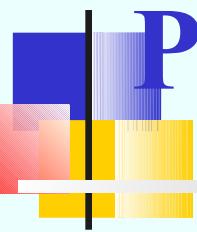


Chapter 1 The Scope of Software Engineering

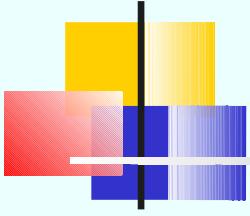
Công nghệ phần mềm



Phạm vi của công nghệ phần mềm

Giảng viên: TS. Nguyễn Mạnh Hùng

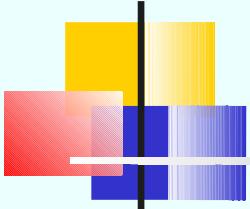
Học viện Công nghệ Bưu chính Viễn thông (PTIT)



Khía cạnh lịch sử (1)

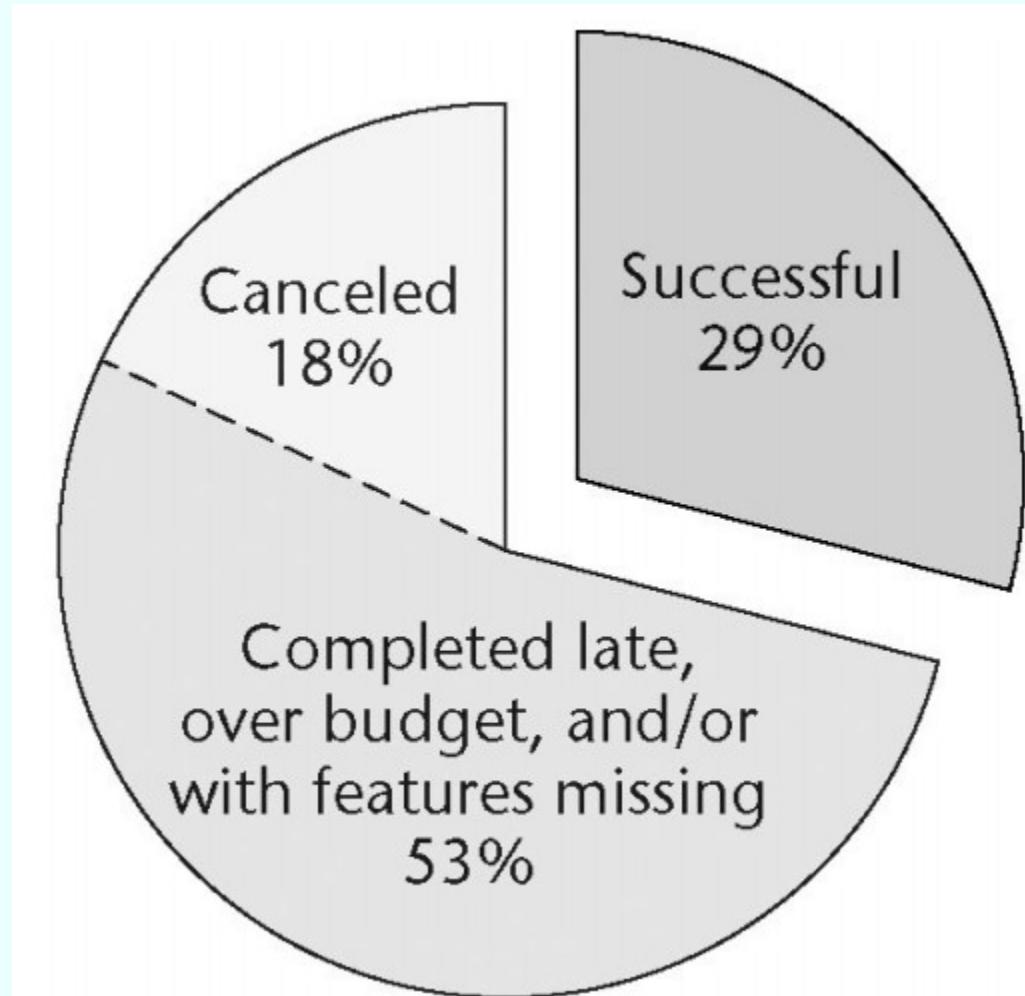
Năm 1968, NATO đã nhóm họp ở Đức để tìm giải pháp thoát khỏi khủng hoảng phần mềm:

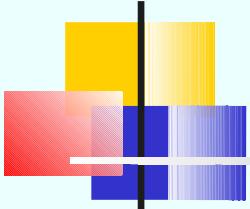
- Phần mềm hoàn thành và chuyển giao trễ thời hạn
- Vượt chi phí dự đoán
- Vẫn còn tiềm tàng lỗi



Khía cạnh lịch sử (2)

Dữ liệu thống kê từ
9236 dự án phần
mềm năm 2004:





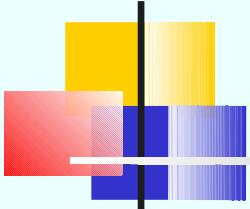
Khía cạnh lịch sử (3)

Khảo sát năm 2000 trên các cty phần mềm:

- 78% dự án có tranh chấp đều kết thúc bằng kiện tụng

Với các dự án bị kiện:

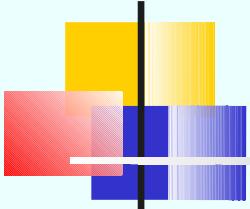
- 67% dự án bị kiện do chức năng không đúng yêu cầu khách hàng
- 56% dự án bị kiện vì dời hẹn giao sản phẩm quá nhiều lần
- 45% dự án bị kiện là do còn lỗi nghiêm trọng đến mức sản phẩm không sử dụng được



Khía cạnh lịch sử (4)

Sự khủng hoảng phần mềm không thể giải quyết dứt điểm:

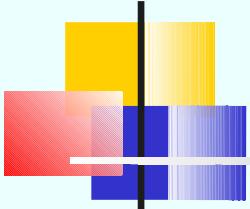
- Nó còn có thể tồn tại lâu dài
- Hiện nay vẫn chưa có dự đoán chính xác thời điểm kết thúc



Khía cạnh kinh tế

Xem xét khía cạnh kinh tế của các tình huống:

- Có ngôn ngữ lập trình mới, có nên dùng ngôn ngữ mới này cho dự án?
- Có công cụ phân tích thiết kế mới, dễ dùng hơn, có nên sử dụng cho dự án?
- Có công nghệ code/test mới, có nên ứng dụng vào dự án?

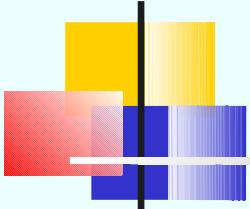


Khía cạnh bảo trì (1)

Mô hình vòng đời phát triển phần mềm:

- Ví dụ: mô hình thác nước (waterfall)

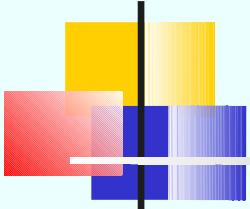
1. Requirements phase
2. Analysis (specification) phase
3. Design phase
4. Implementation phase
5. Postdelivery maintenance
6. Retirement



Khía cạnh bảo trì (2)

Các dạng bảo trì:

- Bảo trì sửa chữa (corrective)
- Bảo trì phát triển (perfective)
- Bảo trì tương thích (adaptive)



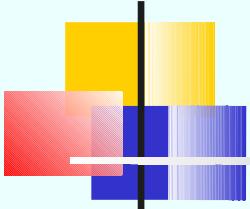
Khía cạnh bảo trì (3)

Khi nào thì coi một hành động là của pha bảo trì?

- **Định nghĩa cổ điển** (dựa trên thời gian): một hành động là của pha bảo trì khi nó thực hiện sau khi bàn giao và cài đặt sản phẩm

Ví dụ:

- Nếu một lỗi được phát hiện sau khi bàn giao phần mềm thì việc sửa lỗi là của pha bảo trì
- Nếu cùng lỗi đó nhưng được phát hiện trước khi bàn giao phần mềm thì việc sửa lỗi thuộc pha cài đặt



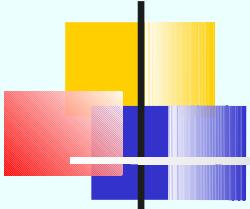
Khía cạnh bảo trì (4)

Khi nào thì coi một hành động là của pha bảo trì?

- Định nghĩa hiện đại: một hành động là của pha bảo trì khi nó làm thay đổi phần mềm vì lí do hoàn thiện hay tương thích

Ví dụ:

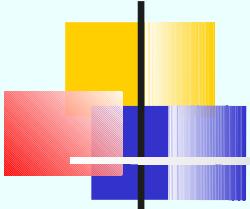
- Nếu khách hàng bổ sung thêm yêu cầu từ pha phân tích, thiết kế hay cài đặt thì việc thay đổi đó sẽ được coi là bảo trì sản phẩm



Khía cạnh bảo trì (5)

Tầm quan trọng của pha bảo trì:

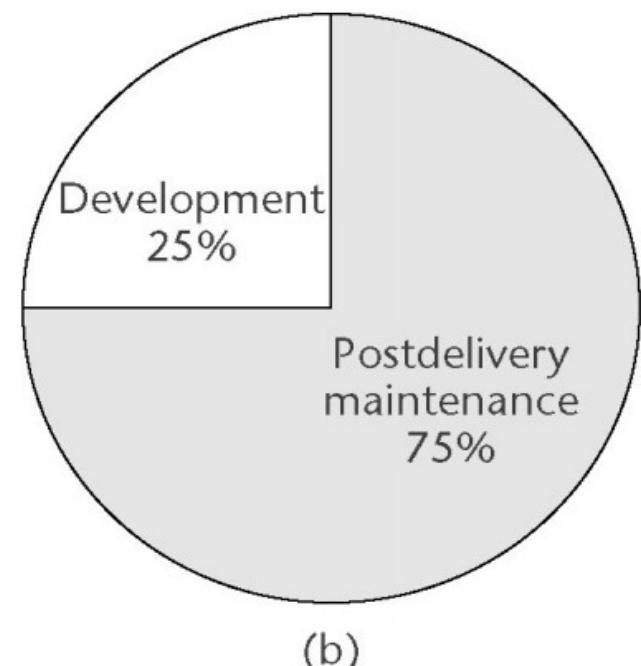
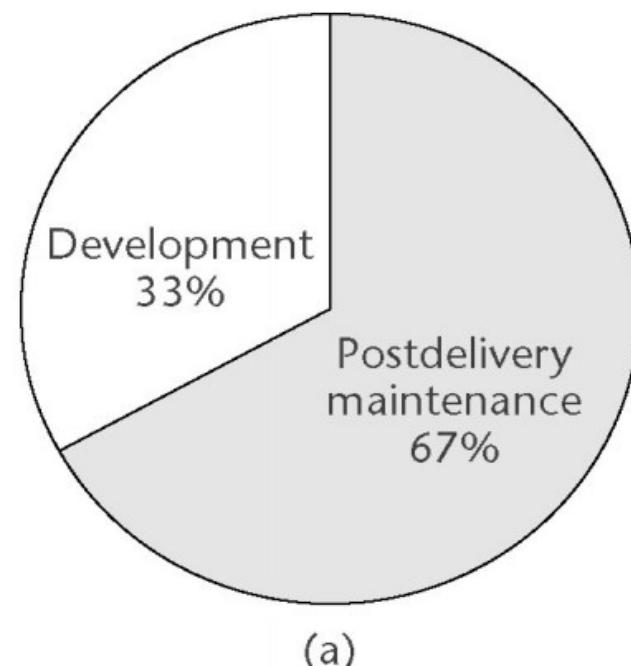
- Phần mềm không tốt thì sẽ bị vứt bỏ, chứ không được bảo trì
- Chỉ những phần mềm tốt mới được bảo trì, thời gian bảo trì có thể 10- 20 năm, có thể cả đời
- Bản thân phần mềm là một công cụ hỗ trợ, hoặc phương tiện làm việc, do đó nó sẽ thay đổi thường xuyên theo yêu cầu công việc

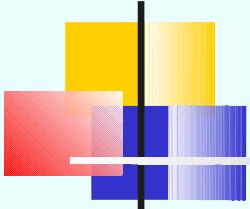


Khía cạnh bảo trì (6)

Thời gian (chi phí) cho bảo trì luôn chiếm tỉ trọng lớn nhất:

- (a): 1976 – 1981
- (b): 1992 - 1998

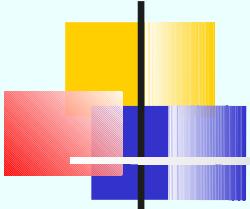




Khía cạnh bảo trì (7)

Thời gian (chi phí) cho các pha gần như không thay đổi nhiều:

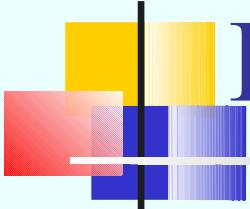
	Various Projects between 1976 and 1981	132 More Recent Hewlett-Packard Projects
Requirements and analysis (specification) phases	21%	18%
Design phase	18	19
Implementation phase		
Coding (including unit testing)	36	34
Integration	24	29



Khía cạnh bảo trì (8)

Chi phí tương quan giữa các pha:

- Nếu giảm 10% chi phí cho pha cài đặt → sẽ giảm được khoảng 0.85% chi phí cho dự án
- Nếu giảm 10% chi phí cho bảo trì → sẽ giảm được 7.5% chi phí toàn bộ dự án!



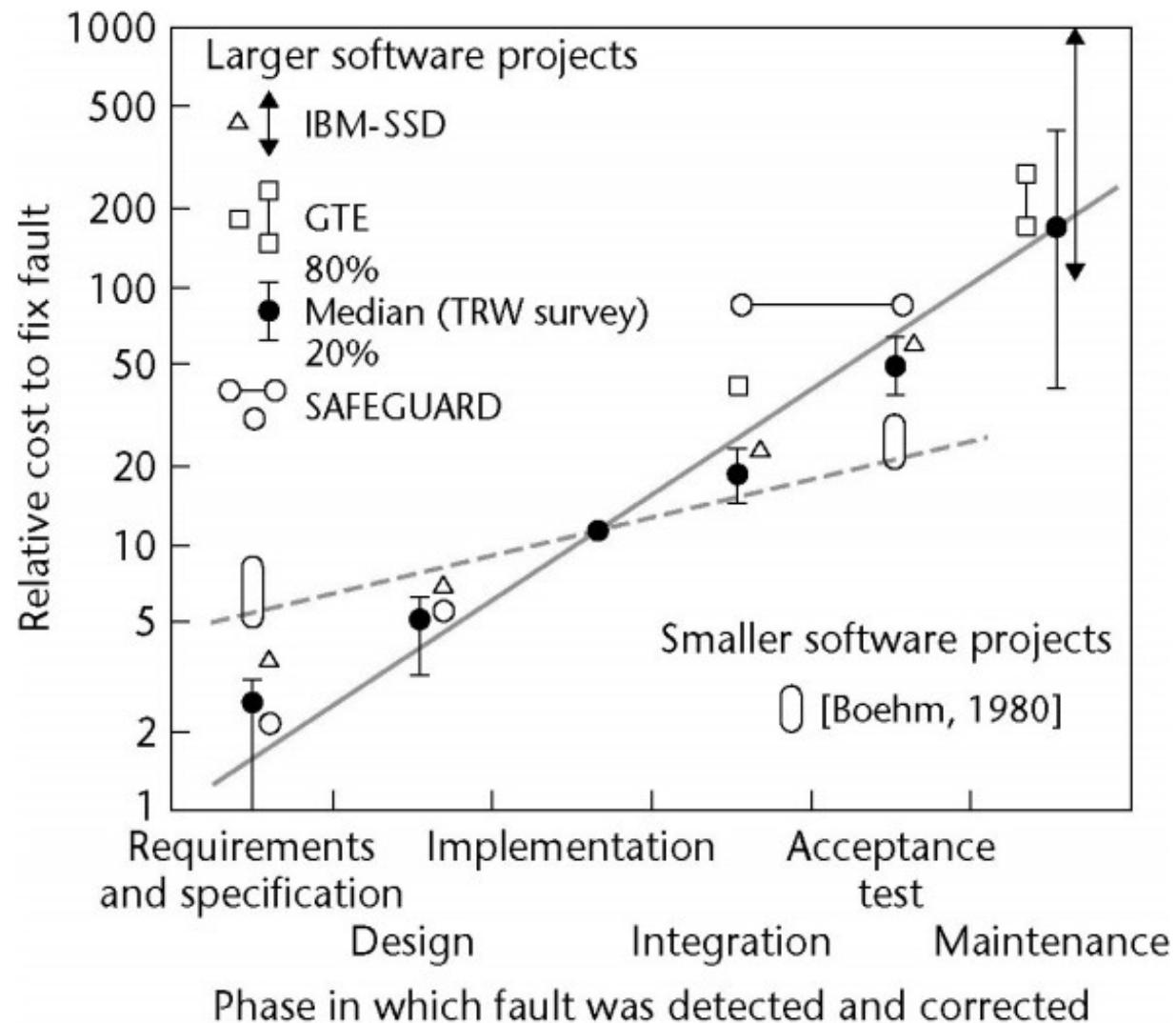
Khía cạnh phân tích- thiết kế (1)

Nguyên tắc cơ bản:

- Lỗi được phát hiện càng sớm thì chi phí để sửa lỗi càng thấp!
- Và ngược lại!

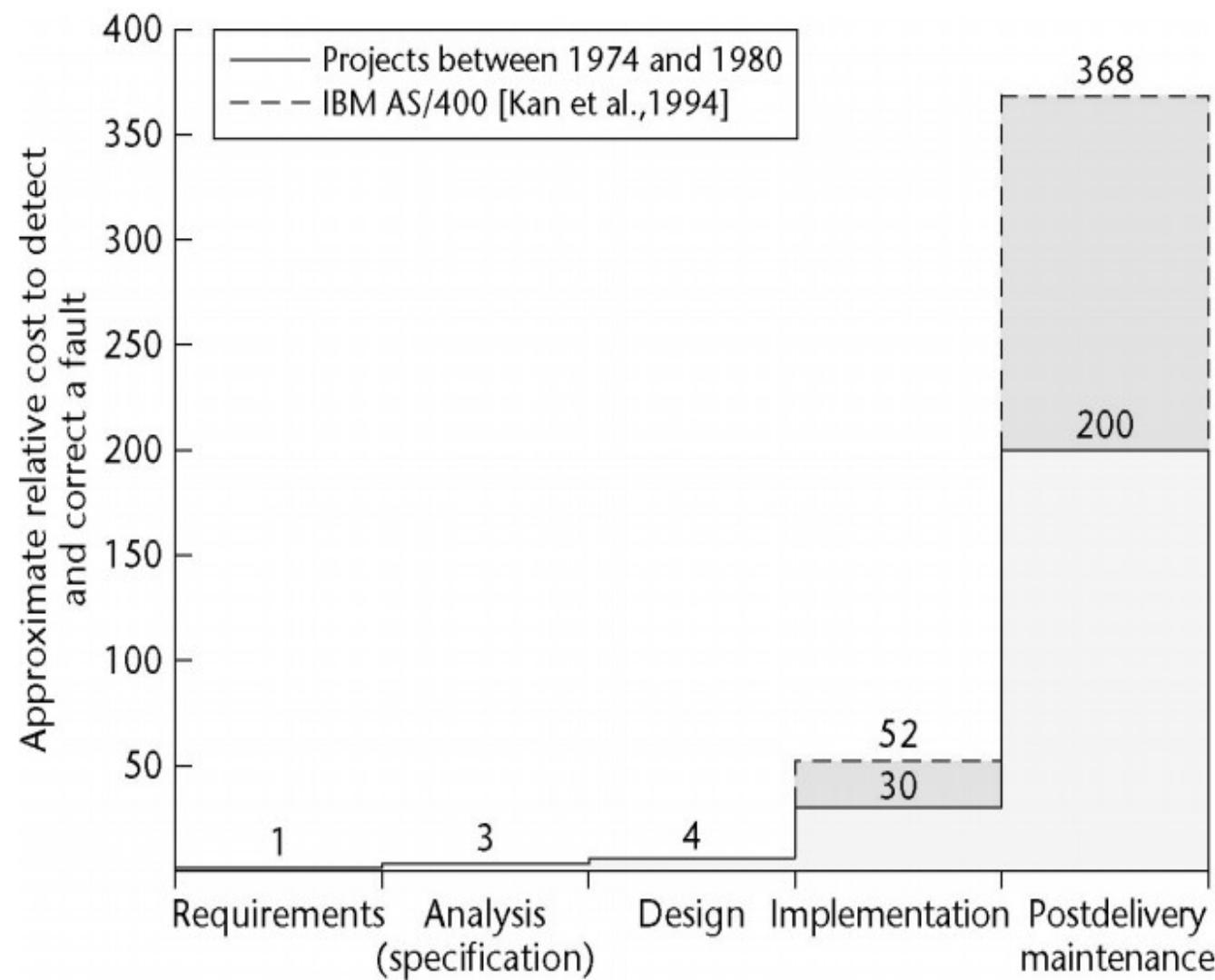
Khía cạnh phân tích- thiết kế (2)

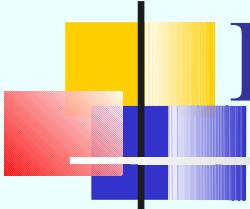
Ví dụ:



Khía cạnh phân tích- thiết kế (3)

Ví dụ (tt):





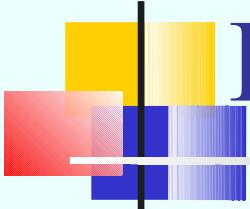
Khía cạnh phân tích- thiết kế (4)

Để sửa một lỗi phát hiện sớm trong các pha yêu cầu, phân tích, và thiết kế:

- Chỉ cần thay đổi tài liệu các pha tương ứng

Để sửa một lỗi phát hiện muộn trong cài đặt hoặc bảo trì:

- Lần ngược lại các pha trước để sửa lại tài liệu
- Sửa lại code vì phân tích thiết kế đã bị sửa
- Test lại phần sửa/ test phần tương thích với phần còn lại
- Cài đặt lại hệ thống cho khách hàng



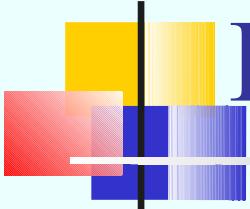
Khía cạnh phân tích- thiết kế (5)

Thông kê cho thấy:

- 60-70% lỗi phát hiện ra là nằm trong các pha yêu cầu, phân tích, và thiết kế
- Nhưng thời điểm phát hiện ra các lỗi đấy là trong pha cài đặt và bảo trì

Ví dụ của công ty Jet Propulsion Laboratory:

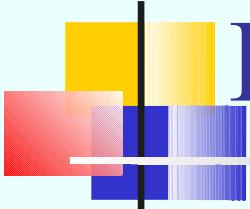
- 1.9 lỗi/ trang đặc tả (specification)
- 0.9 lỗi/ trang thiết kế
- 0.3 lỗi/ trang code



Khía cạnh phân tích- thiết kế (6)

Kết luận:

- Phải cải thiện chất lượng của pha lấy yêu cầu, phân tích và thiết kế
- Phát hiện lỗi càng sớm càng tốt
- Giảm thiểu tổng số lỗi phát hiện của toàn dự án



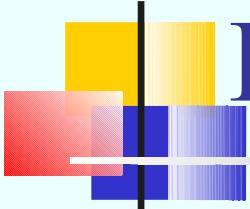
Khía cạnh nhóm phát triển

Có thể:

- Phát triển một phần mềm lớn, với chỉ một người làm tất cả các khâu!

Nhưng, phần mềm thường phát triển bởi một nhóm phát triển:

- Vấn đề tương tác, tích hợp giữa các modul
- Vấn đề giao tiếp và cộng tác giữa các thành viên của nhóm



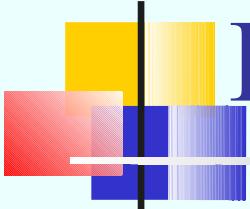
Khía cạnh lập kế hoạch

Câu hỏi:

- Tại sao không có pha lập kế hoạch?

Trả lời:

- ...



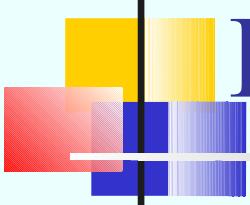
Khía cạnh kiểm thử

Câu hỏi:

- Tại sao không có pha kiểm thử (test)?

Trả lời:

- ...



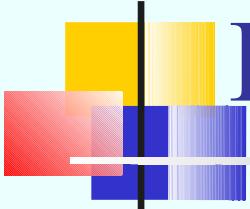
Khía cạnh làm tài liệu

Câu hỏi:

- Tại sao không có pha làm tài liệu?

Trả lời:

- ...



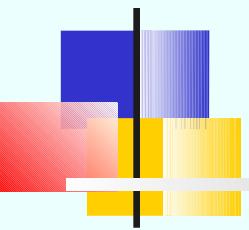
Kết luận

Để phát triển được một phần mềm không dễ dàng:

- Các vấn đề về kinh tế
- Các vấn đề về kỹ thuật
- Các vấn đề cho mỗi giai đoạn phát triển
- Các vấn đề về con người

Với mục đích:

- Phát hiện lỗi càng sớm càng tốt
- Giảm thiểu được số lượng lỗi
- Giảm thiểu thời gian (và chi phí) phát triển



Questions?
