

module-1

① what is SDLC?

⇒ Software Development Life Cycle.

SDLC is a structure imposed on the development of software product that define the process for planning, implementation, testing, documentation, deployment and ongoing maintenance and support.

SDLC phases

Requirements Collection / Gathering

↓ (Establish customer needs)

Analysis

↓ (Model and specify the requirements "what")

Design

↓ (Model and specify a solution - "why")

Implementation

↓ (Construct a solution In software)

Testing

↓ (Validate the solution against the requirements)

Maintenance

↓ (Repair defects and adapt the solution to the new requirements)

② what is Testing?

⇒ Software testing is a process used to

identify the correctness, completeness and quality of developed computer software

③ what is agile methodology?

⇒ Agile model believes that every project needs to be handled differently and the existing method need to be tailored to best suit the project requirements. In agile the tasks are divided to time boxes (small time frames) to deliver specific features for a release.

→ Agile thought process had started early in the software development and started becoming popular with time due to its flexibility and adaptability.

④ what is SRS?

⇒ Software Requirement Specification

→ SRS is a complete description of the behavior of the system to be developed.

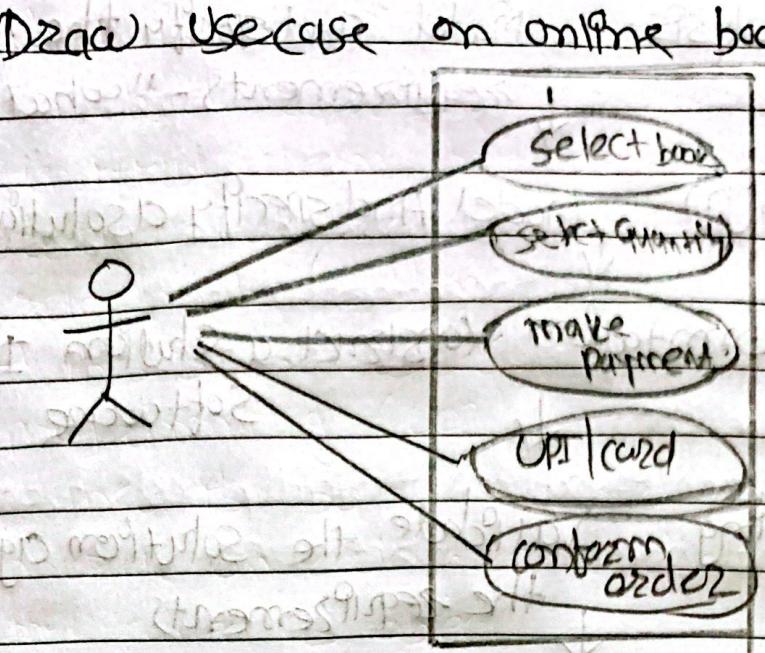
→ Types of Requirements

① Customer Requirements

② Functional Requirements

③ Non functional Requirements

(12) Draw Use case on online book shopping



(13) Draw Use case on online bill payment system



(Buytm)



(14) write SDLC phases with basic introduction



Requirements
collection/gathering

Establish customer
need



Analysis Model and specify the requirements - "what"

Design Model And specify a solution why.

Implementation Construct a Solution In Software

Testing Validate the solution against the requirements

Maintenance Repair defects and adapt the solution to the new requirements.

(15)

Explain phases of the waterfall model

→ The waterfall is unsuitable for many reason

→ Requirements must be frozen too early in the life cycle

+ Requirements are validated too late

Requirement Collection

Analysis

Design

Implementation

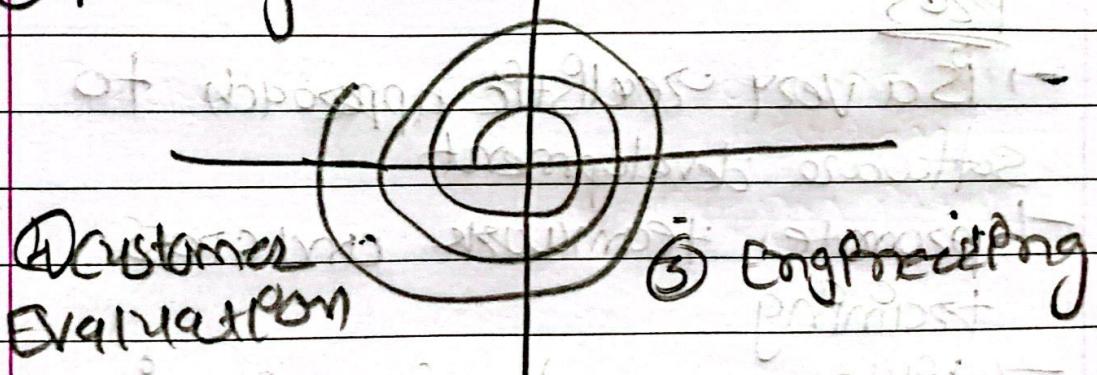
Testing

Maintenance

(16) write phases of spiral model

- ⇒ Spiral model is very widely used in the software industry as it is sync with the natural development process of any product
- For medium to high-risk projects
 - When costs there are a budget constraint and risk evaluation is important

① Planning ② Risk analysis



(17) write agile manifesto principles

- ⇒ Agile SDLC model is a combination of iterative and incremental process models with focus on process adaptability and customer satisfaction by rapid delivery of working software product
- Agile methods break the product into small incremental builds
 - These builds are provided for iteration which empirically lasts from

(18)

Explain working methodology of agile model and also write pros and cons

- ⇒ Agile methods break the product into small incremental builds
 - These builds are planned in iterations
 - Each iteration typically lasts from about one to three weeks

Pros

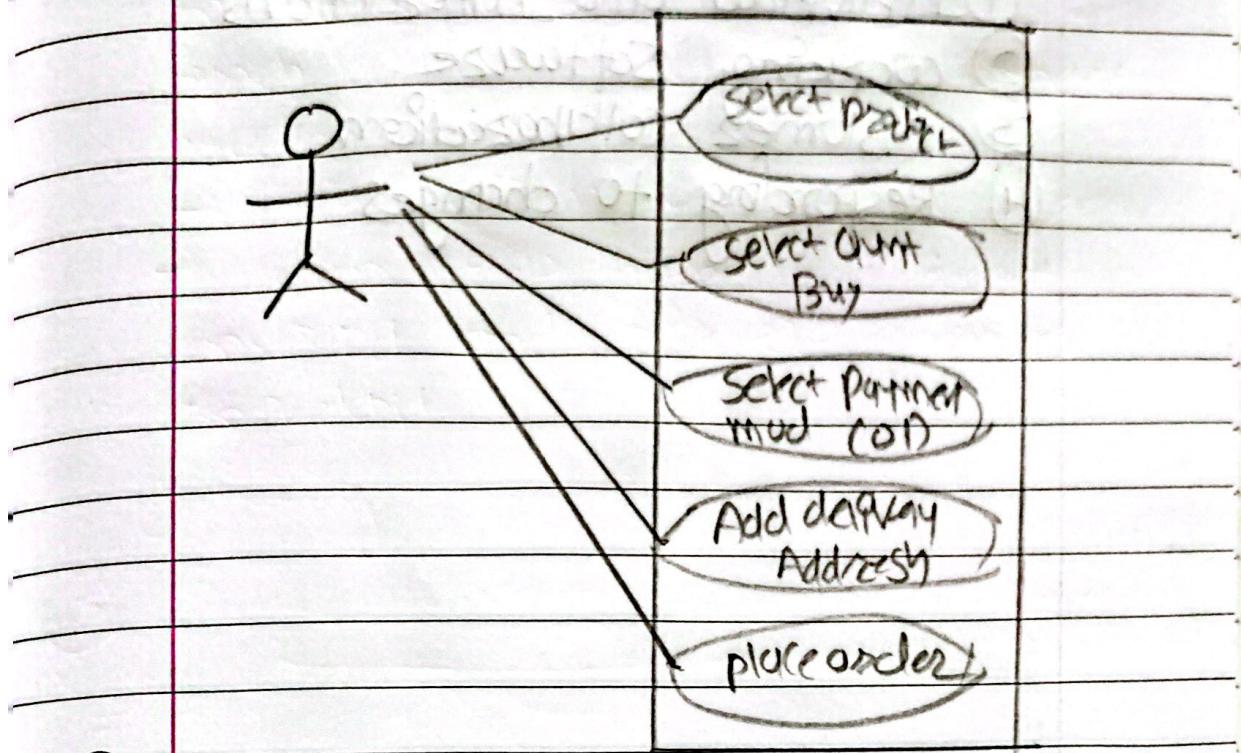
- It's a very realistic approach to software development
- promotes teamwork and cross training
- little or no planning required
 - Easy to manage
- gives flexibility to develop

Cons

- Not suitable for handling complex dependencies
- more risk of sustainability, maintainability and extensibility.
- There is very high dependency dependency since there is minimum documentation generated

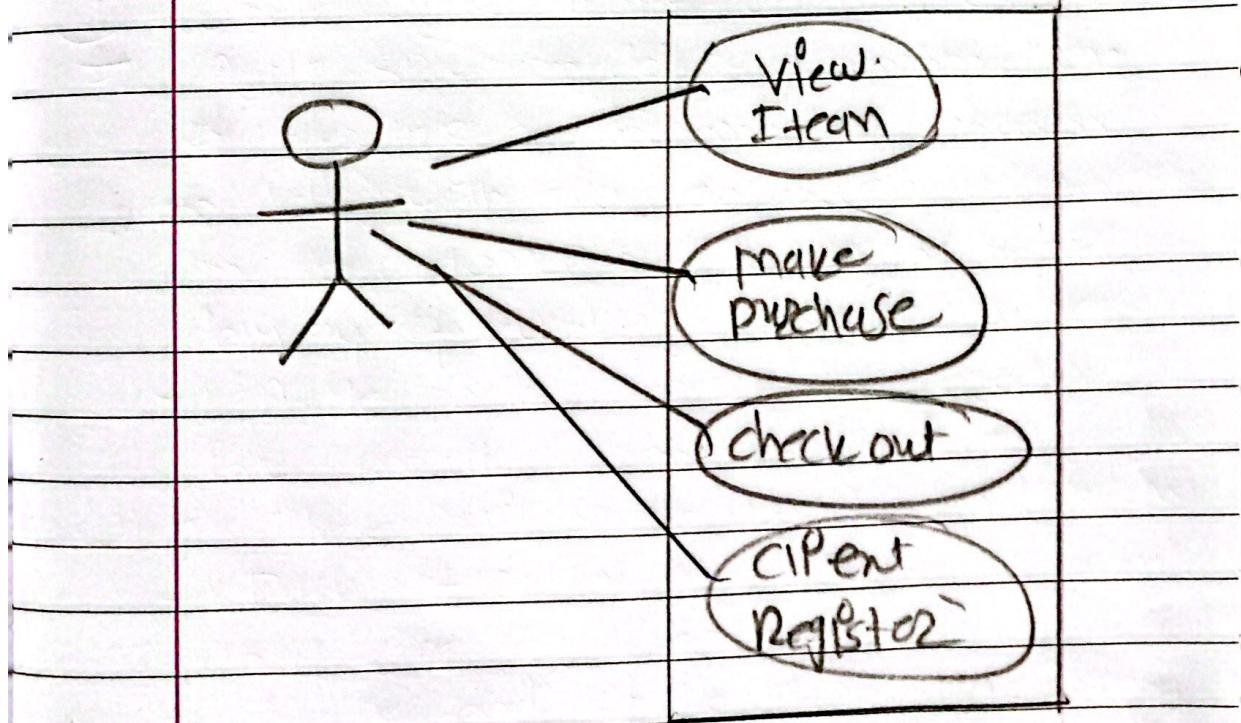
(14)

Draw use case on online shopping product using CO.



(20)

Draw use case on online shopping product using payment gateway.



17)

write agile manifesto principles
there are four manifesto in Agile

- ① Individual and Interactions
- ② working software.
- ③ customer collaboration
- ④ responding to changes

⑤ what is OOPS?

⇒ Object Oriented Programming.

OOP is Black Box

→ OOP is a computer programming model that organizes software design around data or objects rather than functions and logic. An object can be defined as a data field that has unique attributes and behavior.

there are four types of oop

① Encapsulation

② Inheritance

③ Polymorphism

④ Data Abstraction

⑥ what is Basic Concepts of OOPS?

⇒ OOP is an approach or a programming pattern where the programs are structured around objects rather than functions and logic. It makes the code partitioned into two memory areas. i.e. data and functions and helps make the code flexible and modular.

(7) What is an object?

⇒ An object represent an individual identifiable item, unit or entity either real or abstract with a well defined role in the problem domain.

→ This is the basic unit of object oriented programming (oop).

→ That is both data and function that operate on data are bundled as a unit called as object.

Object = Data + methods

(8) What is class?

⇒ Class is a blueprint for an object.

→ Class represents an abstraction of the object and abstracts the properties and behaviors of that object.

→ An object is a particular instance of a class which has actual existence and there can be many objects for a class.

(9) What is Encapsulation?

⇒ Encapsulation is the practice of including in an object everything it needs hidden from other objects. The internal state is usually not accessible by other objects.

⑩

what is Inheritance?

⇒ Inheritance means that one class inherits the characteristics of another class. This is also called a 'Is a' relationship.

→ In general, Java supports single-parent, multiple-children inheritance and multi-level inheritance.

① Single-level

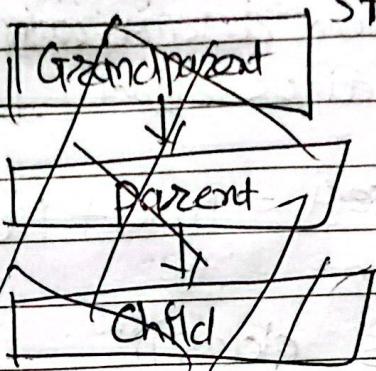
② Multi - "

③ Multiple "

④ Hierarchical "

⑤ Hybrid "

STYLES OF INHERITANCE



⑪

what is polymorphism?

⇒ polymorphism means "having many forms"

→ It allows different objects to respond to the same message. In different ways the response specific to the type of the object.

→ Most important aspect of an object is its behaviour.

→ A behaviour is exhibited by sending a message to the object by calling a method.

→ Two types of polymorphism

① Compile time polymorphism (overloading)

② Runtime polymorphism (overriding)