

Nhập môn công nghệ phần mềm Tiến trình phần mềm

GV: ThS. Ngô Tiến Đức



Nội dung chính (1)

- Luồng lấy yêu cầu
- Luồng phân tích
- Luồng thiết kế
- Luồng cài đặt
- Luồng kiểm thử



Nội dung chính (2)

- Tiến trình thống nhất
- Các pha trong tiến trình thống nhất
- Cải tiến tiến trình phần mềm



Workflow

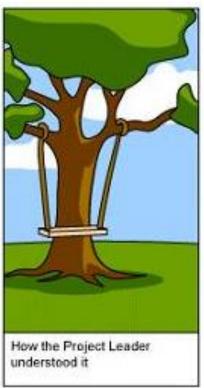
- Workflow (tạm dịch: luồng): sự sắp xếp có hệ thống các bước và hoạt động để hoàn thành một nhiệm vụ.
- Xác định cách thức công việc được thực hiện từ đầu đến cuối.



Luồng lấy yêu cầu (1)

Mục tiêu: Xác định nhu cầu của khách hàng









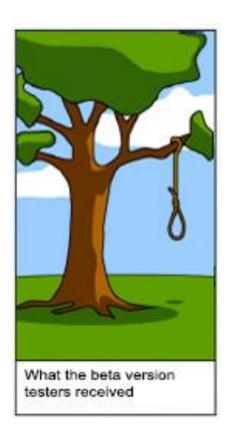


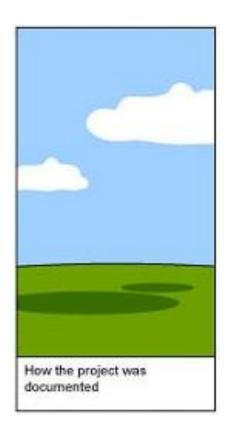


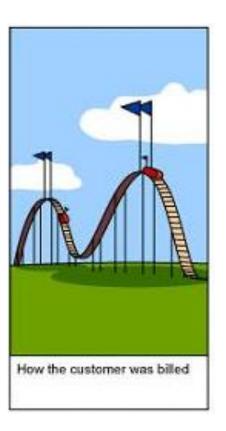
Luồng lấy yêu cầu (1)

(J4F)











Luồng lấy yêu cầu (2)

- Cần xác định lĩnh vực phần mềm
- Xây dựng mô hình nghiệp vụ:
 - Làm việc với chuyên gia nghiệp vụ
 - Sử dụng công cụ UML tiếp cận theo hướng nghiệp vụ
 - Đánh giá tính khả thi
 - Khách hàng cảm thấy chi phí không hợp lý?



Luồng lấy yêu cầu (3)

Kết quả cần đạt được:

- Thời hạn bàn giao
- Độ tin cậy
- Chi phí
- Một số yêu cầu khác: độ tin cậy, thời gian phản hồi, portability...



Luồng phân tích (1)

Mục tiêu:

- Phân tích và làm mịn (refine) yêu cầu của khách hàng
- Mô tả yêu cầu bằng ngôn ngữ kỹ thuật



Luồng phân tích (2)

- Tài liệu phải thống nhất được cả bên khách hàng và đội phát triển
- Artifact của luồng lấy yêu cầu: Dành cho khách hàng
- -> Tài liệu mô tả bằng ngôn ngữ tự nhiên không chính xác
- Artifact của luồng lấy phân tích: Dành cho đội phát triển
- -> Tài liệu mô tả bằng ngôn ngữ kỹ thuật chính xác và khoa học
- -> Cần 2 loại tài liệu đặc tả riêng biệt



Luồng phân tích (3)

Yêu cầu về tài liệu:

- Đầy đủ yêu cầu
- Không có mâu thuẫn
- Không nhập nhằng
- Theo dõi được (Traceability)



Luồng phân tích (4)

Sau khi tài liệu đặc tả đã được thông qua -> lập kế hoạch quản lý dự án phần mềm (tạm thời):

- Ước lượng chi phí
- Phân chia công việc
- Ước lượng thời gian
- Xác định các mốc (milestone) và sản phẩm sau từng mốc



Luồng thiết kế (1)

- Mục tiêu: Mô hình hóa kết quả phân tích cho đến khi lập trình viên có thể code được
- Các yêu cầu phi chức năng:
 - Chọn ngôn ngữ lập trình
 - Tính tái sử dụng
 - Tính khả chuyển



Luồng thiết kế (2)

Thiết kế cổ điển:

- Thiết kế kiến trúc: Chia nhỏ sản phẩm thành các module
- Thiết kế chi tiết: Lựa chọn cấu trúc dữ liệu và giải thuật



Luồng thiết kế (3)

Thiết kế hướng đối tượng:

- Các class đã được xác định từ pha phân tích
- Thiết kế kiến trúc: Xác định quan hệ giữa các class
- Thiết kế chi tiết: Thiết kế thuộc tính và phương thức cho các class
- -> Kết quả cần đạt được: Bản mẫu các class, thuộc tính và phương thức + thuật toán



Luồng cài đặt

- Mục tiêu: Cài đặt hệ thống theo kết quả từ luồng thiết kế
- Phương pháp: Cài đặt theo class và module -> tích hợp
- Kết quả đạt được: Hệ thống hoàn chỉnh theo yêu cầu



Luồng kiểm thử

- Lấy yêu cầu: Test tài liệu
- Phân tích: Test tài lệu, test kế hoạch
- Thiết kế: Review tài liệu thiết kế
- Cài đặt:
 - Unit test
 - Integration test
 - Product test
 - Acceptance test



Tiến trình thống nhất (1)

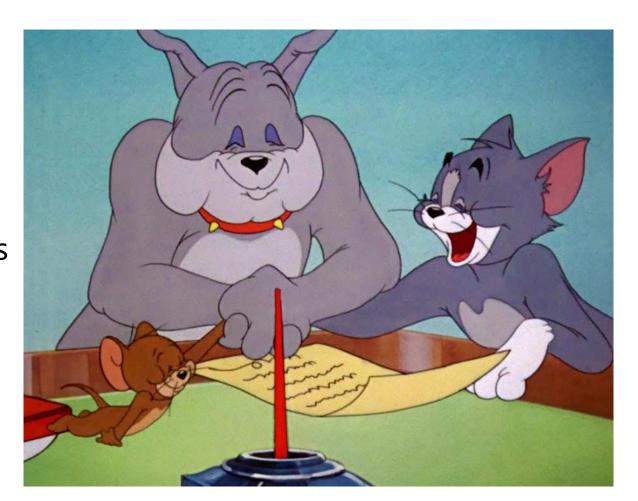
Lịch sử: Có hơn 50 phương pháp mô hình hướng đối tượng (1994)

- Quy trình và ký hiệu đồ họa khác nhau
- Ba phương pháp luận hướng đối tượng thành công nhất:
 - Booch's method
 - Jacobson's OOSE (Object-oriented Software Engineering)
 - Rumbaugh's OMT (Object Modeling Technique)



Tiến trình thống nhất (2)

- 1995: Thống nhất 3 phương pháp luận
- 1997: UML ra đời
- 1999: Công bố Rational Unified Process
 - → Unified Software Development Process
 - → Unified Process





Tiến trình thống nhất (3)

- Tiến trình thống nhất là phương pháp luận có thể áp dụng linh hoạt (adaptable methodology)
 - Cần được chỉnh sửa tùy theo phần mềm
- UML (Unified Modeling Language): các ký hiệu và biểu đồ được sử dụng để thiết kế hệ thống (hướng đối tượng)
 - Giúp giao tiếp nhanh và chính xác hơn



Tiến trình thống nhất (4)

Các biểu đồ UML – UML Diagrams:

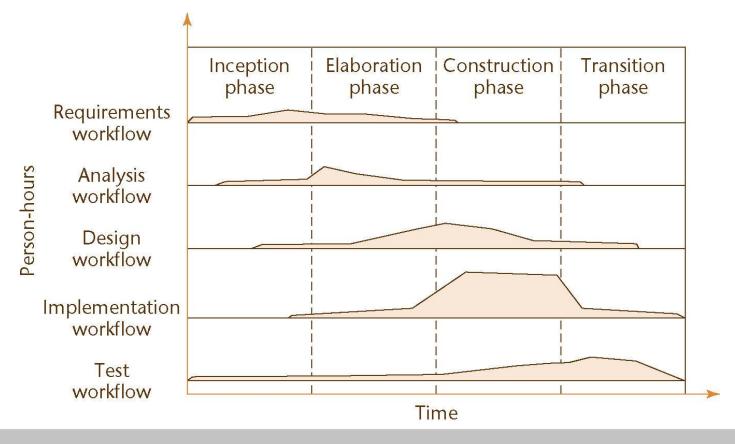
- Biểu đồ ca sử dụng (Use Case)
- Biểu đồ hoạt động (Activity)
- Biểu đồ tương tác:
 - Biểu đồ tuần tự (Sequence)
 - Biểu đồ giao tiếp/cộng tác (Communication/Collaboration)
- Biểu đồ trạng thái (Statechart)
- Biểu đồ cấu trúc tĩnh (Static Structure)
 - Biểu đồ lớp (Class)
 - Biểu đồ đối tượng (Object)
- Biểu đồ thực thi (Implementation)
 - Biểu đồ thành phần (Component)
 - Biểu đồ triển khai (Deployment)



Các pha trong tiến trình thống nhất (1)

Quá trình phát triển trải qua 4 bước tăng trưởng, mỗi bước tương ứng với

một pha:





Các pha trong tiến trình thống nhất (2)

- Các pha:
 - Inception: Khởi đầu/Đánh giá
 - Elaboration: Khảo sát/Thiết lập
 - Construction: Xây dựng
 - Transition: Chuyển giao



Các pha trong tiến trình thống nhất (3)

- Mỗi bước tăng trưởng đều rơi vào:
 - Một trong 5 luồng và
 - Một trong 4 pha
- Tại sao mỗi bước phải chia ra luồng và pha?
 - Luồng: Cách nhìn kỹ thuật
 - Pha: Cách nhìn nghiệp vụ



Pha khởi đầu/đánh giá (1)

- Mục tiêu: Xác định tính khả thi và kinh tế của phần mềm
- Thực hiện:
 - Tìm hiểu lĩnh vực chuyên môn
 - Xây dựng mô hình nghiệp vụ
 - Xác định phạm vi (scope) của dự án
 - Đưa ra bản phân tích về kinh doanh



Pha khởi đầu/đánh giá (2)

Bản phân tích về kinh doanh:

- Giá sản phẩm có hợp lý không?
- Bao lâu vốn được quay vòng?
- Nếu từ bỏ dự án thì chi phí thế nào?
- Chiến dịch tiếp thị sản phẩm cho phần mềm COTS?
 - COTS: Commercial-Off-The-Shelf
- Sản phẩm có thể bàn giao đúng hạn không? Nếu muộn thì thiệt hại gì?



Pha khởi đầu/đánh giá (3)

Đánh giá rủi ro:

- Nhân lực
- Phần cứng
 - Phần cứng hiện tại có đủ tốt không? Có cần phần cứng mới không?
 - Giao phần cứng trễ hẹn? Phương án dự phòng?
- Công cụ hỗ trợ
 - Có sẵn không?
 - Có đủ các chức năng cần thiết không?



Pha khởi đầu/đánh giá (4)

Quan hệ với các luồng:

- Bao gồm phần lớn luồng lấy yêu cầu
- Một phần luồng phân tích, thiết kế
- Gần như không bao gồm cài đặt
 - Chỉ khi cần xây dựng bản mẫu khái niệm (proof-of-concept prototype)
- Kiểm thử yêu cầu



Pha khảo sát/thiết lập

- Mục tiêu:
 - Min hóa các kết quả của pha trước
 - Phân tích rủi ro theo mức độ nghiêm trọng
 - Đưa ra kế hoạch quản lý dự án
- Quan hệ với các luồng: Bao gồm phần lớn luồng phân tích và bắt đầu thiết kế kiến trúc



Pha xây dựng

- Mục tiêu: Xây dựng phiên bản đầu tiên hoạt động được của sản phẩm (bản beta)
- Làm tài liệu hướng dẫn sử dụng cho khách hàng
- Cập nhật bản phân tích kinh doanh (nếu cần)
- Quan hệ với các luồng:
 - Bao gồm thiết kế chi tiết và phần lớn luồng cài đặt
 - Luồng kiểm thử: unit test, integration test, product test



Pha chuyển giao

- Mục tiêu: Đảm bảo tất cả yêu cầu của khách hàng đã được đáp ứng
- Dựa vào phản hồi từ bản beta đã được cài đặt
 - Tất cả lỗi được xác định đã được sửa
 - Nỗ lực phát hiện các rủi ro chưa xác định trước đó
 - Tài liệu HDSD hoàn chỉnh cho người dùng
 - Tài liệu cho toàn bộ dự án



Cải tiến tiến trình phần mềm (1)

CMM (Capability Maturity Model)

- Chuẩn quản lý quy trình chất lượng của các sản phẩm phần mềm
- Bộ các tiêu chuẩn đánh giá chiến lược cải tiến tiến trình phần mềm: SW-CMM
- Các tài liệu HDSD đã hoàn chỉnh
- Phiên bản nâng cấp: CMMI (Capability Maturity Model Integration)



Cải tiến tiến trình phần mềm (2)

SW - CMM

- Ra đời năm 1986 bởi SEI
- Mục tiêu: Cải tiến tiến trình phần mềm
 - Cải thiện chất lượng
 - Giao hàng đúng hạn, phù hợp ngân sách
 - Cải tiến kỹ thuật
- 5 cấp độ



Cải tiến tiến trình phần mềm (3)

SW – CMM level 1: Initial

- Không thể đoán trước toàn bộ tiến trình
- Việc quản lý chỉ bao gồm xử lý các rủi ro gặp phải
- Gần như tất cả các công ty đều đạt chuẩn level 1



Cải tiến tiến trình phần mềm (4)

SW – CMM level 2: Repeatable

- Các quyết định đưa ra dựa vào các dự án trước đó
- Tạo ra các thước đo (metrics)
- Sử dụng để ước lượng cho các dự án tiếp theo
- Sửa lỗi ngay khi phát hiện
- Mất khoảng 3 5 năm để từ level 1 -> level 2



Cải tiến tiến trình phần mềm (5)

SW – CMM level 3: Defined

- Các khía cạnh quản lý và kỹ thuật được xác định rõ ràng
- Liên tục nâng cao chất lượng và năng suất (productivity)
- Việc đánh giá (reviews) được thực hiện thường xuyên
- Mất khoảng 1.5 đến 3 năm để từ level 2 -> level 3



Cải tiến tiến trình phần mềm (6)

SW – CMM level 4: Managed

- Chất lượng và năng suất được đặt ra cho mỗi dự án
 - Liên tục được giám sát
- Điều chỉnh dựa theo kết quả thống kê chất lượng



Cải tiến tiến trình phần mềm (7)

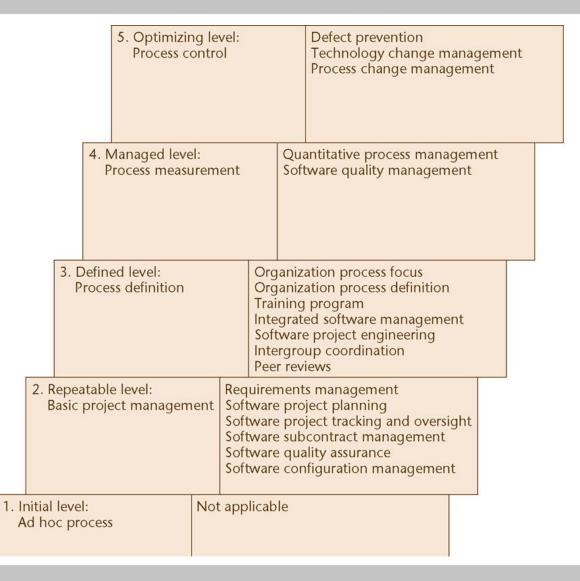
SW – CMM level 5: Optimizing

- Tiến trình được cải tiến liên tục:
 - Kiểm soát quy trình
 - Kiểm soát chất lượng thống kê
 - Ghi nhận phản hồi và kinh nghiệm sau mỗi sản phẩm



Cải tiến tiến trình phần mềm (8)

Key Process Areas cho từng level:





Bài tập về nhà

Trả lời câu hỏi từ 21 đến 30 trong ngân hàng câu hỏi thi



Tài liệu tham khảo

- Stephen R. Schach. *Object-Oriented and Classical Software Engineering*. 8th Edition, WCB/McGraw-Hill, 2010
- T. Đ. Quế, N. M. Hùng. *Bài giảng Nhập môn công nghệ phần mềm*. HVCNBCVT, 2020