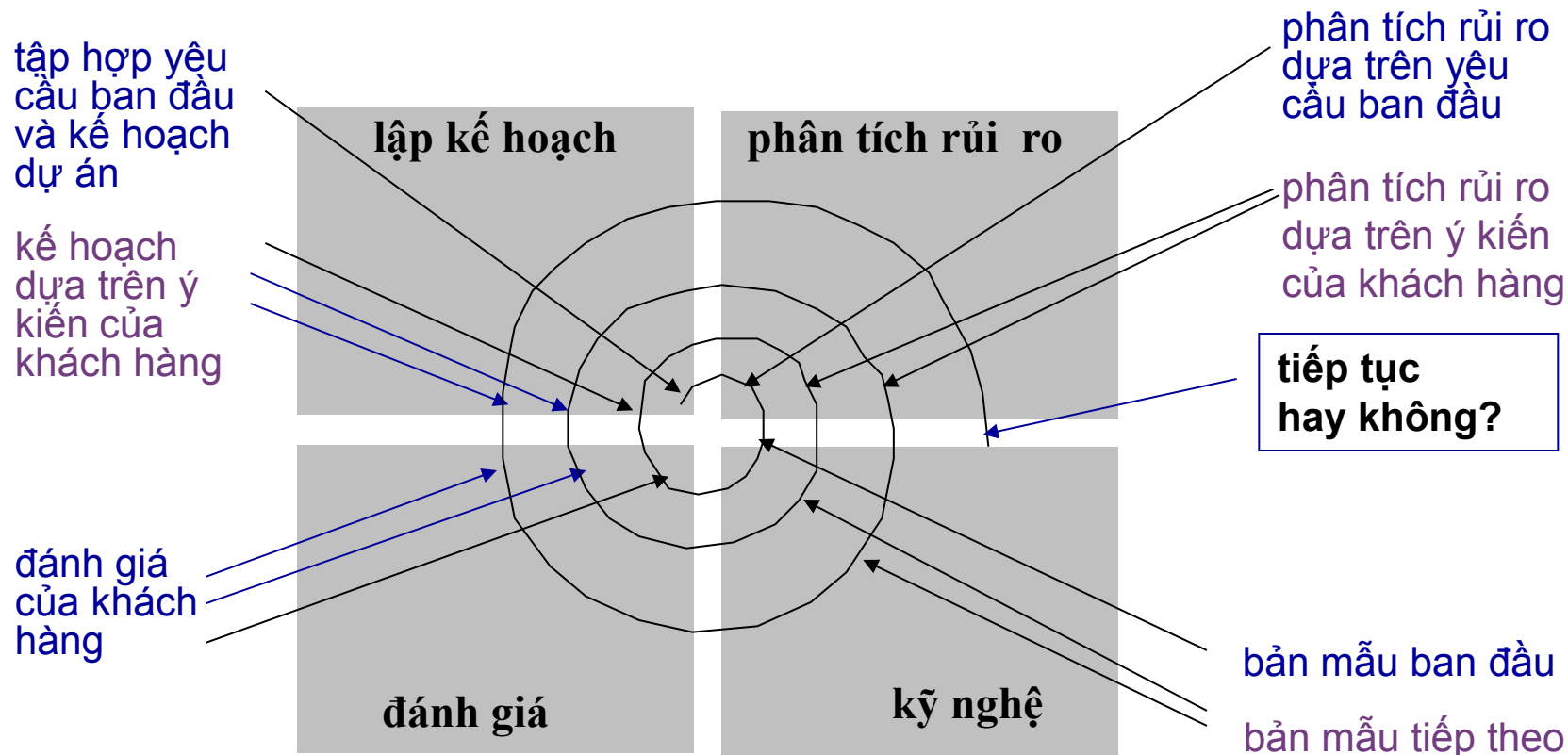
Kết hợp ưu điểm 2 mô hình:

- Trình tự rõ ràng
- Loại trừ rủi ro sớm

MÔ HÌNH XOẮN ỐC (*Spiral Model*)

- Là quá trình lặp theo hướng mở rộng, hoàn thiện dần qua 4 hoạt động:
 - *Lập kế hoạch*: xác lập tài nguyên, thời hạn... cho dự án
 - *Phân tích rủi ro*: xem xét các mạo hiểm có thể xảy ra
 - *Công nghệ*: phát triển một phiên bản của phần mềm (lựa chọn một mô hình thích hợp)
 - *Giao tiếp với khách hàng*: khách hàng đánh giá về phiên bản đã phát triển; làm mịn, sửa đổi các yêu cầu

SƠ ĐỒ HOẠT ĐỘNG VỚI MÔ HÌNH XOẮN ỐC



ƯU ĐIỂM MÔ HÌNH XOẮN ỐC

KHẢ NĂNG LINH HOẠT

- Các vòng tròn được lặp để đáp ứng yêu cầu thay đổi của người dùng
- Kiểm soát rủi ro ở từng mức tiến hóa
- Có thể sử dụng các mô hình khác nhau để tránh các rủi ro đã xác định
- Đánh giá chi phí chính xác hơn các phương pháp khác

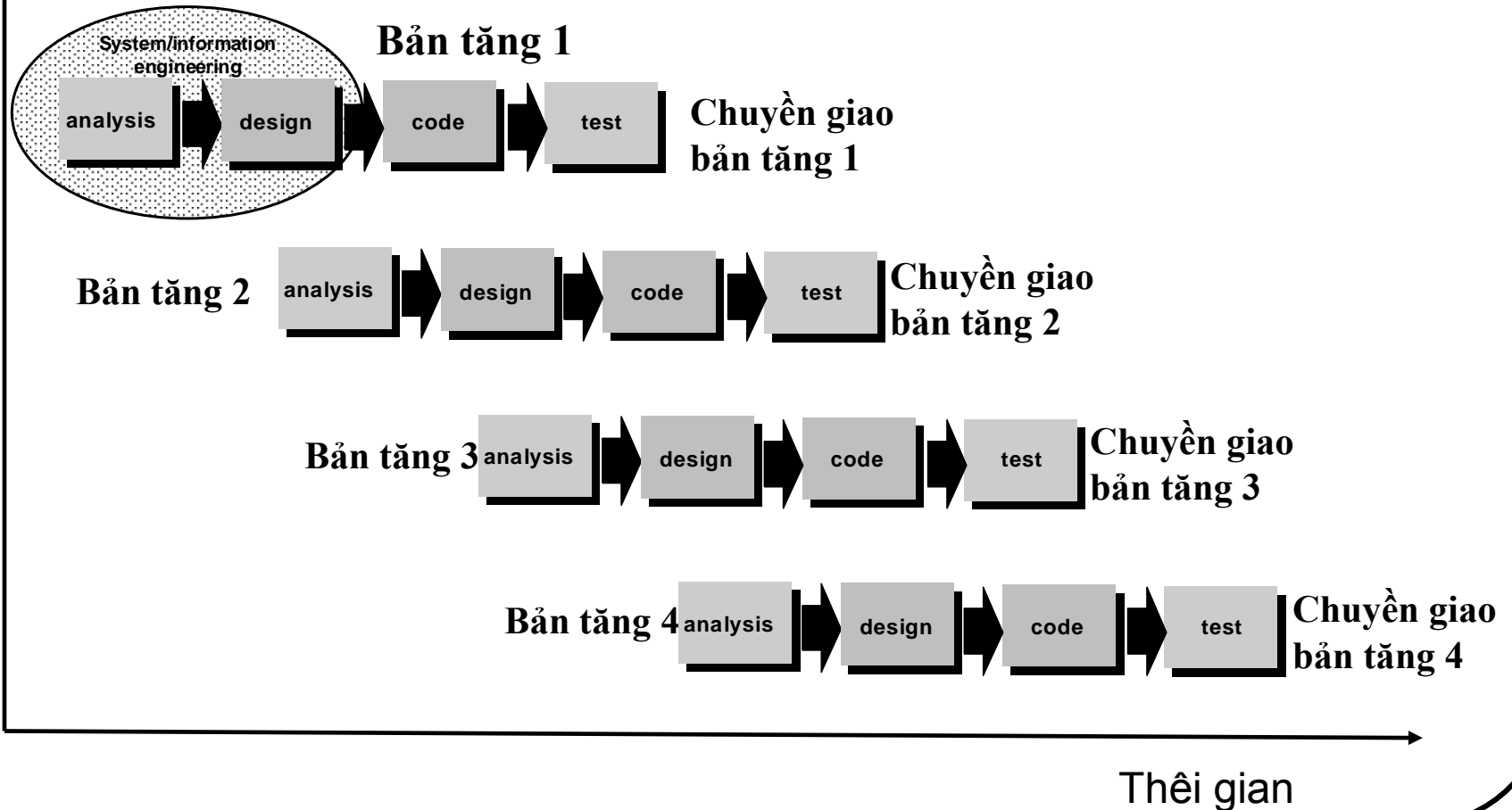
NHƯỢC ĐIỂM MÔ HÌNH XOẮN ỐC

- Yêu cầu thay đổi thường xuyên dẫn đến lặp vô hạn và thất bại
- Khó thuyết phục khách hàng là phương pháp tiến hóa có thể kiểm soát được
- Đòi hỏi năng lực quản lý, năng lực phân tích rủi ro cao -> cần chi phí chuyên gia
- Chưa được dùng rộng rãi như mô hình thác nước hoặc làm mẫu

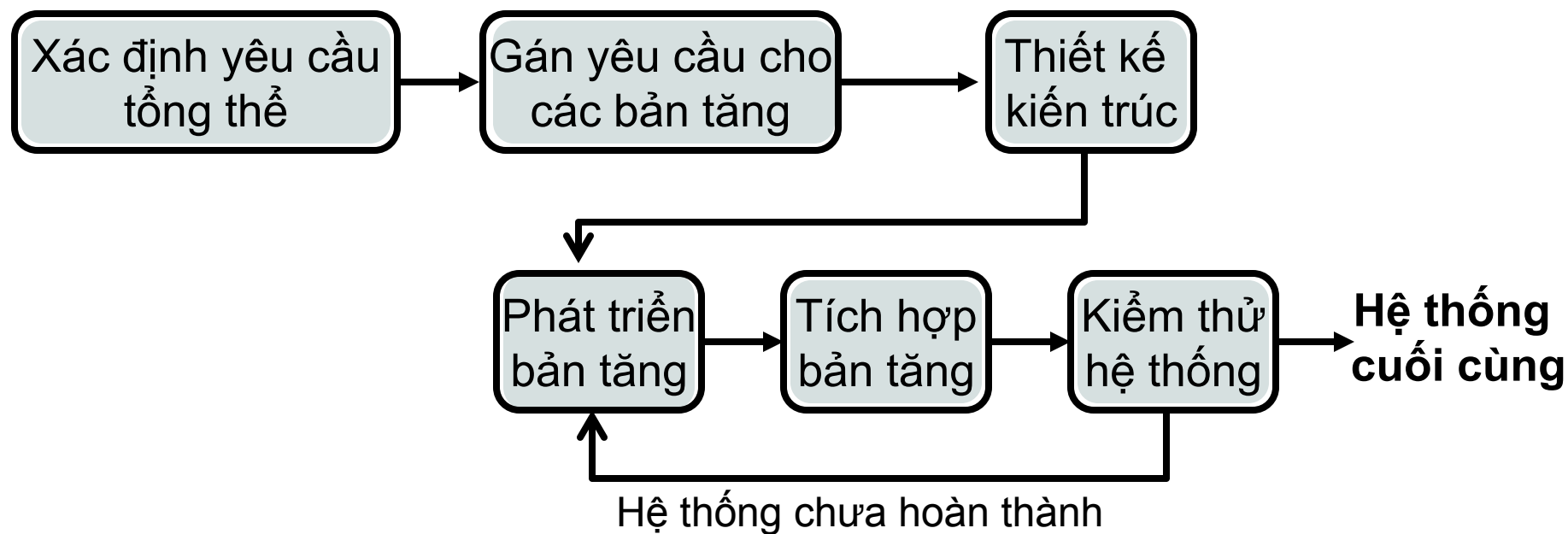
MÔ HÌNH PHÁT TRIỂN TĂNG TRƯỞNG (*Incremental development*)

- Thay vì chuyển giao một lần, quá trình phát triển và chuyển giao được chia làm nhiều lần, mỗi chuyển giao đáp ứng một phần chức năng
- Yêu cầu người dùng được phân loại ưu tiên, mức cao sẽ thuộc phần chuyển giao sớm

MÔ HÌNH PHÁT TRIỂN TĂNG TRƯỞNG



HOẠT ĐỘNG PHÁT TRIỂN TĂNG TRƯỞNG



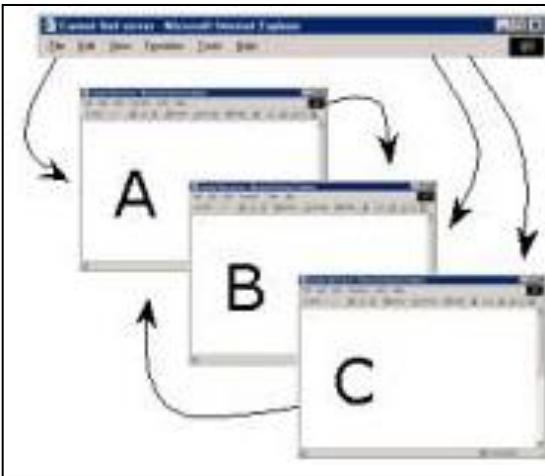
ƯU ĐIỂM MÔ HÌNH TĂNG TRƯỞNG

- Có sản phẩm dùng được trong thời gian ngắn
 - đáp ứng nhanh yêu cầu của khách hàng
 - chiếm lĩnh thị trường
- Rủi ro được loại bỏ sớm
- Dịch vụ hệ thống ưu tiên mức cao nhất được kiểm thử nhiều nhất

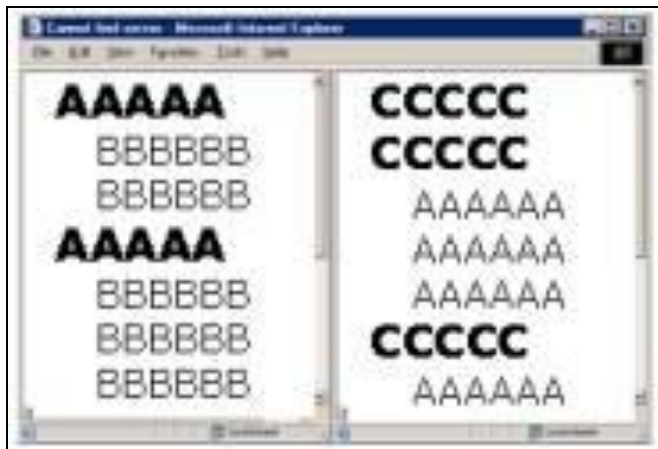
NHƯỢC ĐIỂM MÔ HÌNH TĂNG TRƯỞNG

- Tổng chi phí phát triển là cao hơn bình thường
- Tổng thời gian để chuyển giao toàn bộ chức năng là lớn hơn

VẤN ĐỀ VỚI MÔ HÌNH TĂNG TRƯỞNG – Ví dụ



- Theo từng bản tăng cho ra các thuộc tính giao diện: A, B, C
- Khó khăn khi cần tích hợp giao diện: A và B, A và C



➔ Vấn đề tương tự với tổng thể hệ thống (Kiến trúc nghèo nàn)

TỔNG KẾT MÔ HÌNH TIẾN HÓA

- Bao gồm: Mô hình bản mẫu, Mô hình xoắn ốc, Mô hình tăng trưởng
- Phát triển sơ bộ
 - Làm việc với khách hàng để có được đặc tả khái quát ban đầu của hệ thống. Có thể bắt đầu với hiểu biết chưa đầy đủ.
- Thực hiện phát triển bằng cách làm mẫu
 - Mục tiêu là để hiểu hệ thống. Bản mẫu ban đầu được phát triển có thể còn sơ sài.

TỔNG KẾT MÔ HÌNH TIẾN HÓA

- Hạn chế
 - Hệ thống thường có cấu trúc nghèo nàn
- Khả năng ứng dụng
 - Ứng dụng cho nhiều lớp hệ thống khác nhau

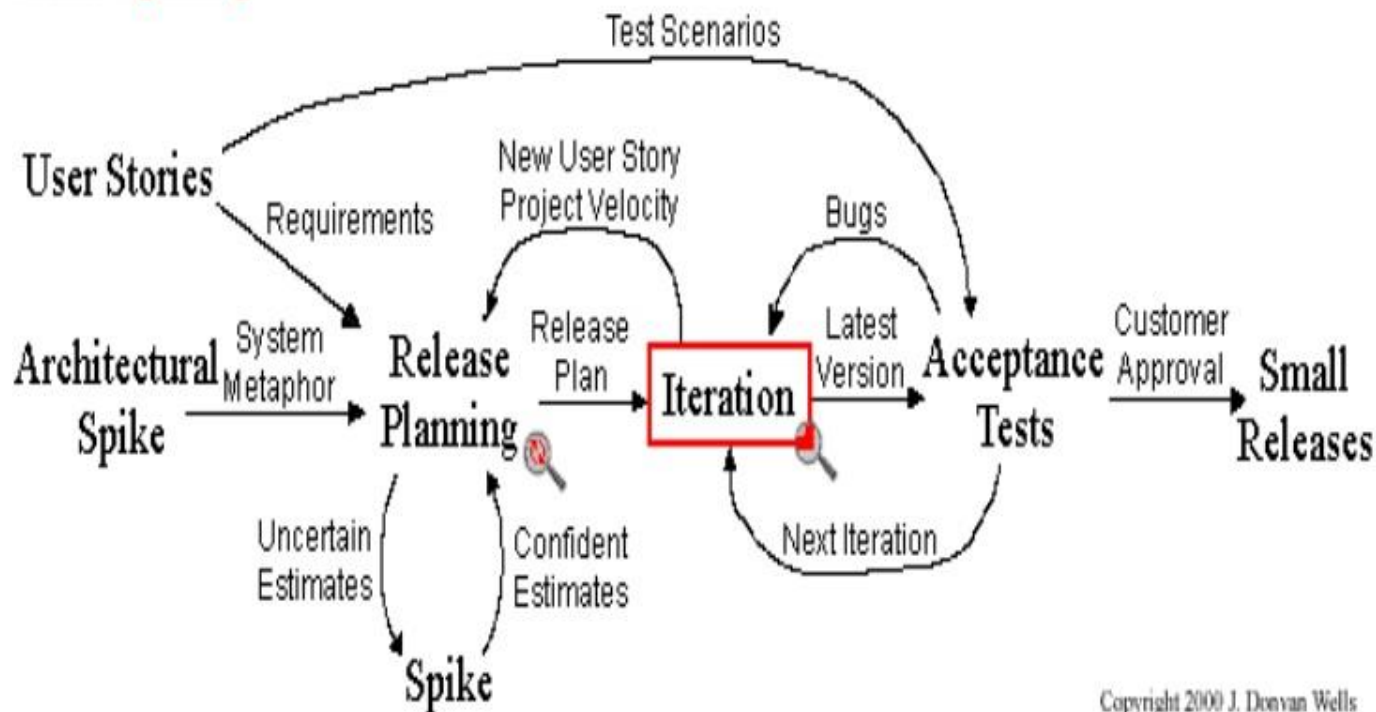
Mô hình lập trình cực đoan XP/Agile

- Tiếp cận theo lối làm việc cộng tác. Có các đặc trưng sau:
 - Lập trình đôi (cặp).
 - Mã nguồn sẽ được giao nộp theo cá nhân
 - Dùng “câu chuyện người dùng” (user stories) để mô tả yêu cầu ở dạng phi hình thức, mô tả này trên các flash card.
 - Rất nhấn mạnh kiểm thử, trong đó kế hoạch kiểm thử được xây dựng ở giai đoạn đầu phát triển chứ không phải ở cuối.
- Lập trình cực đoan là qui trình lặp.
- Mỗi vòng lặp, như xác định yêu cầu, rút ra từ câu chuyện người dùng; thiết kế rút ra từ các kế hoạch kiểm thử; phát triển rút ra từ lập trình đôi; và cài đặt rút ra từ các kiểm thử của người dùng

XP/Agile



Extreme Programming Project

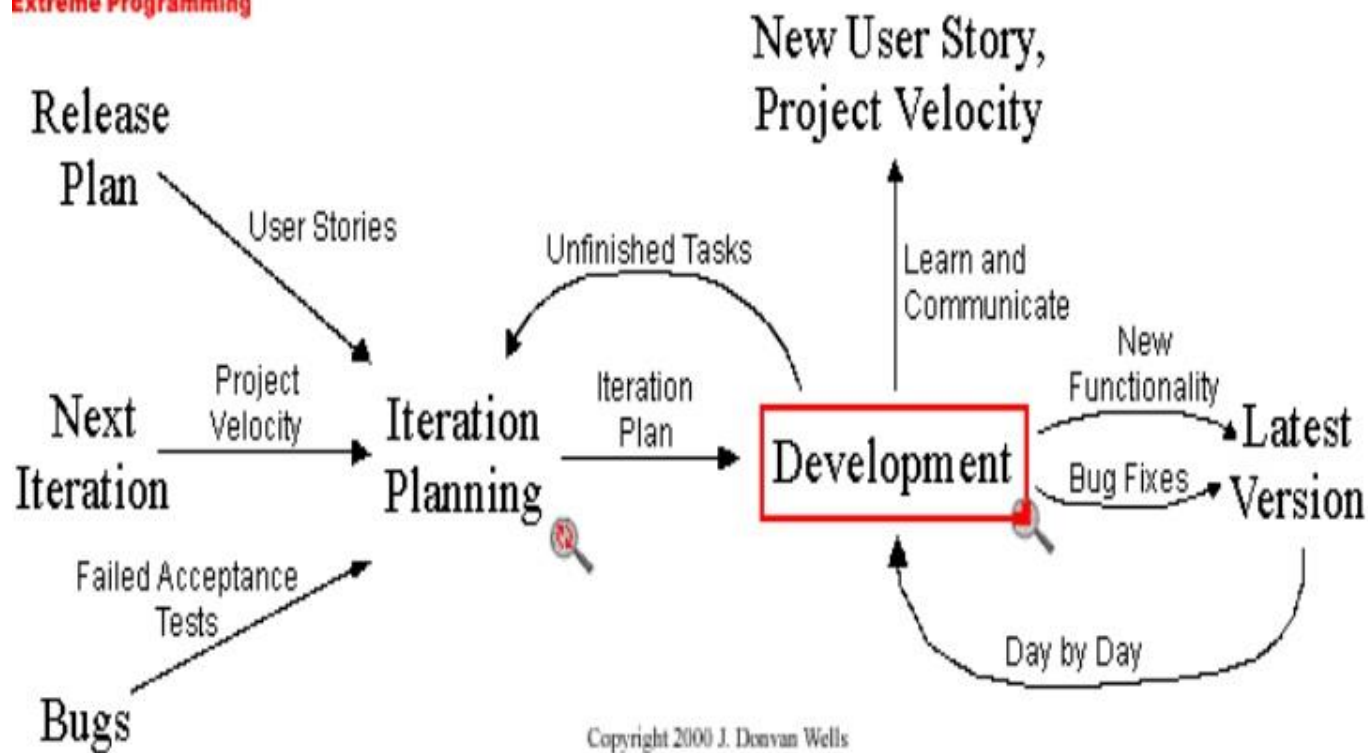


Copyright 2000 J. Donovan Wells

XP/Agile



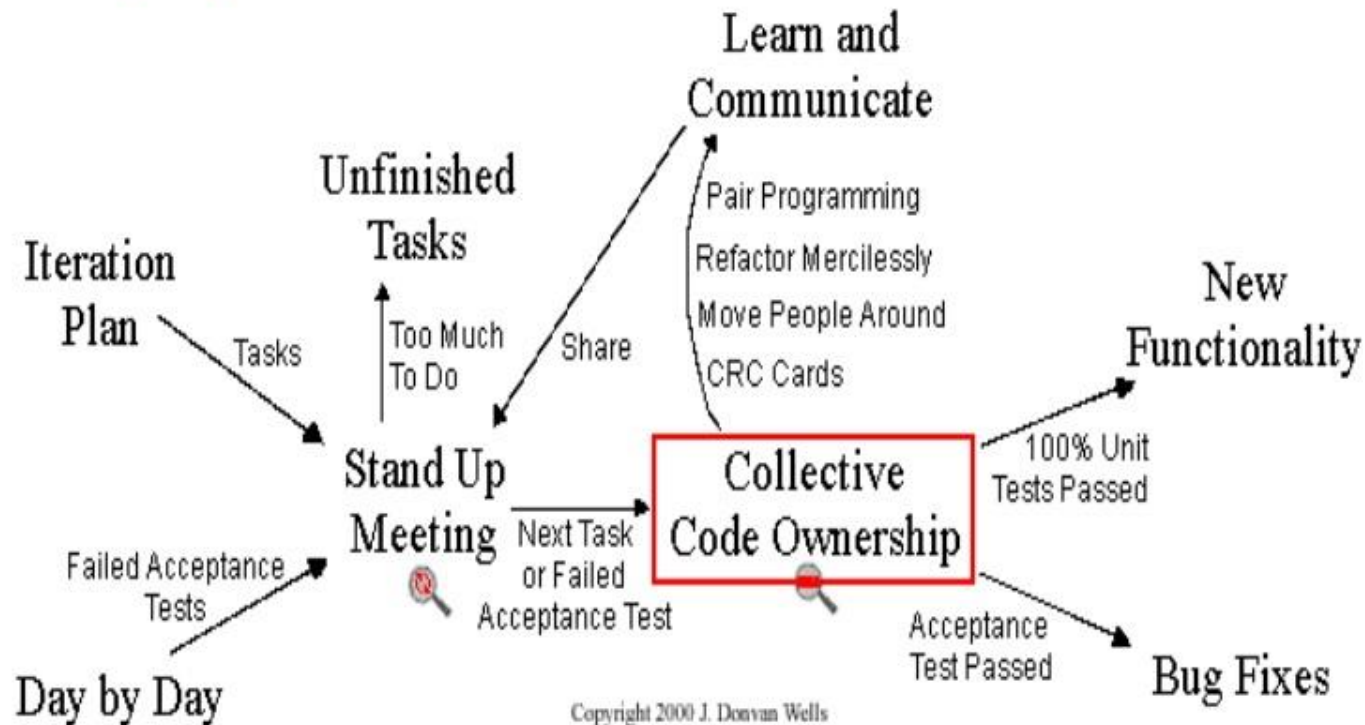
Iteration



XP/Agile



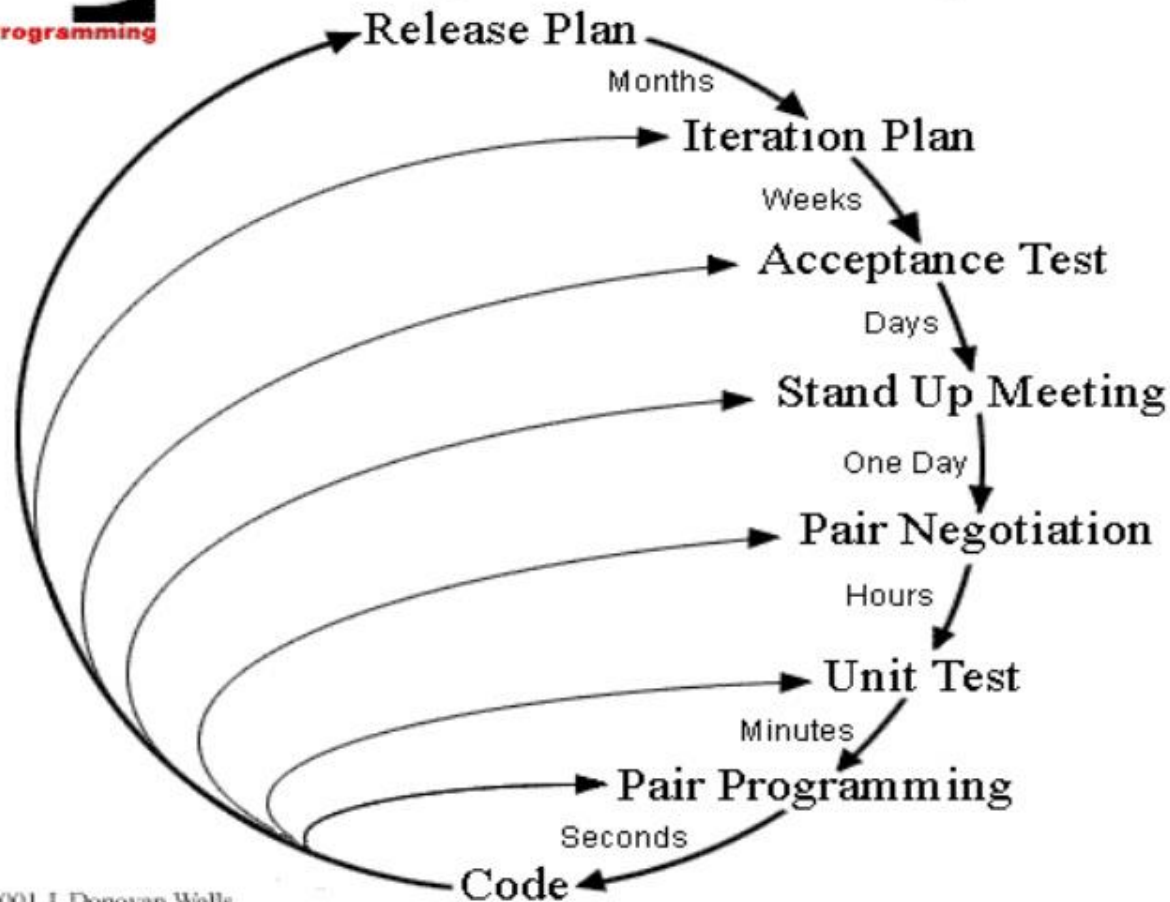
Development



XP/Agile

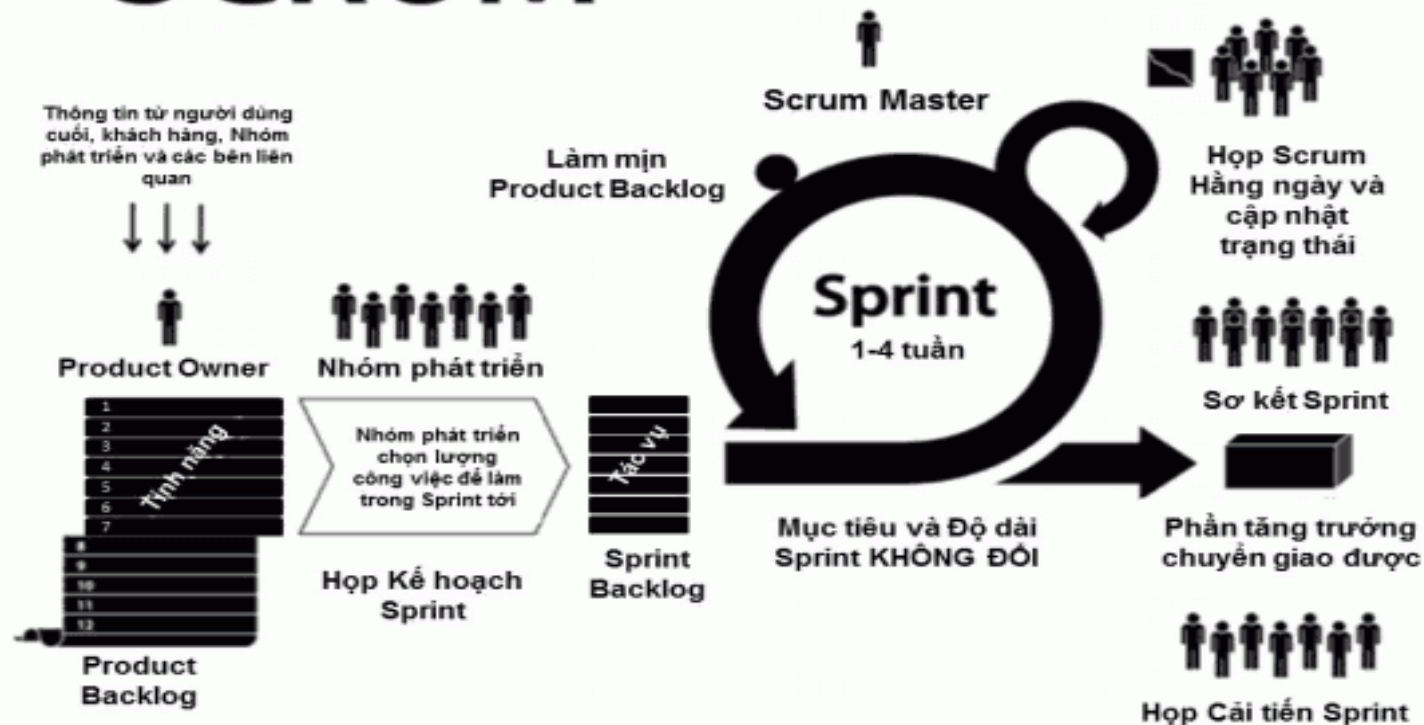


Planning/Feedback Loops



Scrum/Agile

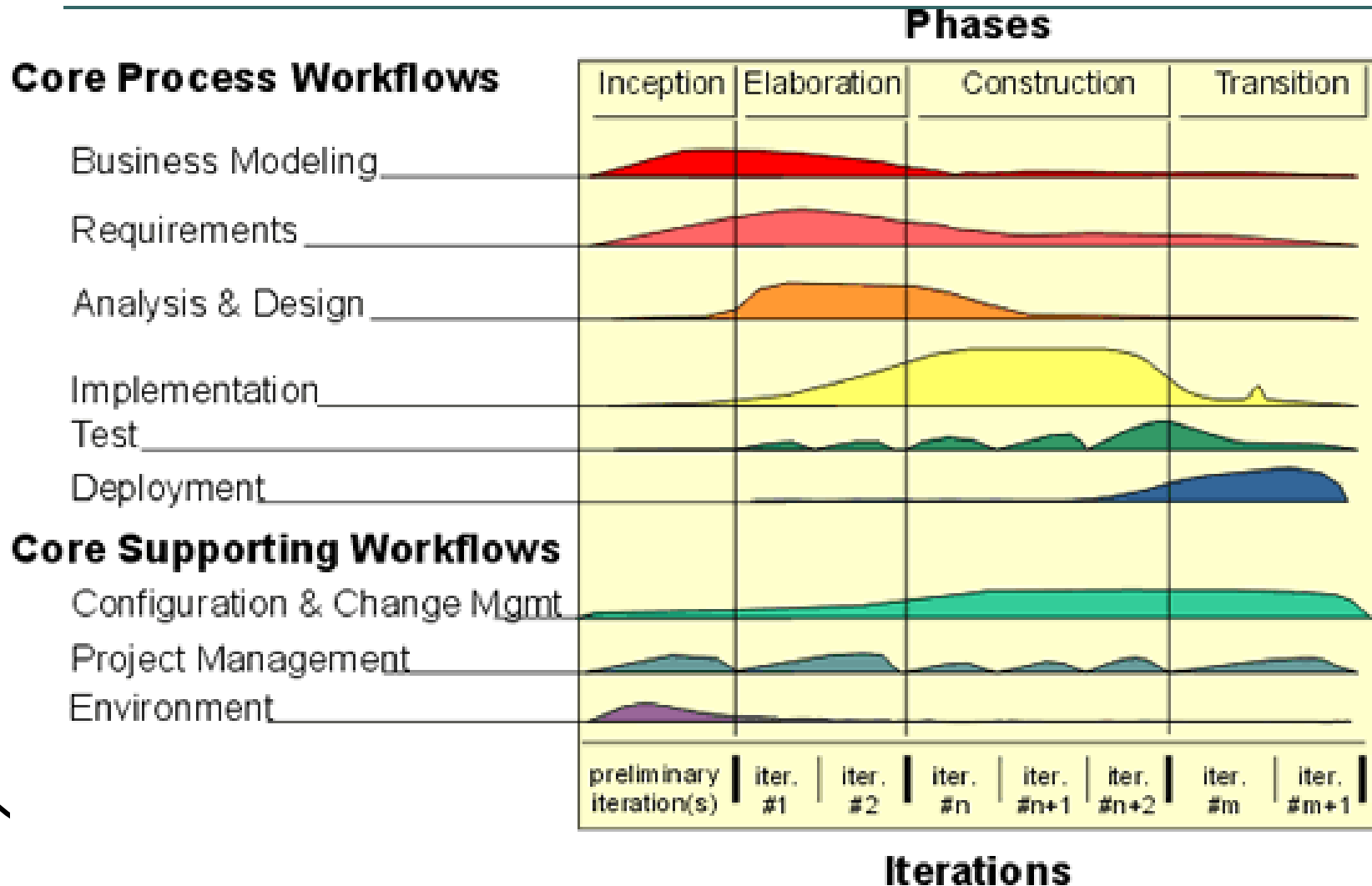
SCRUM



Quy Trình RUP

- RUP là một quy trình vòng lặp phát triển phần mềm.
- RUP là một nền tảng quy trình thích ứng với sự phát triển các tổ chức và các nhóm dự án phần mềm.
- RUP hỗ trợ các hoạt động giữa các nhóm, phân chia công việc cho từng thành viên trong nhóm, trong từng giai đoạn khác nhau của quá trình phát triển phần mềm.
- RUP sử dụng hệ thống ký hiệu trực quan của UML và RUP được phát triển song song với UML.
- RUP là một sản phẩm tiến trình có thể tùy biến.

Kiến trúc của RUP



YÊU CẦU SINH VIÊN

- Hiểu các mô hình quy trình
 - Khi nào áp dụng từng loại mô hình
 - Cách thức áp dụng mô hình
- Với mô hình quy trình đã học, áp dụng liệt kê các công việc phải làm cho đề tài cuối kỳ, phân định vai trò cho những thành viên trong **nhóm**.
- Tìm hiểu nội dung số tài liệu liên quan trong quá trình phát triển phần mềm