

### **1) What is SDLC**

Ans. SDLC or the Software Development Life Cycle is a process that produces software with the highest quality and lowest cost in the shortest time possible. SDLC provides a well-structured flow of phases that help an organization to quickly produce high-quality software which is well-tested and ready for production use.

### **2). What is software testing?**

Ans. Software Testing is a method to check whether the actual software product matches expected requirements and to ensure that software product is [Defect](#) free.

### **3). What is agile methodology.**

Ans. 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.

### **4). what is SRS**

Ans. SRS document describes what a software system is supposed to do, how it should behave, and what constraints or limitations it must adhere to. It serves as a blueprint for software development, ensuring that all stakeholders have a clear understanding of what needs to be built.

### **5). What is oops**

The **object-oriented programming** is basically a computer programming design philosophy or methodology that organizes/ models software design around data, or objects rather than functions and logic.

Object Oriented Programming (OOP) is a programming paradigm in computer science that relies on the concept of **classes** and **objects**. It is used to structure a software program into simple, reusable pieces of code blueprints (usually called classes), which are used to create individual instances of objects. There are many object-oriented programming languages, including JavaScript, c++, java, and python.

## 6) . Write Basic Concepts of oops

- Object
- Class
- Encapsulation
- Inheritance
- Polymorphism

**object:** Any Entity which has own state and behavior that is called an object.

**class:** Collection of objects that is called class

**Encapsulation:** Wrapping up of data or binding of data that is called encapsulation

**Inheritance:** When one object acquires all the properties and behavior of parents class

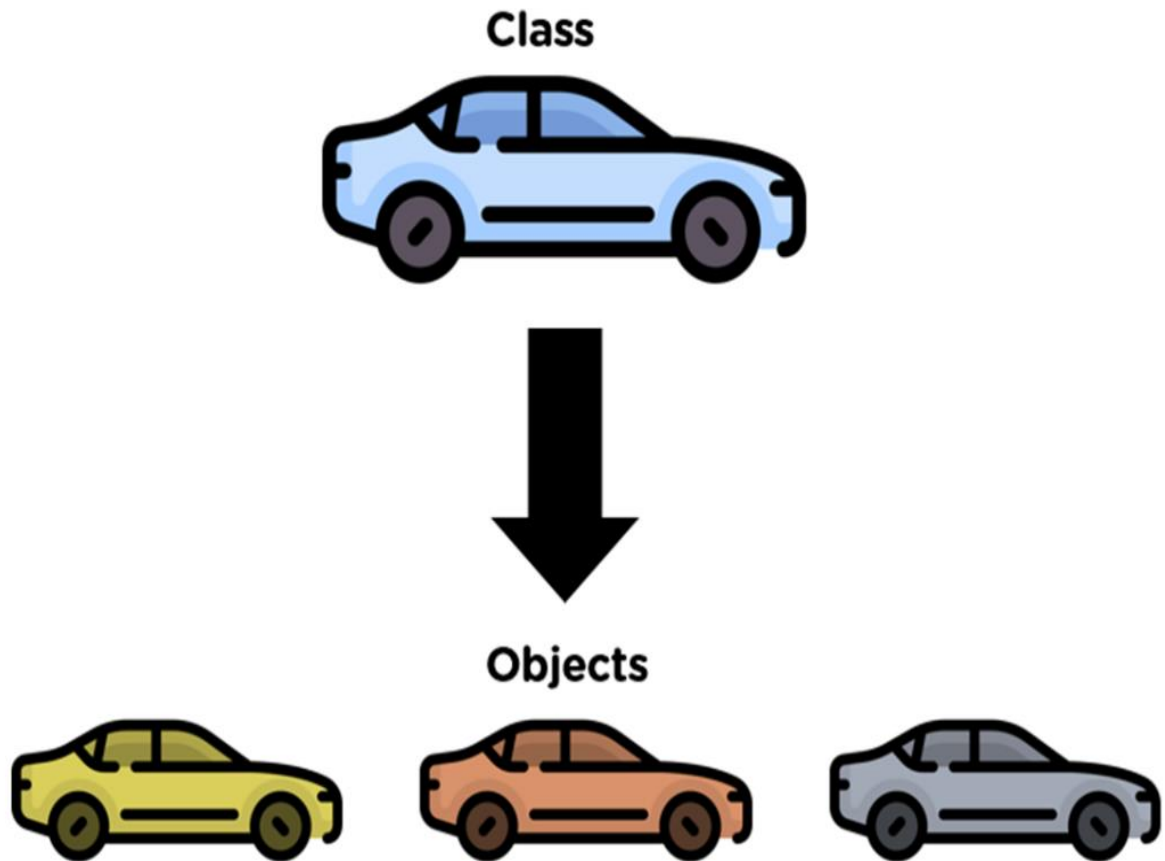
**Polymorphism:** Many ways to perform anything

## 7). What is object

Ans. An object can be defined as an entity that has a state and behavior, or in other words, anything that exists physically in the world is called an object. It can represent a dog, a person, a table, etc. An object means a combination of data and programs, which further represent an entity.

### 8). What is class

**Ans.** class can be defined as a blueprint of the object. It is basically a collection of objects which act as building blocks.



### 9) . What is encapsulation

**Ans .** Classes allow encapsulating data and related functions into a single unit, providing data hiding and preventing direct access.

### 10). What is inheritance

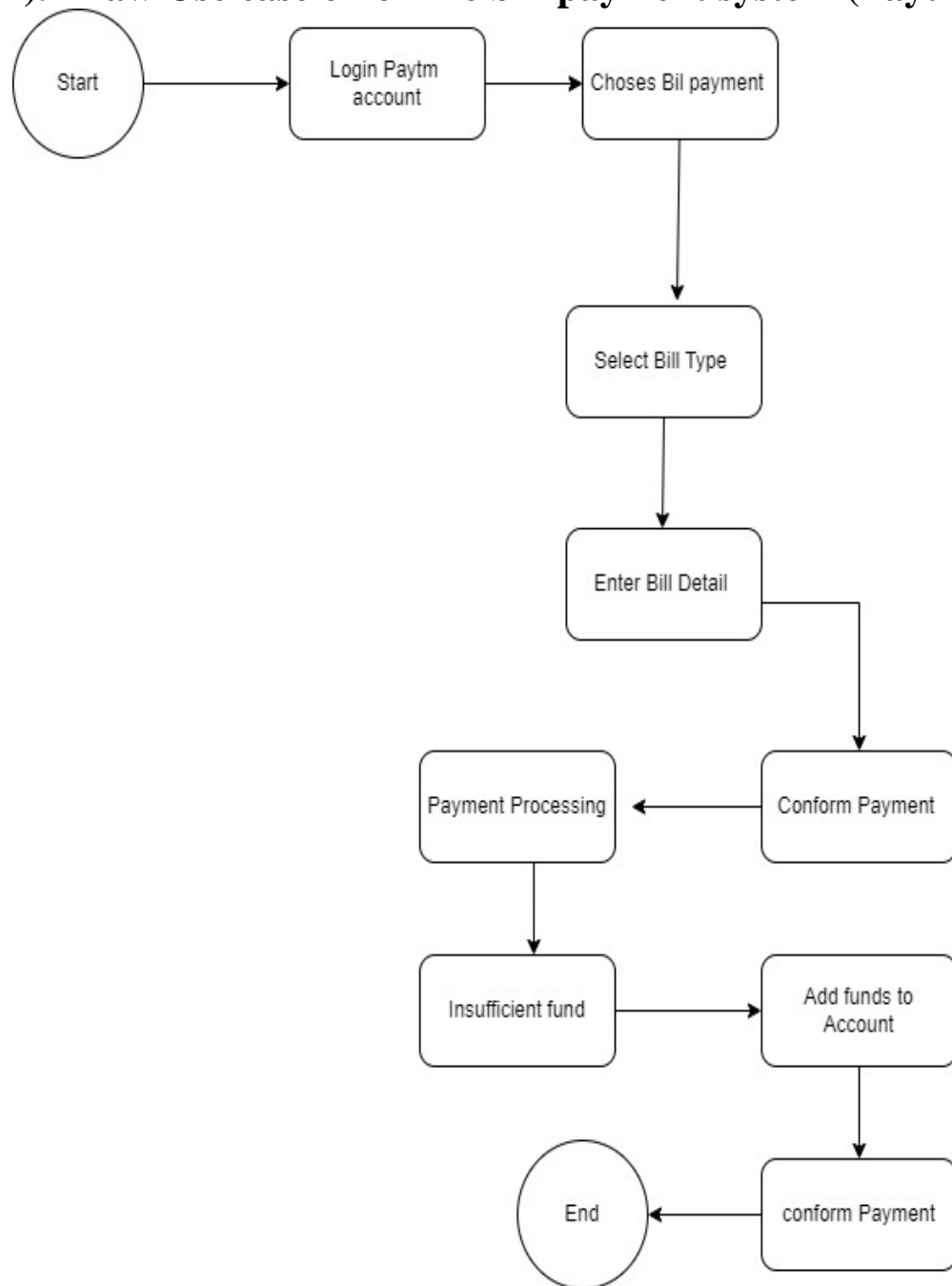
**Ans.** inheritance is the process in which two classes have an is-a relationship among each other and objects of one class acquire properties and features of the other class. The class which inherits the features is known as the child class, and the class whose features it inherited is called the parent class. For example, Class Vehicle is the parent class, and Class Bus, Car, and Bike are child classes.

### 11). What is polymorphism

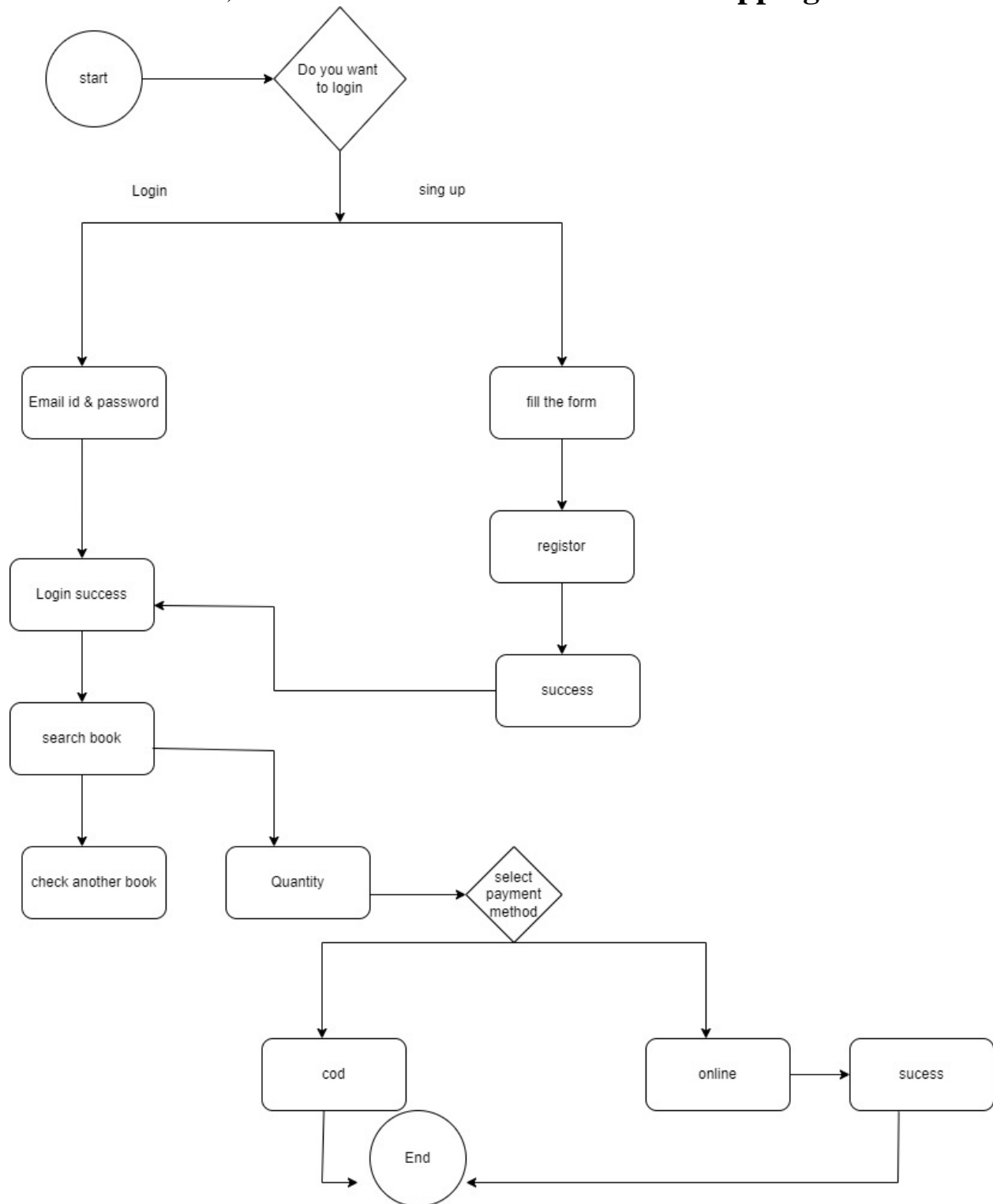
**Ans.** polymorphism is an object-oriented programming concept that refers to the ability of a *variable, function, or object* to take on *multiple* forms. In a programming language exhibiting polymorphism, class objects belonging to the same hierarchical tree (inherited from a common *parent class*) may have functions with the same name, but with different behaviors.

Polymorphism is one of the most important concepts in OOP. It describes the ability of something to have or to be displayed in more than one form. The different forms arise because these entities can be assigned different meanings and used in various ways in multiple contexts.

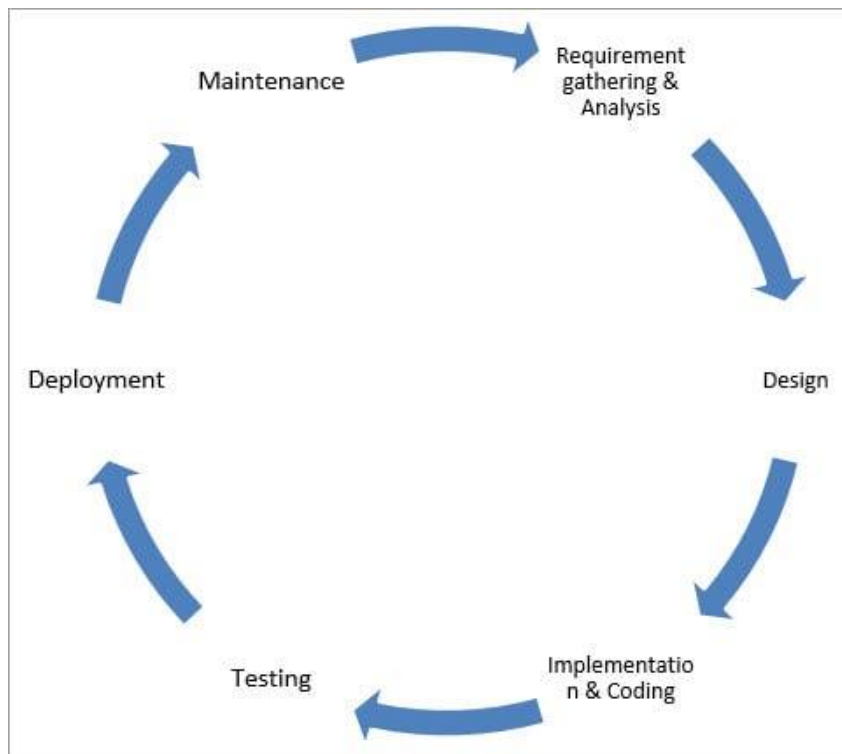
**12). Draw Use case on online bill payment system (Paytm)**



### 13.) Draw Use case on Online book shopping



**14). Write SDLC phases with basic introduction.**



**Ans. 1) Requirement Gathering and Analysis**

During this phase, all the relevant information is collected from the customer to develop a product as per their expectation. Any ambiguities must be resolved in this phase only.

Business analyst and Project Manager set up a meeting with the customer to gather all the information like what the customer wants to build, who will be the end-user, what is the purpose of the product. Before building a product, a core understanding or knowledge of the product is very important.

**For Example,** A customer wants to have an application which involves money transactions. In this case, the requirement has to be clear like

what kind of transactions will be done, how it will be done, in which currency it will be done, etc.

Once the requirement gathering is done, an analysis is done to check the feasibility of the development of a product. In case of any ambiguity, a call is set up for further discussion.

Once the requirement is clearly understood, the SRS (Software Requirement Specification) document is created. This document should be thoroughly understood by the developers and also should be reviewed by the customer for future reference.

**3) Design:** In this phase, the requirement gathered in the SRS document is used as an input and software architecture that is used for implementing system development is derived.

**2) Implementation:** Implementation starts once the developer gets the Design document. The Software design is translated into source code.

**4) Testing:** Testing starts once the coding is complete and the modules are released for testing. In this phase, the developed software is tested thoroughly and any defects found are assigned to developers to get them fixed.

Retesting, regression testing is done until the point at which the software is as per the customer's expectation. Testers refer SRS document to make sure that the software is as per the customer's standard.

### **5) Deployment**

Once the product is tested, it is deployed in the production environment or first **UAT (User Acceptance testing)** is done depending on the customer expectation.

In the case of UAT, a replica of the production environment is created and the customer along with the developers does the testing. If the

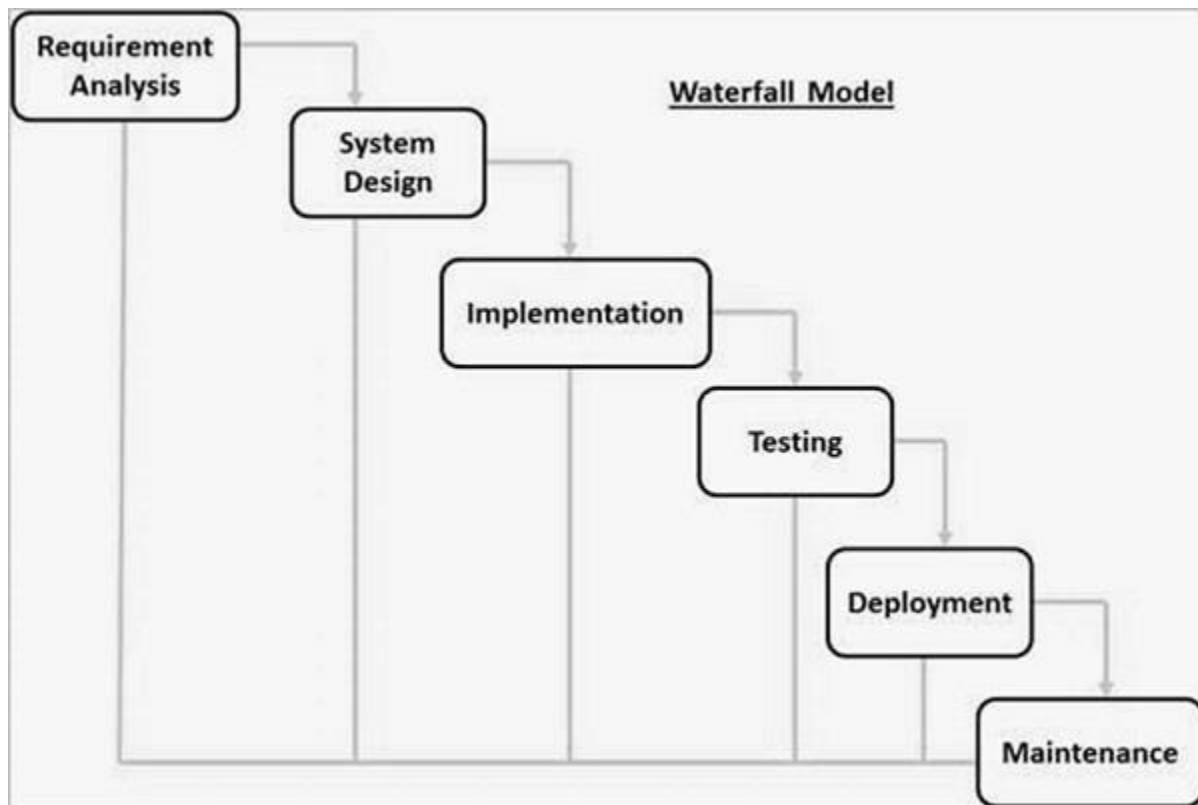


customer finds the application as expected, then sign off is provided by the customer to go live.

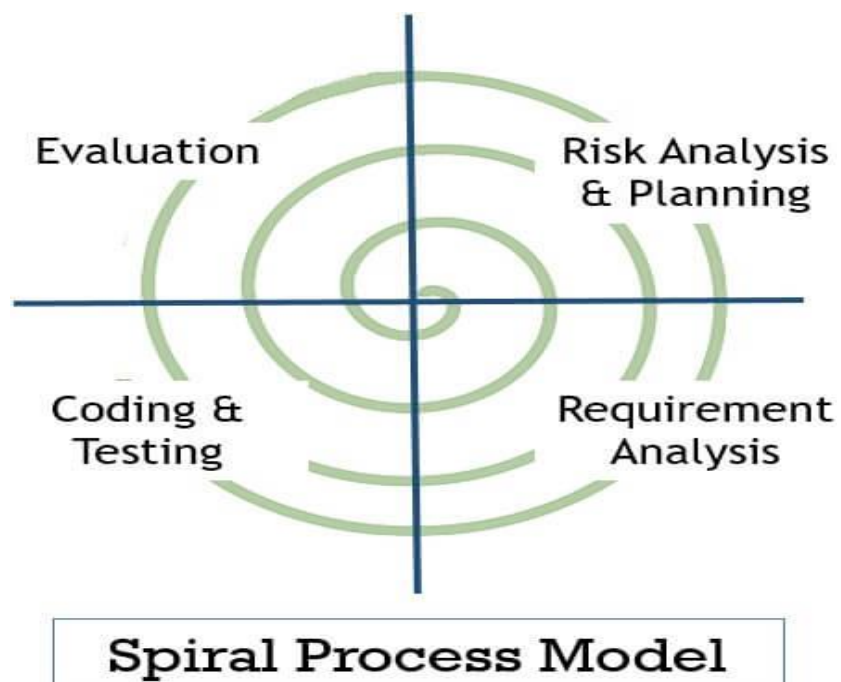
**6) Maintenance:** After the deployment of a product on the production environment, maintenance of the product i.e. if any issue comes up and needs to be fixed or any enhancement is to be done is taken care by the developers

## 15.) Explain Phases of the waterfall model

- **Ans. Requirement Gathering and analysis** – All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification document.
- **System Design** – The requirement specifications from first phase are studied in this phase and the system design is prepared. This system design helps in specifying hardware and system requirements and helps in defining the overall system architecture.
- **Implementation** – With inputs from the system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality, which is referred to as Unit Testing.
- **Integration and Testing** – All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.
- **Deployment of system** – Once the functional and non-functional testing is done; the product is deployed in the customer environment or released into the market.
- **Maintenance** – There are some issues which come up in the client environment. To fix those issues, patches are released. Also to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.



16. Write phases of spiral model



### **1. Risk Analysis & Planning**

Developers identify risks involved in the current iteration and also corrective measures to minimize risk. Then they evaluate corrective measures against objectives and constraints. Thus, the deadline is set for the next stage.

### **2. Requirement Analysis**

This phase analyzes the client's requirements. Also, the requirements for developing the product are also established.

### **3. Coding & Testing**

Developers develop multiple programs and integrated them to form a software or prototype. Like, in the early cycles the product of this stage would be a prototype. In subsequent cycles, the product of this stage becomes **developed software**. The product of this stage is tested to find any error in coding.

### **4. Evaluation**

This stage evaluates whether the project is going as per planning or not. It evaluates whether the objective set at the first stage has been achieved or not. The evaluation phase also helps developers decide the number of cycles required to complete the project.

The spiral model allows using other process models in one or more of its cycles. This is either to reduce the risk at some stage or to get the requirements of the user clear or for resolving technical risks.

### **17). Write agile manifesto principles**

Ans.1. Individuals and interactions - in agile development, self-organization and motivation are important, as are interactions like co-location and pair programming.

2. Working software - Demo working software is considered the best means of communication with the customer to understand their requirement, instead of just depending on documentation.

3.Customer collaboration - As the requirements cannot be gathered completely in the beginning of the project due to various factors, continuous customer interaction is very important to get proper product requirements.

4. Responding to change - agile development is focused on quick responses to change and continuous development.

**18). Explain working methodology of agile model and also write pros and cons.**

Ans. 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 in iterations.
- Each iteration typically lasts from about one to three weeks.
- Every iteration involves cross functional teams working simultaneously on various areas like planning, requirements analysis, design, coding, unit testing, and acceptance testing.
- At the end of the iteration a working product is displayed to the customer and important stakeholders. What is Agile?
- Agile model believes that every project needs to be handled differently and the existing methods 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.

**Pros:** - Is a very realistic approach to software development

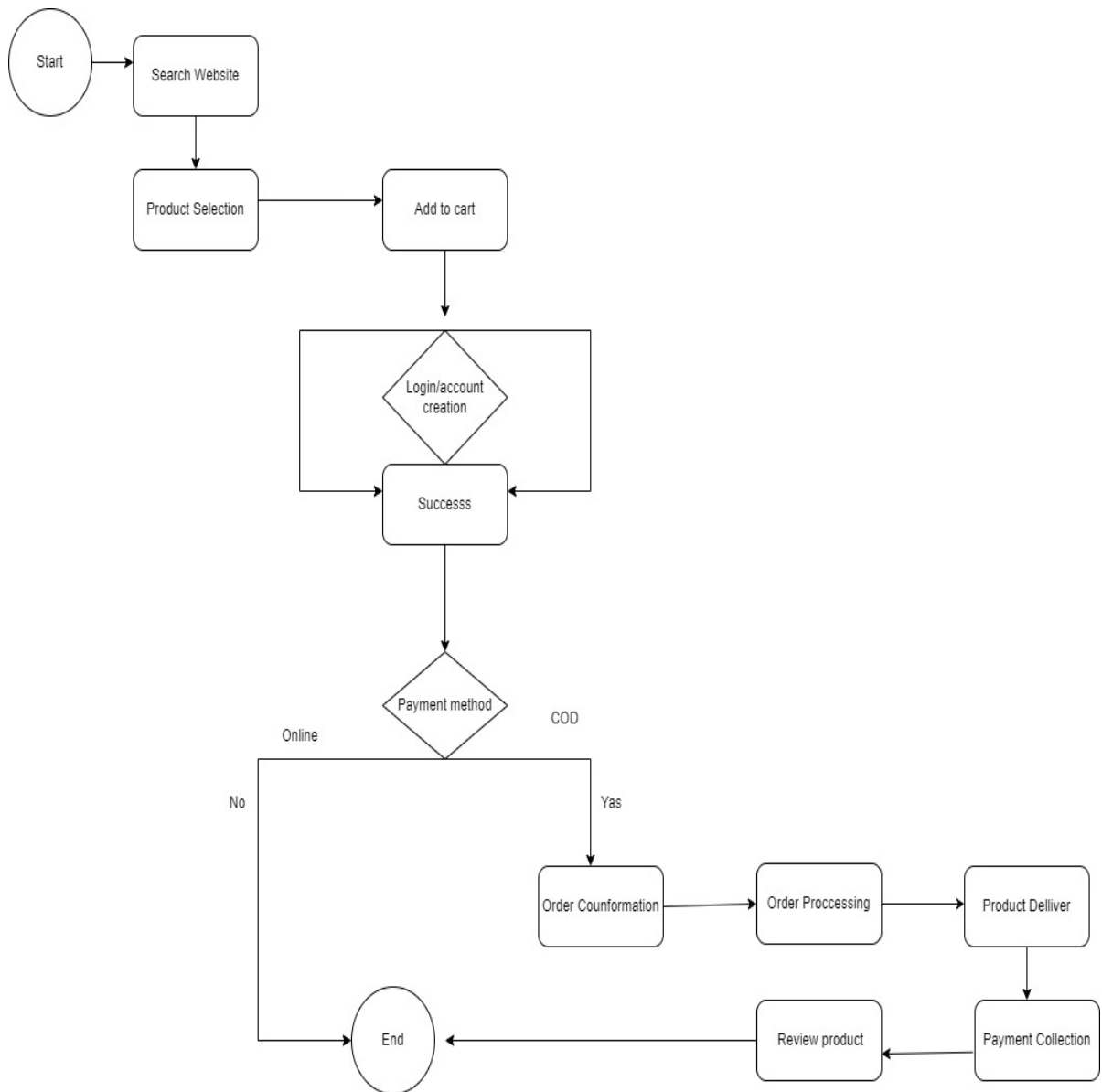
- Promotes teamwork and cross training.
- Functionality can be developed rapidly and demonstrated.
- Resource requirements are minimum.
- Suitable for fixed or changing requirements
- Delivers early partial working solutions.
- Good model for environments that change steadily
- Minimal rules, documentation 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.
- An overall plan, an agile leader and agile PM practice is a must without which it will not work.
- 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

**19. • Draw use case on Online shopping product using COD.**



## 20.) Draw use case on Online shopping product using payment

