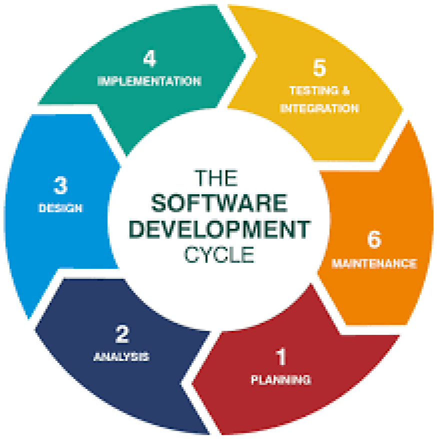
1) What is SDLC

SDLC is a step by step process to develop any software with high quality, within the budget and in shortest possible time.

timeline + budget + quality = SDLC



2) What is software testing?

Process to verify and validate the developed software.

Software Testing is the process of checking that a software program works as expected
free of bugs before release.

3) What is agile methodology?

It is iterative, incremental, flexible, collaborative, and adjustable.

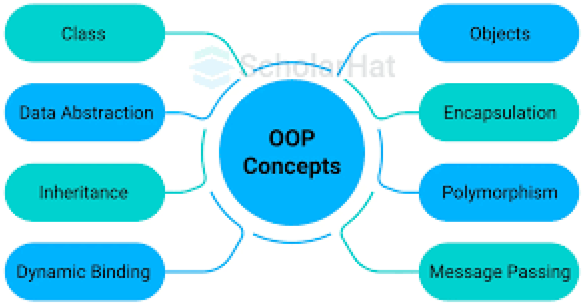
Able to move quickly and easily.
Agile methodology shortens project timelines by delivering working software in incremental releases, reducing the risk of long development cycles.

4) What is SRS ?

A software requirements specification is a description of a software system to be developed. It is modeled after the business requirements specification.

5) What is oops

Object-Oriented Programming System (OOP) is a way of organizing code by grouping data and functions (actions) into "objects", which can then interact with each other.



6) Write Basic Concepts of oops

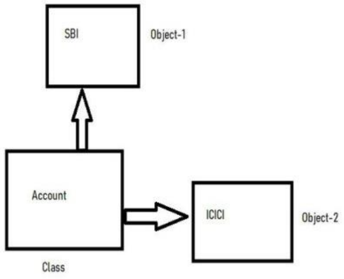
Object-oriented programming aims to implement real-world entities like inheritance, hiding, polymorphism, etc in programming.

An object is a component of a program that knows how to perform certain actions and how to interact with other elements of the program. Objects are the basic units of object-oriented programming.

A simple example of an object would be a person. Logically, you would expect a person to have a name.

7) What is object

In Object-Oriented Programming (OOP), an object is an instance of a class that combines data (attributes) and methods (functions) into a single entity.



8) What is class

In **Object-Oriented Programming (OOP)**, a **class** is a blueprint or template used to create objects. It defines the **attributes** (data) and **methods** (functions) that the objects created from the class will have.

9) What is encapsulation

Encapsulation is **a programming concept that bundles data and methods into a single unit**

Encapsulation is one of the core principles of Object-Oriented Programming (OOP). It refers to the practice of restricting direct access to certain parts of an object and controlling how the data is accessed or modified.

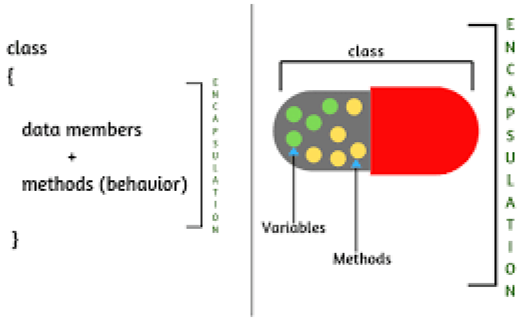
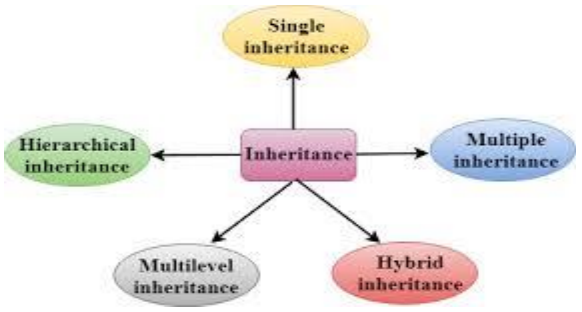


Fig: Encapsulation

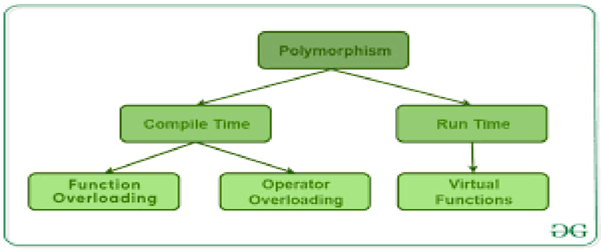
10) What is inheritance

Inheritance is an Object-Oriented Programming (OOP) concept where a new class (child/subclass) derives the properties and behaviors of an existing class (parent/superclass).

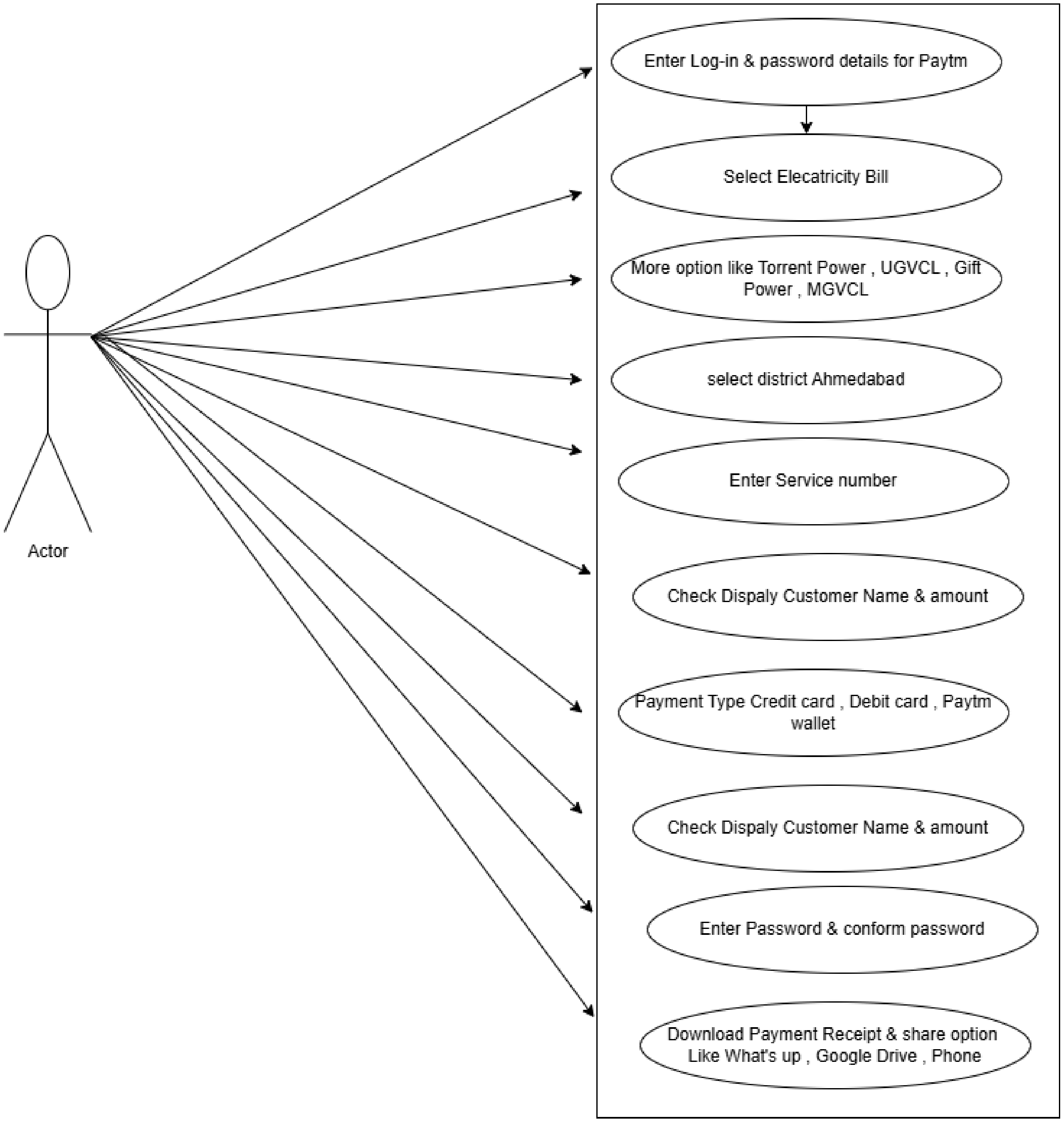


11) What is polymorphism

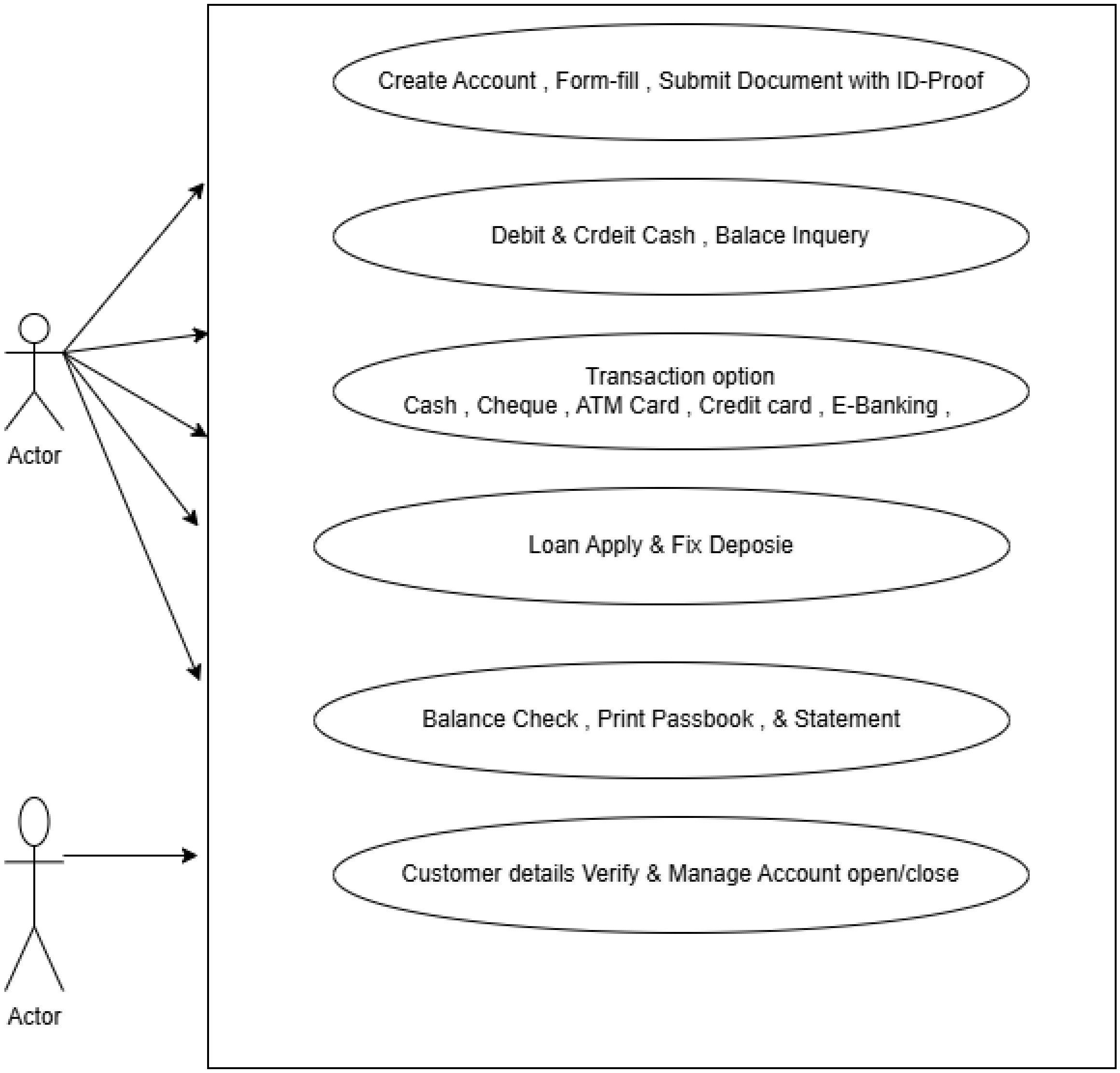
In object-oriented programming (OOP), polymorphism means "many forms" and allows objects of different classes to be treated as objects of a common type, enabling flexible and reusable code.



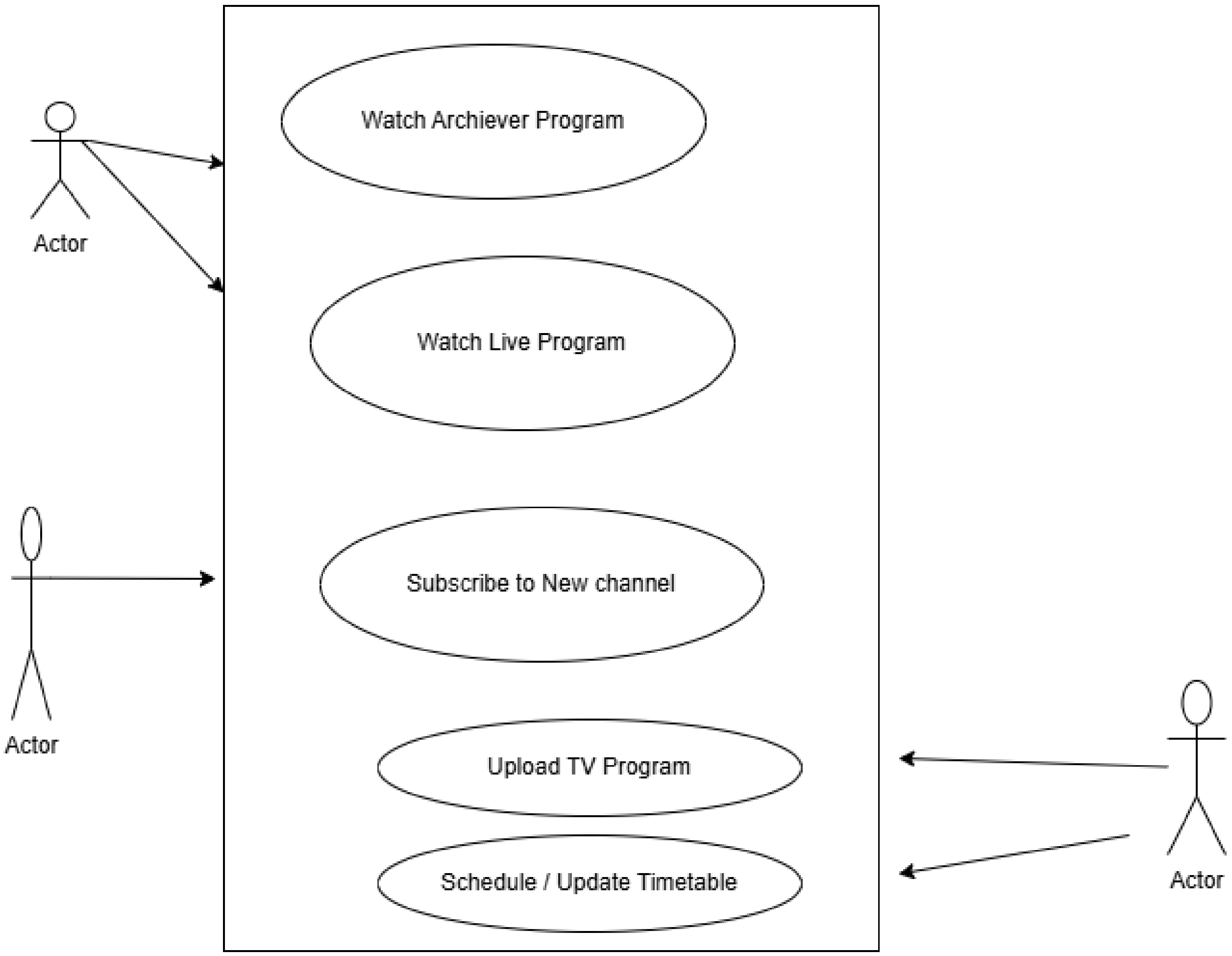
12) Draw Usecase on online bill payment system (paytm)


Actor
Enter Log-in & password details for Paytm
Select Elecatricity Bill
More option like Torrent Power , UGVCL , Gift Power , MGVCL
select district Ahmedabad
Enter Service number
Check Dispaly Customer Name & amount
Payment Type Credit card , Debit card , Paytm wallet
Check Dispaly Customer Name & amount
Enter Password & conform password
Download Payment Receipt & share option Like What's up , Google Drive , Phone

13) Draw Usecase on banking system for customers.


Create Account , Form-fill , Submit Document with ID-Proof
Debit & Crdeit Cash , Balace Inquery
Transaction option
Cash , Cheque , ATM Card , Credit card , E-Banking ,
Loan Apply & Fix Deposie
Balance Check , Print Passbook , & Statement
Customer details Verify & Manage Account open/close
Actor
Actor

14) Draw Usecase on Broadcasting System.



15) Write SDLC phases with basic introduction

1)) Planning / Requirement Gathering (What)

Problems can be arised while gathering the req. :

- lack of clarity :
- req. confusion : functional vs. non-functional
- req. amalgamation (together) :

2)) Analysis (How)

Prepare the SRS : System/software requirement specification.

A software requirements specification (SRS) is a complete description of the behavior of the system to be developed.

Types of Requirements:

-customer req
-functional req.
-nonfunctional req.

3)) Designing : (view the system - prototype)
multiple designs for the product architecture are present in the Design Document Specification (DDS)

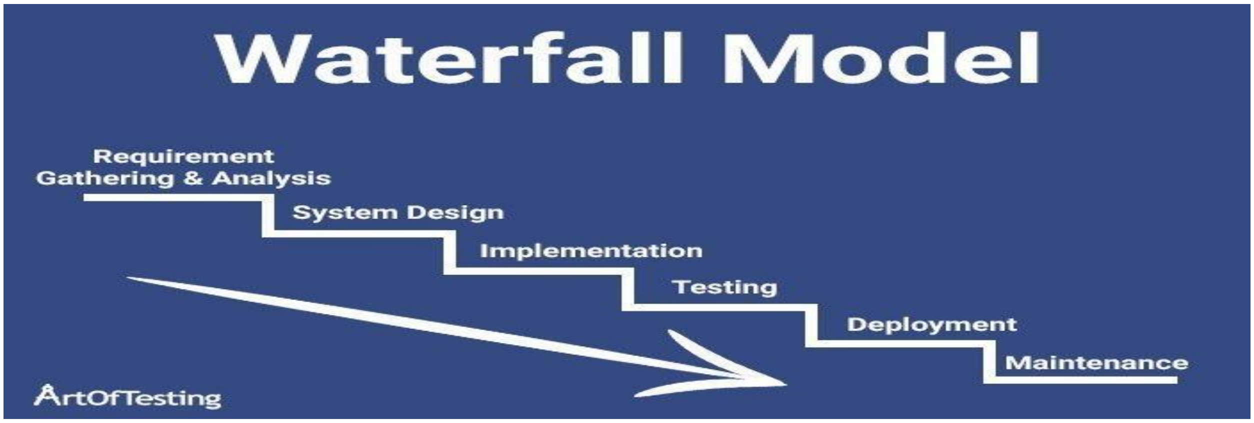
4)) Implementation / building /coding (h/w, s/w)
The actual coding of the software takes place in this phase

5)) Testing (QA)
The developed software is tested for bugs and defects.

6)) Maintenance
the software is monitored and updated based on user feedback.
After retrieving beneficial feedback, the company releases it as it is or with auxiliary improvements to make it further helpful for the customers.

16) Explain Phases of the waterfall model

requirements, design, implementation, verification, and maintenance



Requirements: The first phase involves gathering requirements from stakeholders and analyzing them to understand the scope and objectives of the project.

Design: Once the requirements are understood, the design phase begins. This involves creating a detailed design document that outlines the software architecture, user interface, and system components.

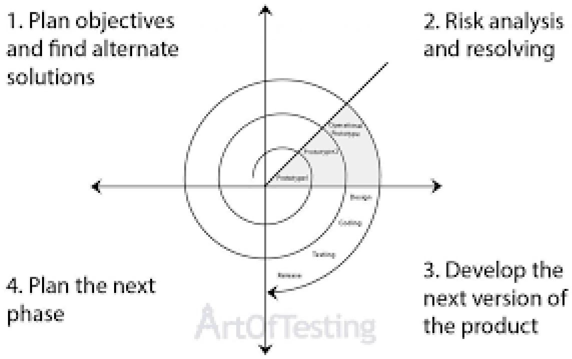
Development: The Development phases include implementation involves coding the software based on the design specifications. This phase also includes unit testing to ensure that each component of the software is working as expected.

Testing: In the testing phase, the software is tested as a whole to ensure that it meets the requirements and is free from defects.

Deployment: Once the software has been tested and approved, it is deployed to the production environment.

Maintenance: The final phase of the Waterfall Model is maintenance, which involves fixing any issues that arise after the software has been deployed and ensuring that it continues to meet the requirements over time.

17) Write phases of spiral model


1. Plan objectives and find alternate solutions
2. Risk analysis and resolving
Operational Prototype
Prototype2
Prototype1
Design
Coding
Testing
Release
4. Plan the next phase
3. Develop the next version of the product
ArtOfTesting

The Spiral Model is a risk-driven model, meaning that the focus is on managing risk through multiple iterations of the software development process. It consists of the following phases:

Objectives Defined: In first phase of the spiral model we clarify what the project aims to achieve, including functional and non-functional requirements.

Risk Analysis: In the risk analysis phase, the risks associated with the project are identified and evaluated.

Engineering: In the engineering phase, the software is developed based on the requirements gathered in the previous iteration.

Evaluation: In the evaluation phase, the software is evaluated to determine if it meets the customer’s requirements and if it is of high quality.

Planning: The next iteration of the spiral begins with a new planning phase, based on the results of the evaluation.

The Spiral Model is often used for complex and large software development projects, as it allows for a more flexible and adaptable approach to software development. It is also well-suited to projects with significant uncertainty or high levels of risk.

18) Write agile manifesto principles.

1) Satisfy customers through early, continuous improvement and delivery.

2) Welcome changing requirements, even late in the project.

3) Deliver value frequently.

4) Break the silos of your projects.

5) Build projects around motivated individuals.

6) The most effective way to communicate is face-to-face.

7) Working software is the primary measure of progress.

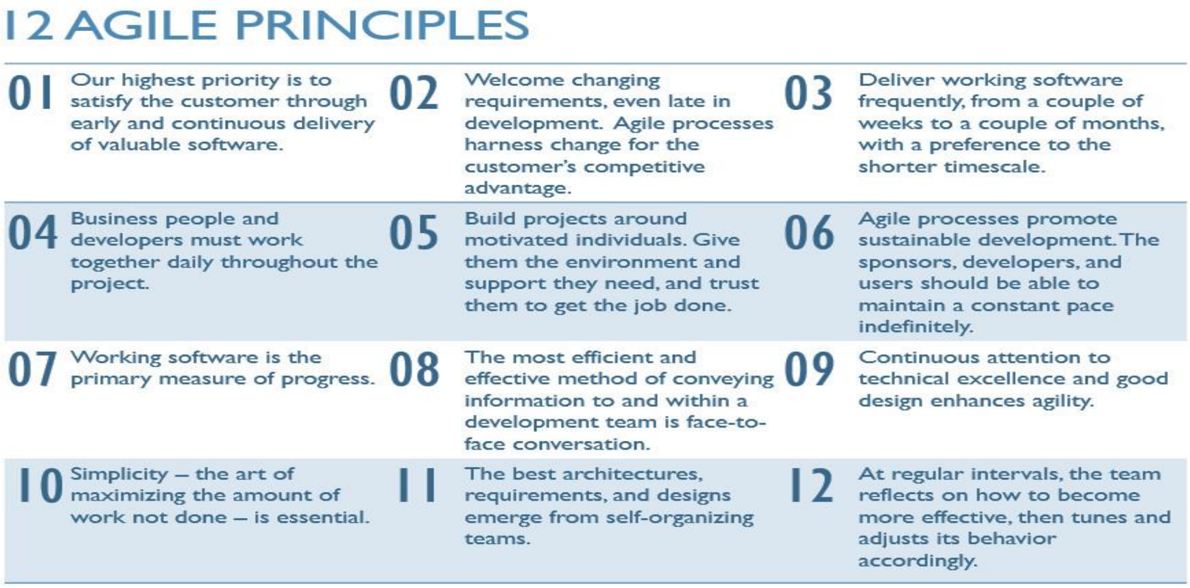
8) Maintain a sustainable working pace.

9) Continuous excellence enhances agility.

10) Simplicity is essential.

11) Self-organizing teams generate the most value.

12) Regularly reflect and adjust your way of work to improve effectiveness.

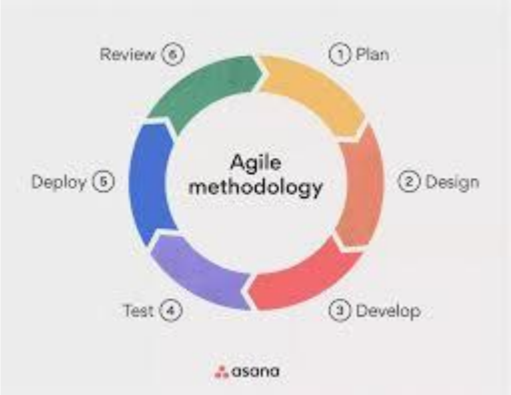


12 AGILE PRINCIPLES

01	Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.	02	Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.	03	Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
04	Business people and developers must work together daily throughout the project.	05	Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.	06	Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
07	Working software is the primary measure of progress.	08	The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.	09	Continuous attention to technical excellence and good design enhances agility.
10	Simplicity – the art of maximizing the amount of work not done – is essential.	11	The best architectures, requirements, and designs emerge from self-organizing teams.	12	At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

19) Explain working methodology of agile model and write pros and cons.

Pros



Agile methodology is a project management framework that breaks projects down into several dynamic phases, commonly known as sprints.

Pros

Is a very realistic approach to software development

Promotes teamwork and cross training.

Functionality can be developed rapidly and demonstrated.

Resource requirements are minimum.

Suitable for fixed or changing requirements

Delivers early partial working solutions.

Good model for environments that change steadily.

Minimal rules, documentation easily employed.

Enables concurrent development and delivery within an overall

planned context.

Little or no planning required

Easy to manage & Gives flexibility to developers

Cons

Not suitable for handling complex dependencies.

More risk of sustainability, maintainability and extensibility.

Strict delivery management dictates the scope, functionality to be

delivered, and adjustments to meet the deadlines.

Depends heavily on customer interaction, so if customer is not clear, team can be driven in the wrong direction.

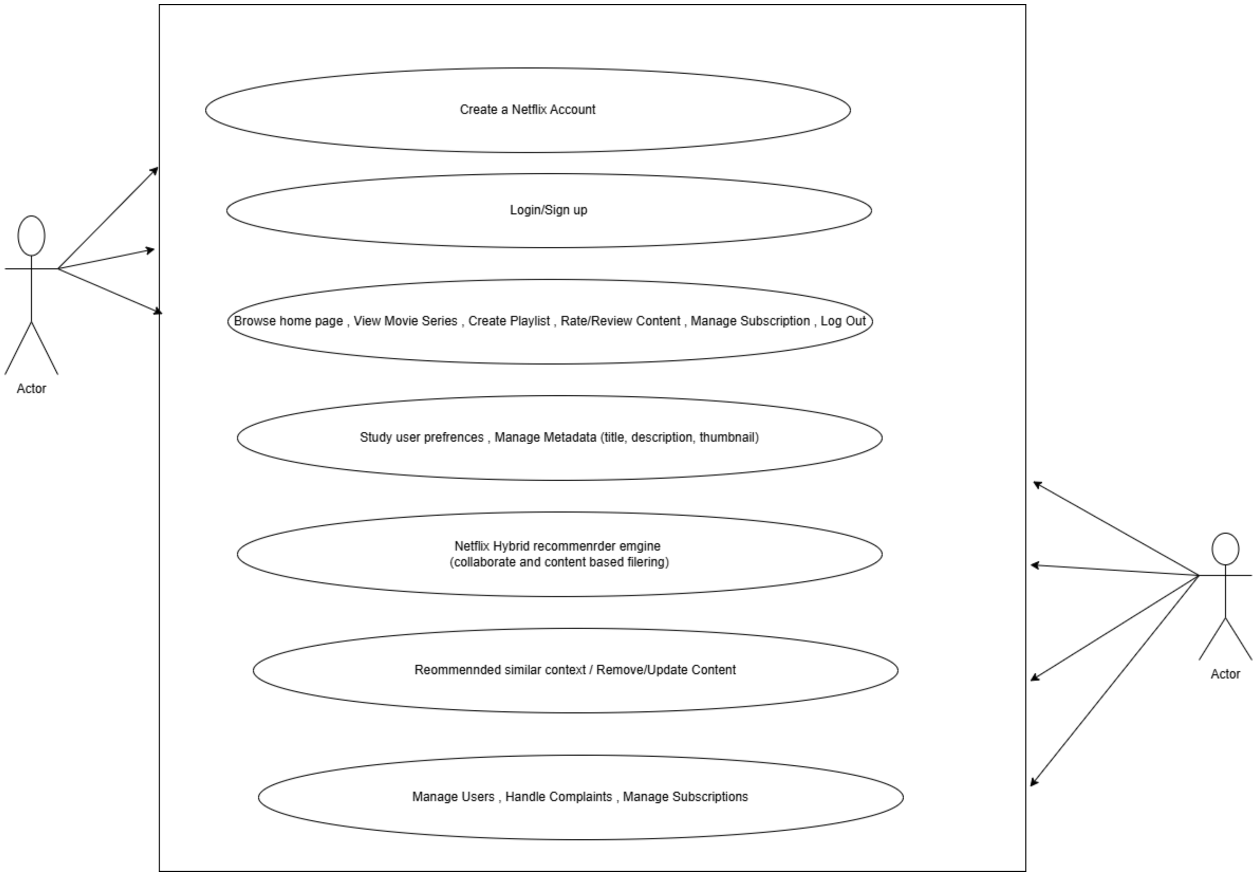
There is very high individual dependency, since there is minimum

documentation generated.

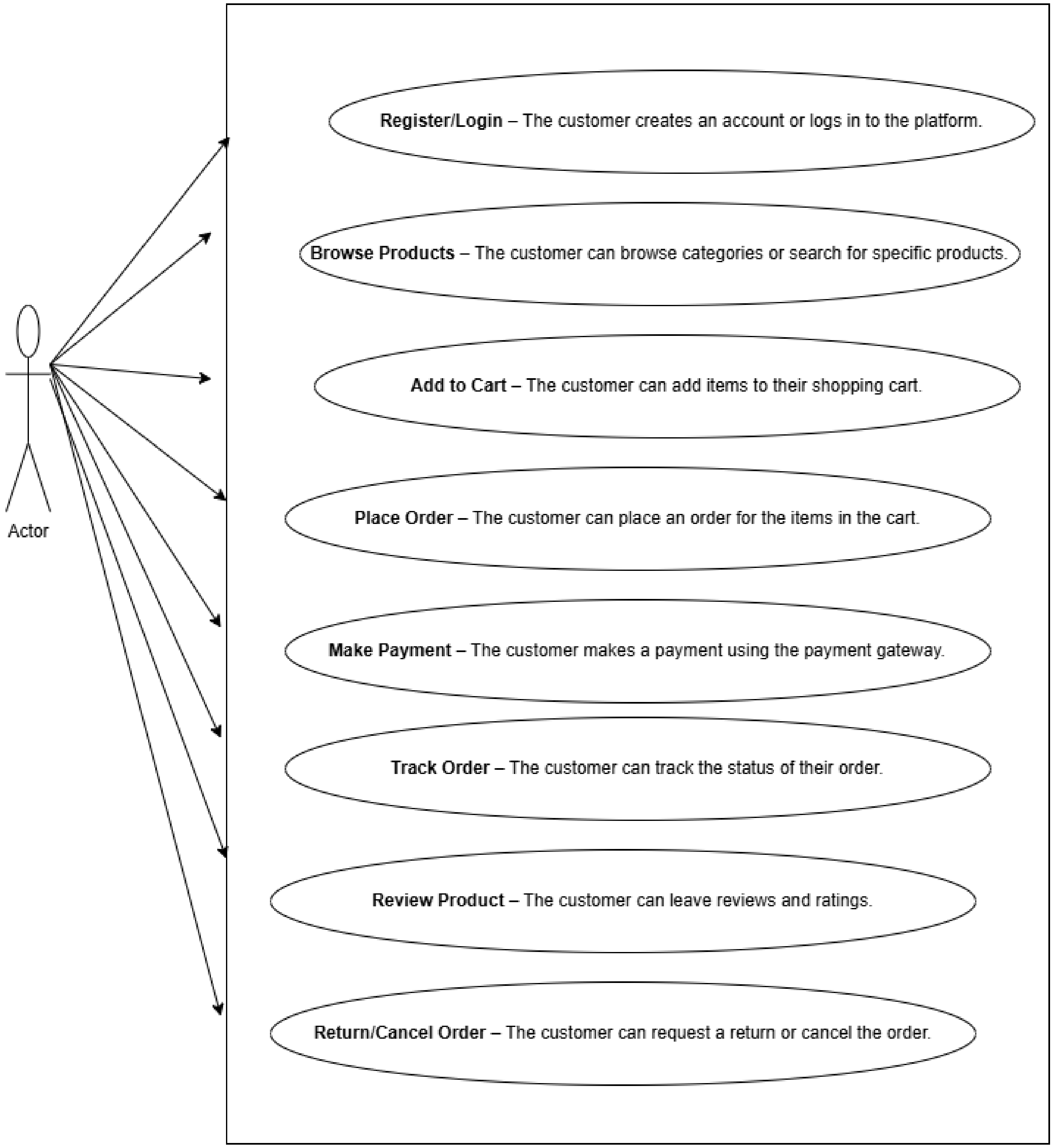
Transfer of technology to new team members may be quite challenging

due to lack of documentation.

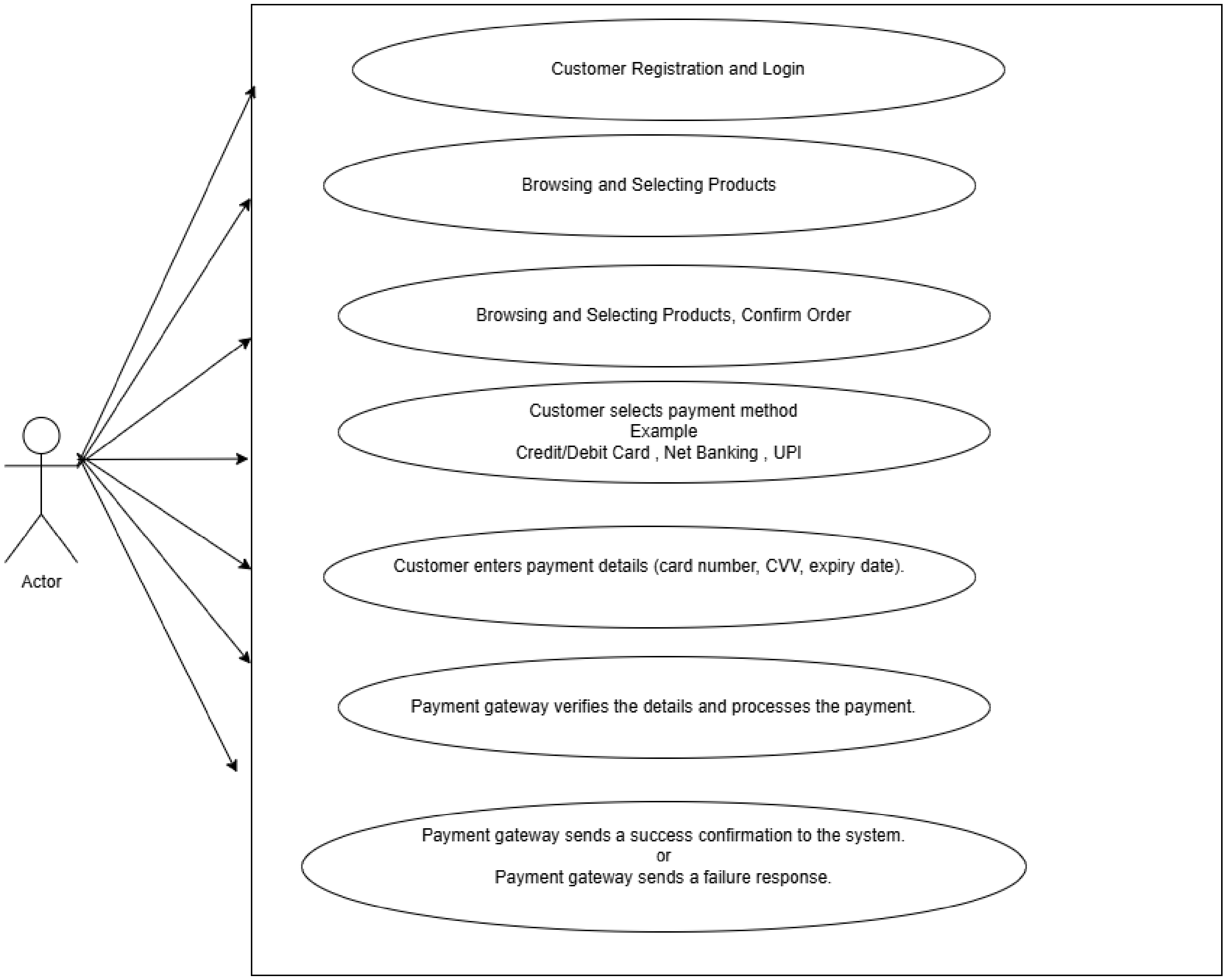
20) Draw usecase on OTT Platform.


Create a Netflix Account
Login/Sign up
Browse home page , View Movie Series , Create Playlist , Rate/Review Content , Manage Subscription , Log Out
Study user prefrences , Manage Metadata (title, description, thumbnail)
Netflix Hybrid recommenrder emgine
(collaborate and content based filering)
Reommennded similar context / Remove/Update Content
Manage Users , Handle Complaints , Manage Subscriptions
Actor
Actor

21) Draw usecase on E-commerce application.


Register/Login – The customer creates an account or logs in to the platform.
Browse Products – The customer can browse categories or search for specific products.
Add to Cart – The customer can add items to their shopping cart.
Place Order – The customer can place an order for the items in the cart.
Make Payment – The customer makes a payment using the payment gateway.
Track Order – The customer can track the status of their order.
Review Product – The customer can leave reviews and ratings.
Return/Cancel Order – The customer can request a return or cancel the order.
Actor

22) Draw usecase on Online shopping product using payment gateway.


Customer Registration and Login
Browsing and Selecting Products
Browsing and Selecting Products, Confirm Order
Customer selects payment method
Example
Credit/Debit Card , Net Banking , UPI
Customer enters payment details (card number, CVV, expiry date).
Payment gateway verifies the details and processes the payment.
Payment gateway sends a success confirmation to the system.
or
Payment gateway sends a failure response.
Actor