

Lý thuyết

Contents

1. Trình bày tóm tắt các thành phần trong giai đoạn thiết kế (Design) của chu trình phát triển hệ thống (Systems Development Life Cycle - SDLC).....	4
2. Trình bày tóm tắt các thành phần trong giai đoạn thực hiện (Implementation) của chu trình phát triển hệ thống (Systems Development Life Cycle - SDLC)	4
3. Trình bày và giải thích các phương pháp hướng đối tượng OOA, OOD, OOP. Cho ví dụ minh họa.	4
4. Rapid Application Development là gì? Trình bày các thành phần có trong RAD. Vẽ mô hình.	4
5. Kiến trúc mô hình khung nhìn (view) 4+1 gồm có những sơ đồ nào? Giải thích ngắn gọn các sơ đồ của Process view.....	5
6. Trình bày và giải thích các khái niệm (principles) của phương pháp Agile.....	5
7. Sơ đồ lớp (class diagram) là gì? Có những khái niệm phần tử nào kèm ví dụ minh họa?.....	5
8. Trình bày tóm tắt và ví dụ minh họa các thành phần trong giai đoạn triển khai (Implementation) của chu trình phát triển hệ thống (Systems Development Life Cycle - SDLC).....	6
9. Trình bày và giải thích kèm ví dụ minh họa cho yêu cầu hệ thống (System request) trong giai đoạn 1 của chu trình phát triển hệ thống (Systems Development Life Cycle - SDLC).....	6
10. Giải thích và ví dụ minh họa 3 vai trò trong Scrum.....	6
11. Sơ đồ tuần tự (Sequence diagram) là gì? Có những phần tử nào? Giải thích.	6
12. Giải thích ngắn gọn các sơ đồ của Logical view và Implementation view kèm hình minh họa.	7
13. Trình bày khái niệm các sơ đồ cùng các đối tượng, trạng thái thuộc khung nhìn xử lý (process view).....	7
14. Khi nào thì áp dụng một trong các cách tiếp cận vòng đời phát triển ứng dụng (Software Development Life Cycle), phương pháp Agile (Agile Methodologies), phương pháp hướng đối tượng (Object-Oriented Methodologies)?	8
15. Scrum là gì? Có những giai đoạn nào? Giải thích.....	8
16. Trình bày và cho ví dụ minh họa các đặc tính của câu hỏi mở (Opened) và câu hỏi đóng (Closed question).	9
17. Kiến trúc mô hình khung nhìn (view) 4+1 gồm có những sơ đồ nào? Giải thích ngắn gọn các sơ đồ của khung nhìn xử lý (Process view).....	9

18.	Chất lượng phần mềm được đánh giá qua các đặc tính gì? Giải thích.....	9
19.	Phát triển ứng dụng nhanh (Rapid Application Development) là gì? Vẽ mô hình RAD và giải thích các thành phần.	10
20.	Trình bày các khái niệm trong phương pháp phân tích hướng đối tượng (Class – Object). Cho ví dụ minh họa.	10
21.	Trình bày và giải thích các loại tài liệu có trong quá trình phân tích thiết kế hệ thống.	11
22.	Trình bày và cho ví dụ các khái niệm về kịch bản người sử dụng (user scenario), câu chuyện người sử dụng (user story) và trường hợp sử dụng (use case)	11
23.	Trình bày các khái niệm về tác nhân (actor), trường hợp sử dụng (Use case) và mối quan hệ giữa các use case. Cho một ví dụ minh họa và diễn giải ngắn gọn.	12
24.	Giải thích và ví dụ minh họa 3 vai trò trong Scrum.	12
25.	Trình bày và cho ví dụ các khái niệm về tài liệu thiết kế mức cao (high-level) trong giai đoạn thiết kế của chu trình phát triển phần mềm (SDLC).	12
26.	Trình bày và cho ví dụ các hình thức kiểm tra (test) trong giai đoạn kiểm tra (Testing phase) của chu trình phát triển phần mềm (SDLC).	13
27.	Trình bày và cho ví dụ minh họa các đặc tính của câu hỏi mở (Opened) và câu hỏi đóng (Closed question).	13
28.	Scrum là gì? Có những giai đoạn nào? Giải thích.....	13
29.	Giải thích khi nào thì sử dụng phương pháp chu trình phát triển phần mềm (System Development Life Cycle) – phương pháp Agile	13
30.	Trình bày ba vai trò của Scrum team: Product owner – Development team – Scrum master.....	14
31.	Sơ đồ trạng thái (State diagram).....	17
32.	Các mối quan hệ trong use case	19
a.	Mối quan hệ extend.....	19
b.	Mối quan hệ include	19
	Là mối quan hệ giữa A và B, để thực hiện A cần phải thực hiện B trước	19
	VD: Một từ điển Anh-Việt trực tuyến có chức năng tra từ cho phép người dùng nhập vào từ tiếng Anh để tìm nghĩa tiếng Việt. Từ điển còn có chức năng cho phép người dùng thêm từ đã tra vào danh sách các từ thường gặp. Theo mô hình use-case, mối liên hệ giữa chức năng tra từ và thêm từ đã tra vào danh sách là extend.....	19
33.	Mối quan hệ giữa actor.....	19
34.	Mối quan hệ giữa actor và use case Association.....	19
35.	Use case Scenario: Một chuỗi các hành động cụ thể được mô tả theo trình tự từ đầu cho đến kết thúc một use-case	19

36. Lược đồ use-case cho thấy Một số chức năng của hệ thống phần mềm và một số actor tương tác với các chức năng này	19
37. Actor là tác nhân bên ngoài của hệ thống, phát họa ai và cái gì sẽ tương tác với hệ thống và hệ thống cung cấp các chức năng gì	19
38. Trong qui trình phát triển phần mềm, giai đoạn phân tích tập trung vào những công việc trả lời cho câu hỏi What.....	20
43. d. Bài học được rút ra trong lần lặp trước có thể được áp dụng cho lần lặp sau để cải thiện qui trình thực hiện tốt hơn.....	20
45.	20
Bài toán thư viện.....	22

1. Trình bày tóm tắt các thành phần trong giai đoạn thiết kế (Design) của chu trình phát triển hệ thống (Systems Development Life Cycle - SDLC)
 - Thiết kế đầu vào (input design)
 - Thiết kế đầu ra (output design)
 - Thiết kế cơ sở dữ liệu (database design)
 - Thiết kế chương trình (program design)
 - Kiến trúc chương trình (Architectural design)

2. Trình bày tóm tắt các thành phần trong giai đoạn thực hiện (Implementation) của chu trình phát triển hệ thống (Systems Development Life Cycle - SDLC)
 - Lập trình (coding)
 - Cài đặt (Installation)
 - Huấn luyện (training)
 - Chuyển đổi tập tin dữ liệu (file conversion)

3. Trình bày và giải thích các phương pháp hướng đối tượng OOA, OOD, OOP. Cho ví dụ minh họa.

OOA: Phân tích hướng đối tượng là tìm kiếm và miêu tả các khái niệm hay đối tượng.

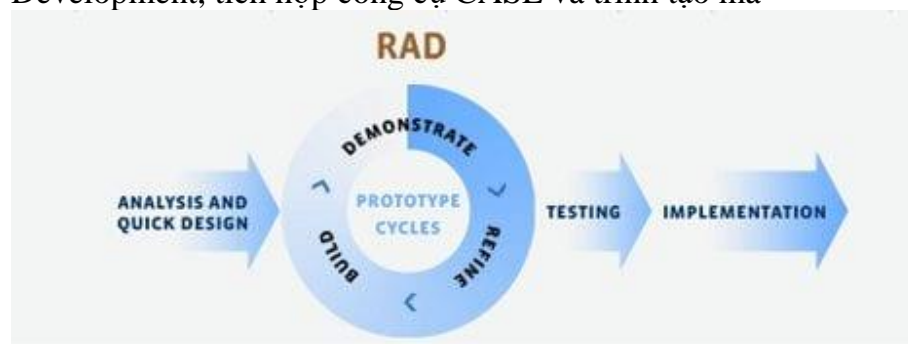
OOD: Thiết kế hướng đối tượng là xác định cách các đối tượng phần mềm này cộng tác để đáp ứng các yêu cầu. Gồm có thuộc tính (Attribute) và phương thức (method)

OOP: Triển khai (lập trình) các thiết kế đối tượng trên các phần mềm ngôn ngữ lập trình như C++, C#, Java, ...

4. Rapid Application Development là gì? Trình bày các thành phần có trong RAD.
Về mô hình.

Rapid Application Development một mô hình tiến trình phần mềm gia tăng với chu kỳ phát triển ngắn. Mô hình RAD dựa vào sử dụng thành phần (component) và sử dụng các ứng dụng tạo mã tự động.

Gồm có: sự tham gia của người dùng, tạo mẫu (prototyping), Join Application Development, tích hợp công cụ CASE và trình tạo mã



5. Kiến trúc mô hình khung nhìn (view) 4+1 gồm có những sơ đồ nào? Giải thích ngắn gọn các sơ đồ của Process view.

Use Case View: Use Case Diagram

Logical view: Class Diagram, Object diagram

Process view:

- **Sequence diagram:** xác định các trình tự diễn ra sự kiện của một nhóm đối tượng nào đó.
- **Collaboration diagram:** minh họa về các mối quan hệ và tương tác giữa các đối tượng của hệ thống.
- **Activity diagram:** mô tả các bước thực hiện, các hành động, các nút quyết định và điều kiện rẽ nhánh để điều khiển luồng thực hiện của hệ thống. Đối với những luồng thực thi có nhiều tiến trình chạy song song thì biểu đồ hoạt động là sự lựa chọn tối ưu cho việc thể hiện.
- **State diagram:** xác định các trạng thái khác nhau của một đối tượng trong suốt thời gian tồn tại của nó và các trạng thái này được thay đổi bởi các sự kiện.

Implementation view: Component diagram, Package diagram

Deployment view: Deployment diagram

6. Trình bày và giải thích các khái niệm (principles) của phương pháp Agile..

Trình bày, giải thích được các ý:

- Sự tham gia của khách hàng (customer involvement)
- Phân phối gia tăng (incremental delivery)
- Tập trung vào con người không phải qui trình (people, not process)
- Nắm lấy thay đổi (Embrace change)
- Duy trì sự đơn giản (maintain simplicity)

7. Sơ đồ lớp (class diagram) là gì? Có những khái niệm phân tử nào kèm ví dụ minh họa?

Trình bày, giải thích và ví dụ minh họa được các ý:

- Class: Name - Attribute - Operation()
- Đa hình - Interface
- Relationships: Association, Aggregation, Composition, Generalization (inheritance) và Realization
- Multiplicity và Navigation

8. Trình bày tóm tắt và ví dụ minh họa các thành phần trong giai đoạn triển khai (Implementation) của chu trình phát triển hệ thống (Systems Development Life Cycle - SDLC)

Trình bày, giải thích và ví dụ minh họa được các ý:

- Coding
- Installation
- Training
- File Conversion

9. Trình bày và giải thích kèm ví dụ minh họa cho yêu cầu hệ thống (System request) trong giai đoạn 1 của chu trình phát triển hệ thống (Systems Development Life Cycle - SDLC).

Trình bày, giải thích và ví dụ minh họa được các ý:

- Improved service
- Better performance
- More information
- Stronger controls
- Reduced cost

10. Giải thích và ví dụ minh họa 3 vai trò trong Scrum.

BA VAI TRÒ

Trong Scrum, đội ngũ tham gia phát triển phần mềm được phân chia ra ba vai trò với trách nhiệm rõ ràng để đảm bảo tối ưu hóa các công việc đặc thù. Ba vai trò này bao gồm: Product Owner(chủ sản phẩm), Scrum Master và Development Team (Đội sản xuất hay Nhóm Phát triển).

Product Owner (chủ sản phẩm): Là người chịu trách nhiệm về sự thành công của dự án, người định nghĩa các yêu cầu và đánh giá cuối cùng đầu ra của các nhà phát triển phần mềm.

Scrum Master: Là người có hiểu biết sâu sắc về Scrum và đảm bảo nhóm có thể làm việc hiệu quả với Scrum.

Development Team (Đội sản xuất, hay Nhóm phát triển) Một nhóm liên chức năng (cross-functional) tự quản lý để tiến hành chuyển đổi các yêu cầu được tổ chức trong Product Backlog thành chức năng của hệ thống.

11. Sơ đồ tuần tự (Sequence diagram) là gì? Có những phần tử nào? Giải thích.

Giải thích các ký hiệu thông điệp (message).

Trình bày, giải thích được các ý trong sơ đồ tuần tự::

xác định các trình tự diễn ra sự kiện của một nhóm đối tượng nào đó. Nó miêu tả chi tiết các thông điệp được gửi và nhận giữa các đối tượng đồng thời cũng chú trọng đến việc trình tự về mặt thời gian gửi và nhận các thông điệp đó.

- Object – Message

- Synchronous Message
- Asynchronous Message
- Self Message
- Reply or Return Message
- Create Message
- Delete Message

12. Giải thích ngắn gọn các sơ đồ của Logical view và Implementation view kèm hình minh họa.

Use Case View: Use Case Diagram

Logical view:

- **Class Diagram:** mô tả cấu trúc tĩnh của hệ thống bằng cách hiển thị các lớp của hệ thống, các thuộc tính, hoạt động (hoặc phương thức) của chúng và mối quan hệ giữa các đối tượng.
- **Object diagram:** trình bày mối liên hệ cụ thể giữa các đối tượng có trong các lớp của hệ thống mà hệ thống cần lưu trữ thông tin.

Implementation view:

- **Component diagram:** mô tả các thành phần và sự phụ thuộc của chúng trong hệ thống. Các biểu đồ thành phần giúp chi tiết việc triển khai mô hình và kiểm tra kỹ xem mọi khía cạnh của các chức năng yêu cầu của hệ thống.
- **Package diagram:** nhóm các lớp/ đối tượng vào một phần tử ở mức độ cao hơn. Và giữa các gói cũng có các mối quan hệ với nhau (như các lớp/đối tượng)

13. Trình bày khái niệm các sơ đồ cùng các đối tượng, trạng thái thuộc khung nhìn xử lý (process view).

Trình bày được các ý:

Sequence diagram

- Xác định các trình tự diễn ra sự kiện của một nhóm đối tượng nào đó. Nó miêu tả chi tiết các thông điệp được gửi và nhận giữa các đối tượng
- Gồm có: Đối tượng (object or class): biểu diễn bằng các hình chữ nhật, Đường đời đối tượng (Lifelines): biểu diễn bằng các đường gạch rời thẳng đứng bên dưới các đối tượng và Thông điệp (Message): biểu diễn bằng các đường mũi tên

Collaboration diagram (communication diagram)

- Tương tự sơ đồ tuần tự nhưng chỉ tập trung vào các đối tượng.

State diagram

- Mô tả các trạng thái có thể có và sự chuyển đổi giữa các trạng thái đó khi có các sự kiện tác động của một đối tượng.
- Gồm có: Trạng thái bắt đầu (Initial State), Sự kiện (Event) hoặc Chuyển đổi (Transition), Trạng thái đối tượng (State) và Trạng thái kết thúc (Final State)

Activity diagram

- Mô tả các bước thực hiện, các hành động, các nút quyết định và điều kiện rẽ nhánh để điều khiển luồng thực hiện của hệ thống. Đối với những luồng thực thi có nhiều

tiến trình chạy song song thì biểu đồ hoạt động là sự lựa chọn tối ưu cho việc thể hiện.

- Gồm có:
 - Trạng thái khởi tạo hoặc điểm bắt đầu (Initial State or Start Point)
 - Hoạt động hoặc trạng thái hoạt động (Activity or Action State)
 - Nút quyết định (Decision)
 - Thanh chia nhánh (Fork) hay thanh đồng bộ (Join)
 - Luồng hoạt động (Action Flow)
 - Phân làn (Swimlanes)
 - Trạng thái kết thúc hoặc điểm cuối (Final State or End Point)

14. Khi nào thì áp dụng một trong các cách tiếp cận vòng đời phát triển ứng dụng (Software Development Life Cycle), phương pháp Agile (Agile Methodologies), phương pháp hướng đối tượng (Object-Oriented Methodologies)?

Trình bày được các ý:

Software Development Life Cycle:

- Các hệ thống đã được phát triển và lập tài liệu qua việc sử dụng SDLC
- Điều quan trọng để lập tài liệu là phải ghi lại từng bước
- Việc quản lý mức cao cảm thấy thoải mái hoặc an toàn hơn khi sử dụng SDLC
- Có đủ nguồn lực và thời gian để hoàn thành đầy đủ SDLC
- Truyền thông về cách thức hoạt động của các hệ thống mới là quan trọng

Agile Methodologies:

- Có một trong các phương pháp nhanh nhất trong tổ chức
- Các ứng dụng cần được phát triển nhanh chóng để đáp ứng với môi trường năng động
- Một cuộc xử lý nhanh chóng diễn ra (hệ thống bị lỗi và không có thời gian để tìm ra điều gì đã xảy ra)
- Khách hàng hài lòng với những cách cải tiến gia tăng
- Các người quản lý thực hiện và nhà phân tích đồng ý với các nguyên tắc của phương pháp Agile

Object-Oriented Methodologies:

- Các vấn đề được mô hình hóa theo dạng các lớp (class)
- Một tổ chức hỗ trợ việc học theo UML
- Hệ thống có thể được thêm dần dần, một hệ thống con tại một thời điểm
- Có khả năng tái sử dụng phần mềm đã viết trước
- Có thể chấp nhận giải quyết các vấn đề khó khăn trước

15. Scrum là gì? Có những giai đoạn nào? Giải thích.

Trình bày được các ý:

Scrum là một phương pháp Agile chỉ tập trung vào việc quản lý sự phát triển lặp đi lặp lại hơn là các thực tiễn nhanh nhẹn cụ thể.

Có ba giai đoạn trong Scrum.

Giai đoạn đầu tiên là giai đoạn lập kế hoạch phác thảo, nơi bạn thiết lập các mục tiêu chung cho dự án và thiết kế kiến trúc ứng dụng.

Tiếp theo là một loạt các chu kỳ chạy nước rút, trong đó mỗi chu kỳ phát triển một sự gia tăng của hệ thống.

Giai đoạn kết thúc dự án kết thúc dự án, hoàn thành các tài liệu cần thiết như khung trợ giúp hệ thống, hướng dẫn sử dụng và đánh giá các bài học thực tiễn được rút ra từ dự án.

16. Trình bày và cho ví dụ minh họa các đặc tính của câu hỏi mở (Opened) và câu hỏi đóng (Closed question).

- Độ tin cậy dữ liệu
- Hiệu quả việc sử dụng thời gian
- Độ chính xác của dữ liệu
- Tính bề rộng và chiều sâu
- Kỹ năng phỏng vấn
- Có dễ dàng phân tích không

17. Kiến trúc mô hình **khung nhìn (view) 4+1 gồm có những sơ đồ nào? Giải thích ngắn gọn các sơ đồ của khung nhìn xử lý (Process view).**

Use Case View: Use Case Diagram

Logical view: Class Diagram, Object diagram

Process view:

- **Sequence diagram:** xác định các trình tự diễn ra sự kiện của một nhóm đối tượng nào đó.
- **Collaboration diagram:** minh họa về các mối quan hệ và tương tác giữa các đối tượng của hệ thống.
- **Activity diagram:** mô tả các bước thực hiện, các hành động, các nút quyết định và điều kiện rẽ nhánh để điều khiển luồng thực hiện của hệ thống. Đối với những luồng thực thi có nhiều tiến trình chạy song song thì biểu đồ hoạt động là sự lựa chọn tối ưu cho việc thể hiện.
- **State diagram:** xác định các trạng thái khác nhau của một đối tượng trong suốt thời gian tồn tại của nó và các trạng thái này được thay đổi bởi các sự kiện.

Implementation view: Component diagram, Package diagram

Deployment view: Deployment diagram

18. Chất lượng phần mềm được đánh giá qua các đặc tính gì? Giải thích.

- Tính di động (Portability): khả năng của phần mềm được chuyển dễ dàng từ máy tính này sang máy tính khác.

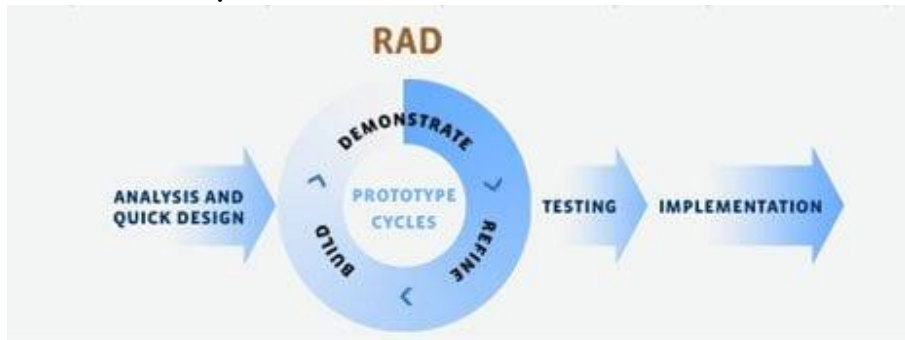
- Tính hiệu quả (Efficiency): khả năng thực hiện của phần mềm với mức sử dụng tối thiểu tài nguyên máy tính.
- Tính tiện dụng (Usability): khả năng dễ hiểu và sử dụng.
- Khả năng kiểm tra (Testability): khả năng của phần mềm dễ dàng được xác minh bằng cách thực hiện.
- Tính dễ hiểu (Understandability, dễ bảo trì): khả năng đọc dữ liệu của phần mềm bảo trì.
- Khả năng hiệu chỉnh (Modifiability): khả năng sửa đổi hiệu chỉnh bởi phần mềm bảo trì.

19. Phát triển ứng dụng nhanh (Rapid Application Development) là gì? Vẽ mô hình RAD và giải thích các thành phần.

Rapid Application Development một mô hình tiến trình phần mềm gia tăng với chu kỳ phát triển ngắn. Mô hình RAD dựa vào sử dụng thành phần (component) và sử dụng các ứng dụng tạo mã tự động.

Gồm có: sự tham gia của người dùng, tạo mẫu (prototyping), Join Application Development, tích hợp công cụ CASE và trình tạo mã

- Giai đoạn phân tích và thiết kế nhanh.
- Chu trình xây dựng mẫu: minh họa – tinh chỉnh theo yêu cầu – lập trình
- Giai đoạn kiểm tra
- Giao đoạn triển khai



20. Trình bày các khái niệm trong phương pháp phân tích hướng đối tượng (Class – Object). Cho ví dụ minh họa.

Khái niệm lớp: gồm có tên lớp, các thuộc tính và phương thức (xử lý) – đối tượng: thể hiện cụ thể của một lớp

Tính kế thừa: các lớp con sẽ kế thừa các thuộc tính hoặc/và các phương thức (xử lý) từ lớp cha – có 2 loại kế thừa: kế thừa đơn và đa kế thừa.

Tính đa hình: là khái niệm mà hai hoặc nhiều lớp có những phương thức giống nhau nhưng có thể thực thi theo những cách thức khác nhau.

Các mối quan hệ: kết hợp (tụ hợp – hợp thành) – bản số – hướng

21. Trình bày và giải thích các loại tài liệu có trong quá trình phân tích thiết kế hệ thống.

- Tài liệu mô tả chức năng (Functional Description)
- Tài liệu giới thiệu hệ thống (Introductory Manual)
- Tài liệu miêu tả các chi tiết của hệ thống (System Reference Manual)
- Tài liệu hướng dẫn cài đặt hệ thống (System Installation Manual)
- Tài liệu quản trị hệ thống (System Administrator's Manual)

22. Trình bày và cho ví dụ các khái niệm về kịch bản người sử dụng (user scenario), câu chuyện người sử dụng (user story) và trường hợp sử dụng (use case) .

- User scenario:

User Scenario là một cách tường thuật về quá trình thực hiện mục tiêu của người dùng trong thế giới thật. User Scenario sẽ nêu rõ mục tiêu, quy trình và kết quả bằng ngôn ngữ của người dùng.

- User story

User Story được sử dụng để nói về một chức năng cụ thể từ yêu cầu của người dùng. Có những User Story nói về các chức năng nhỏ và cũng có User Story để nói về các chức năng rất lớn. User Story lớn có phạm vi rộng và có thể chứa nhiều User Story nhỏ.

- Use Case

Được sử dụng một cách tương tự như User Story nhưng Use Case được dùng để nói một cách cụ thể, chi tiết cho một chức năng nhất định.

Use Case có khả năng hiệu quả khi được sử dụng để kiểm tra thực tế theo yêu cầu nhất định. Do đó, Use Case bị giới hạn trong một luồng tác vụ đơn giản.

	User scenario	User story	Use Case
Người viết	Các nhà nghiên cứu user UX designer	Quản lý dự án, Business Analyst	Người phân tích hệ thống, user
Định nghĩa	Là mục tiêu động lực và nhiệm vụ trong hoàn cảnh nhất định	Là các câu ngắn gọn liên quan đến các yêu cầu của user về hệ thống chức năng	Mô tả chi tiết cách mà hệ thống/phần mềm sẽ tương tác với user
Chức năng	Tìm ra nhu cầu của user. Lập kế hoạch phát triển từ những nhu cầu thực tế đó	Sử dụng mô hình Agile/Scrum để mô tả nhu cầu của user	Giúp người lập trình nắm bắt các tương tác của user với phần mềm. Loại bỏ các thao tác không mong muốn
Ngôn ngữ	User	User	User

Hình thức	Dài - là một dòng các sự kiện qua các Use Case	Ngắn - thường được viết trên card	Dài - thường được viết thành tài liệu hoặc văn bản chi tiết
-----------	--	-----------------------------------	---

23. Trình bày các khái niệm về tác nhân (actor), trường hợp sử dụng (Use case) và mối quan hệ giữa các use case. Cho một ví dụ minh họa và diễn giải ngắn gọn.

Tác nhân (actor): user hay hệ thống khác tương tác với hệ thống.

Có 04 dạng tác nhân

- Users
- Applications
- Devices
- External events

Use Case: biểu diễn một hoạt động cụ thể mà hệ thống cần thực hiện để đáp ứng nhu cầu của người dùng

Các mối quan hệ

- Use-case specialization
- <<Extend>>
- <<Include>>

24. Giải thích và ví dụ minh họa 3 vai trò trong Scrum.

BA VAI TRÒ

Trong Scrum, đội ngũ tham gia phát triển phần mềm được phân chia ra ba vai trò với trách nhiệm rõ ràng để đảm bảo tối ưu hóa các công việc đặc thù. Ba vai trò này bao gồm: Product Owner(chủ sản phẩm), Scrum Master và Development Team (Đội sản xuất hay Nhóm Phát triển).

Product Owner (chủ sản phẩm): Là người chịu trách nhiệm về sự thành công của dự án, người định nghĩa các yêu cầu và đánh giá cuối cùng đầu ra của các nhà phát triển phần mềm.

Scrum Master: Là người có hiểu biết sâu sắc về Scrum và đảm bảo nhóm có thể làm việc hiệu quả với Scrum.

Development Team (Đội sản xuất, hay Nhóm phát triển) Một nhóm liên chức năng (cross-functional) tự quản lý để tiến hành chuyển đổi các yêu cầu được tổ chức trong Product Backlog thành chức năng của hệ thống.

25. Trình bày và cho ví dụ các khái niệm về tài liệu thiết kế mức cao (high-level) trong giai đoạn thiết kế của chu trình phát triển phần mềm (SDLC).

Trình bày và cho ví dụ các khái niệm

- Mô tả ngắn gọn và tên của từng module
- Phác thảo chức năng của mỗi module
- Mối quan hệ và sự phụ thuộc giữa các module

- Xác định các bảng cơ sở dữ liệu cùng với các phần tử chính
- Hoàn thiện sơ đồ kiến trúc cùng với các chi tiết kỹ thuật

26. Trình bày và cho ví dụ các hình thức kiểm tra (test) trong giai đoạn kiểm tra (Testing phase) của chu trình phát triển phần mềm (SDLC).

Trình bày và cho ví dụ

- Kiểm tra mức độ thực hiện (Performance testing)
- Kiểm tra chức năng (Unit testing – functional tests)
- Chất lượng mã lập trình (Code quality)
- Kiểm tra bảo mật (Security testing)
- Kiểm tra tích hợp (Integration testing)

27. Trình bày và cho ví dụ minh họa các đặc tính của câu hỏi mở (Opened) và câu hỏi đóng (Closed question).

Trình bày và cho ví dụ

- Độ tin cậy dữ liệu
- Hiệu quả việc sử dụng thời gian
- Độ chính xác của dữ liệu
- Tình bề rộng và chiều sâu
- Kỹ năng phỏng vấn
- Có dễ dàng phân tích không

28. Scrum là gì? Có những giai đoạn nào? Giải thích.

Trình bày được các ý:

Scrum là một phương pháp Agile chỉ tập trung vào việc quản lý sự phát triển lặp đi lặp lại hơn là các thực tiễn nhanh nhẹn cụ thể.

Có ba giai đoạn trong Scrum.

Giai đoạn đầu tiên là giai đoạn lập kế hoạch phác thảo(initial phase), nơi bạn thiết lập các mục tiêu chung cho dự án và thiết kế kiến trúc ứng dụng.

Tiếp theo là một loạt các chu kỳ chạy nước rút (sprint cycle), trong đó mỗi chu kỳ phát triển một sự gia tăng của hệ thống.

Giai đoạn kết thúc dự án kết thúc dự án (project closure phase), hoàn thành các tài liệu cần thiết như khung trợ giúp hệ thống, hướng dẫn sử dụng và đánh giá các bài học thực tiễn được rút ra từ dự án.

29. Giải thích khi nào thì sử dụng phương pháp chu trình phát triển phần mềm (System Development Life Cycle) – phương pháp Agile

- SDLC:

Áp dụng cho các dự án lớn

Các hệ thống đã được phát triển và lập tài liệu theo phương pháp SDLC

Điều quan trọng là phải lập tài liệu sau mỗi bước

Quản lý cấp trên cảm thấy thoải mái hơn hoặc an toàn hơn khi sử dụng phương pháp SDLC

Có đủ nguồn lực và thời gian để hoàn thành dự án theo phương pháp SDLC

Truyền thông về cách thức hoạt động của các hệ thống mới là rất quan trọng

- Agile:

Áp dụng cho các dự án vừa và nhỏ

Có một người dẫn đầu dự án về các phương pháp Agile trong tổ chức

Các ứng dụng cần được phát triển nhanh chóng để đáp ứng với môi trường năng động

Một cuộc giải cứu diễn ra (hệ thống không thành công và không có thời gian để tìm ra điều gì đã xảy ra)

Khách hàng hài lòng với những cải tiến gia tăng

Giám đốc điều hành và nhà phân tích đồng ý với các nguyên tắc của phương pháp Agile

30. Trình bày ba vai trò của Scrum team: Product owner – Development team – Scrum master

- **Product Owner:** Nhiệm vụ của Product Owner là đảm bảo việc quản lý những công việc còn tồn đọng (**Product backlog**) của việc phát triển sản phẩm phần mềm. Product Owner phải liên tục cập nhật thông tin cho các thành viên trong team để họ hiểu về yêu cầu hay các tính năng cần có của sản phẩm ngay cả khi họ không trực tiếp phát triển tính năng đó.
- **Development Team:** là những lập trình viên sẽ tham gia vào việc phát triển từng tính năng cụ thể. Các lập trình viên này có thể sẽ có kỹ năng khác nhau và một số sẽ giỏi về những kỹ năng nhất định. Tuy nhiên khi sử dụng Scrum thì tất cả các thành viên của Development Team yêu cầu phải có khả năng làm việc thay thế vị trí của nhau và không ai chỉ chịu trách nhiệm phát triển một (hoặc một số) tính năng nhất định.
- **Scrum Master:** sẽ chịu trách nhiệm cho việc lên kế hoạch để phân công công việc, sắp xếp thứ tự ưu tiên giải quyết những công việc tồn đọng nào có trong **Backlog** trước, tổ chức các buổi họp với Product Owner để theo dõi tình hình và nắm thông tin cần thiết.

Các nguyên lý của phương pháp Agile

Nguyên lý:

1. Ưu tiên cao nhất của chúng tôi là thỏa mãn khách hàng thông qua việc chuyển giao sớm và liên tục các phần mềm có giá trị.
2. Chào đón việc thay đổi yêu cầu, thậm chí rất muộn trong quá trình phát triển. Các quy trình linh hoạt tận dụng sự thay đổi cho các lợi thế cạnh tranh của khách hàng.
3. Thường xuyên chuyển giao phần mềm chạy tốt tới khách hàng, từ vài tuần đến vài tháng, ưu tiên cho các khoảng thời gian ngắn hơn.
4. Nhà 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 sự hỗ trợ cần thiết, và tin tưởng họ để hoàn thành công việc.
6. Phương pháp hiệu quả nhất để truyền đạt thông tin tới nhóm phát triển và trong nội bộ nhóm phát triển là hội thoại trực tiếp.
7. Phần mềm chạy tốt là thước đo chính của tiến độ.
8. Các quy trình linh hoạt thúc đẩy phát triển bền vững. Các nhà tài trợ, nhà phát triển, và người dùng có thể duy trì một nhịp độ liên tục không giới hạn.
9. Liên tục quan tâm đến các kỹ thuật và thiết kế tốt để gia tăng sự linh hoạt.
10. Sự đơn giản – nghệ thuật tối đa hóa lượng công việc chưa xong – là căn bản.
11. Các kiến trúc tốt nhất, yêu cầu tốt nhất, và thiết kế tốt nhất sẽ được làm ra bởi các nhóm tự tổ chức.
12. Đội sản xuất sẽ thường xuyên suy nghĩ về việc làm sao để trở nên hiệu quả hơn, sau đó họ sẽ điều chỉnh và thay đổi các hành vi của mình cho phù hợp.

Các đặc trưng của phương pháp Agile

1. Tính lặp (Iterative)

Dự án sẽ được thực hiện trong các phân đoạn lặp đi lặp lại. Các phân đoạn (được gọi là Iteration hoặc Sprint) này thường có khung thời gian ngắn (từ một đến bốn tuần). Trong mỗi phân đoạn này, nhóm phát triển thực hiện đầy đủ các công việc cần thiết như lập kế hoạch, phân tích yêu cầu, thiết kế, triển khai, kiểm thử (với các mức độ khác nhau) để cho ra các phần nhỏ của sản phẩm. Các phương pháp agile thường phân chia mục tiêu thành các phần nhỏ với quá trình lập kế hoạch đơn giản và gọn nhẹ nhất có thể, và không thực hiện việc lập kế hoạch dài hạn.

2. Tính tiệm tiến (Incremental) và tiến hóa (Evolutionary)

Cuối các phân đoạn, nhóm phát triển thường cho ra các phần nhỏ của sản phẩm cuối cùng. Các phần nhỏ này thường là đầy đủ, có khả năng chạy tốt, được kiểm thử cẩn thận và có thể sử dụng ngay (gọi là potentially shippable product increment of functionality). Theo thời gian, phân đoạn này tiếp nối phân đoạn kia, các phần chạy được này sẽ được tích lũy, lớn dần lên cho tới khi toàn bộ yêu cầu của khách hàng được thỏa mãn. Khác với mô hình phát triển Thác nước – vốn chỉ cho phép nhìn thấy toàn bộ các chức năng tại thời điểm kết thúc dự án, sản phẩm trong các dự án agile lớn dần lên theo thời gian, tiến hóa cho tới khi đạt được trạng thái đủ để phát hành.

3. Tính thích ứng (hay thích nghi – adaptive)

Do các phân đoạn chỉ kéo dài trong một khoảng thời gian ngắn, và việc lập kế hoạch cũng được điều chỉnh liên tục, nên các thay đổi trong quá trình phát triển (yêu cầu thay đổi, thay đổi công nghệ, thay đổi định hướng về mục tiêu v.v.) đều có thể được đáp ứng theo cách thích hợp. Ví dụ, trong Scrum – phương pháp phổ biến nhất hiện nay – trong khi nhóm phát triển sản xuất ra các gói phần mềm, khách hàng có thể đưa thêm các yêu cầu mới, chủ sản phẩm (Product Owner) có thể đánh giá các yêu cầu này và có thể đưa vào làm việc trong phân đoạn (được gọi là Sprint trong Scrum) tiếp theo. Theo đó, các quy trình agile thường thích ứng rất tốt với các thay đổi.

4. Nhóm tự tổ chức và liên chức năng

Cấu trúc nhóm agile thường là liên chức năng(cross-functionality) và tự tổ chức(self-organizing). Theo đó, các nhóm này tự thực hiện lấy việc phân công công việc mà không dựa trên các mô tả cứng về chức danh (title) hay làm việc dựa trên một sự phân cấp rõ ràng trong tổ chức. Các nhóm này cộng tác với nhau để ra quyết định, theo dõi tiến độ, giải quyết các vấn đề mà không chờ mệnh lệnh của các cấp quản lý. Họ không làm việc theo cơ chế “mệnh lệnh và kiểm soát” (command and control).

Nhóm tự tổ chức có nghĩa là nó đã đủ các kỹ năng (competency) cần thiết cho việc phát triển phần mềm, do vậy nó có thể được trao quyền để tự ra quyết định, tự quản lý và tổ chức lấy công việc của chính mình để đạt được hiệu quả cao nhất.

5. Quản lý tiến trình thực tiễn (Empirical Process Control)

Các nhóm agile ra các quyết định dựa trên các dữ liệu thực tiễn thay vì tính toán lý thuyết hay các tiên giả định (prescription). Việc phân nhỏ dự án thành các phân đoạn ngắn góp phần gia tăng các điểm mốc để nhóm phát triển thu thập dữ kiện cho phép điều chỉnh các chiến lược phát triển của mình. Nói cách khác, Agile rút ngắn vòng đời phản hồi (short feedback life cycle) để dễ dàng thích nghi và gia tăng tính linh hoạt. Theo thời gian, các chiến lược này sẽ tiến gần đến trạng thái tối ưu, nhờ đó nhóm có thể kiểm soát được tiến trình, và nâng cao năng suất lao động.

6. Giao tiếp trực diện(face-to-face communication)

Về yêu cầu của khách hàng, agile khuyến khích nhóm phát triển trực tiếp nói chuyện với khách hàng để hiểu rõ hơn về cái khách hàng thực sự cần, thay vì phụ thuộc nhiều vào các loại văn bản.

Trong giao tiếp giữa nội bộ nhóm phát triển với nhau, thay vì một lập trình viên (thực hiện việc code) và một kỹ sư (thực hiện việc thiết kế) giao tiếp với nhau thông qua bản thiết kế, agile khuyến khích hai người này trực tiếp trao đổi và thống nhất với nhau về thiết kế của hệ thống và cùng nhau triển khai thành các chức năng theo yêu cầu.

Bản thân các nhóm agile thường nhỏ để đơn giản hóa quá trình giao tiếp, thúc đẩy việc cộng tác hiệu quả. Các dự án lớn muốn dùng agile thường phải phân thành nhóm nhỏ đồng thời làm việc với nhau hướng đến một mục tiêu chung. Việc này có thể đòi hỏi một số nỗ lực đáng kể trong việc điều phối các mức ưu tiên giữa các nhóm.

Các nhóm phát triển thường tạo ra các thói quen và cơ chế trao đổi trực diện một cách thường xuyên. Một trong các cơ chế thường thấy là các cuộc họp tập trung hằng ngày (daily meeting, Daily Scrum, standup meeting). Tại đây, tất cả các thành viên được yêu cầu nói rõ cho nhóm của mình biết mình đã làm gì, đang làm gì, sắp làm gì và đang gặp phải khó khăn nào trong quá trình làm việc. Khi cơ chế này được thực hiện hiệu quả,

nhóm luôn luôn nắm được tình hình công việc của mình, có các hành động thích hợp để vượt qua các trở lực để thực hiện thành công mục tiêu của dự án.

7. Phát triển dựa trên giá trị (value-based development)

Một trong các nguyên tắc cơ bản của agile là “phần mềm chạy tốt chính là thước đo của tiến độ”. Nguyên tắc này giúp nhóm dám loại bỏ đi các công việc dư thừa không trực tiếp mang lại giá trị cho sản phẩm.

Đề vận hành được cơ chế “làm việc dựa trên giá trị”, nhóm agile thường làm việc trực tiếp và thường xuyên với khách hàng (hay đại diện của khách hàng), cộng tác trực tiếp với họ để biết yêu cầu nào có độ ưu tiên cao hơn, mang lại giá trị hơn sớm nhất có thể cho dự án. Nhờ đó các dự án agile thường giúp khách hàng tối ưu hóa được giá trị của dự án. Một cách gần như trực tiếp, agile gia tăng đáng kể độ hài lòng của khách hàng.

Trình bày 3 giá trị cốt lõi của Scrum

BA GIÁ TRỊ CỐT LÕI CỦA SCRUM

Scrum là một phương pháp linh hoạt (agile), vì thế nó tuân thủ các nguyên tắc của Tuyên ngôn Agile (Manifesto for Agile Software Development). Ngoài ra Scrum hoạt động dựa trên ba giá trị cốt lõi, còn gọi là Ba chân của Scrum bao gồm Minh bạch, Thanh tra và Thích nghi.

Minh bạch (transparency): Trong Scrum, tính minh bạch được đề cao như là giá trị cốt lõi cơ bản nhất. Muốn thành công với Scrum, thông tin liên quan tới quá trình phát triển phải minh bạch và thông suốt. Các thông tin đó có thể là: tầm nhìn (vision) về sản phẩm, yêu cầu khách hàng, tiến độ công việc, các khúc mắc và rào cản v.v. Từ đó mọi người ở các vai trò khác nhau có đủ thông tin cần thiết để tiến hành các quyết định có giá trị để nâng cao hiệu quả công việc. Các công cụ và cuộc họp trong Scrum luôn đảm bảo thông tin được minh bạch cho các bên.

Thanh tra/Kiểm tra (inspection): Công tác thanh tra liên tục các hoạt động trong Scrum đảm bảo cho việc phát lộ các vấn đề cũng như giải pháp để thông tin đa dạng và hữu ích đến được với các bên tham gia dự án. Truy xét kỹ càng và liên tục là cơ chế khởi đầu cho việc thích nghi và các cải tiến liên tục trong Scrum.

Thích nghi (adaptation): Scrum rất linh hoạt như các phương pháp phát triển linh hoạt (agile software development) khác. Nhờ đó nó mang lại tính thích nghi rất cao. Dựa trên các thông tin minh bạch hóa từ các quá trình thanh tra và làm việc, Scrum có thể phản hồi lại các thay đổi một cách tích cực, nhờ đó mang lại thành công cho dự án.

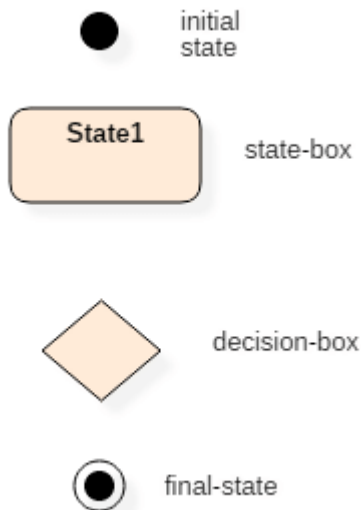
31. Sơ đồ trạng thái (State diagram)

Khái niệm

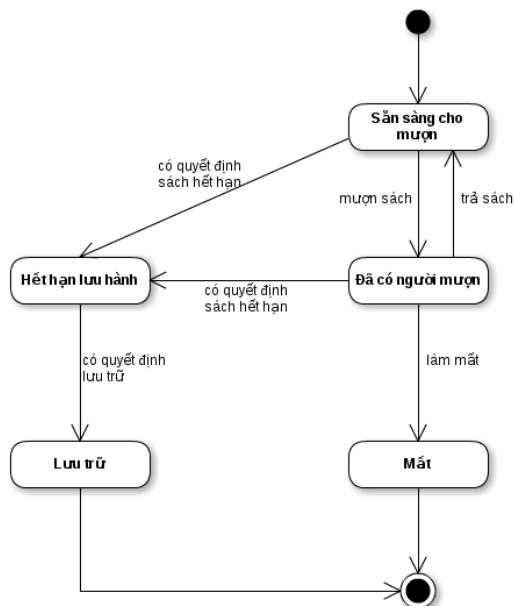
Biểu đồ trạng thái được sử dụng để mô tả trừu tượng về hoạt động của hệ thống. Hành vi này được phân tích và biểu diễn bằng một chuỗi các sự kiện có thể xảy ra ở một hoặc nhiều trạng thái có thể xảy ra. Bằng cách này "mỗi sơ đồ thường đại diện cho các đối

trạng của một lớp duy nhất và theo dõi các trạng thái khác nhau của các đối tượng của nó thông qua hệ thống".

Ký hiệu



Biểu đồ trạng thái thể hiện lớp Sách trong một hệ thống quản lý thư viện điện tử:



Biểu đồ trạng thái của lớp Sách trên có thể diễn tả lại như sau: Biểu đồ có 5 trạng thái chính là sẵn sàng cho mượn, đã có người mượn, hết hạn lưu hành, đã mượn, mất. và hai trạng thái phụ là trạng thái khởi tạo và trạng thái kết thúc.

1. Sách khởi tạo ở trạng thái "sẵn sàng cho mượn" .
2. Sách chuyển từ trạng thái "sẵn sàng cho mượn" sang trạng thái "Đã mượn" khi có người mượn sách.

3. Sách chuyển từ trạng thái "sẵn sàng cho mượn" sang trạng thái "Hết hạn lưu hành" khi có quyết định hết hạn lưu hành.
4. Sách "đã có người mượn" chuyển sang trạng thái "Hết hạn lưu hành" khi có quyết định hết hạn lưu hành.
5. Sách chuyển từ trạng thái "hết hạn lưu hành" sang trạng thái "lưu trữ" khi có quyết định lưu trữ .
6. Sách chuyển từ trạng thái "đã có người mượn" sang trạng thái "mất" khi làm mất.
7. Sách chuyển từ trạng thái "đã có người mượn" sang trạng thái "sẵn sàng cho mượn" khi trả sách.

32. Các mối quan hệ trong use case

a. Mối quan hệ extend

Xác định mối quan hệ giữa A và B, trong khi thực hiện A có thể kéo theo việc thực hiện B trong một điều kiện nào đó.

b. Mối quan hệ include

Là mối quan hệ giữa A và B, để thực hiện A cần phải thực hiện B trước

VD: Một từ điển Anh-Việt trực tuyến có chức năng **tra từ** cho phép người dùng nhập vào từ tiếng Anh để tìm nghĩa tiếng Việt. Từ điển còn có chức năng cho phép người dùng **thêm từ đã tra vào danh sách** các từ thường gặp. Theo mô hình use-case, mối liên hệ giữa chức năng tra từ và thêm từ đã tra vào danh sách là extend

33. Mối quan hệ giữa actor

Generalization

34. Mối quan hệ giữa actor và use case Association

35. Use case Scenario: Một chuỗi các hành động cụ thể được mô tả theo trình tự từ đầu cho đến kết thúc một use-case

36. Lược đồ use-case cho thấy Một số chức năng của hệ thống phần mềm và một số actor tương tác với các chức năng này

37. Actor là tác nhân bên ngoài của hệ thống, phát họa ai và cái gì sẽ tương tác với hệ thống và hệ thống cung cấp các chức năng gì

- a. Xây dựng Domain Model là để Xác định danh sách các đối tượng (ở mức khái niệm) làm ứng cử viên tham gia xây dựng hệ thống, các thuộc tính của đối tượng (nếu có) và mối quan hệ giữa các đối tượng, được xây dựng trong giai đoạn Phân tích yêu cầu

38. Trong qui trình phát triển phần mềm, giai đoạn phân tích tập trung vào những công việc trả lời cho câu hỏi What
39. Trong qui trình phát triển phần mềm, giai đoạn thiết kế tập trung vào những công việc trả lời cho câu hỏi How
Giai đoạn phân tích và giai đoạn thiết kế trong qui trình phát triển phần mềm khác nhau ở chỗ
Phân tích tập trung vào việc tìm hiểu về yêu cầu khách hàng và thiết kế tập trung vào việc đưa ra giải pháp để thực hiện các yêu cầu đó.
Phân tích hướng đối tượng là gì (Finding and describing the objects or concepts in the problem domain)
Thiết kế hướng đối tượng là gì (Defining logical software objects and how they collaborate to fulfill the requirements)
40. Model là gì? (A model is a complete description of a system from a particular perspective)
41. Unified Process là gì (Unified Process is an agile approach, not a series of steps for constructing a software product)
42. Phát triển lặp (Iterative development) mang lại lợi ích gì
- Tỷ lệ lỗi thấp
 - Giảm nhẹ rủi ro
 - Thông tin được phản hồi sớm dẫn đến hệ thống được điều chỉnh phù hợp hơn với nhu cầu thực sự của các bên liên quan
 - Bài học được rút ra trong lần lặp trước có thể được áp dụng cho lần lặp sau để cải thiện qui trình thực hiện tốt hơn
43. Lược đồ trình tự thể hiện
- a. Sự phân chia công việc giữa các đối tượng để cùng hoàn thành một công việc nào đó
 - b. Sự trao đổi thông điệp giữa các đối tượng để cùng hoàn thành một công việc nào đó
 - c. Sự cộng tác giữa các đối tượng để cùng hoàn thành một công việc nào đó
44. Ưu điểm của lược đồ trình tự là Trình tự thực hiện của các thông điệp được thể hiện rõ ràng
45. Nhược điểm của lược đồ cộng tác là Khó thấy được trình tự thực hiện của các thông điệp
46. Lược đồ trạng thái và lược đồ hoạt động được dùng để mô tả Business rules của hệ thống
47. Ký hiệu “Fork” trong lược đồ hoạt động biểu diễn nhiều hoạt động đang diễn ra Đồng thời

Bài tập 1

Trung tâm mở nhiều lớp học khác nhau. Học viên sẽ đăng ký thành viên thông qua trang web với các thông tin tên đăng nhập (khóa), mật khẩu, họ, tên, ngày sinh, địa chỉ, số điện thoại liên lạc và email.

Sau khi đăng nhập thành công học viên sẽ tìm và chọn lớp mà mình muốn tham gia để thực hiện đăng ký. Học viên sẽ điền 1 phiếu đăng ký gồm có số phiếu, ngày đăng ký, thông tin học viên và thông tin lớp để đăng ký học.

Nhân viên giáo vụ sẽ tiếp nhận xử lý và gửi email xác nhận đăng ký thành công, đồng thời trên phiếu đăng ký sẽ có thêm thông tin phiếu này do ai xử lý.

Mỗi lớp sẽ do một giáo viên phụ trách, giáo viên cũng được xem như là nhân viên của trung tâm. Trong quá trình học giáo viên sẽ chấm và nhập điểm bài tập và một bài kiểm tra kết thúc lớp. Một giáo viên sẽ dạy nhiều lớp.

Trưởng trung tâm sẽ quản lý các lớp học, khi mở 1 lớp học mới trưởng trung tâm sẽ nhập thông tin 1 lớp học gồm mã lớp học, tên lớp học, số giờ và học phí, ngày bắt đầu, ngày kết thúc, thời gian học (ca 1 2-4-6, ca 2 3-5-7, ...) do một giáo viên phụ trách. Đồng thời trưởng trung tâm còn quản lý thông tin nhân viên (thêm 1 nhân viên mới (mã nhân viên, họ tên, account, pass, vị trí công việc), sửa thông tin nhân viên, xóa nhân viên)

Định kỳ và cuối tháng, trưởng trung tâm sẽ tổng hợp và in báo cáo (nếu có) quá trình hoạt động của trung tâm.

Một Gara xe hơi muốn xây dựng một hệ thống để quản lý hoạt động sửa chữa, bảo trì xe. Gara được tổ chức như sau :

Gara gồm nhiều thợ và có nhiều hạng mục công việc sửa chữa. Quản lý gara sẽ là người thêm, sửa, xóa thông tin cần về thợ: Mã số (duy nhất cho mỗi thợ), họ tên và thuộc một tổ. Tổ trưởng của mỗi tổ là một trong số các người thợ của tổ đó.

Quản lý gara cũng là người cập nhật (thêm, sửa, xóa) các mục công việc sửa chữa: Mã công việc, tên chi tiết công việc, giá công thợ.

Mỗi lần sửa xe hơi khách hàng mang xe đến gara. Nhân viên quản lý gara sẽ lập hợp đồng sửa chữa cho khách. Trên hợp đồng sẽ ghi nhận các thông tin Một khách hàng có thể ký hợp đồng sửa chữa nhiều xe khác nhau . Các thông tin cần lưu trữ về khách hàng và hợp đồng gồm :

Mỗi hợp đồng đều có Mã số hợp đồng, Ngày lập hợp đồng, các xe sửa, giá trị hợp đồng, ngày nghiệm thu. (biển số xe là số đăng bộ của xe đem đến sửa, do phòng

CSGT đường bộ cấp, nếu xe đổi chủ thì xem như là một xe khác), các công việc sửa chữa. Mỗi công việc tương ứng với 1 hạng mục sửa chữa và tính giá = phụ kiện + công thợ. Các xe sửa ghi nhận lại một số thông tin: biển số xe, tên xe, loại xe.

Sau khi ký hợp đồng với khách gara sẽ phân công thợ phụ trách các hạng mục của hợp đồng. Một hạng mục công việc trên hợp đồng do 1 thợ đảm nhận. Một thợ có thể đảm nhận nhiều hạng mục trên hợp đồng

Mỗi khách hàng đều có Mã số khách hàng, họ tên, địa chỉ, điện thoại.

Mỗi khách hàng thanh toán tiền của một hợp đồng sửa chữa làm nhiều lần trước hay sau khi nghiệm thu (trong cùng ngày hay khác ngày) phải lập một phiếu thu và một phiếu bảo hành. Các thông tin cần lưu trữ về phiếu thu và phiếu bảo hành gồm :

- Mỗi phiếu thu đều có mã số phiếu thu, ngày phát hành phiếu và số tiền thu, mã số hợp đồng, đợt thu.

- Mỗi phiếu bảo hành đều có Mã số phiếu bảo hành, ngày bắt đầu bảo hành, thời gian

bảo hành, mã số hợp đồng, xe bảo hành, phụ kiện bảo hành. Gara có 1 danh mục các phụ kiện, mỗi phụ kiện có các thông tin: mã phụ kiện, tên phụ kiện, nhà sản xuất, ngày sản xuất, giá.

Bài toán thư viện

Thư viện giới thiệu sách đến đọc giả thông qua website. Thông tin về mỗi sách gồm: mã sách, tên sách, tên tác giả, nhà xuất bản, năm xuất bản, số trang, tóm tắt nội dung. Để thuận tiện cho việc tra cứu thì sách sẽ được phân loại theo từng chủ đề. Mỗi chủ đề gồm mã chủ đề, tên chủ đề

Để trở thành đọc giả của thư viện bạn đọc sẽ điền phiếu thông tin đăng ký làm đọc giả. Phiếu đăng ký thông tin đọc giả phải điền đầy đủ các thông tin: họ tên, giới tính, ngày sinh, địa chỉ, số điện thoại, email. Thư viện sẽ cấp 1 mã số duy nhất cho từng đọc giả.

Ở mỗi đợt nhập sách, mỗi đầu sách sẽ được thư viện nhập về 1 số lượng cuốn sách. Mỗi cuốn sách được nhập về sẽ được đánh mã số (mã cuốn sách), thư viện cũng ghi nhận lại ngày nhập và giá nhập của từng cuốn sách.

Khi đọc giả đến thư viện mượn sách, đọc giả sẽ đến các kệ sách chọn các cuốn sách mượn rồi mang các cuốn sách này đến quầy mượn sách. Nhân viên thư viện sẽ ghi nhận lại lượt mượn sách của đọc giả gồm: lượt mượn (số thứ tự tăng dần), mã đọc giả, mã từng cuốn sách, ngày mượn).

Khi trả sách cho thư viện, đọc giả mang cuốn sách đến quầy trả sách. Nhân viên thư viện sẽ ghi nhận lại đọc giả đã trả cuốn sách này và ngày trả sách.