

## Phát triển phần mềm ITSS

### Bài 1: Quy trình phát triển phần mềm

1

## Giáo trình

[1] ISO/IEC FDIS 12207, *Systems and software engineering — Software life cycle processes*.

2

2

## Nội dung

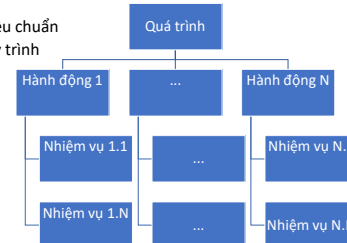
1. Quy trình vòng đời phần mềm
2. Quy trình triển khai phần mềm
3. Phân tích thiết kế hướng đối tượng
4. Mô hình phát triển phần mềm

3

3

## 1. Quy trình vòng đời phần mềm

- ❖ Theo “ISO / IEC 12207: 2008, Hệ thống và kỹ thuật phần mềm – Quy trình vòng đời phần mềm”
  - Quy trình phát triển phần mềm tiêu chuẩn quốc tế và mới nhất
- ❖ “Vòng đời bắt đầu với một ý tưởng hoặc một nhu cầu có thể được phần mềm đáp ứng toàn bộ hoặc một phần và kết thúc bằng việc phần mềm ngừng hoạt động”.
- ❖ Các công việc thực hiện tiêu chuẩn
  - Phân cấp như các quy trình



4

4

## Tiêu chuẩn quốc tế là gì?

- ❖ Trong ISO, tất cả các tiêu chuẩn ngành, bao gồm cả Công nghệ thông tin, đều được phát triển.
- ❖ Trong lĩnh vực Công nghệ thông tin, trong ISO / IEC JTC1, các tiêu chuẩn quốc tế được xây dựng.
- ❖ ISO / IEC JTC có 32 cơ quan thành viên chính phát triển các tiêu chuẩn quốc tế và 44 cơ quan thành viên quan sát viên.
- ❖ Một số viết tắt
  - ISO: Tổ chức tiêu chuẩn quốc tế
  - IEC: Ủy ban kỹ thuật điện quốc tế
  - JTC1: Ủy ban kỹ thuật chung



5

## Quy trình vòng đời

- ❖ “Tiêu chuẩn này nhóm các hoạt động có thể được thực hiện trong vòng đời của hệ thống phần mềm thành 7 nhóm quá trình” [1]\*:
  1. Các quy trình thỏa thuận: 2 quy trình
  2. Các quy trình hỗ trợ dự án của tổ chức: 5 quy trình
  3. Quy trình dự án: 7 quy trình
  4. Quy trình kỹ thuật: 11 quy trình
  5. Quy trình triển khai: 6 quy trình
    - Mục đích: “để tạo ra một phần tử hệ thống cụ thể được triển khai như một sản phẩm hoặc dịch vụ phần mềm” [1]\*\*.
  6. Quy trình hỗ trợ phần mềm: 8 quy trình
  7. Quy trình tái sử dụng phần mềm: 3 quy trình

[1]\*: clause 5.2.1; pp. 13, [1]\*\*: clause 7.1.1.1; pp. 57,



6

## Nội dung

1. Quy trình vòng đời phần mềm
2. **Quy trình triển khai phần mềm**
3. Phân tích thiết kế hướng đối tượng
4. Mô hình phát triển phần mềm



7

## 2. Quy trình triển khai phần mềm

Quy trình Phân tích Yêu cầu Hệ thống và Quy trình Thiết kế Kiến trúc Hệ thống đạt được ngay trước Quy trình Triển khai Phần mềm.

Quy trình triển khai phần mềm bao gồm các quy trình cấp thấp hơn sau:

1. Quy trình phân tích yêu cầu phần mềm
2. Quy trình thiết kế kiến trúc phần mềm
3. Quy trình thiết kế chi tiết phần mềm
4. Quy trình xây dựng phần mềm
5. Quy trình tích hợp phần mềm
6. Quy trình kiểm tra chứng chỉ phần mềm



8

## 2.1 Quy trình phân tích yêu cầu phần mềm

- ❖ Mục đích: “Thiết lập các yêu cầu của các phần tử phần mềm của hệ thống” [1]
- ❖ Các mục chính được viết trên mô tả yêu cầu ngắn gọn.
  - Điều kiện môi trường hệ thống mà phần mềm sẽ thực hiện.
  - Các yêu cầu chức năng và các yêu cầu giao diện.
  - Định nghĩa dữ liệu và yêu cầu cơ sở dữ liệu.
  - Một số mục yêu cầu phi chức năng như độ tin cậy, khả năng sử dụng, hiệu quả thời gian
  - Yêu cầu về tiêu chuẩn: Các yêu cầu được sử dụng làm tiêu chí hoặc điều kiện để đủ điều kiện cho một sản phẩm phần mềm tuân thủ các thông số kỹ thuật của nó.

[1]: Session 7.1.2.1; pp. 59

## 2.2 Quy trình thiết kế kiến trúc phần mềm

- ❖ Mục đích: “Cung cấp thiết kế cho phần mềm triển khai và có thể xác minh được theo các yêu cầu” [1]
- ❖ Kiến trúc phần mềm được thiết kế từ các yêu cầu phần mềm
- ❖ Các mục chính:
  - Cấu trúc cấp cao nhất của phần mềm và các thành phần phần mềm tạo nên phần mềm
  - Thiết kế cấp cao nhất cho các giao diện ngoài phần mềm và giữa các thành phần phần mềm
  - Một thiết kế cấp cao nhất cho cơ sở dữ liệu.

[1]: Session 7.1.3.1; pp. 61

## 2.3 Quy trình thiết kế chi tiết phần mềm

- ❖ Mục đích: “Cung cấp một thiết kế cho phần mềm triển khai và có thể được xác minh theo các yêu cầu và kiến trúc phần mềm và đủ chi tiết để cho phép mã hóa và thử nghiệm” [1]
- ❖ Một thiết kế chi tiết cho từng thành phần phần mềm được phát triển. Trong thiết kế chi tiết, các hạng mục sau được phát triển:
  - Mỗi thành phần được tinh chỉnh thành các đơn vị phần mềm có thể được mã hóa, biên dịch và kiểm tra
  - Các giao diện bên ngoài mục phần mềm, giữa các thành phần phần mềm và giữa các đơn vị phần mềm

[1]: Session 7.1.4.1; pp. 62

## 2.4 Quy trình xây dựng phần mềm

- ❖ Mục đích: “Tạo ra các đơn vị phần mềm thực thi phản ánh đúng thiết kế phần mềm” [1]
- ❖ Các hạng mục chính sẽ được phát triển:
  - Mỗi đơn vị phần mềm và cơ sở dữ liệu.
  - Quy trình kiểm tra và kiểm tra dữ liệu cho đơn vị phần mềm và cơ sở dữ liệu.
  - Kiểm tra đơn vị và kiểm tra cơ sở dữ liệu.
- ❖ Người triển khai phải đánh giá mã phần mềm và kết quả thử nghiệm xem xét tính nhất quán bên trong bên ngoài, phạm vi kiểm tra của các đơn vị và khả năng truy nguyên các yêu cầu và thiết kế của phần mềm.

[1]: Session 7.1.5.1; pp. 63

## 2.5 Quy trình tích hợp phần mềm

- ❖ Mục đích: “kết hợp các đơn vị phần mềm và các thành phần phần mềm, tạo ra các hạng mục phần mềm tích hợp, phù hợp với thiết kế phần mềm, chứng minh rằng các yêu cầu phần mềm chức năng và phi chức năng được thỏa mãn trên một nền tảng hoạt động tương đương hoặc hoàn chỉnh” [1]
- ❖ Nhiệm vụ chính:
  - Một kế hoạch tích hợp, bao gồm các yêu cầu thử nghiệm, quy trình thử nghiệm và các trường hợp / dữ liệu thử nghiệm.
  - Tích hợp các đơn vị / thành phần phần mềm
  - Chương trình / Phần mềm / kiểm tra tích hợp.

## 2.6 Quy trình kiểm tra chứng chỉ phần mềm

- ❖ Mục đích: “Để xác nhận rằng sản phẩm phần mềm tích hợp đáp ứng các yêu cầu xác định của nó”[1].
- ❖ Kiểm tra chất lượng phù hợp với các yêu cầu chất lượng cho mục phần mềm được tiến hành
  - Kiểm tra trường hợp kiểm thử và quy trình kiểm tra.
- ❖ Người triển khai hỗ trợ (các) kiểm tra để phù hợp phần mềm đáp ứng các yêu cầu về trình độ.
  - Nếu hoàn thành thành công các cuộc đánh giá, người triển khai chuẩn bị sản phẩm phần mềm có thể phân phối cho quá trình Xây dựng Hệ thống.

[1]: Session 7.1.7.1; pp. 66

## Tổng kết

- ❖ “Quy trình Vòng đời Phần mềm – SLCP” là các quy trình tiêu chuẩn quốc tế tập trung vào việc phát triển và hỗ trợ Phần mềm Ứng dụng.
- ❖ SLCP có thể được sử dụng như một ngôn ngữ chung giữa các bên liên quan như bên mua và nhà cung cấp. Họ có thể giao tiếp hoặc ra lệnh phát triển phần mềm bằng SLCP. Ví dụ, chúng ta có thể nói “Để đặt hàng quy trình thiết kế chi tiết phần mềm hoặc các quy trình triển khai phần mềm sau này của hệ thống thư viện mới”.

## Đánh giá

- ❖ “Thành công trong kinh doanh của khách hàng phụ thuộc vào thành công của việc phát triển hệ thống.”
- ❖ Phần mềm BA là một trong những thành phần chính của Hệ thống BA.
- ❖ Các yếu tố chính để phát triển hệ thống thành công là gì?
  1. Phần mềm được phát triển đáp ứng các yêu cầu chức năng ?
  2. Để giữ ngày giao hàng đã hẹn ?
  3. Đáp ứng các yêu cầu chất lượng như Độ tin cậy, Khả năng sử dụng, Hiệu suất, Khả năng bảo trì ?
  4. Cần thiết để cung cấp hoạt động bảo trì trong suốt thời gian vận hành hệ thống ?

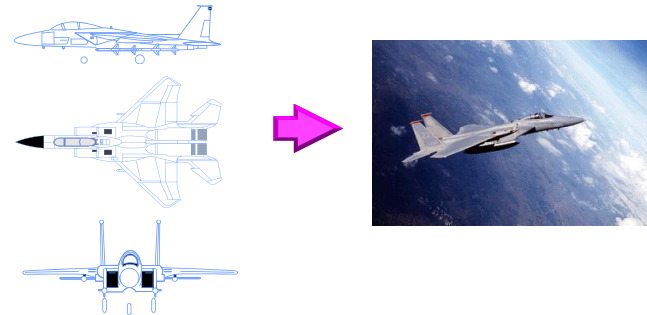
## Nội dung

1. Quy trình vòng đời phần mềm
2. Quy trình triển khai phần mềm
3. Phân tích thiết kế hướng đối tượng
4. Mô hình phát triển phần mềm



## 3.1 Mô hình hóa

❖ Mô hình là sự đơn giản hóa hệ thống.



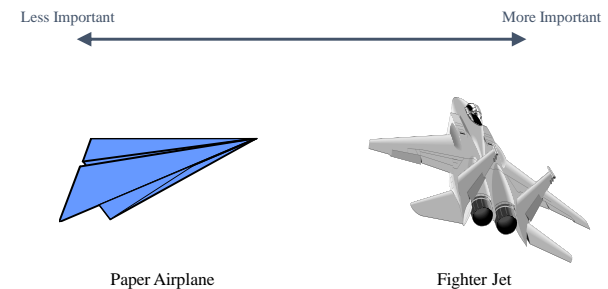
## Tại sao Mô hình?

- ❖ Mô hình hóa đạt được bốn mục tiêu [1]:
  - Giúp bạn “hình dung một hệ thống như bạn muốn”.
  - Cho phép bạn “chỉ định cấu trúc hoặc hành vi của hệ thống”.
  - Cung cấp cho bạn “một khuôn mẫu hướng dẫn bạn xây dựng một hệ thống”.
  - “Ghi lại các quyết định bạn đã thực hiện”.
- ❖ Bạn xây dựng mô hình của các hệ thống phức tạp bởi vì bạn không thể hiểu toàn bộ một hệ thống như vậy.
- ❖ Bạn xây dựng mô hình để hiểu rõ hơn về hệ thống bạn đang phát triển.

[1]: Chapter 1, Section 1.1



## Tầm quan trọng của mô hình hóa



### 3.2 Ngôn ngữ mô hình thống nhất (UML)

- ❖ “UML là ngôn ngữ dành cho
  - Hình dung
  - Xác định
  - Cấu tạo
- ❖ Lập hồ sơ tạo tác của một hệ thống sử dụng nhiều phần mềm”[1].

[1]: Chapter 2, Section 2.1

21

### UML là một ngôn ngữ để hình dung

- ❖ Việc truyền đạt các mô hình khái niệm cho người khác dễ bị sai sót trừ khi tất cả mọi người liên quan đều nói cùng một ngôn ngữ.
- ❖ Có những điều về hệ thống phần mềm mà bạn không thể hiểu trừ khi bạn xây dựng mô hình.
- ❖ Một mô hình rõ ràng tạo điều kiện giao tiếp.



22

### UML là một ngôn ngữ để chỉ định

- ❖ UML xây dựng các mô hình chính xác, rõ ràng và hoàn chỉnh.



23

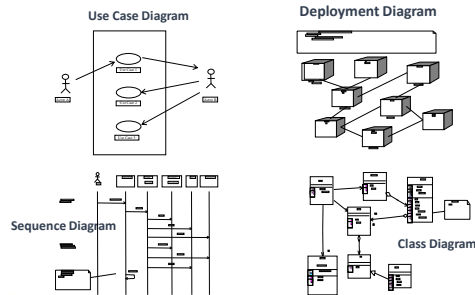
### UML là một ngôn ngữ để xây dựng

- ❖ Các mô hình UML có thể được kết nối trực tiếp với nhiều ngôn ngữ lập trình khác nhau.
  - Tham chiếu tới Java, C ++, Visual Basic, v.v.
  - Các bảng trong RDBMS hoặc lưu trữ liên tục trong OODBMS
  - Cho phép kỹ thuật chuyển tiếp
  - Cho phép thiết kế ngược

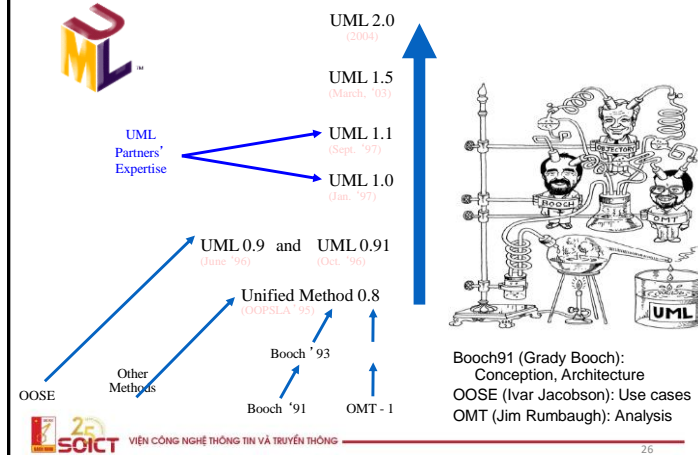
24

## UML là một ngôn ngữ lập tài liệu

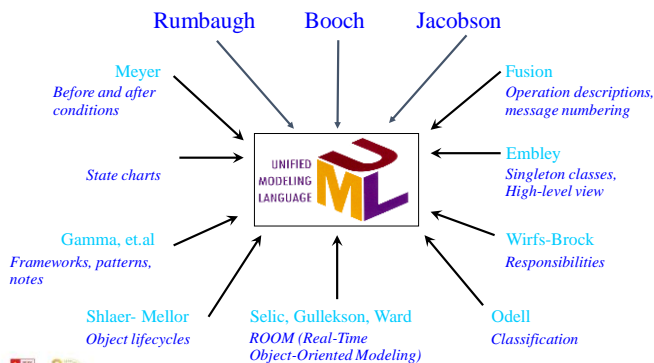
- ❖ UML giải quyết tài liệu về kiến trúc hệ thống, các yêu cầu, kiểm tra, lập kế hoạch dự án và quản lý phát hành



## Lịch sử của UML



## Đầu vào cho UML



## Chế độ xem UML

- ❖ “Một khung nhìn chỉ đơn giản là một tập hợp con của các cấu trúc mô hình hóa UML đại diện cho một khía cạnh của hệ thống” [2]
- ❖ Bốn lĩnh vực phân loại
  - cấu trúc,
  - hành vi năng động
  - bố trí vật lý,
  - và quản lý mô hình.

[2]: Part 2, Chapter 3, Section 3.1

## Chế độ xem UML

Khu vực chính	Chế độ xem	Biểu đồ
Cấu trúc	Chế độ xem tĩnh	Biểu đồ lớp
	Chế độ xem thiết kế	Sơ đồ nội bộ, Sơ đồ cộng tác Sơ đồ thành phần
	Chế độ xem ca sử dụng	Use case diagram
Động	Chế độ xem trạng thái	Sơ đồ trạng thái
	Chế độ xem hoạt động	Sơ đồ hoạt động
	Chế độ xem tương tác	Sơ đồ trình tự Sơ đồ giao tiếp
Vật lý	Chế độ xem triển khai	Sơ đồ triển khai
Mô hình Quản lý	Quản lý mô hình xem	Sơ đồ gói
	Hồ sơ	Sơ đồ gói

[2]: Part 2, Chapter 3, Section 3.1, Table 3.1 (extracted)

## Chế độ xem tĩnh so với chế độ xem động

Vai trò của các bộ phận thành phần và cách chúng liên quan với nhau

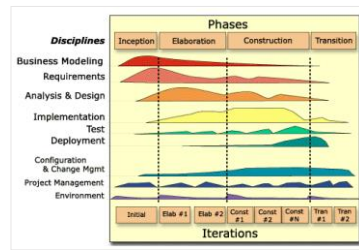
Cách các thành phần tương tác với nhau và / hoặc thay đổi trạng thái nội bộ theo thời gian



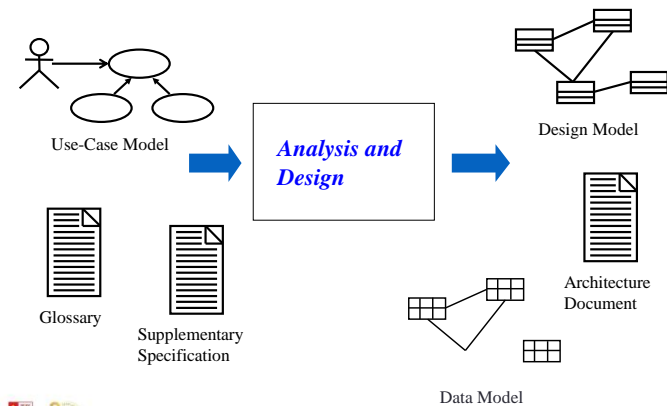
## 3.2 Phân tích và thiết kế hệ thống

Mục đích của Phân tích và Thiết kế là:

- Chuyển đổi các yêu cầu thành một thiết kế của hệ thống tương lai.
- Phát triển một kiến trúc mạnh mẽ cho hệ thống.
- Điều chỉnh thiết kế để phù hợp với môi trường thực hiện, thiết kế để thực hiện.

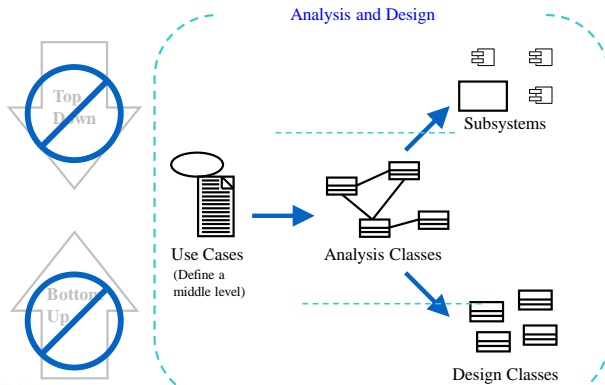


## Phân tích và thiết kế hướng đối tượng





## Phân tích và thiết kế không từ trên xuống hoặc từ dưới lên



33

## Các bước phân tích và thiết kế

Hoạt động	Bước	Mô tả	Người thực hiện
Xác định kiến trúc ứng viên	1. Phân tích kiến trúc	<ul style="list-style-type: none"> <li>Một lần ở giai đoạn đầu</li> <li>Bỏ qua nếu rủi ro kiến trúc thấp</li> </ul>	Kiến trúc sư
Phân tích hành vi	2. Phân tích ca sử dụng	<ul style="list-style-type: none"> <li>Mỗi trường hợp ca sử dụng</li> </ul>	Nhà thiết kế
Tinh chỉnh kiến trúc	3. Xác định các yếu tố thiết kế	<ul style="list-style-type: none"> <li>Khớp nối và gắn kết</li> <li>Khả năng tái sử dụng</li> </ul>	Kiến trúc sư
	4. Xác định cơ chế thiết kế	<ul style="list-style-type: none"> <li>Mẫu thiết kế</li> </ul>	
	5. Mô tả kiến trúc thời gian chạy	<ul style="list-style-type: none"> <li>Bỏ qua nếu không phải đa luồng</li> <li>Chế độ xem</li> </ul>	
	6. Mô tả phân phối	<ul style="list-style-type: none"> <li>Kiến trúc vật lý</li> </ul>	
Thành phần thiết kế	7. Thiết kế ca sử dụng	<ul style="list-style-type: none"> <li>Mỗi trường hợp sử dụng</li> </ul>	Nhà thiết kế
	8. Thiết kế hệ thống con		
	9. Thiết kế lớp		
Thiết kế DB	10. Thiết kế CSDL		

34

## Nội dung

1. Quy trình vòng đời phần mềm
2. Quy trình triển khai phần mềm
3. Phân tích thiết kế hướng đối tượng
4. Mô hình phát triển phần mềm

35

## Bài tập về nhà

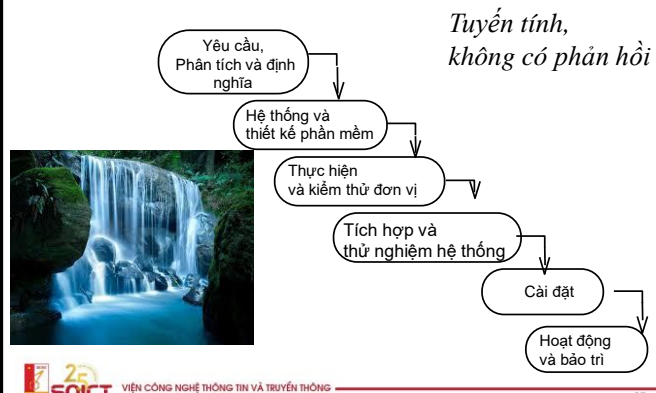
❖ So sánh một số hình phát triển phần mềm phổ biến

- Mô hình thác nước
- Mô hình lặp lại
- Mô hình nguyên mẫu
- Mô hình xoắn ốc
- Quy trình hợp nhất khẩu phần (RUP)
- Phương pháp nhanh nhẹn

→ Gửi báo cáo cá nhân

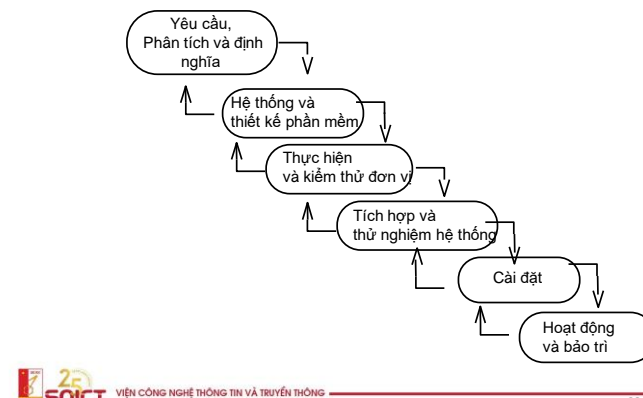
36

## Mô hình thác nước / tuyến tính



37

## Thác nước lặp lại / Mô hình Tuyến tính



38

## Mô hình lặp lại

❖ Mỗi lần lặp lại tạo ra một tệp thực thi



39

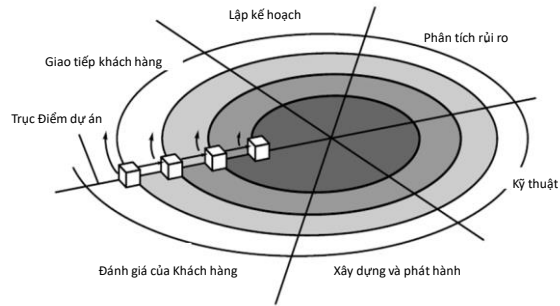
## Mô hình nguyên mẫu

1. Thu thập yêu cầu
2. Thiết kế và xây dựng mẫu SW
3. Đánh giá nguyên mẫu với khách hàng
4. Xây dựng / sửa đổi nguyên mẫu



40

## Mô hình xoắn ốc



41

## Phương pháp nhanh nhen

❖ “Sự nhanh nhen là khả năng vừa tạo ra vừa phản ứng với sự thay đổi để thu lợi nhuận trong một môi trường đầy biến động.”

- [Jim Highsmith, Hệ sinh thái phát triển phần mềm Agile, Lời nói đầu XXIII]

=> Mục tiêu: Vạch ra các giá trị và nguyên tắc để cho phép nhóm phần mềm:

- phát triển nhanh chóng và
- đáp ứng với sự thay đổi.



42

## Theo Agile

- ❖ Các **cá nhân** và **tương tác** qua các quy trình và công cụ
- ❖ **Làm việc phần mềm** trên tài liệu toàn diện
- ❖ Sự **cộng tác của khách hàng** trong quá trình đàm phán hợp đồng
- ❖ Đáp ứng sự **thay đổi** so với việc tuân theo một kế hoạch

43

## Các nguyên tắc của Agile [1]

- ❖ 1. Ưu tiên cao nhất của chúng tôi là làm **hài lòng khách hàng** thông qua việc phân phối sớm và liên tục các phần mềm có giá trị.
- ❖ 2. Hoan nghênh các yêu cầu **thay đổi**, thậm chí muộn trong quá trình phát triển. Các quy trình nhanh nhen khai thác sự thay đổi vì lợi thế cạnh tranh của khách hàng.
- ❖ 3. Cung cấp **phần mềm hoạt động thường xuyên**, từ vài tuần đến vài tháng, với ưu tiên trong quy mô thời gian ngắn hơn.

44

## Các nguyên tắc của Agile [2]

- ❖ 4. **Người kinh doanh** và **nhà phát triển** phải làm việc cùng nhau hàng ngày trong suốt dự án.
- ❖ 5. Xây dựng các dự án xung quanh những cá nhân có động lực. Cung cấp cho họ môi trường và hỗ trợ nhu cầu của họ, và **tin tưởng** họ hoàn thành công việc.
- ❖ 6. Phương pháp hiệu quả và hiệu quả nhất để truyền tải thông tin đến và trong nhóm phát triển là **trò chuyện trực tiếp**.



## Các nguyên tắc của Agile [3]

- ❖ 7. **Phần mềm làm việc** là thước đo chính của sự tiến bộ.
- ❖ 8. Các quy trình Agile thúc đẩy sự phát triển **bền vững**.
- ❖ 9. Các nhà tài trợ, nhà phát triển và người dùng có thể duy trì **tốc độ liên tục vô thời hạn**.
- ❖ 10. Liên tục chú ý đến sự xuất sắc về kỹ thuật và thiết kế tốt giúp tăng cường sự nhanh nhẹn.



## Các nguyên tắc của Agile [4]

- ❖ 11. Sự **đơn giản** - nghệ thuật tối đa hóa khối lượng công việc chưa hoàn thành - là điều cần thiết.
- ❖ 12. Các kiến trúc, yêu cầu và thiết kế tốt nhất xuất hiện từ **các nhóm tự tổ chức**.
- ❖ 13. Định kỳ, nhóm **phản ánh** cách trở nên hiệu quả hơn, sau đó điều chỉnh và **điều chỉnh** hành vi của mình cho phù hợp.

