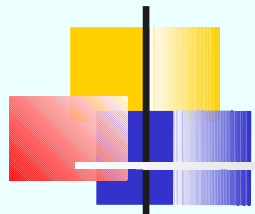


# Công nghệ phần mềm

# Tiến trình phần mềm

---

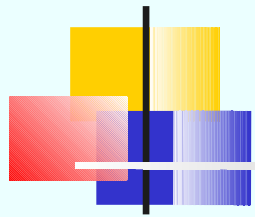
*Giảng viên: TS. Nguyễn Mạnh Hùng*  
*Học viện Công nghệ Bưu chính Viễn thông (PTIT)*



# Nội dung tham khảo từ

---

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

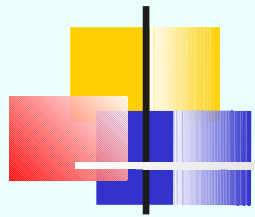


# Requirement workflow (1)

---

Mục đích:

- Xác định rõ cái mà khách hàng cần
- Không phải cái khách hàng muốn

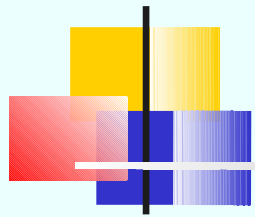


# Requirement workflow (2)

---

## Phương pháp:

- Xác định rõ hiểu lĩnh vực ứng dụng của phần mềm:
  - Làm rõ các khái niệm chuyên ngành trong lĩnh vực tương ứng
- Xây dựng mô hình nghiệp vụ của khách hàng:
  - Làm việc với chuyên gia nghiệp vụ
  - Sử dụng công cụ UML
  - Đánh giá tính khả thi: kĩ thuật, chi phí...

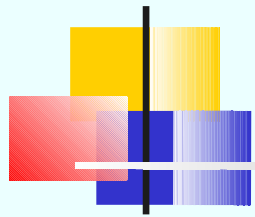


# Requirement workflow (3)

---

Kết quả cần đạt được:

- Thời hạn giao sản phẩm (deadline)
- Độ tin cậy (reliability)
- Chi phí (cost)
- Ngoài ra còn phải thống nhất một số yêu cầu khác: portability, respond time, parallel running



# Analysis workflow (1)

---

Mục tiêu:

- Phân tích, phân rã và mịn hóa yêu cầu của khách hàng

→ Tại sao không làm việc này ngay trong pha requirement?

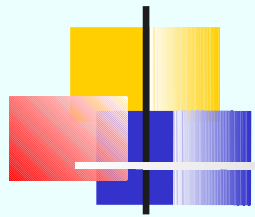


# Analysis workflow (2)

---

Vấn đề ngôn ngữ:

- Tài liệu phải thống nhất được cả hai bên khách hàng và đội phát triển
  - Khách hàng chỉ hiểu ngôn ngữ tự nhiên: ngôn ngữ không chính xác
  - Đội phát triển chỉ làm việc được trên ngôn ngữ kĩ thuật: chính xác và khoa học
- Tạo ra hai loại tài liệu đặc tả: bằng ngôn ngữ tự nhiên và bằng ngôn ngữ kĩ thuật



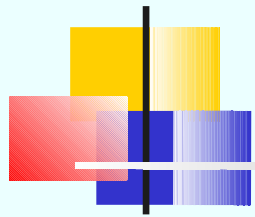
# Analysis workflow (3)

---

→ Tại sao việc làm tài liệu đặc tả lại quan trọng đến vậy?

- Trả lời: ...





# Analysis workflow (4)

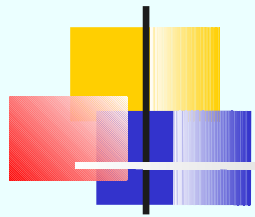
---

Kết quả cần đạt được:

- Tài liệu đặc tả đúng yêu cầu của khách hàng

Yêu cầu về tài liệu không được:

- Mâu thuẫn
- Có khái niệm và định lượng mờ
- Không đầy đủ



# Analysis workflow (5)

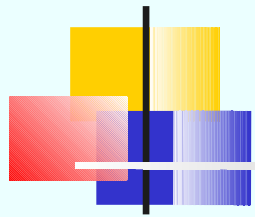
---

Kết quả cần đạt được:

- Bản kế hoạch (tạm thời) về quản lí dự án phần mềm

Yêu cầu về bản kế hoạch:

- Ước lượng chi phí
- Ước lượng thời gian
- Các điểm mốc quan trọng (milestone)
- Các sản phẩm phải có sau mỗi điểm mốc

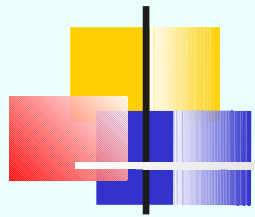


# Design workflow (1)

---

Mục đích:

- Mịn hóa và mô hình hóa kết quả pha phân tích cho đến khi có thể code được từng modul trên một ngôn ngữ lập trình tương ứng

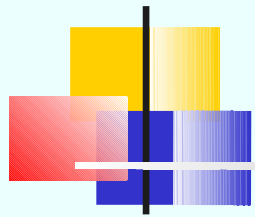


# Design workflow (2)

---

Các vấn đề xem xét:

- Chọn ngôn ngữ lập trình
- Tính sử dụng lại (reuseability)
- Tính thiết kế mở (open-design)
- Tính khả chuyển (portability)

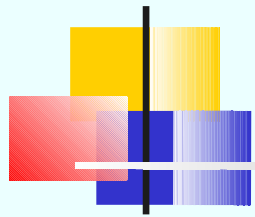


# Design workflow (3)

---

Phương pháp:

- Trích các lớp
- Xác định quan hệ giữa các lớp (thiết kế kiến trúc)
- Thiết kế các thuộc tính và phương thức (method) cho mỗi lớp (thiết kế chi tiết)

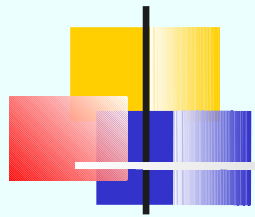


# Design workflow (4)

---

Kết quả cần đạt được:

- Bản mẫu các lớp, thuộc tính và phương thức + thuật toán xử lý trong các phương thức để có thể cài đặt được ngay

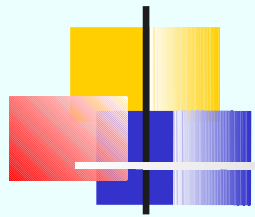


# Implementation workflow (1)

---

Mục tiêu:

- Cài đặt hệ thống theo kết quả pha thiết kế



# Implementation workflow (2)

---

Phương pháp:

- Cài đặt theo class, modul
- Tích hợp các class, modul





# Test workflow (1)

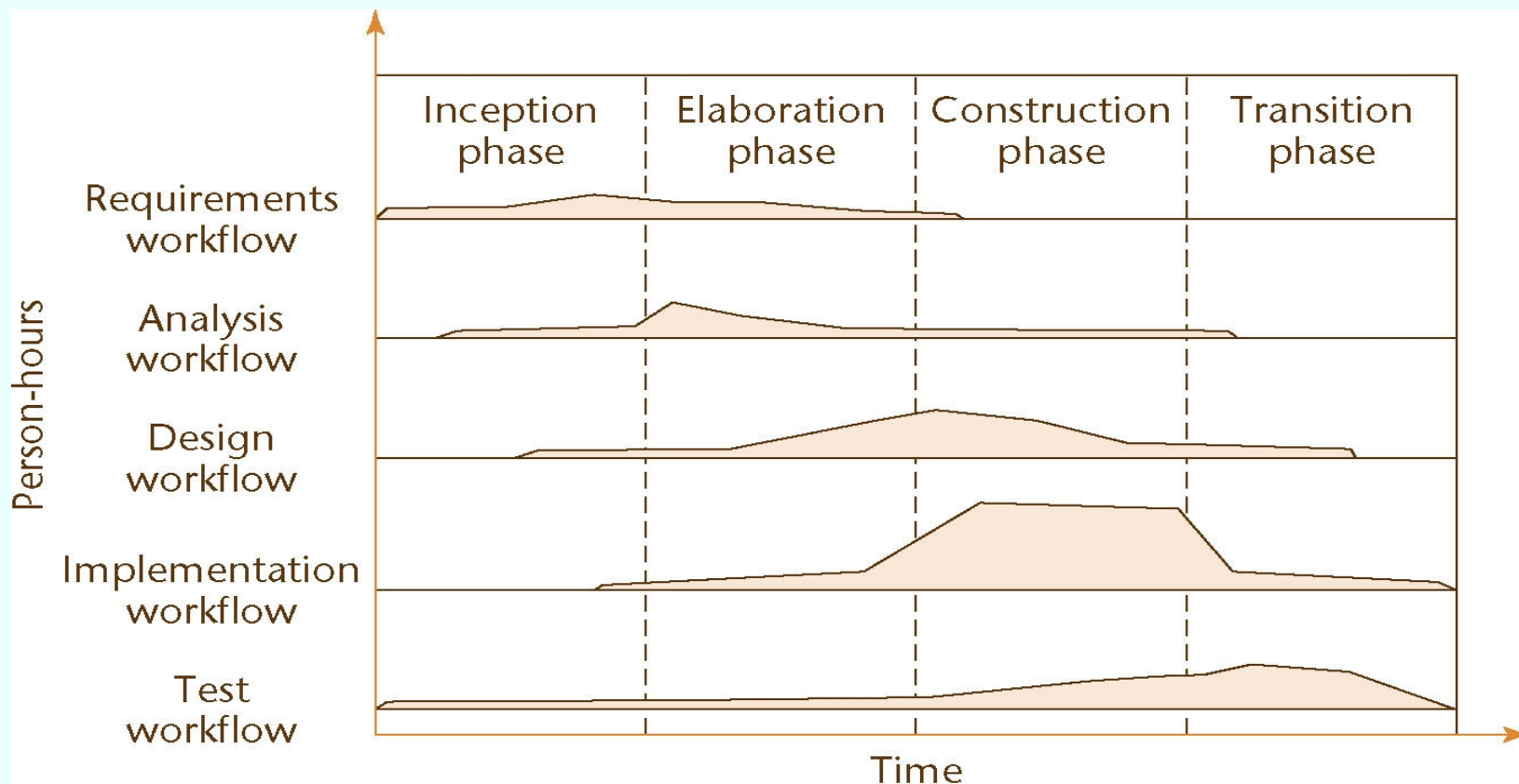
---

Nội dung test các sản phẩm đầu ra của từng pha:

- Yêu cầu: test tài liệu
- Phân tích: test tài liệu, kế hoạch và ước lượng
- Thiết kế: test tài liệu
- Cài đặt: unit test, integrated test, product test, acceptance test. Đối với phần mềm COTS thì test phản alpha và beta.

# Unified Process (1)

Mỗi pha tương ứng một bước trong chu kì tăng trưởng (increasement):





# Unified Process (2)

---

Các pha phát triển:

- Inception: Đánh giá
- Elaboration: Thiết lập
- Construction: Xây dựng
- Transition: Chuyển tiếp



# Unified Process (3)

---

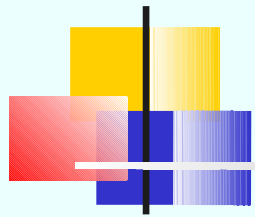
Mỗi bước thực hiện tương ứng với:

- 1 trong 5 workflows
- 1 trong 4 pha

Quan hệ:

- Workflow tương ứng với cách nhìn kĩ thuật
- Pha tương ứng cách nhìn nghiệp vụ

→ Tại sao mỗi bước phải có hai cách nhìn khác nhau?

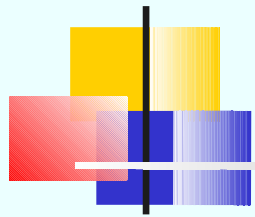


# Inception phase (1)

---

Mục tiêu:

- Xác định xem phần mềm làm ra có kinh tế và khả thi hay không

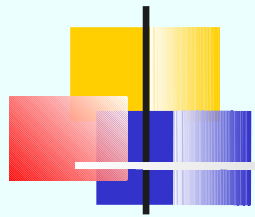


# Inception phase (2)

---

Thực hiện:

- Tìm hiểu lĩnh vực chuyên môn
- Xây dựng mô hình nghiệp vụ
- Nêu rõ giới hạn của sản phẩm
- Bắt đầu xây dựng phân tích kinh doanh

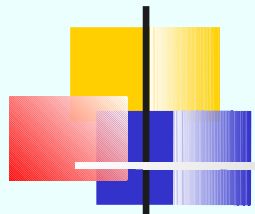


# Inception phase (3)

---

Phân tích kinh doanh:

- Giá phát triển có mang tính kinh tế?
- Bao lâu sẽ quay vòng vốn?
- Nếu từ bỏ dự án thì chi phí hết bao nhiêu?
- Nếu sản phẩm dạng COTS, có cần có chiến dịch tiếp thị sản phẩm?
- Sản phẩm có thể giao đúng hạn không?
- Thiệt hại gì nếu giao sản phẩm cho khách hàng trễ hạn?



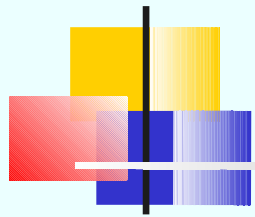
# Inception phase (4)

---

Phân tích rủi ro khi phát triển phần mềm:

- Liệu team có đủ kinh nghiệm cần thiết?
- Có cần phần cứng mới cho sản phẩm?
- Nếu có, thì thiệt hại gì nếu người ta giao phần cứng trễ hẹn?
- Trong trường hợp đó, có nên đặt hàng một nhà cung cấp phần cứng khác để dự phòng không?
- Có cần công cụ hỗ trợ nào không?
- Nếu có, liệu chúng có sẵn hay không, hay có cần toàn bộ chức năng của nó hay không?





# Elaboration phase (1)

---

Mục tiêu:

- Mịn hóa các kết quả sau pha inception và requirement
- Phân tích rủi ro theo mức độ nghiêm trọng
- Mịn hóa bản phân tích kinh doanh có trong pha inception
- Xem xét lại SPMP



# Elaboration phase (2)

---

Phương pháp:

- Sử dụng các kĩ thuật và phương pháp trong pha inception và requirement



# Construction phase (1)

---

Mục tiêu:

- Xây dựng phiên bản đầu tiên hoạt động được của sản phẩm



# Construction phase (2)

---

Phương pháp:

- Cài đặt
- Kiểm thử
- Làm tài liệu



# Transition phase (1)

---

Mục tiêu:

- Đảm bảo tất cả các yêu cầu của khách hàng đã được thực hiện một cách đúng đắn
- Các lỗi đã được sửa
- Các tài liệu hướng dẫn sử dụng đã hoàn chỉnh



# Capability maturity model - CMM

---

- Không phải một mô hình vòng đời phát triển phần mềm
- Một bộ các tiêu chuẩn đánh giá chiến lược hoàn thiện tiến trình phần mềm: SW-CMM
- Các tài liệu hướng dẫn sử dụng đã hoàn chỉnh



# SW - CMM

---

- Ra đời năm 1986 bởi SEI
- Hoàn thiện tiến trình phần mềm
- Hoàn thiện quản lí tiến trình, hoàn thiện kĩ thuật
- Có 5 levels



# SW – CMM: level 1

---

Mức khởi đầu (initial):

- Các tiến trình phần mềm là không dự đoán được
- Việc quản lí chỉ bao gồm việc xử lí các rủi ro gặp phải

→ Tất cả các cty đều đạt chuẩn level 1





## SW – CMM: level 2

---

Mức có khả năng lặp lại (repeatable):

- Các quyết định quản lí dựa vào các dự án tương tự trước đó
- Có phương pháp đo các tiêu chí
- Kết quả dự án này có thể được dùng để ước lượng chi phí và thời gian cho các dự án tiếp theo
- Khi có lỗi xảy ra, việc khắc phục lỗi được thực hiện ngay



# SW – CMM: level 3

---

Mức có được định nghĩa (defined):

- Có tài liệu kĩ thuật và quản lí
- Liên tục có cố gắng để nâng cao chất lượng sản phẩm
- Việc xem xét lại luôn được thực hiện để đảm bảo chất lượng sản phẩm



# SW – CMM: level 4

---

Mức có được quản lí (managed):

- Chất lượng và quy trình sản xuất luôn được giám sát
- Việc điều chỉnh chất lượng sản phẩm theo kết quả thống kê cũng được thực hiện



# SW – CMM: level 5

---

Mức có tối ưu hóa (optimize):

- Không ngừng cải thiện: chất lượng sản phẩm theo thống kê và điều khiển quy trình phát triển
- Ghi nhận phản hồi và kinh nghiệm có được sau mỗi sản phẩm để cải tiến các sản phẩm tiếp theo



# Questions?

---