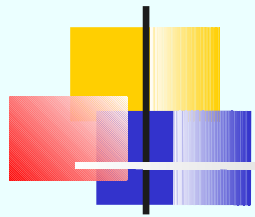


Công nghệ phần mềm

Một số mô hình vòng đời phát triển phần mềm

Giảng viên: TS. Nguyễn Mạnh Hùng
Học viện Công nghệ Bưu chính Viễn thông (PTIT)



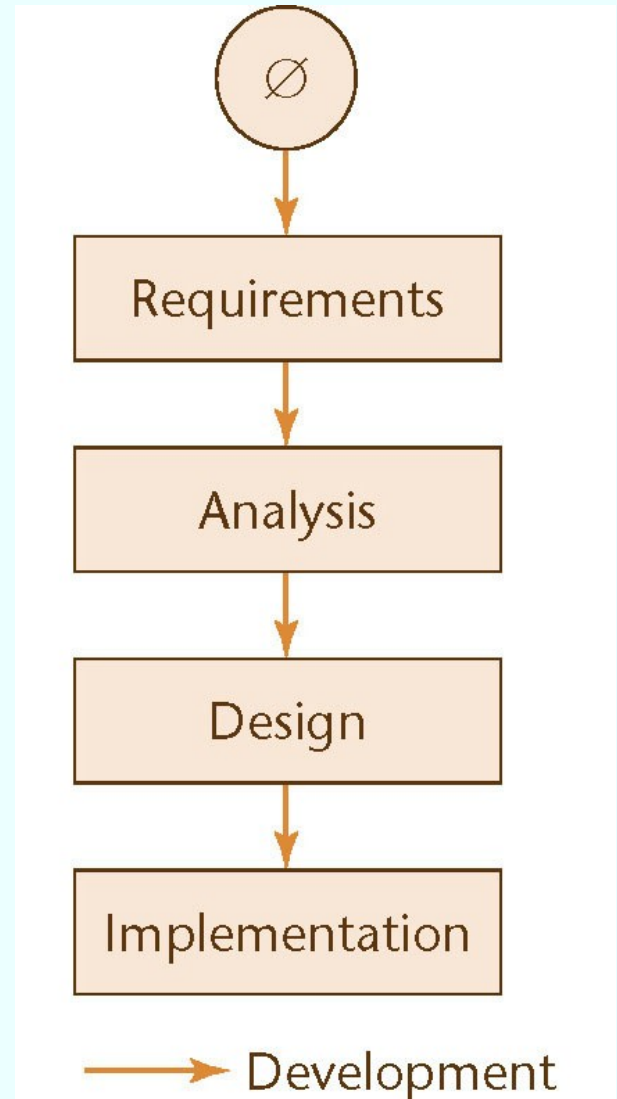
Nội dung tham khảo từ

Stephen R. Schach. *Object-Oriented and Classical Software Engineering*. Seventh Edition, WCB/McGraw-Hill, 2007

Mô hình trên lí thuyết

Trên lí thuyết thì:

- Các pha được tiến hành tuần tự
- Bắt đầu phát triển hoàn toàn từ không có gì

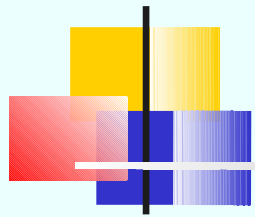




Thực tế...

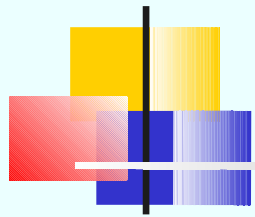
Phát triển phần mềm hoàn toàn khác:

- Lỗi có thể xảy ra mọi lúc mọi nơi trong tiến trình phát triển
- Khách hàng thay đổi hoặc không nắm rõ yêu cầu



Vấn đề thay đổi yêu cầu (1)

- Khách hàng có thể thay đổi yêu cầu ngay khi phần mềm đang được phát triển
- Ngay cả khi thay đổi có lí do hợp lí, thì mọi thay đổi đều ảnh hưởng đến phần mềm
- Các thay đổi có thể dẫn đến lỗi hồi quy (regression fault)
- Nếu thay đổi quá nhiều → phải thiết kế và cài đặt lại phần mềm



Vấn đề thay đổi yêu cầu (2)

Yêu cầu thay đổi là việc không tránh khỏi:

- Khách hàng là công ty đang phát triển thì yêu cầu thay đổi thường xuyên
- Mỗi cá nhân/khách hàng đều có quyền thay đổi yêu cầu của mình
- → hiện chưa có giải pháp triệt để để giải quyết vấn đề này!



Mô hình lặp và tăng trưởng (1)

Thực tế:

- Các pha phát triển không kết thúc khi chuyển sang pha khác, nó kéo dài liên tục trong suốt vòng đời phát triển → gọi là các workflow
- Bản chất của tiến trình phát triển phần mềm là lặp: lặp lại các bước nhiều lần, kết quả lần sau sẽ tốt hơn lần trước

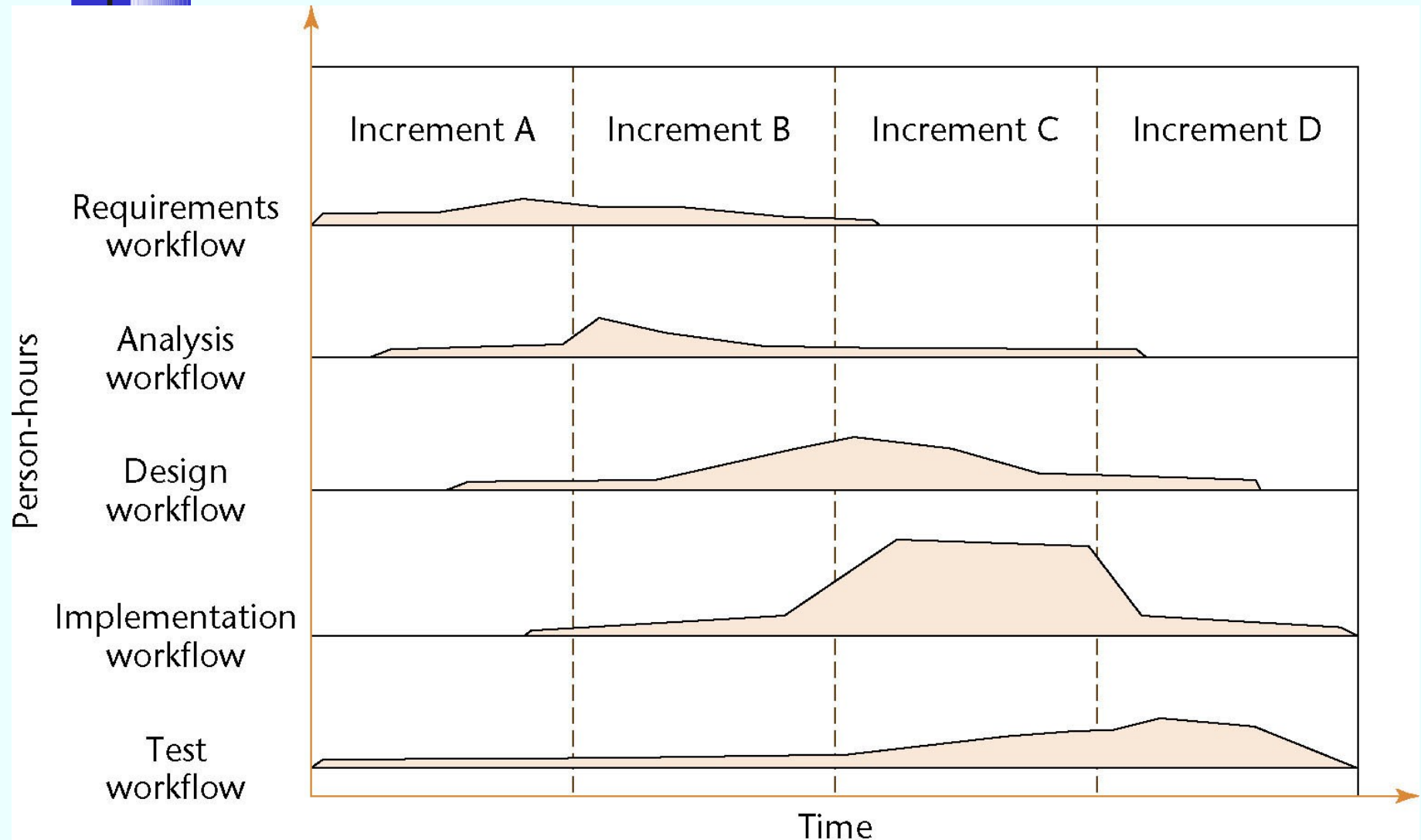


Mô hình lập và tăng trưởng (2)

Luật Miller:

- Tại mỗi thời điểm, người ta chỉ có thể tập trung vào tối đa khoảng 7 vấn đề
- để xử lý các vấn đề lớn, sử dụng phương pháp làm mịn từng bước:
- Tập trung xử lý các việc quan trọng trước
 - Các việc ít quan trọng hơn xử lý sau
- gọi là tiến trình tăng trưởng

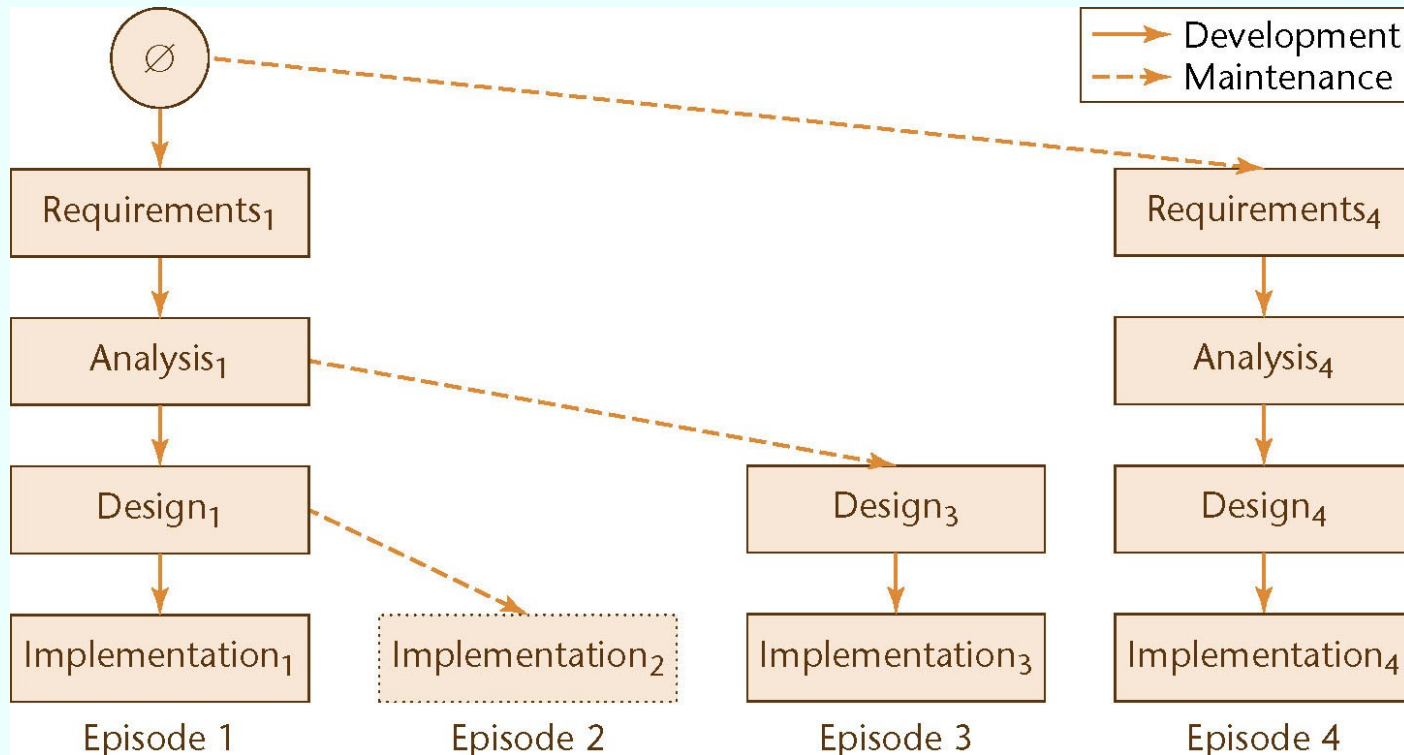
Mô hình lặp và tăng trưởng (3)



Mô hình lặp và tăng trưởng (4)

Lặp và tăng trưởng kết hợp nhau:

- Không có các pha đơn lẻ, mà mỗi pha được lặp lại nhiều lần





Mô hình lập và tăng trưởng (5)

Dùng khái niệm workflow thay vì phase:

- Thực tế không tồn tại tuần tự các pha
- Tất cả 5 workflow đều hoạt động trong suốt vòng đời PTPM
- Tại mỗi giai đoạn, có một workflow chiếm vị trí trọng tâm



Mô hình lặp và tăng trưởng (6)

- Có thể coi dự án là một tập các dự án nhỏ, mỗi dự án nhỏ tương ứng với một lần tăng trưởng
- Mỗi dự án nhỏ đều có artifact cho mỗi workflow:
 - Mở rộng các artifacts (tăng trưởng)
 - Kiểm thử các artifacts (test)
 - Thay đổi artifacts (lặp)
- Mỗi dự án nhỏ thực hiện một phần của dự án ban đầu



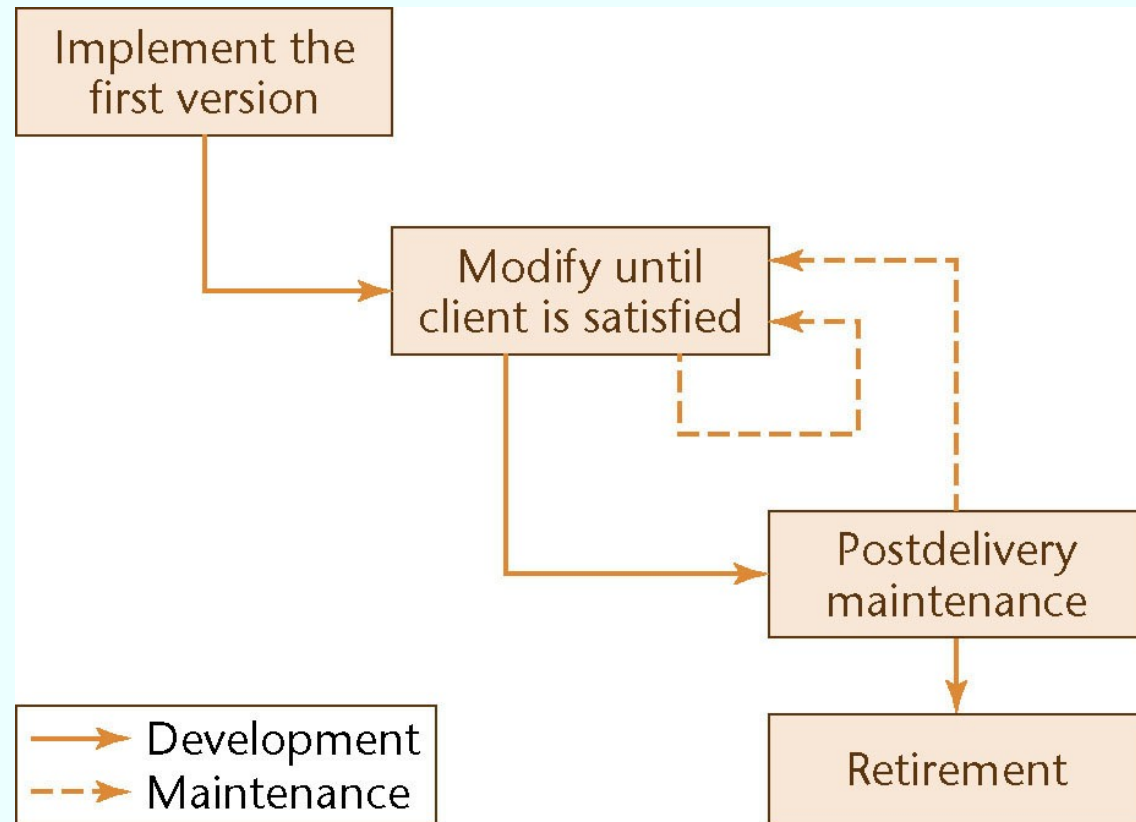
Mô hình lặp và tăng trưởng (7)

Ưu điểm:

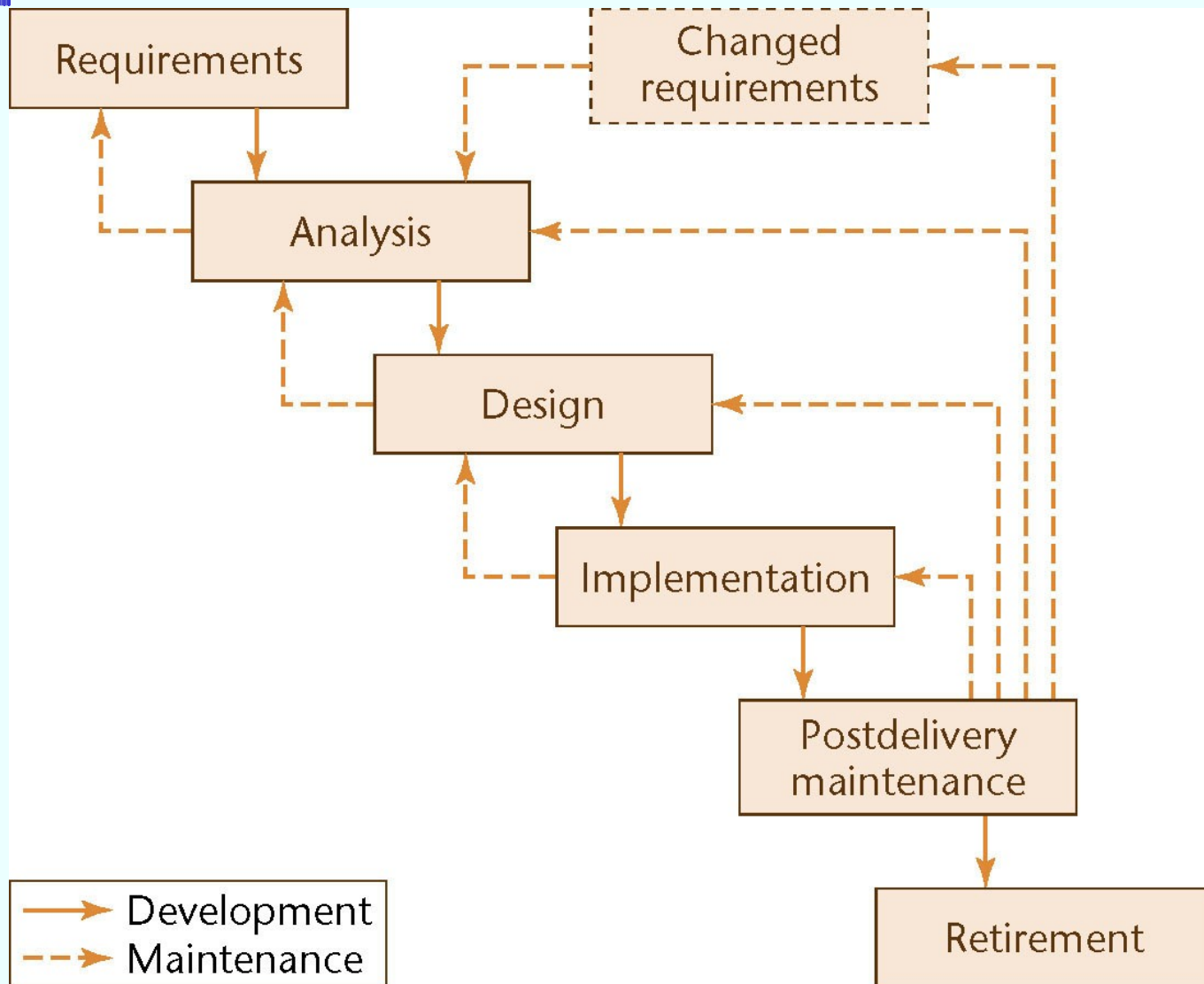
- Mỗi bước lặp đều có test workflow → có thể phát hiện và sửa lỗi sớm
- Thiết kế kiến trúc ngay từ đầu: mô hình theo modul ngay từ đầu, và việc tính toán tránh lỗi khi kết hợp được tính toán trước
- Có thể có phiên bản dùng được của sản phẩm ngay từ giai đoạn đầu
- Khách hàng có thể dựa vào phần đã hoàn thành để nêu chính xác yêu cầu trong những modul sau

Mô hình xây và sửa

- Không có đặc tả, không có thiết kế
- Chỉ cài đặt
- Bảo trì mò mẫm
- Ưu điểm?
- Nhược điểm?



Mô hình thác nước (1)





Mô hình thác nước (2)

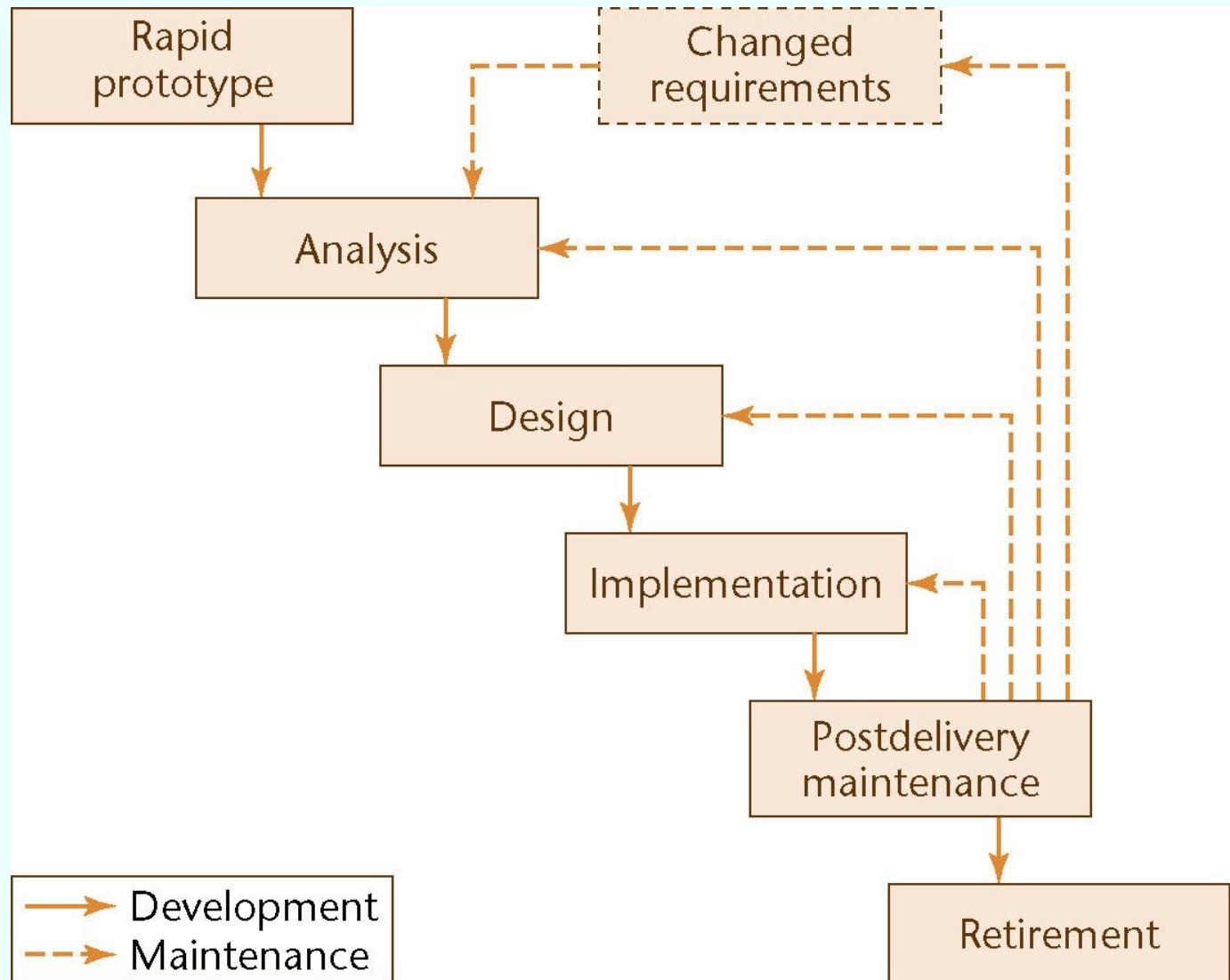
Đặc trưng:

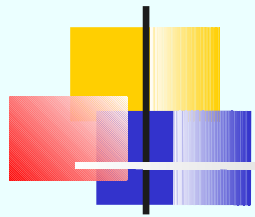
- Các vòng lặp phản hồi sau mỗi pha
- Làm tài liệu cuối mỗi pha

Ưu và nhược điểm?

- ...

Mô hình bản mẫu nhanh (1)





Mô hình bản mẫu nhanh (2)

Đặc trưng:

- Tiến hành làm bản mẫu nhanh (rapid prototype) trong pha lấy yêu cầu
- Các pha còn lại làm theo thứ tự tuyến tính

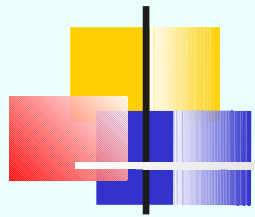
Ưu và nhược điểm?



Tiến trình linh hoạt (1)

Trích chọn các story của sản phẩm:

- Ước lượng thời gian và chi phí
- Chọn story tiếp theo để phát triển
- Mỗi story được chia nhỏ thành các task
- Viết các test case cho các task trước khi cài đặt



Tiến trình linh hoạt (2)

Phát triển mỗi story:

- Không có pha đặc tả
- Thiết kế linh hoạt và có thể thay đổi theo yêu cầu của khách hàng
- Luôn có đại diện của khách hàng trong team
- Lập trình theo cặp
- Liên tục tích hợp các task



Tiến trình linh hoạt (3)

Sử dụng phương pháp họp đứng (stand-up meeting):

- Toàn bộ team đứng vòng tròn và nhìn thấy được nhau
- Thời gian họp cố định hàng ngày, nhưng không quá 15p
- Họp để thấy vấn đề, nếu có, chứ không giải quyết vấn đề
- Lần lượt mỗi người trả lời các câu hỏi giống nhau



Tiến trình linh hoạt (4)

Các câu hỏi khi họp đứng:

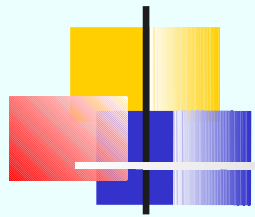
- Tôi đã làm được gì từ buổi họp hôm qua?
- Hôm nay tôi đang làm cái gì?
- Có vấn đề gì với việc đang làm hôm nay?
- Chúng ta có quên làm phần nào không?
- Tôi đã học thêm được gì khi làm việc với team?



Tiến trình linh hoạt (5)

Chiến lược:

- Mỗi story chỉ phát triển liên tục và phải hoàn thiện sau 2-3 tuần
 - Cứ sau 3 tuần hoàn thành một bước lặp và bàn giao tính năng mới cho khách hàng
 - Sử dụng kĩ thuật timeboxing để quản lí thời gian
- mô hình này yêu cầu cố định thời gian, không cố định tính năng của sản phẩm

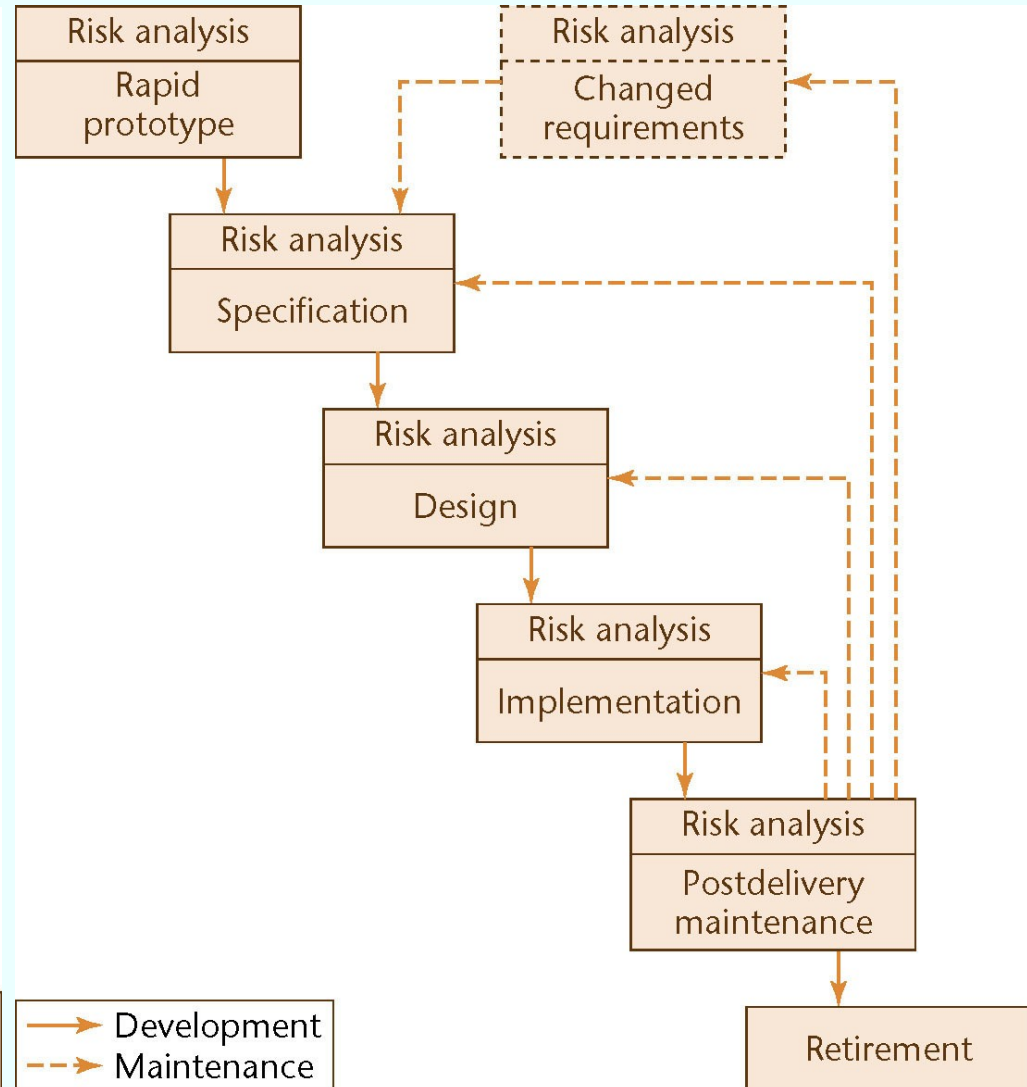
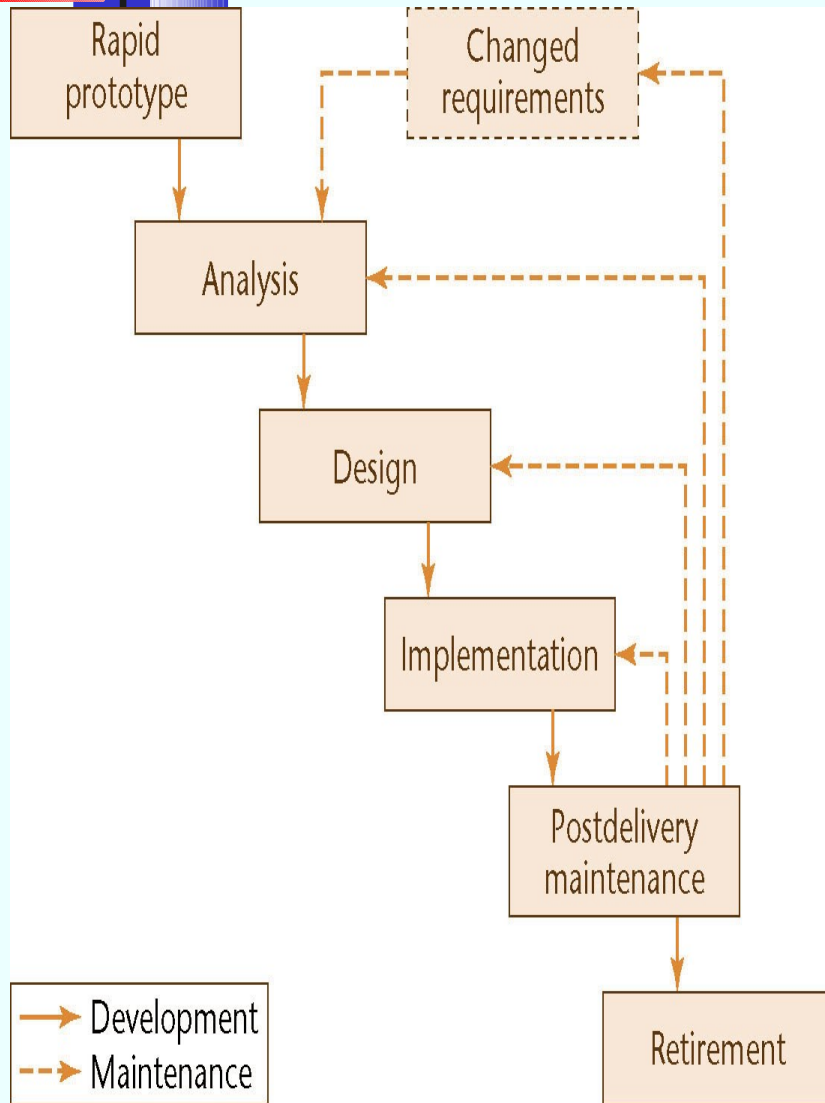


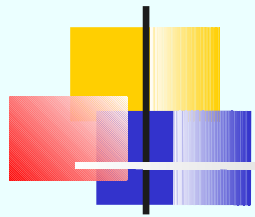
Tiến trình linh hoạt (6)

Ưu điểm và nhược điểm:

- Phương pháp họp đứng (stand-up meeting)?
- Kỹ thuật timeboxing?

Mô hình xoắn ốc (1)



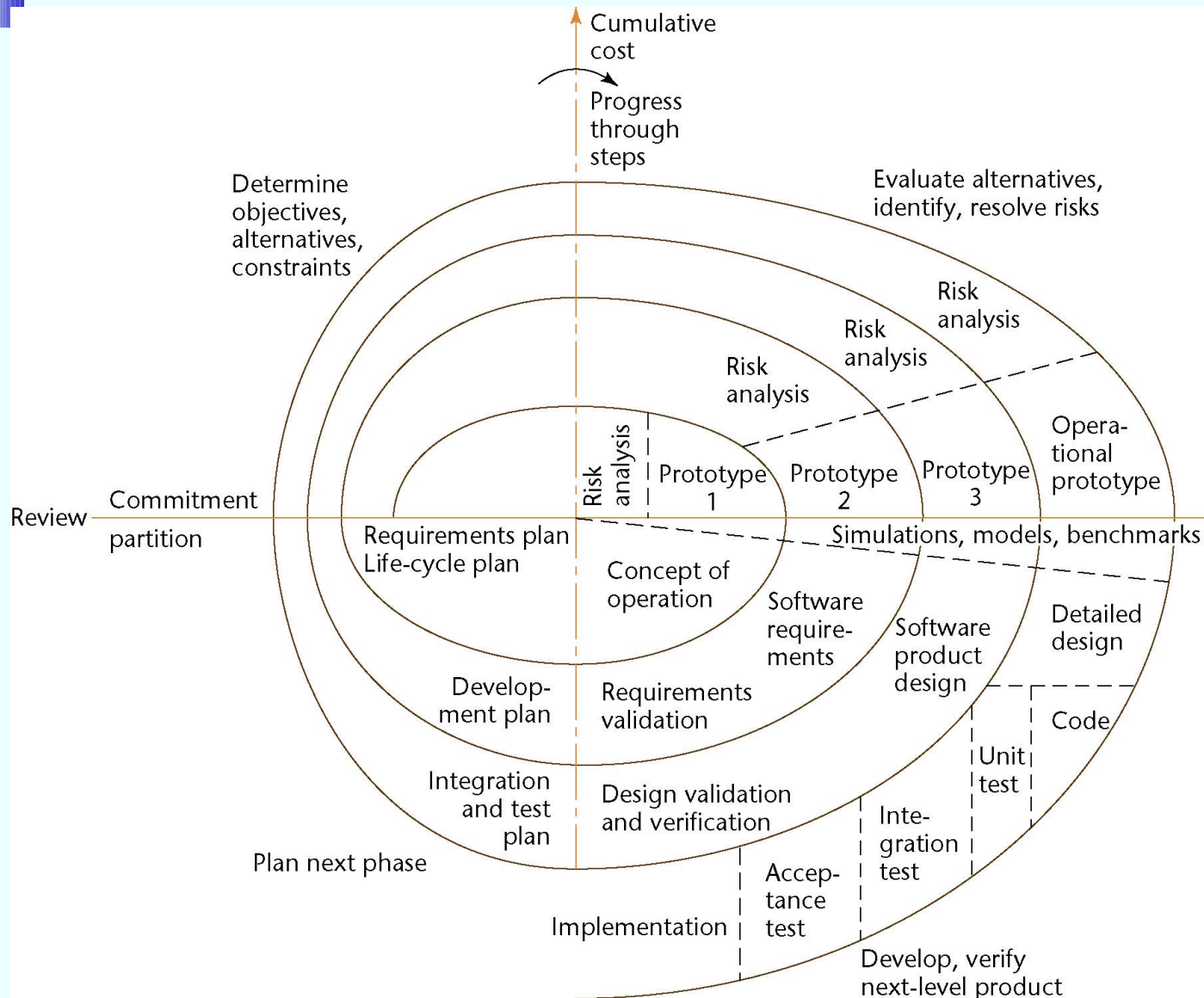


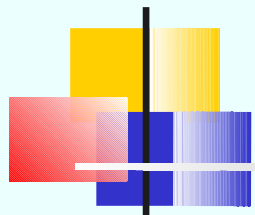
Mô hình xoắn ốc (2)

Đặc trưng:

- Có nhiều vòng lặp nhau, nhưng vòng lặp sau phát triển rộng hơn vòng trước
- Mỗi pha của mỗi lần lặp:
 - Bắt đầu bằng việc quyết định và phân tích rủi ro
 - Kết thúc bằng việc đánh giá lỗi và lập kế hoạch cho pha tiếp theo
 - Nếu các rủi ro đều không xử lí được thì dừng lại ngay lập tức

Mô hình xoắn ốc (3)





Mô hình xoắn ốc (4)

Ưu điểm và nhược điểm?

- ...



Questions?
