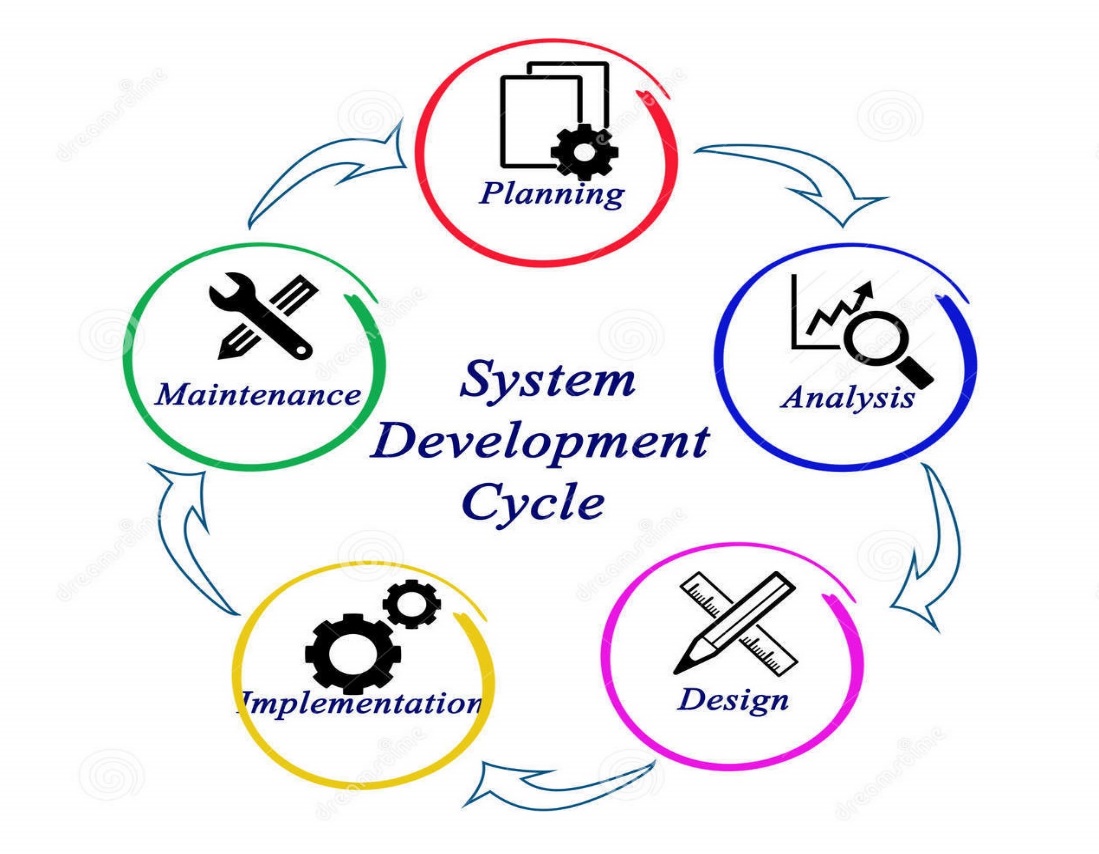
1. SDLC: System Development Life Cycle: Vòng đời phát triển hệ thống:

* Là một quá trình theo sau cho một dự án phần mềm, trong một tổ chức phần mềm. Nó bao gồm một kế hoạch chi tiết mô tả làm thế nào để phát triển, duy trì, thay đổi hoặc nâng cấp phần mềm cụ thể.
* Khái niệm vòng đời phát triển hệ thống áp dụng cho một loạt cấu hình phần cứng và phần mềm, vì một hệ thống có thể được tạo nên từ chỉ phần cứng, chỉ phần mềm, hoặc kết hợp cả hai



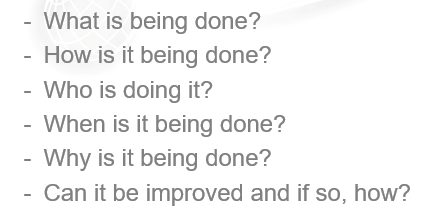
* Planning – Lập kế hoạch: định nghĩa, phân tích và đánh giá ưu tiên của việc cần thiết khi xây dựng mới, hoặc nâng cấp hệ thống phần mềm.
* Analysis - Phân tích: khảo sát và cấu trúc các yêu cầu của một hệ thống thông tin
* Design - Thiết kế: chi tiết hóa hoặc chuyển đổi các giải pháp được đề xuất ở bước phân tích tới hệ thống xây mới hoặc nâng cấp.
* Implementation – Triển khai: lập trình, kiểm thử, đánh giá và cài đặt hệ thống mới hoặc nâng cấp.
* Maintenance – Bảo trì: nâng cấp và phát triển thêm hệ thống.

1. Pha Lập kế hoạch:

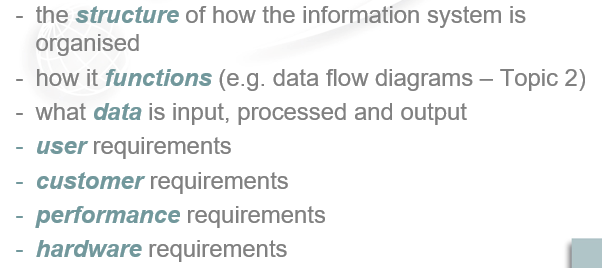
* Giải quyết các vấn đề, câu hỏi
* Tại sao phải xây dựng HTTT ?
* Nhóm dự án phát triển hệ thống thông tin như thế nào?
* Các công việc cụ thể
* Tìm hiểu dự án được bắt đầu và được đánh giá ban đầu như thế nào?
* Xác định các vấn đề, cơ hội, mục tiêu
* Phân tích SWOT (Strength – Weakness – Opportunity – Threat)

1. Pha Phân tích hệ thống

* Giải quyết các vấn đề, câu hỏi
* Ai sẽ sử dụng hệ thống?
* Hệ thống sẽ thực hiện gì, khi nào, ở đâu?



* Các công việc cụ thể
* Phân tích chiến lược: phân tích hiện trạng, phương pháp sử dụng
* Thu thập yêu cầu: mô hình hóa và phân tích các yêu cầu
* Đề xuất mô hình hệ thống



1. Pha thiết kế hệ thống

* Giải quyết các vấn đề, câu hỏi
* Hệ thống sẽ hoạt động như thế nào (phần cứng, phần mềm, mạng, giao diện người dùng, module chương trình, CSDL, tệp, …)
* Các công việc cụ thể
* Chiến lược thực hiện
* Kiến trúc hệ thống: phần cứng, phần mềm, mạng
* Thiết kế dữ liệu
* Thiết kế chương trình
* Thiết kế giao diện

axure

1. Pha triển khai/ cài đặt hệ thống

* Giải quyết các vấn đề, câu hỏi
* Lập trình, kiểm thử, đánh giá và cài đặt hệ thống
* Các công việc cụ thể
* Phát triển ứng dụng: lập trình, kiểm thử, lập tài liệu cho các modun chương trình Kiến trúc hệ thống: phần cứng, phần mềm, mạng
* Cài đặt và đánh giá
* Xây dựng kế hoạch hỗ trợ và bảo trì hệ thống

1. Pha Bảo trì

* Nâng cấp và phát triển thêm hệ thống.

1. Phương pháp phát triển hệ thống

* Cách thực hiện vòng đời phát triển hệ thống.

Hiện tại có 3 nhóm phương pháp chính:

|  |  |  |
| --- | --- | --- |
| 1 | **Các phương pháp hướng quy trình** | * Tập trung định nghĩa các hoạt động gắn với hệ thống * Mô hình hóa các quy trình với luồng vào/ra |
| 2 | **Các phương pháp hướng dữ liệu** | * Tập trung định nghĩa nội dung dữ liệu lưu trữ * Mô hình hóa dữ liệu |
| 3 | **Các phương pháp hướng đối tượng** | * Cân bằng giưa dữ liệu và quy trình * UML (ngôn ngữ mô hình hóa) |

Một số kiểu phát triển hệ thống:

* Phân tích thiết kế có cấu trúc:
* Quy trình thác nước - Waterfall Development Model
* Quy trình tăng trưởng/ quy trình song song - Parallel Development Model
* Phát trienr nhanh ứng dụng (Rapid Application Development – RAD)
* Thực hiện phát triển từng phần hệ thống, mục đích chuyển giao sớm cho người dùng.
* Quy trình xoắn ốc
* Quy trình làm bản mẫu
* Hướng lập trình ứng dụng (Agile Development)
* Tập trung vào phát triển phần mềm linh hoạt, phân giai đoạn thực hiện ( sprint) thể hiện đầy đủ các công việc cần thiết như lập kế hoạch, phân tích, thiết kế, triển khai, kiểm thử tại mỗi sprint.
* Phát triển ứng dụng đơn giản, lăp đi lặp lại
* eXtreme-Programming based

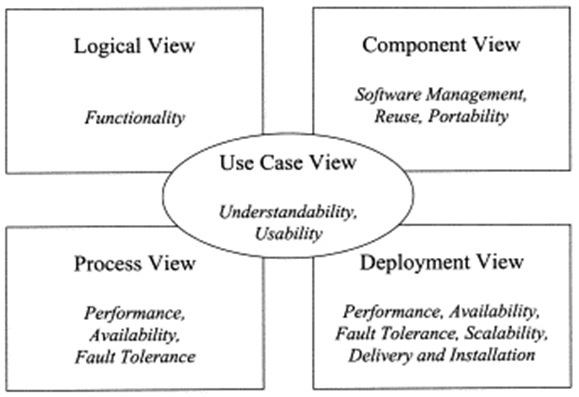
1. Ôn lại kiến thức lập trình hướng đối tượng

* Đối tượng: biểu diễn 1 thực thể thực tế, có thể là thực thể vật lý, thực thể trừu tượng, thực thể phần mềm.
* Đối tượng là thực thể có phạm vi và định danh được xác định rõ rang, có trạng thái và hành vi.
* Trạng thái: biểu diễn thuộc tính và đặc điểm
* Hành vi: các thao tác, các phương thức hoặc hành động của đối tượng
* Lớp là đại diện của tập các đối tượng, khuôn mẫu đặc tả cho các đối tượng, tập trung vào các đặc tinh chung, không có giá trị cụ thể thể hiện.
* 1 đối tượng là 1 thể hiện của lớp.
* 4 tính chất của OOP:
* Đóng gói
* Kế thừa
* Đa hình
* Trừu tượng
* Lợi ích:
* Giúp tái sử dụng mã nguồn và kiến trúc
* Phản ánh gần hơn các mô hình trong thế giới thực
* Ổn định và dễ dùng
* Có khả năng thích ứng với thay đổi
* …
* Ứng dung:
* Hệ thống web hoặc hệ thống client/ server: jhipster, spring eco…
* Hệ thống thời gian thực
* Hệ thống destop
* Mobile

…

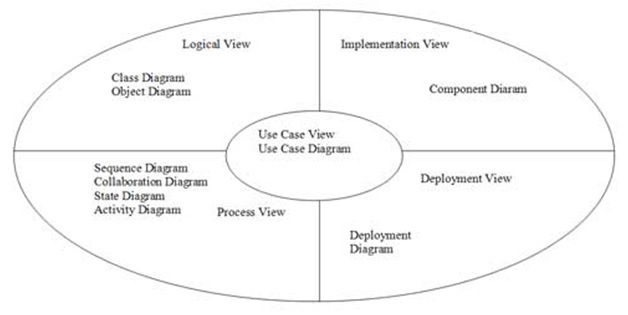
1. Ngôn ngữ mô hình hóa thống nhất ( trực quan ) UML

* Định nghĩa: UML là Unified Modeling Laguage để:
* Trực quan hóa
* Xác định, đặc tả hệ thống
* Xây dựng
* Tài liệu hóa
* Giúp công việc được xử lý nhất quán, giảm thiểu lỗi xảy ra
* Các khung nhìn của UML



* Use Case View: cung cấp góc nhìn về các ca sử dụng giúp chúng ta hiểu hệ thống có gì? ai dùng và dùng nó như thế nào.
* Logical View: cung cấp góc nhìn về cấu trúc hệ thống, xem nó được tổ chức như thế nào. Bên trong nó có gì.
* Process View: cung cấp góc nhìn động về hệ thống, xem các thành phần trong hệ thống tương tác với nhau như thế nào.
* Component View: Cũng là một góc nhìn về cấu trúc giúp chúng ta hiểu cách phân bổ và sử dụng lại các thành phần trong hệ thống ra sao.
* Deployment View: cung cấp góc nhìn về triển khai hệ thống, nó cũng ảnh hưởng lớn đến kiến trúc hệ thống.

Tương ứng với Diagram:



**Biểu đồ use case:**

Mỗi hệ thống tương tác với con người hoặc các hệ

thống khác để thực hiện nhiệm vụ  
• Các hành vi của hệ thống có thể được mô tả trong các use case.  
▫ What, not How  
▫ Các use case mô tả các tương tác giữa hệ thống và môi trường của nó  
• Biểu đồ mô tả các yêu cầu chức năng của hệ thống dưới dạng các use case

**Biểu đồ lớp:**

Mục đích:

• Mô tả khung nhìn tĩnh của hệ thống và cách đưa nó  
vào trong một mô hình.

• Minh họa cách đọc và hiểu một biểu đồ lớp.

• Mô hình hóa mối liên kết (association) và kết tập  
(aggregation) và chỉ ra cách mô hình chúng vào biểu  
đồ lớp.  
• Mô hình tổng quát hóa (generalization) trên một  
biểu đồ lớp

**Biểu đồ tương tác:**

Mô hình hóa phương diện động của hệ thống, mô tả  
tương tác giữa các đối tượng

• Thường dùng để mô tả kịch bản của use case

Biểu đồ tương tác là gì?

• Thuật ngữ chung cho các biểu đồ thể hiện tương  
tác giữa các đối tượng.

▫ Biểu đồ tuần tự (Sequence diagram)  
▫ Biểu đồ giao tiếp (Communication diagram)

• Các biến thể chuyên dụng

▫ Biểu đồ thời gian (Timing Diagram)

▫ Biểu đồ tương tác tổng quát (Interaction Overview  
Diagram

**Biểu đồ trình tự:**

Được sử dụng để xác định và chỉ rõ vai trò của  
các đối tượng tham gia vào luồng sự kiện của  
use case

• Là một loại biểu đồ tương tác, mô tả mô hình  
tương tác giữa các đối tượng, trong đó nhấn  
mạnh vào trình tự thời gian của các thông điệp  
trao đổi giữa các đối tượng đó.

**Biểu đồ giao tiếp:**

Biểu đồ giao tiếp nhấn mạnh vào việc tổ chức các  
đối tượng tham gia vào tương tác.

• Biểu đồ giao tiếp chỉ ra:

▫ Các đối tượng tham gia vào tương tác.

▫ Các liên kết giữa các đối tượng.  
▫ Các thông điệp trao đổi giữa các đối tượng.

1. Phân tích và thiết kế hệ thống hướng đối tượng (OOAD)
2. Tầm quan trọng của OOAD

Nguyên nhân:

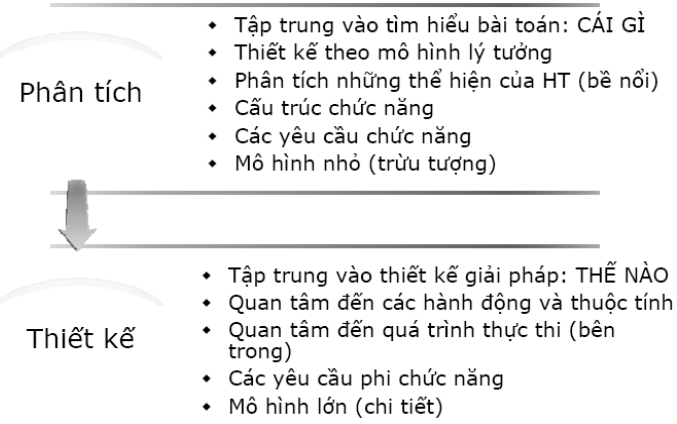
* Nhiều người phát triển dự án:
* Cho rằng phần mềm chủ yếu được xây dựng bằng cách gõ lệnh/ code
* Không dành đủ thời gian cho quá trình phân tích và thiết kế phần mềm
* Nên phải “cày bừa” để hoàn thành chương trình vì các nguyên nhân:
* Không hiểu rõ hoặc hiểu sai yêu cầu
* Giao tiếp nhóm/ giữa các thành viên không tốt
* Không tích hợp được các module trong dự án
* Phần tích và Thiết kế cần được coi trọng.

1. Mục đích của OOAD

* Chuyển các yêu cầu của bài toán thành 1 bản thiết kế của hệ thống sẽ được xây dựng
* Tập trung vào quá trình phân tích các yêu cầu của hệ thống và thiết kế các mô hình cho hệ thống đó trước giai đoạn lập trình.
* Các bước thực hiện nhằm đảm bảo mục đích và yêu cầu của hệ thống được ghi lại 1 cách hợp lý trước khi hệ thống được xây dựng

1. Phương pháp OOAD

* OOAD được chia làm 2 giai đoạn:
* Phân tích hướng đối tượng (OOA): nhằm tạo ra các mô hình cơ bản của hệ thống dựa trên những gì khách hàng yêu cầu về hệ thống của họ.
* Thiết kế hướng đối tượng (OOD) sẽ bổ sung thêm các thông tin thiết kế chi tiết cho hệ thống từ bước phân tích.



* Mô hình phân tích: tập trung vào việc trừu tượng hóa các vấn đề nghiệp vụ, xây dựng mô hình bằng các tìm kiếm các lớp, các đối tượng chính có trong hệ thống. Tránh đưa ra các thông tin cài đặt quá chi tiết.
* Mô hình thiết kế:

+ Bao gồm các lớp, đối tượng mức cao ( package, mối quan hệ giữa các lớp )

+ Mô hình phát triển phần mềm.

1. Các bước trong OOAD

