

Nhập môn công nghệ phần mềm Phạm vi của công nghệ phần mềm

GV: ThS. Ngô Tiến Đức



Nội dung chính (1)

- Khía cạnh lịch sử
- Khía cạnh kinh tế
- Khía cạnh bảo trì
- Khía cạnh phân tích thiết kế
- Khía cạnh lập trình nhóm



Nội dung chính (2)

- Khía cạnh lập kế hoạch
- Khía cạnh kiểm thử
- Khía cạnh làm tài liệu
- Phương pháp hướng đối tượng
- Kết luận



Khía cạnh lịch sử (1)

- 1967: NATO đưa ra thuật ngữ Công nghệ phần mềm
- 1968: Hội nghị giải quyết cuộc khủng hoảng phần mềm
- Phần mềm được bàn giao:
 - Trễ thời hạn
 - Vượt quá ngân sách
 - Tiềm ẩn lỗi



Khía cạnh lịch sử (2)



9236 dự án phần mềm năm 2004



Khía cạnh lịch sử (3)

- 2002: 78% công ty IT bị dính vào kiện tụng
 - 67% do chức năng hệ thống không đúng
 - 56% do trễ hẹn giao sản phẩm nhiều lần
 - 45% do hệ thống lỗi nghiêm trọng không thể sử dụng



Khía cạnh lịch sử (4)

- Khủng hoảng phần mềm chưa thể giải quyết dứt điểm
- "Software depression"
 - Long duration
 - Poor prognosis



Khía cạnh kinh tế

- Có công nghệ mới nhanh hơn công nghệ đang sử dụng. Có nên thay thế không?
 - Thông thường: Tất nhiên
 - CNPM: Cân nhắc



Khía cạnh bảo trì (1)

Mô hình vòng đời phát triển phần mềm

- Ví dụ: Mô hình thác nước waterfall
 - 1. Requirements phase
 - 2. Analysis (specification) phase
 - 3. Design phase
 - 4. Implementation phase
 - 5. Postdelivery maintenance
 - 6. Retirement



Khía cạnh bảo trì (2)

- Hành động nào là của pha bảo trì?
 - Sửa lỗi được phát hiện trước khi bàn giao phần mềm
 - Sửa lỗi được phát hiện sau khi bàn giao phần mềm



Khía cạnh bảo trì (2)

- Hành động nào là của pha bảo trì?
 - Sửa lỗi được phát hiện trước khi bàn giao phần mềm
 - Sửa lỗi được phát hiện sau khi bàn giao phần mềm <- định nghĩa bảo trì cổ điển



Khía cạnh bảo trì (3)

- Định nghĩa hiện đại: Một hành động thuộc pha bảo trì khi nó làm thay đổi phần mềm vì một vấn đề hoặc vì lí do cải tiến, tương thích.
- Ví dụ: Khách hàng yêu cầu sửa đổi từ pha phân tích, thiết kế, cài đặt
- Các dạng bảo trì:
 - Corrective
 - Perfective
 - Adaptive

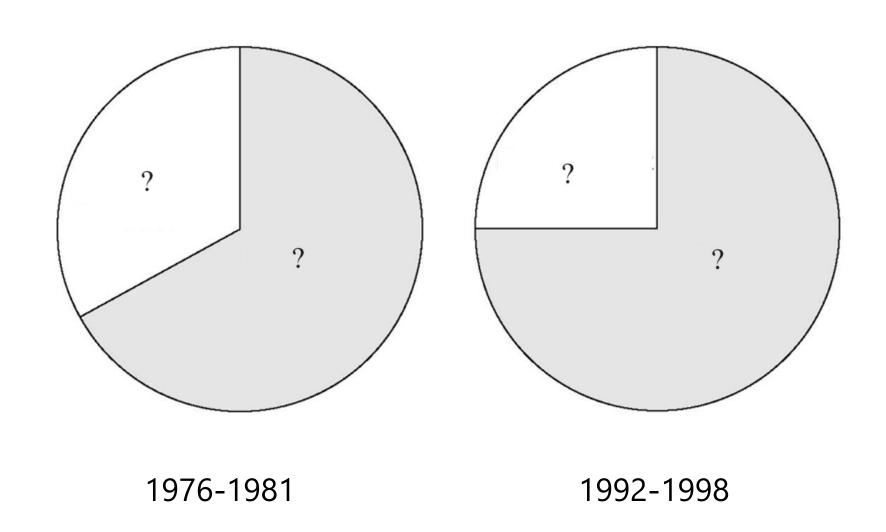


Khía cạnh bảo trì (3)

- Tầm quan trọng của bảo trì:
 - Phần mềm không tốt?
 - Phần mềm tốt được bảo trì trong thời gian dài: 10-20 năm hoặc hơn
 - Phần mềm thay đổi thường xuyên theo yêu cầu công việc



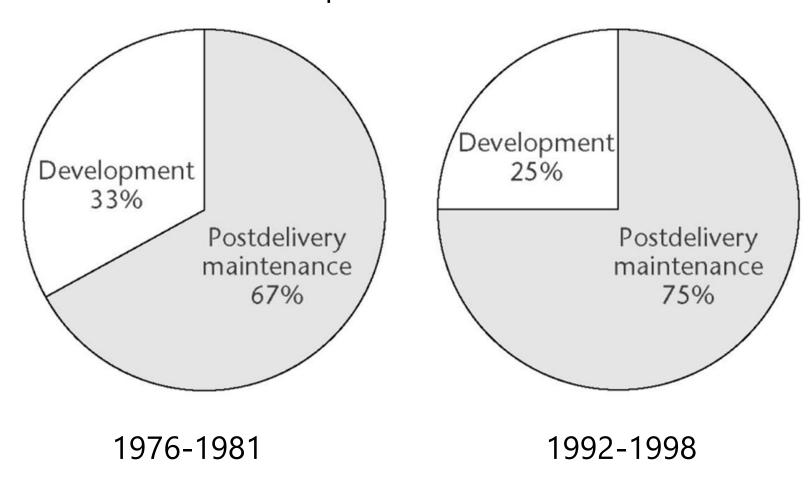
Khía cạnh bảo trì (4)





Khía cạnh bảo trì (4)

Chi phí cho bảo trì





Khía cạnh bảo trì (5)

- Tương quan chi phí giữa các pha:
 - Nếu giảm 10% chi phí pha cài đặt -> giảm khoảng 0.85% chi phí toàn
 dự án
 - Nếu giảm 10% chi phí pha bảo trì -> giảm khoảng 7.5% chi phí toàn dự án

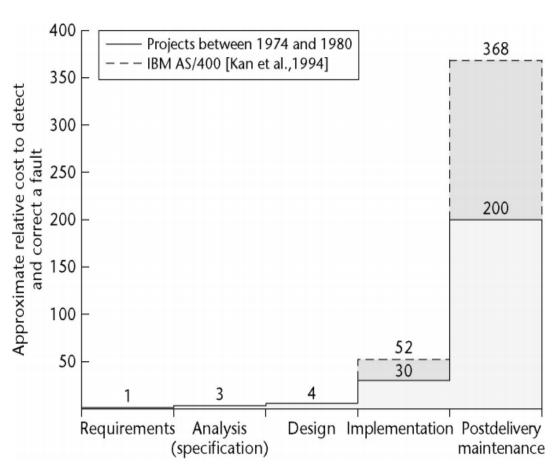


Khía cạnh phân tích - thiết kế (1)

Lỗi được phát hiện càng sớm thì chi phí sửa lỗi càng thấp

VD: Tương quan chi phí sửa chữa khi

lỗi được phát hiện ở các pha





Khía cạnh phân tích - thiết kế (2)

- Sửa lỗi phát hiện trong các pha yêu cầu, phân tích, thiết kế:
 - Sửa tài liệu của pha
- Sửa lỗi phát hiện trong pha cài đặt hoặc bảo trì:
 - Sửa lại tài liệu từ các pha trước
 - Sửa lại code
 - Thực hiện kiểm thử lại phần đã sửa và tương thích với phần còn lại
 - Cài đặt lại hệ thống



Khía cạnh phân tích - thiết kế (3)

- 60-70% lỗi của phần mềm nằm ở đặc tả và thiết kế
- VD: chương trình không gian liên hành tinh của Nasa:
 - 1,9 lỗi trên mỗi trang đặc tả
 - 0,9 lỗi trên mỗi trang thiết kế
 - 0,3 lỗi trên mỗi trang mã nguồn
- Kết luận: Phải cải thiện chất lượng của pha yêu cầu, phân tích và thiết kế
 - Phát hiện lỗi sớm
 - Giảm thiểu tổng số lỗi của dự án -> giảm tổng chi phí



Khía cạnh lập trình nhóm

- Phần mềm thường được phát triển bởi một nhóm
 - Nhiều phần mềm quá lớn để một người có thể thực hiện tất cả các công việc trong thời gian có hạn
 - Vấn đề tương thích giữa các module
 - Vấn đề giao tiếp giữa các thành viên trong nhóm



Khía cạnh lập kế hoạch

- Các hoạt động lập kế hoạch:
 - Lên kế hoạch sơ bộ cho các pha lấy yêu cầu và phân tích khi bắt đầu dự án
 - Kế hoạch quản lý dự án phần mềm được thiết lập khi các đặc tả đã được khách hàng xác nhận
 - Người quản lý sẽ giám sát kế hoạch trong suốt phần còn lại của dự án
- Tại sao không có pha lập kế hoạch?



Khía cạnh kiểm thử

- Các hình thức kiểm thử cổ điển:
 - Verification: Kiểm thử ở cuối mỗi pha
 - Validation: Kiểm thử ở cuối dự án
- Tại sao không có pha kiểm thử?



Khía cạnh làm tài liệu

Tại sao không có pha làm tài liệu?



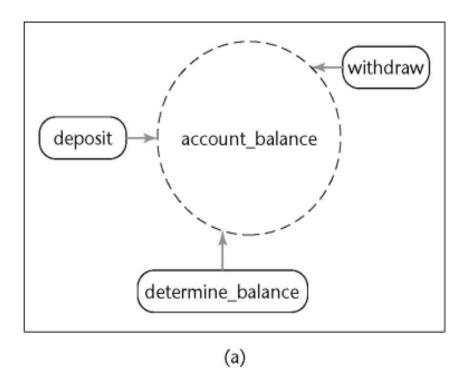
Phương pháp hướng đối tượng (1)

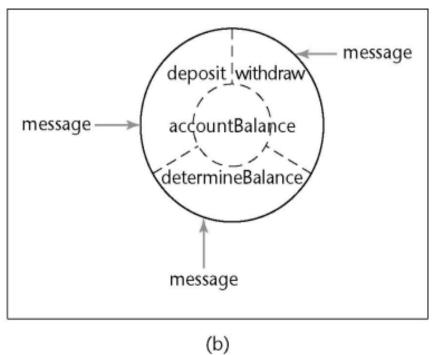
- Phương pháp hướng cấu trúc (hướng thủ tục):
 - Kỹ thuật truyền thống
 - Không phù hợp với phần mềm lớn (> 50.000 dòng code)
 - 70-80% chi phí cho bảo trì
- Phương pháp hướng đối tượng:
 - Dữ liệu + hành động
 - Việc mô hình hóa trở nên đơn giản hơn
 - Bảo mật



Phương pháp hướng đối tượng (2)

• VD:







Phương pháp hướng đối tượng (3)

Phương pháp cổ điển

- Phân tích:
 - Xác định cần làm cái gì

- Thiết kế:
 - Xác định làm như thế nào
 - Xây dựng các module

Phương pháp hướng đối tượng

- Phân tích:
 - Xác định cần làm cái gì
 - Xác định các đối tượng
- Thiết kế:
 - Xác định làm như thế nào
 - Xây dựng các đối tượng



Kết luận

- Phát triển một phần mềm không dễ:
 - Các vấn đề kinh tế
 - Các vấn đề kỹ thuật
 - Các vấn đề trong từng giai đoạn phát triển
 - Các vấn đề về con người
- Công nghệ phần mềm: ngành học nghiên cứu cách sản xuất phần mềm
- Phương pháp hướng đối tượng là cách tiếp cận phổ biến



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