Phân tích thiết kế Hệ thống

Giảng viên: Nguyễn Bá Ngọc

Hà Nội-2022

Các quy trình phát triển

Nội dung

- Các mô hình SDLC
- Các mô hình phát triển linh hoạt

Nội dung

- Các mô hình SDLC
- Các mô hình phát triển linh hoạt

Sản phẩm đầu ra tiêu biểu cho các pha

Pha		**Sản phẩm**
Lập kế hoạch	⇒	Kế hoạch dự án
Phân tích	⇒	Đề xuất hệ thống
Thiết kế	⇒	Đặc tả hệ thống
Thực thi	⇒	Hệ thống mới và hoạt động bảo trì

** SDLC (Chương 1 - Tổng quan)*

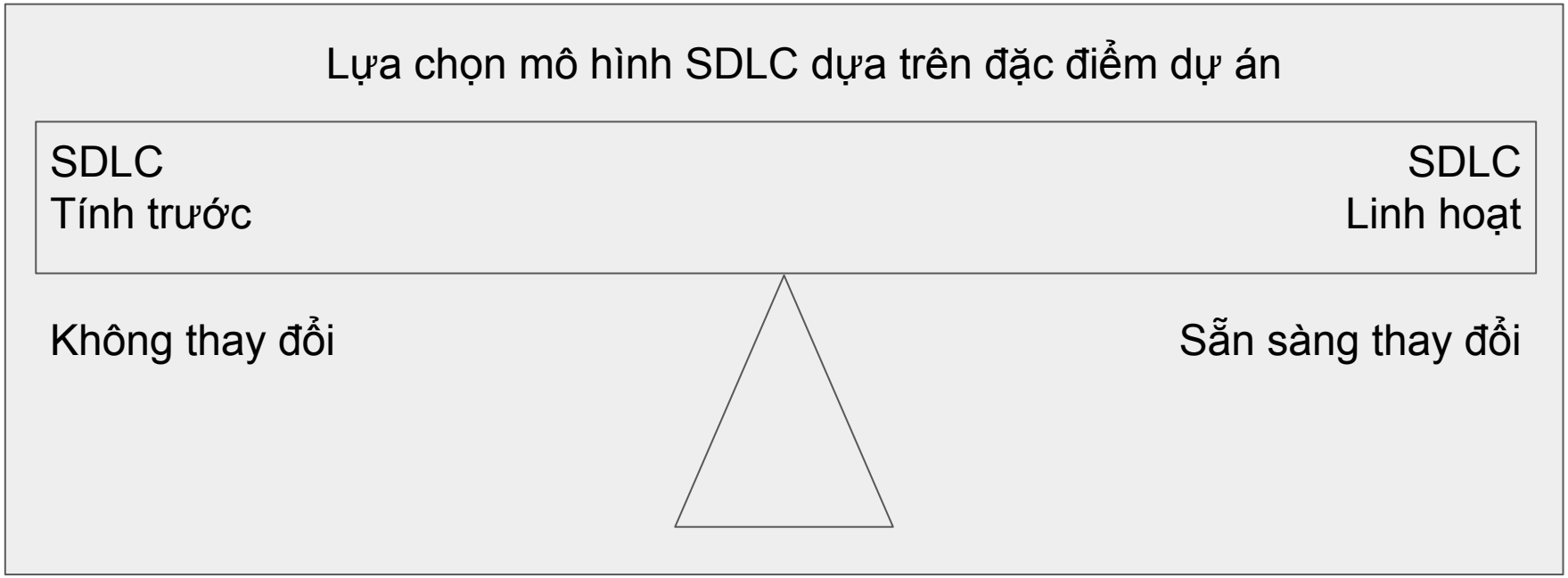
Các mô hình SDLC

Có 2 cách tiếp cận tổng quát đối với SDLC khác nhau về mức độ chi tiết của kế hoạch và khả năng thay đổi kế hoạch:

- Cách tiếp cận tính trước
  - Dự án được lập kế hoạch trước và hệ thống thông tin trong đó được phát triển theo đúng kế hoạch
  - Quản lý dự án cổ điển
  - Không thể (hoặc rất khó) thay đổi kế hoạch
- Cách tiếp cận linh hoạt
  - Dự án có thể được điều chỉnh linh hoạt để đáp ứng những thay đổi cần thiết theo tiến trình phát triển dự án
  - Xu hướng mới hơn
  - Linh hoạt đối với các thay đổi nhưng khó lập kế hoạch tổng thể từ đầu.

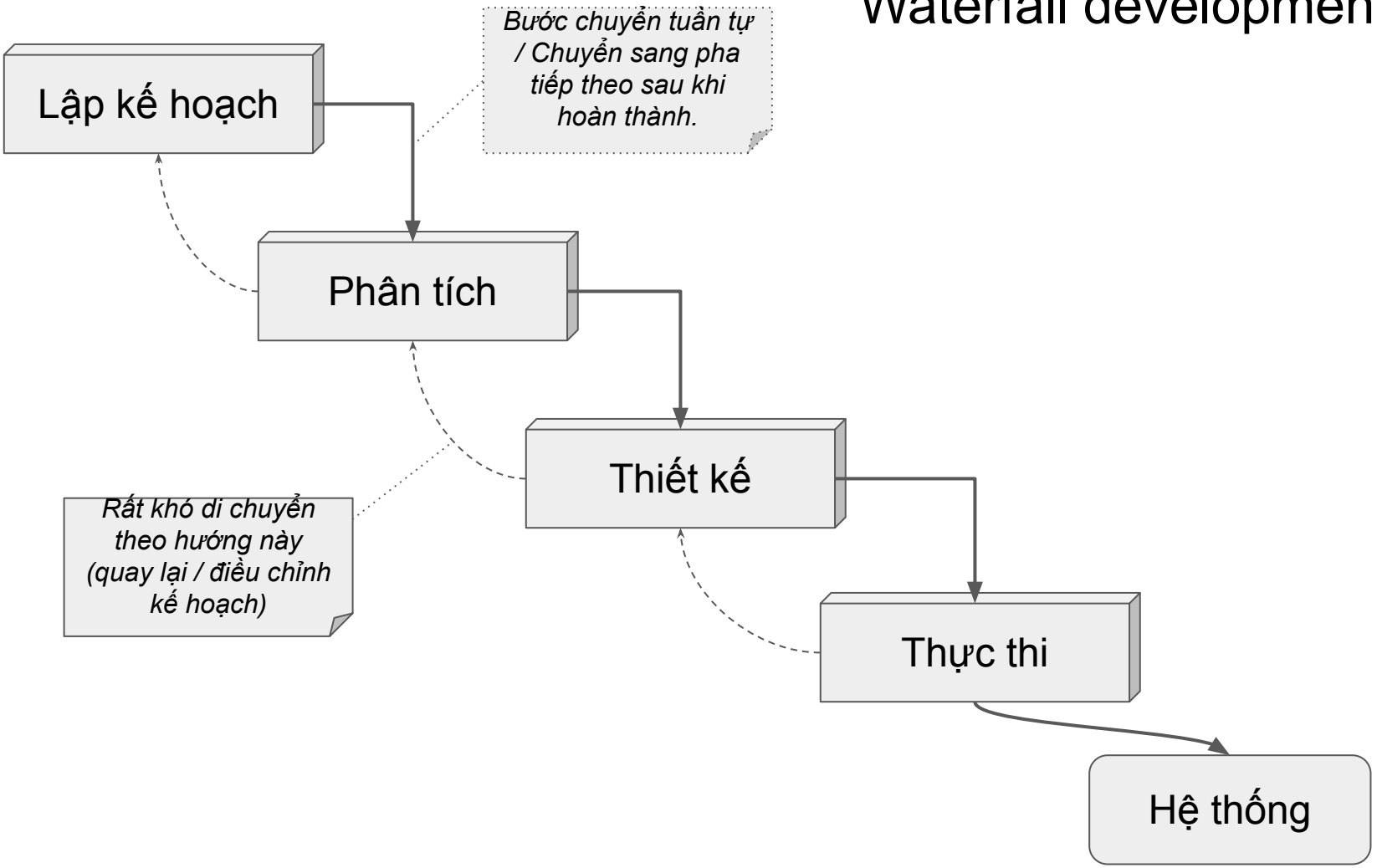
Các mô hình SDLC

- Các dự án thực tế nằm trên một miền liên tục, có cả 2 tính chất cố định (tính trước) và linh hoạt nhưng với mức độ khác nhau

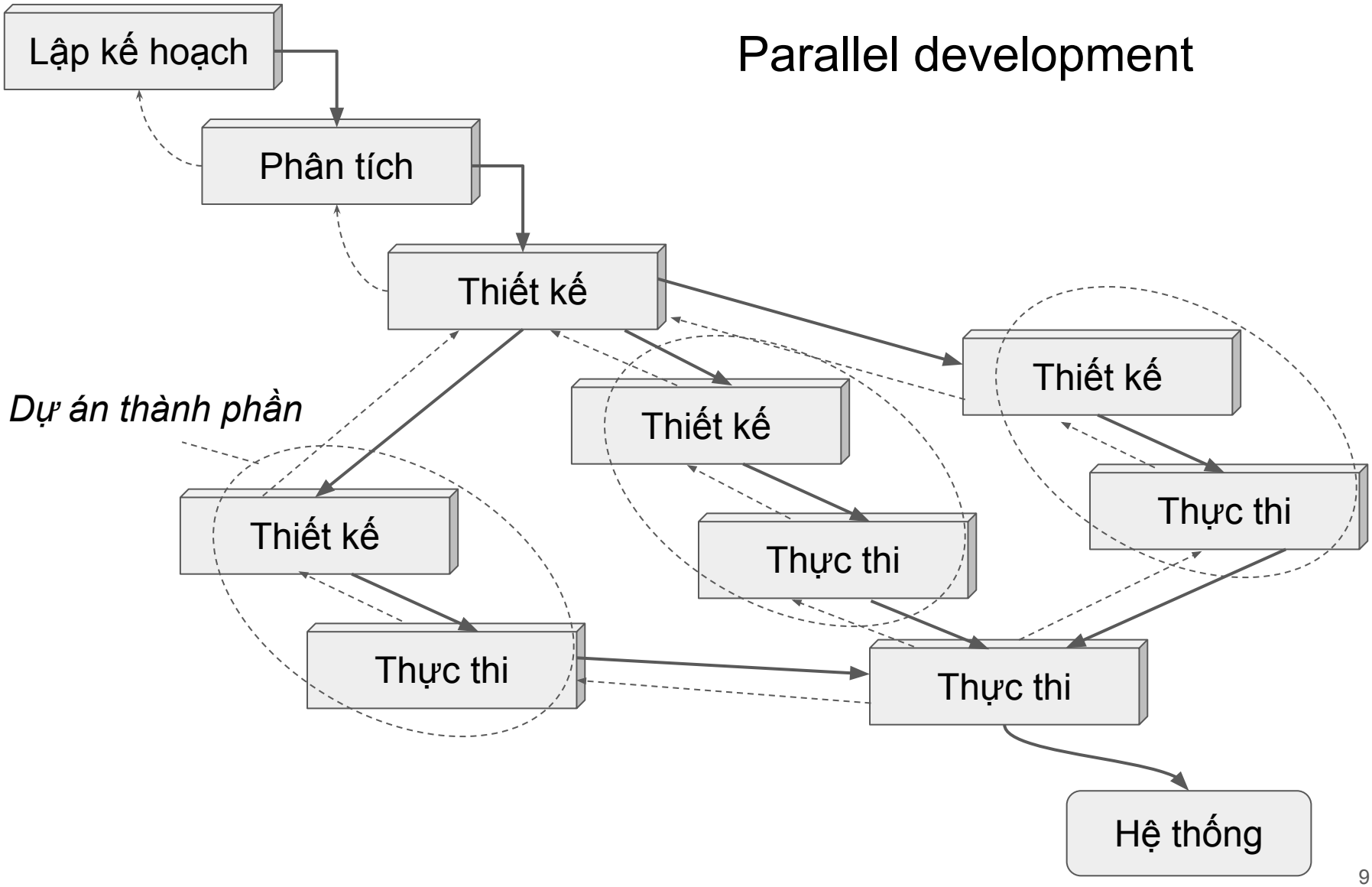


SDLC: Mô hình thác nước

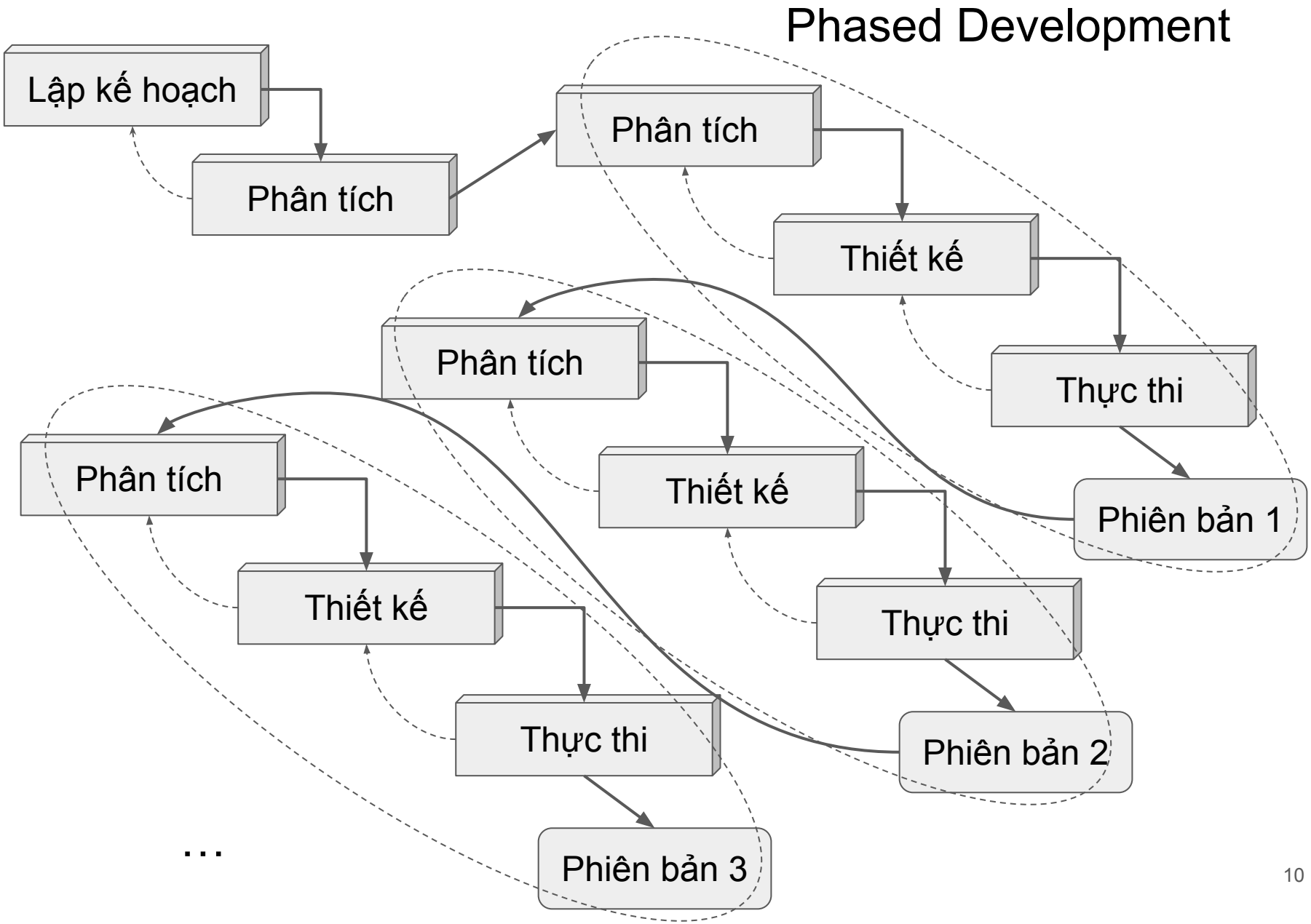
Waterfall development



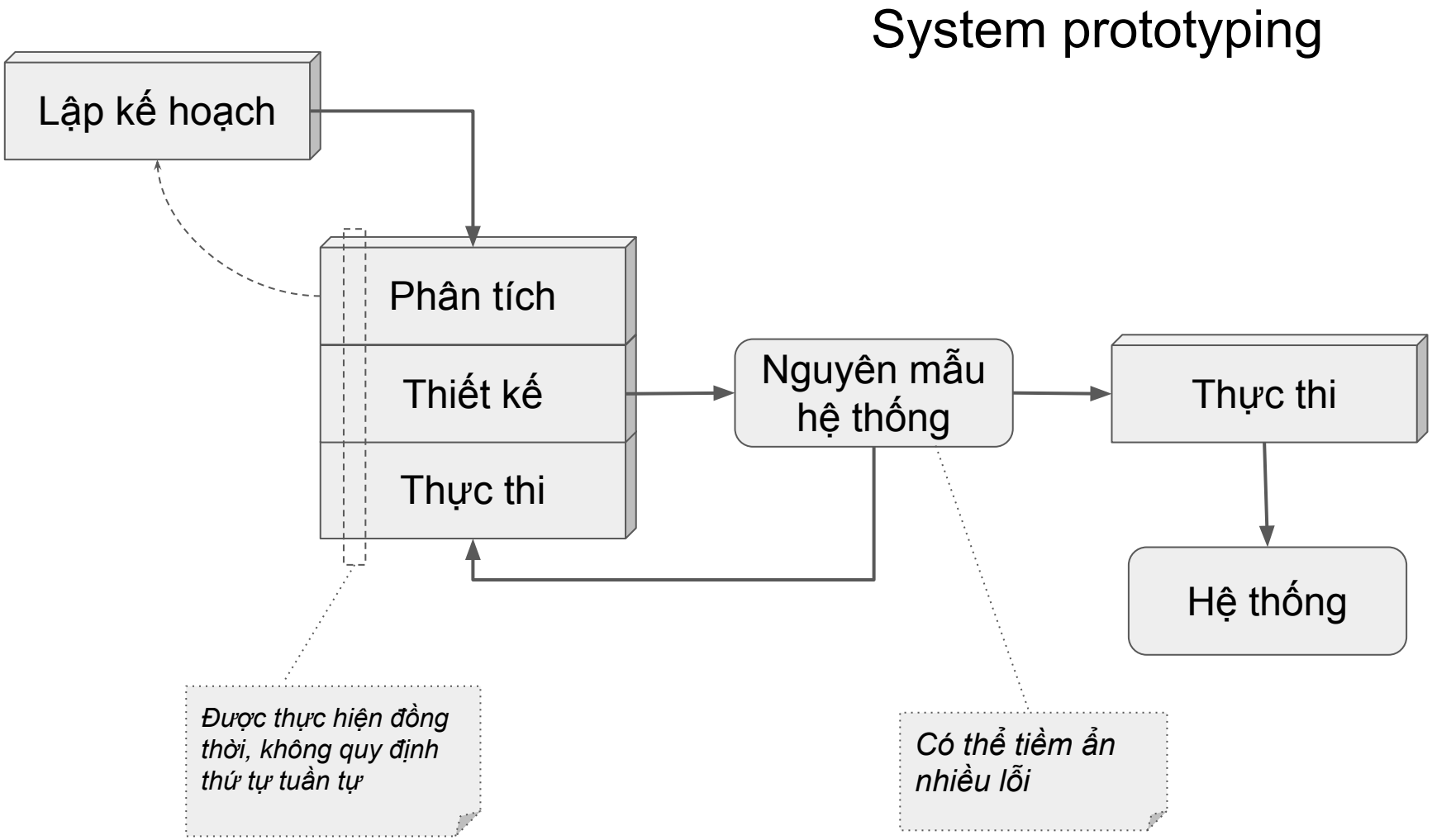
SDLC: Mô hình song song



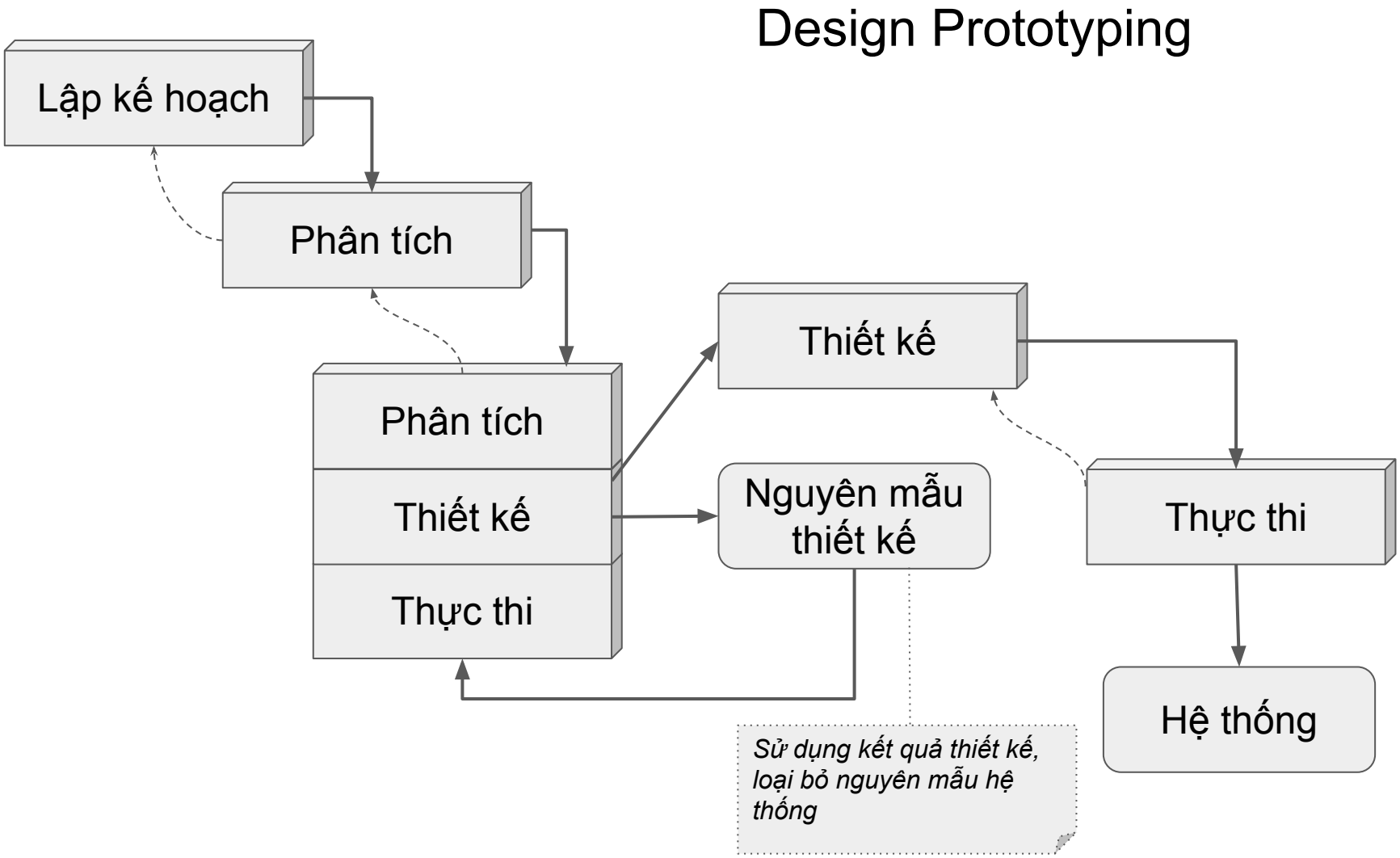
SDLC: Mô hình chia pha



SDLC: Mô hình nguyên mẫu hệ thống

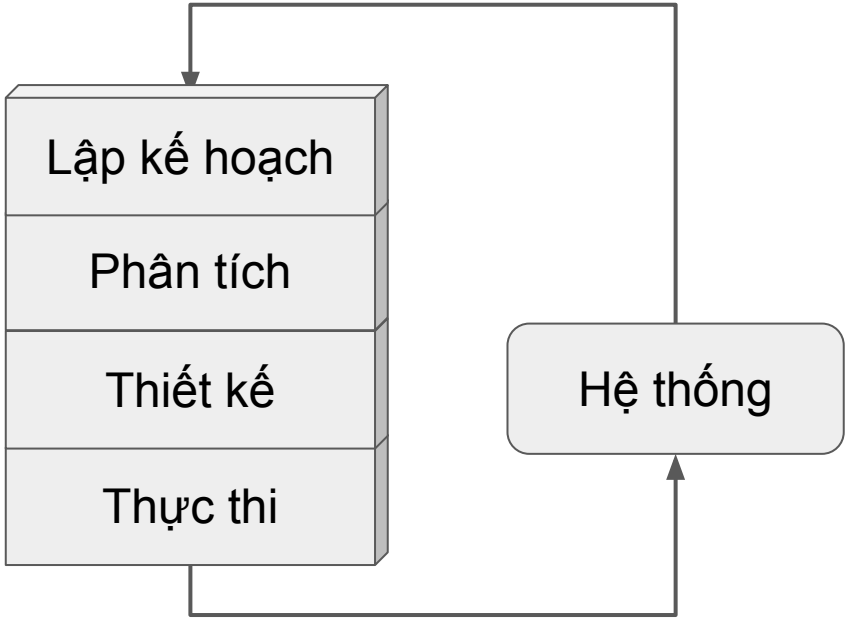


SDLC: Mô hình nguyên mẫu thiết kế



SDLC: Mô hình linh hoạt

Agile Development



Kế hoạch & tính tuần tự

Sử dụng kết quả của mỗi pha SDLC như đầu vào / kế hoạch thực hiện của pha tiếp theo tạo thành 1 tiến trình tuần tự

- Khả năng quay lại điều chỉnh pha trước đó thấp = Tính tuần tự mạnh
- Dựa trên khả năng thay đổi kế hoạch/tính tuần tự, chúng ta có thể phân lớp các mô hình SDLC theo 3 lớp đơn giản:
  - Phát triển có cấu trúc (khả năng thay đổi thấp):
    - Mô hình thác nước
    - Mô hình song song
  - Phát triển ứng dụng nhanh (khả năng thay đổi trung bình)
    - Mô hình chia pha
    - Các mô hình nguyên mẫu
  - Phát triển linh hoạt (sẵn sàng thay đổi)
    - XP
    - SCRUM

Các mô hình SDLC

Các vấn đề dự án tiêu biểu và khả năng xử lý trong SDLC:

Vấn đề/SDLC	**Có cấu trúc / Khả năng thay đổi thấp**		**Phát triển nhanh / Khả năng thay đổi trung bình**			**Linh hoạt/ / Sẵn sàng thay đổi**	
	Thác nước	**Song song**	**Chia pha**	**Nguyên mẫu hệ thống**	**Nguyên mẫu thiết kế**	**XP**	**SCRUM**
Yêu cầu không chắc chắn	Kém	Kém	Tốt	Tuyệt vời	Tuyệt vời	Tuyệt vời	Tuyệt vời
Công nghệ mới	Kém	Kém	Tốt	Kém	Tuyệt vời	Tốt	Tốt
Hệ thống phức tạp	Tốt	Tốt	Tốt	Kém	Tuyệt vời	Tốt	Tốt
Hệ thống tin cậy	Tốt	Tốt	Tốt	Kém	Tuyệt vời	Tuyệt vời	Tuyệt vời
Giới hạn lịch trình	Kém	Tốt	Tuyệt vời	Tuyệt vời	Tốt	Tuyệt vời	Tuyệt vời
Giám sát tiến độ	Kém	Kém	Tuyệt vời	Tuyệt vời	Tốt	Tuyệt vời	Tuyệt vời

...

Người phân tích hệ thống: Các kỹ năng

Người phân tích tìm cách nâng cao hiệu quả hoạt động của tổ chức

- Các kỹ năng cần có
  - Kỹ thuật: Phải hiểu công nghệ
  - Kinh doanh: Phải hiểu các hoạt động kinh doanh
  - Phân tích: Phải có khả năng giải quyết vấn đề
  - Giao tiếp: Môi trường kỹ thuật & phi kỹ thuật
  - Tương tác: Lãnh đạo & quản lý
  - Quy tắc tác nghiệp: Giao dịch trung thực và bảo vệ thông tin mật

Người phân tích hệ thống: Các vai trò

Các vai trò

- Phân tích nghiệp vụ
  - Tập trung vào các vấn đề nghiệp vụ
- Phân tích hệ thống
  - Tập trung vào các vấn đề trong hệ thống thông tin
- Phân tích hạ tầng
  - Tập trung vào các vấn đề kỹ thuật
- Phân tích quản lý thay đổi
  - Tập trung vào các vấn đề nhân lực và quản lý
- Quản lý dự án
  - Đảm bảo dự án hoàn thành đúng thời hạn, trong giới hạn kinh phí, và đáp ứng được các yêu cầu

Nội dung

- Các mô hình SDLC
- Các mô hình phát triển linh hoạt

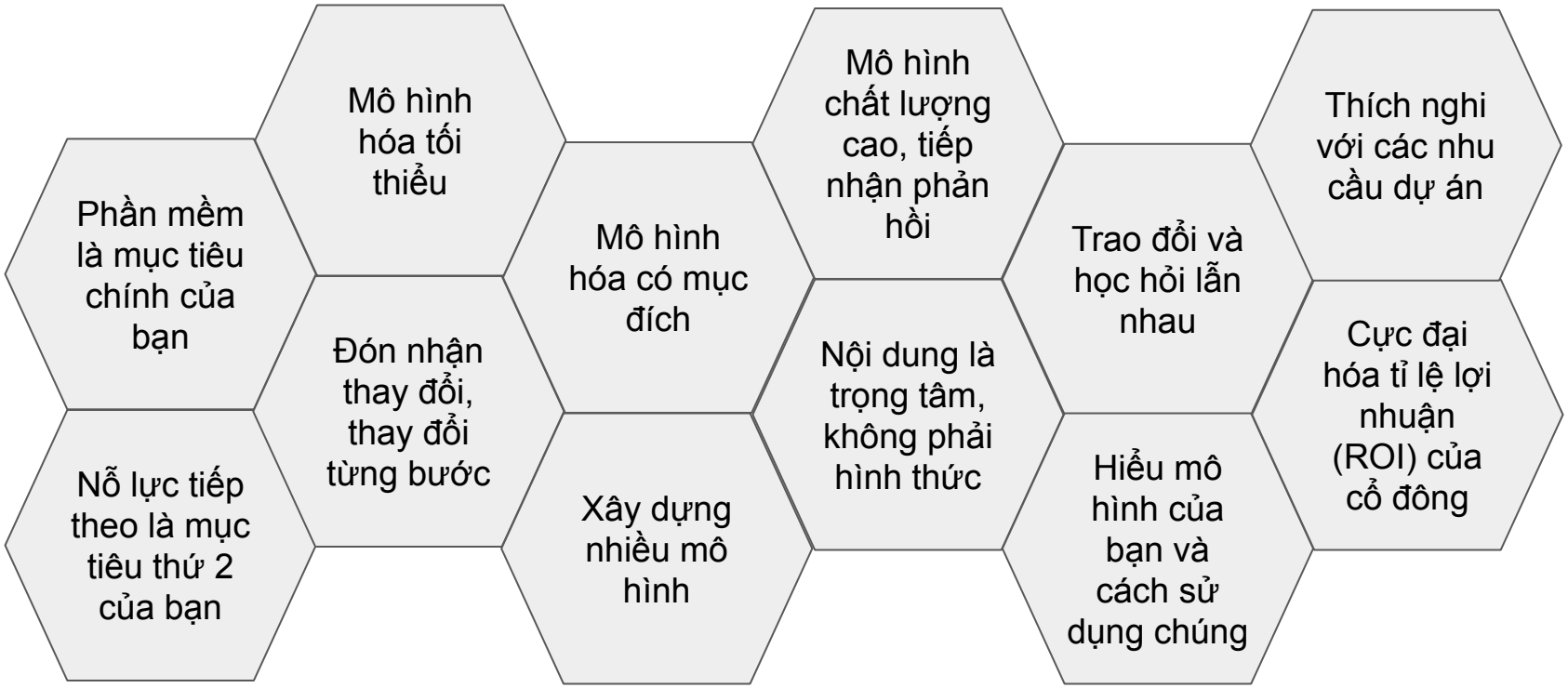
Phát triển linh hoạt

- Các hướng dẫn phát triển hệ thống thông tin trong môi trường với các yêu cầu không chắc chắn, thường xuyên thay đổi.
- Các giá trị trong phát triển linh hoạt
  - Ưu tiên đáp ứng thay đổi hơn thực hiện theo kế hoạch
  - Ưu tiên các cá nhân và tương tác hơn các quy trình và công cụ
  - Ưu tiên phần mềm tốt hơn tài liệu đầy đủ
  - Ưu tiên hợp tác với khách hàng hơn thỏa thuận hợp đồng

[Manifesto for Agile Software Development]
https://agilemanifesto.org/

Các nguyên lý mô hình hóa Linh hoạt

- Mô hình hóa linh hoạt / Agile Modeling (AM) (12 nguyên lý)
  - Định hướng chung: Chỉ xây dựng các mô hình cần thiết và hữu ích ở mức chi tiết vừa đủ.



Các nguyên lý Mô hình hóa Linh hoạt(2)

- Phần mềm là mục tiêu chính
  - Không để bị phân tán bởi tài liệu hoặc mô hình
- Nỗ lực tiếp theo là mục tiêu thứ 2 của bạn
  - Quan tâm tới phiên bản tiếp theo
- Cực tiểu hóa hoạt động mô hình hóa
  - Chỉ xây dựng nhưng gì giúp dự án tiến triển
- Mô hình hóa có mục đích
  - Mô hình hóa để hiểu
  - Mô hình hóa để trao đổi thông tin
- Xây dựng nhiều mô hình
  - Phân tích vấn đề từ nhiều góc độ
- Xây dựng các mô hình chất lượng cao và lấy phản hồi

Các nguyên lý Mô hình hóa Linh hoạt(3)

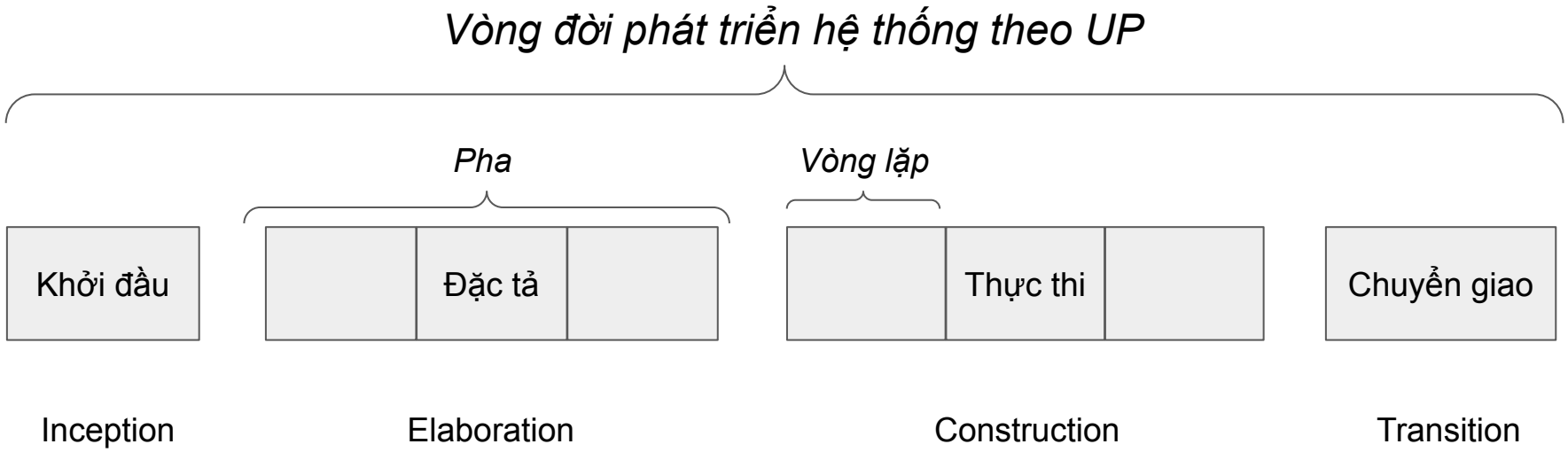
- Nội dung là trọng tâm, không phải hình thức
  - Các mô hình vẽ bằng tay đôi khi cũng tốt
  - Luôn tập trung vào nhu cầu của cổ đông
- Học hỏi lẫn nhau qua giao tiếp cởi mở
- Hiểu mô hình của bạn và cách sử dụng chúng
- Thích nghi với các nhu cầu dự án cụ thể
- Cực đại hóa tỉ lệ lợi nhuận (ROI) của cổ đông

Quy trình thống nhất

- Quy trình thống nhất / The Unified Process (UP)
- UP là tiên phong trong các mô hình thích nghi linh hoạt
- UP và UML được phát triển cùng nhau

Các pha trong quy trình thống nhất

- UP chia SDLC thành 4 pha, mỗi pha bao gồm nhiều vòng lặp, tương ứng với 4 mảng chính của dự án



Các pha trong quy trình thống nhất(2)

- Khởi đầu - Bắt đầu thực hiện dự án
  - Phân tích tính khả thi
  - Thu thập các yêu cầu
  - v.v..
- Đặc tả - Hiểu các yêu cầu hệ thống
  - Phân tích và thiết kế
  - v.v..
- Thực thi - Xây dựng hệ thống
  - Lập trình và kiểm thử
  - v.v..
- Chuyển giao - Chuẩn bị và triển khai hệ thống
  - Kiểm thử tổng thể
  - Triển khai hệ thống, đưa hệ thống vào ứng dụng
  - v.v..

Các luồng công việc trong UP

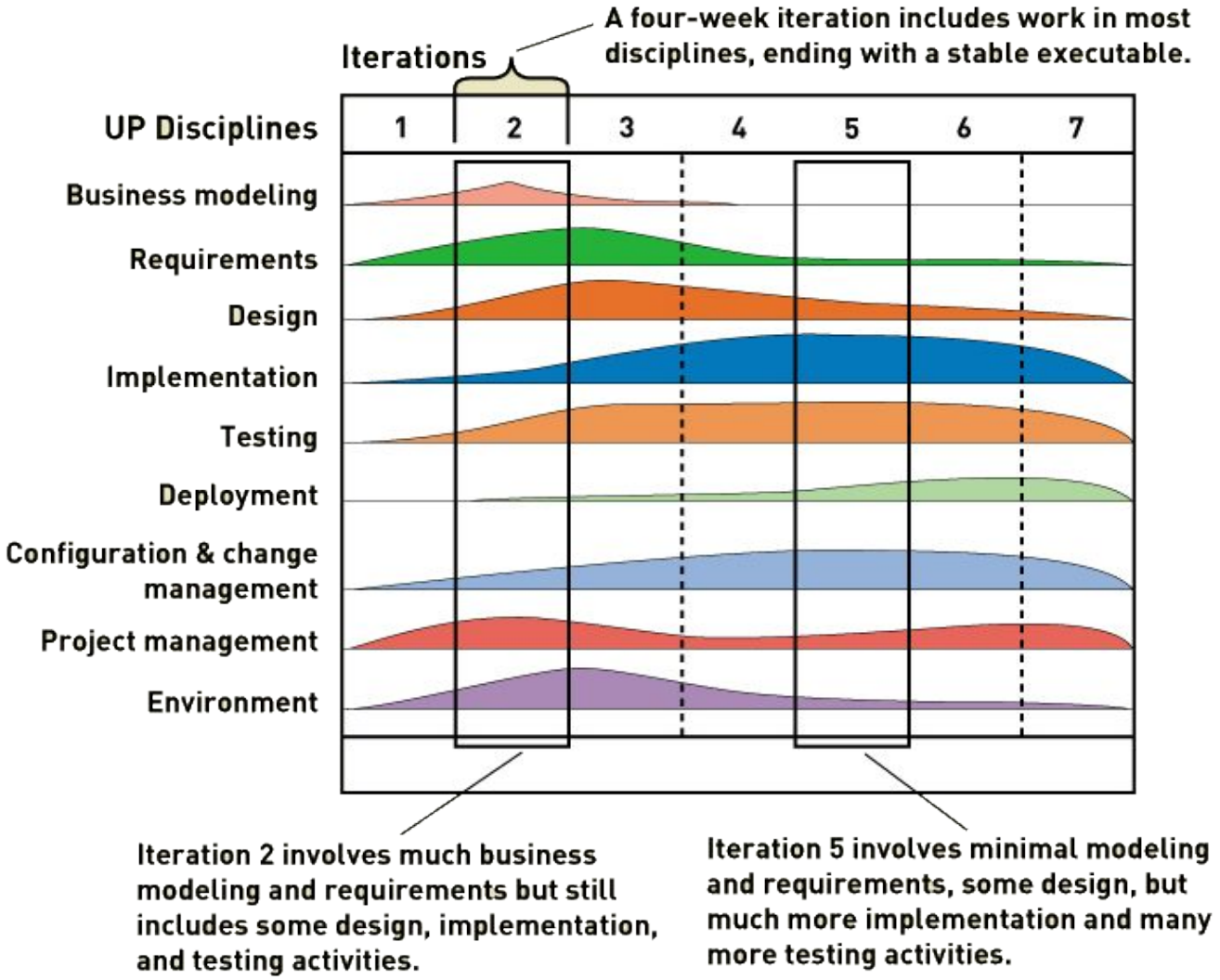
- Luồng công việc / Workflow
- Luồng công việc là 1 tập các hoạt động có chức năng gần nhau
  - Các chức năng liên quan trong 1 luồng công việc cùng hướng tới đạt 1 mục đích chung trong phát triển dự án
- Các loại luồng công việc - Có 2 loại tổng quát
  - Luồng công việc phát triển
  - Luồng công việc quản lý
    - Lập kế hoạch và điều hành

Các luồng công việc trong UP(2)

- Các luồng phát triển
  - Mô hình hóa nghiệp vụ / Business modeling
  - Yêu cầu / Requirements
  - Thiết kế / Design
  - Thực thi / Implementation
  - Kiểm thử / Testing
  - Phân phối / Deployment
- Các luồng quản lý
  - Quản lý cấu hình và thay đổi / Configuration and Change Management
  - Quản lý dự án / Project Management
  - Môi trường / Environment

Các luồng công việc trong UP(3)

- Các luồng công việc được sử dụng xuyên suốt vòng lặp



Quy trình thống nhất tăng cường

Mở rộng quy trình thống nhất:

- Thêm pha sản phẩm để xử lý các vấn đề sau khi đã chuyển giao sản phẩm
- Bổ xung các luồng công việc
  - Vận hành & Hỗ trợ / Operation & Support
  - Quản lý hạ tầng / Infrastructure Management

Lập trình mạo hiểm

Lập trình mạo hiểm / Extreme Programming (XP)

- Sử dụng các kỹ thuật phát triển phần mềm tốt nhất và mở rộng chúng
  - Tập chung chủ yếu vào các kỹ thuật đã được kiểm chứng trong thực tế
  - Kết hợp chúng theo cách hợp lý để có kết quả tốt hơn
- Các giá trị cốt lõi của XP
  - Giao tiếp / Communication
  - Đơn giản / Simplicity
  - Phản hồi / Feedback
  - Can đảm / Courage

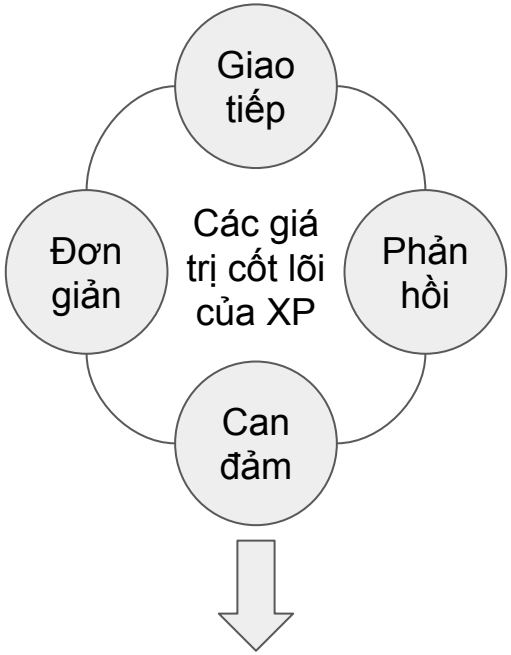
Các thông lệ lập trình mạo hiểm

- Lập kế hoạch - Dựa trên các câu chuyện người dùng
- Kiểm thử - Kiểm thử kỹ lưỡng tất cả các bước
- Lập trình theo cặp - Theo dõi, kiểm tra, cân đối
- Thiết kế đơn giản - Các nguyên lý mô hình hóa Linh hoạt
- Tái cấu trúc - Viết lại mã nguồn theo cách tốt hơn (nếu có) cùng với việc phát triển các tính năng mới
- Đồng sở hữu mã nguồn - Bất kỳ ai cũng có thể kiểm tra và làm mã nguồn tốt hơn.
- Tích hợp liên tục - Phần mềm được phát triển liên tục
- Khách hàng thường trực - Khách hàng tham gia vào tiến trình phát triển
- Hình tượng hệ thống - Quen thuộc với các thành viên trong dự án và mô phỏng hệ thống thực tế

Các thông lệ lập trình mạo hiểm(2)

- Phiên bản nhỏ - Thường xuyên phát hành các phiên bản mới
- Tuần làm việc 40 giờ - Không làm quá tải người lập trình.
- Quy chuẩn mã nguồn - Tương thích với các quy chuẩn mã nguồn

Các giá trị cốt lõi và thông lệ của XP

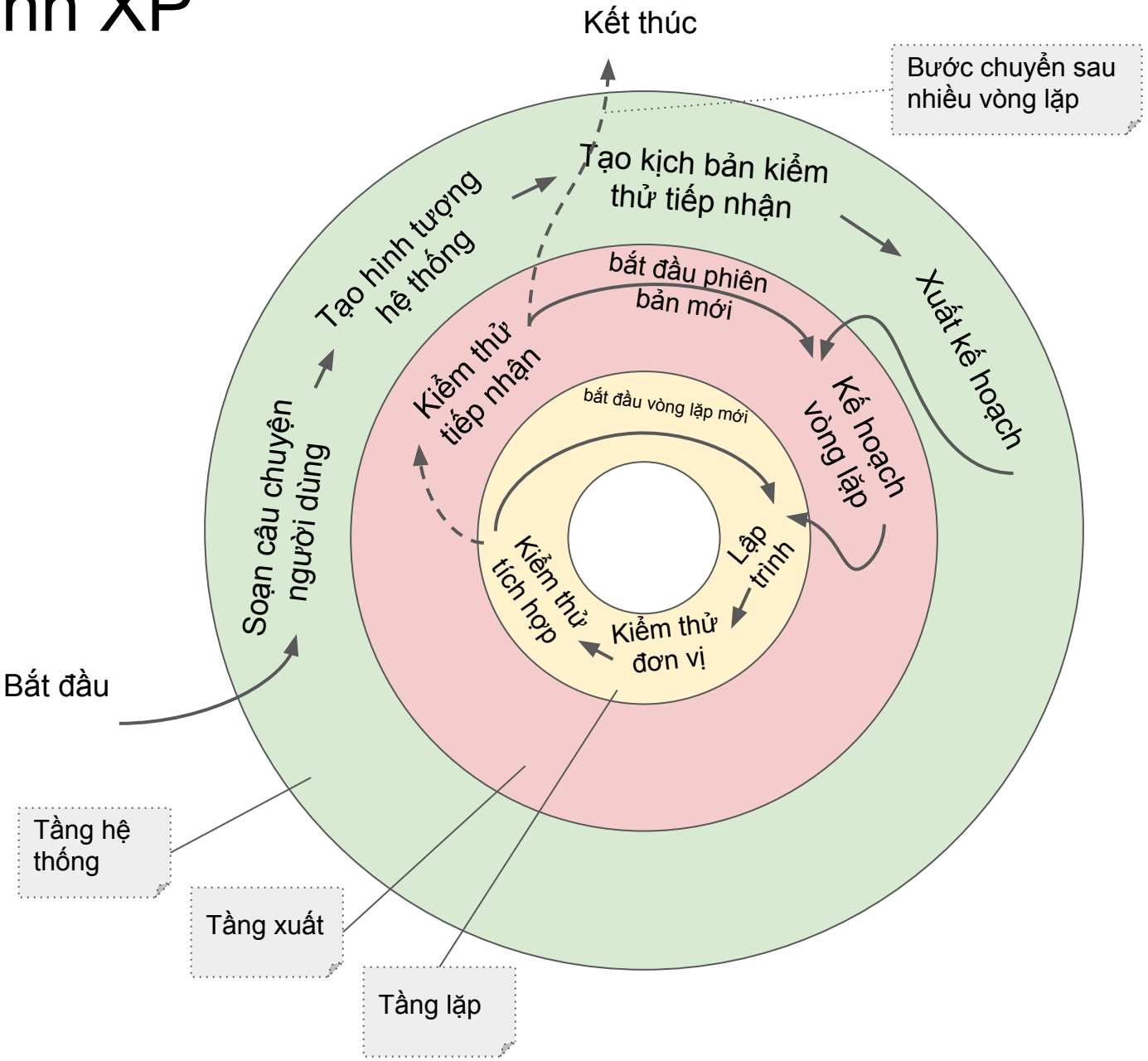


Các thông lệ XP			
Lập kế hoạch	Kiểm thử	Lập trình theo cặp	Thiết kế đơn giản
Tái cấu trúc mã nguồn	Đồng sở hữu mã nguồn	Tích hợp liên tục	Khách hàng thường trực
Hình tượng hệ thống	Phiên bản nhỏ	Tuần làm việc 40 giờ	Quy chuẩn mã nguồn

Mô hình XP

- Mô hình 3 tầng (3 vòng xuyến)
  - Vòng xuyến ngoài cùng - Soạn các câu chuyện người dùng và xác định các kiểm thử tiếp nhận
    - Tầng hệ thống / System level
  - Vòng xuyến ở giữa - Thực hiện kiểm thử và lập kế hoạch tổng thể
    - Tầng xuất / Release level
  - Vòng xuyến ở trong - Các vòng lặp lập trình và kiểm thử
    - Tầng lặp / Iteration level

Mô hình XP

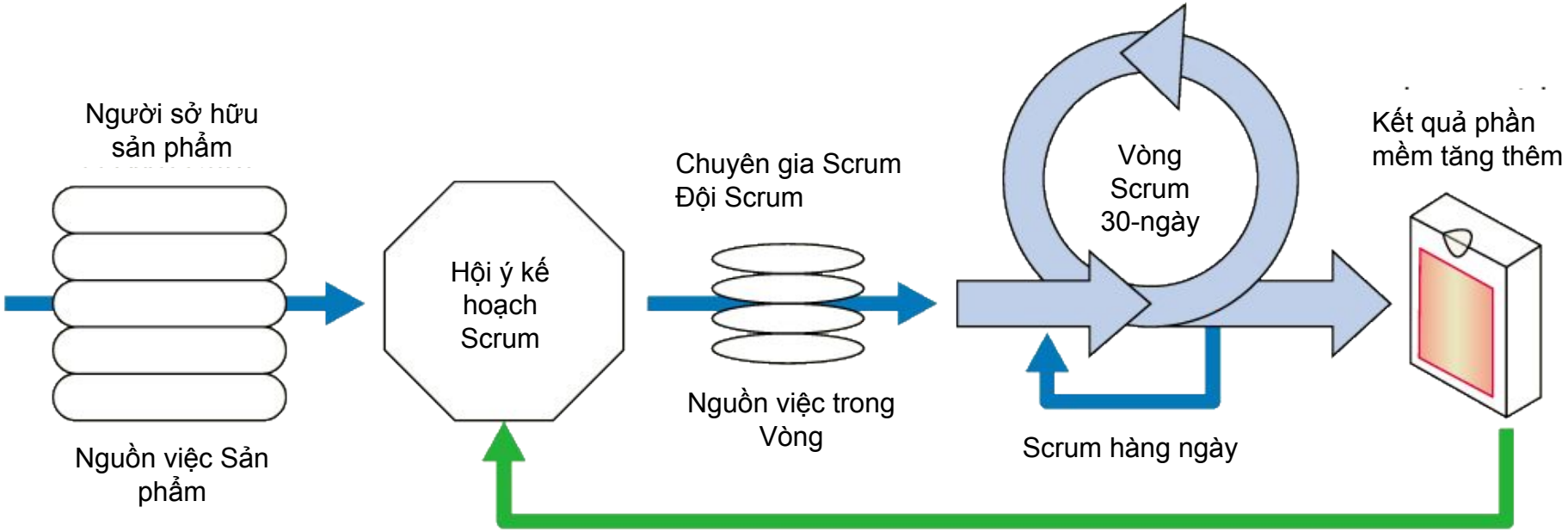

Kết thúc
Bước chuyển sau nhiều vòng lặp
Tạo kịch bản kiểm thử tiếp nhận
Xuất kế hoạch
Tạo hình tượng hệ thống
bắt đầu phiên bản mới
Kiểm thử tiếp nhận
Kế hoạch vòng lặp
bắt đầu vòng lặp mới
Soạn câu chuyện người dùng
Kiểm thử tích hợp
Lập trình
Kiểm thử đơn vị
Bắt đầu
Tầng hệ thống
Tầng xuất
Tầng lặp

SCRUM

- Kết hợp các nguyên lý trong thể thao và phát triển linh hoạt
  - Môn bóng bầu dục / Rugby
- Nguồn việc / Backlog sản phẩm
  - Danh sách yêu cầu có thứ tự ưu tiên
- Người sở hữu sản phẩm / Product owner
  - Quản lý nguồn việc
- Chuyên gia Scrum / Scrum master
  - Người thiết lập Scrum
- Người phát triển / Developer
  - Thực thi các công việc

Scrum Sprint

- Vòng Scrum / Scrum Sprint
- Giống như 1 dự án nhỏ trong giới hạn thời gian (thường <= 1 tháng) để triển khai 1 phần hệ thống



Các thông lệ Scrum

- Phạm vi của mỗi vòng được đóng băng (nhưng có thể được cắt giảm nếu cần)
- Khoảng thời gian được duy trì cố định
- Hội ý Scrum hàng ngày
  - Bạn đã làm gì kể từ cuộc họp trước (trong 24 giờ qua)?
  - Bạn sẽ làm gì tới cuộc họp kế tiếp?
  - Điều gì khiến bạn chưa hoàn thành công việc của mình?