Software Testing (Fundamentals)

Q-1 What is SDLC?

Ans :- SDLC is a methodology or step-by-step approach to produce software with high quality, lowest cost in the shortest possible time by defining the phases like Planning, Analysis & Design, Coding & Implementation, and Testing & Maintenance.

- ☐ Full form of SDLC is software development life cycle.

Q-2 What is software testing?

Ans :- software testing for process for validation and verification for developing product.

Software testing for process to identify correctness and completeness for the quality of developing software and application.

Q- 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 software product.

Q-4 What is SRS?

Ans :- Software requirement specification.

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

Three Types of Requirements:

- ☐ Customer Requirement
- ☐ Functional Requirement
- ☐ Non-Functional Requirement

Software Testing (Fundamentals)

Q-5 What is OOPS?
Ans :- Object Oriented Programming is viewed as a collection of objects. It is used to structure the software program into simple reusable code. Here it is referred to as Functional testing or Black Box Testing.

Q-6 Write Basic concept of OOPS?
Ans :- **Class**:- Class is a collection of a data member and member function with its behavior.Class is a blueprint or a template to describe the properties and behavior of the objects.

Object :- An object is the basic unit of OOP which is accessed by its properties called data member & member function. It creates the memory for the class.

Encapsulation :- A wrapping up of data and functions into a single unit is called Encapsulation. IT hide/include private access of data member & member function.

Abstraction :- Abstraction is the representation of the essential features of an object. Also called data hiding.

Polymorphism:- An ability to take one name having many different forms. Compile time Polymorphism : (Operator Overloading) Method name should be same in single class but its behavior (Arguments & Data type) is different.
Run time Polymorphism (Operator Overriding) Method should be same in super class and sub class but its behavior is different.

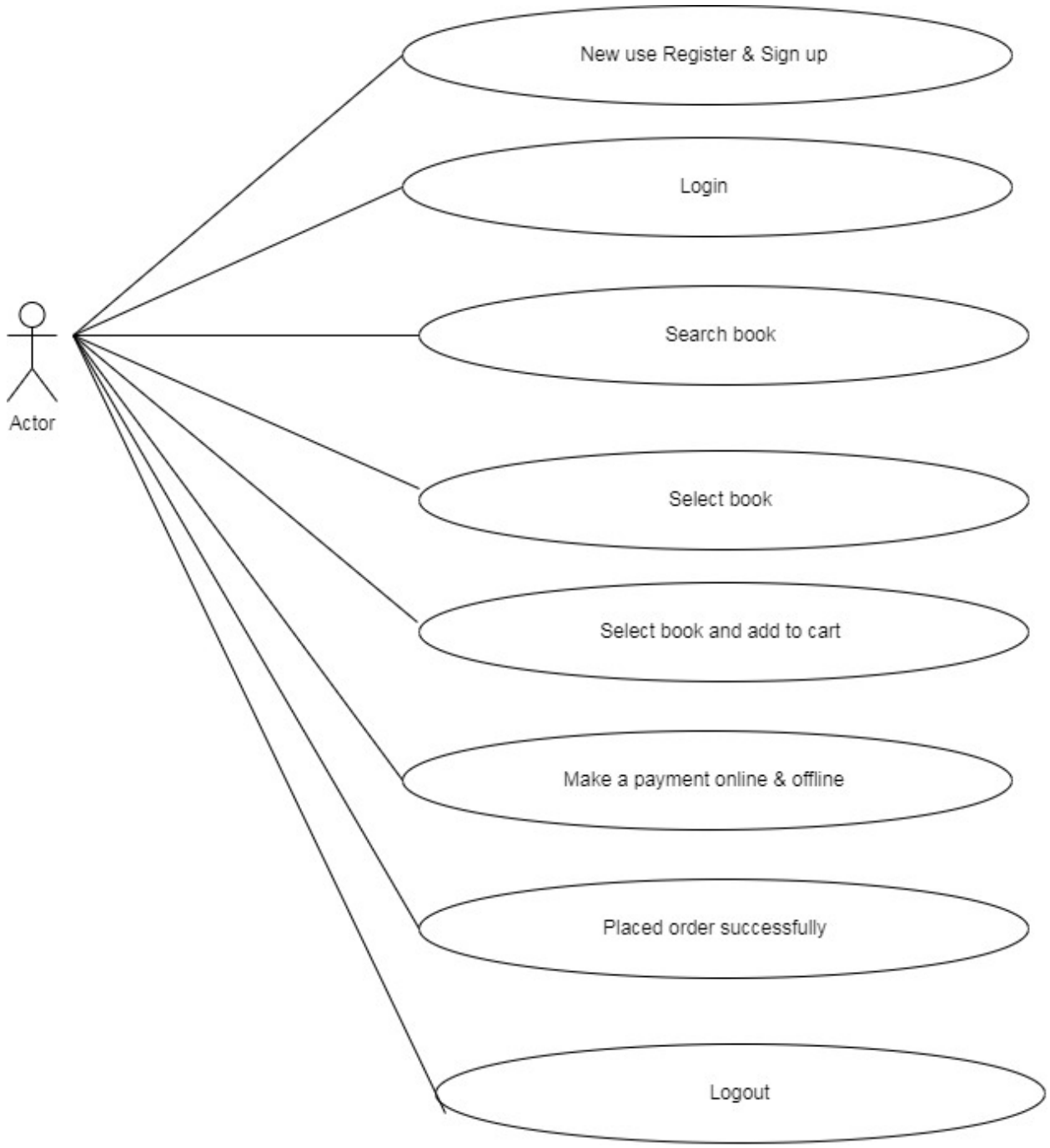
Inheritance: One class (Super, Base) inherits the properties of another class (Sub, Derived).

Types of Inheritance:

- ☐ Single Inheritance
- ☐ Multilevel Inheritance
- ☐ Hierarchical Inheritance
- ☐ Hybrid Inheritance
- ☐ Multiple Inheritance

Software Testing (Fundamentals)

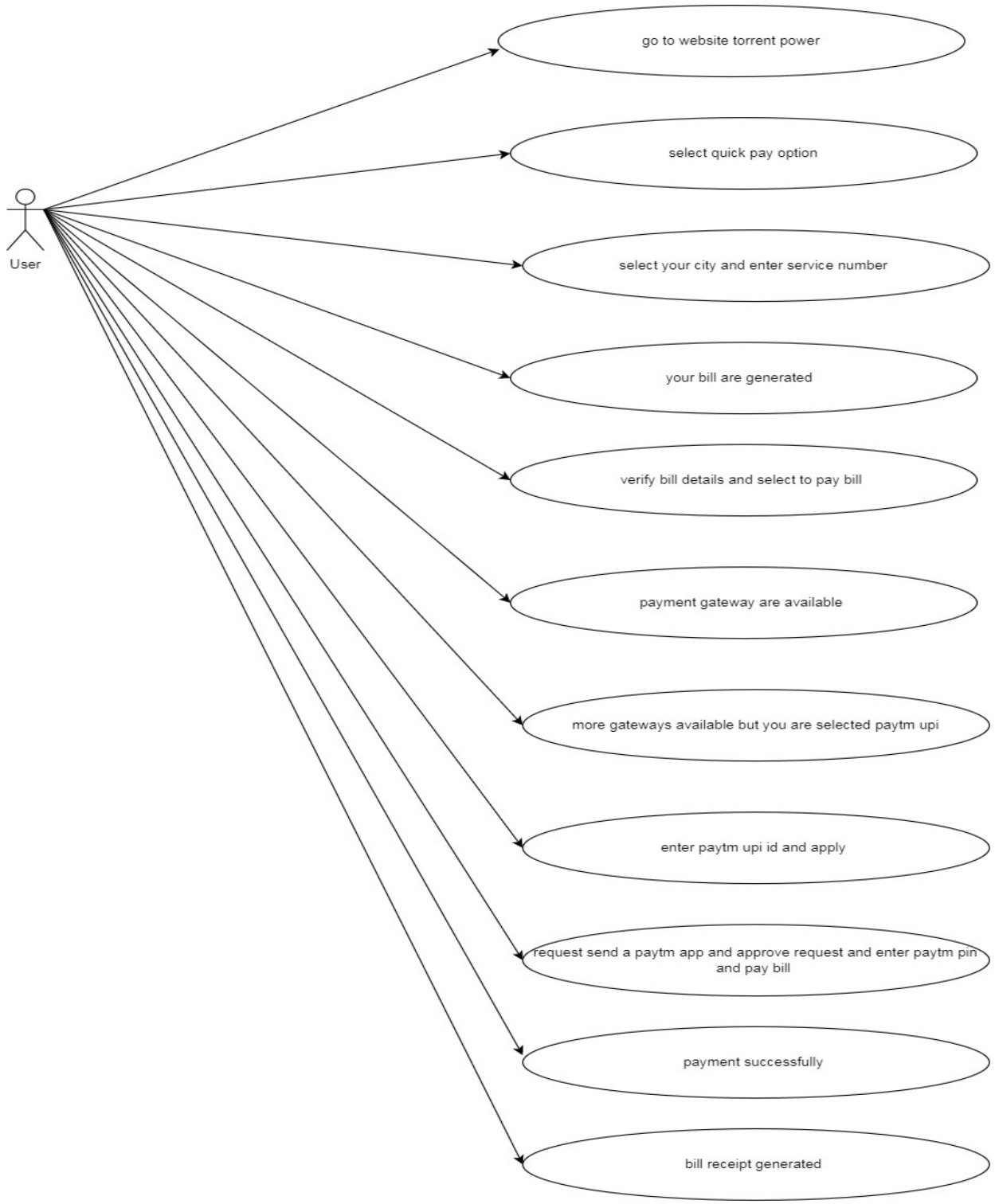
Q-7 Draw Use Case on Online book shopping?
Ans :-



Software Testing (Fundamentals)

Q-8 Draw usecases on online bill payment system (paytm)

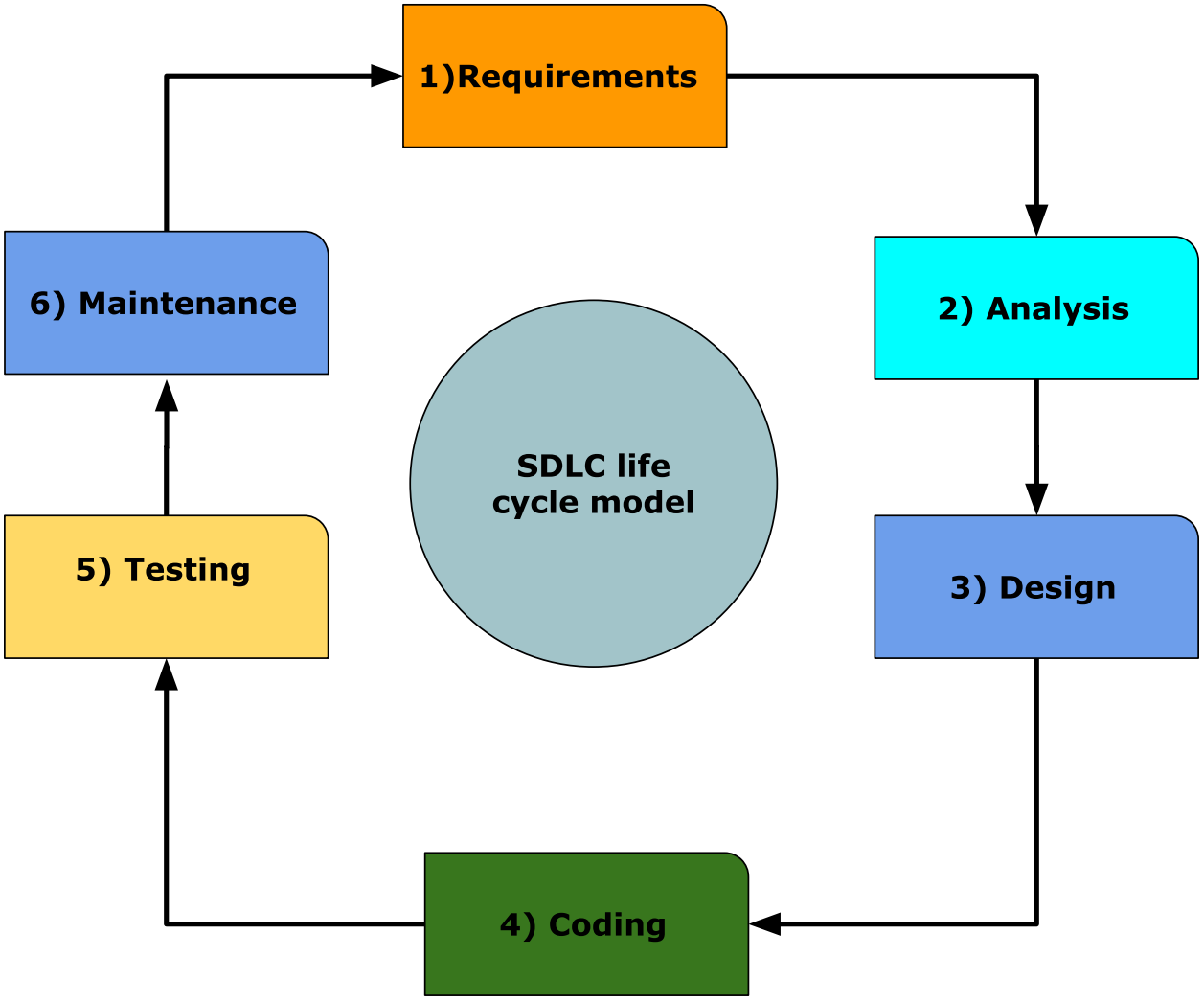
Ans :-



Prepared by Tushar

Q-9 Write SDLC phases with basic introduction?

Ans:-



Ans :-

1) **Requirement gathering** :- requirement gathering for gathering maximum information from the client requirements for the product.Discuss each detail and specification of the product with the customer.

- ☐ Simple word for Establish for customers.

2) **Analysis** :-The development team will then analyze the requirements keeping the design and code of the software in mind. This analysis represents the “what” phase.

- ☐ This phase defines the problem that the customer is trying to solve.
- ☐ The deliverable design document is the architecture.

3) **Design** :- Design Architecture Document.

The Design team can now expand upon the information established in the requirement document.

4) **Coding :-**The developers will then start building the entire system by writing code using the programming languages they chose.

5) **Testing :-** Once the developers build the software, then it is deployed in the testing environment. Then the testing team tests the functionality of the entire system.The testing phase is a separate phase which is performed by a different
team after the implementation is completed.

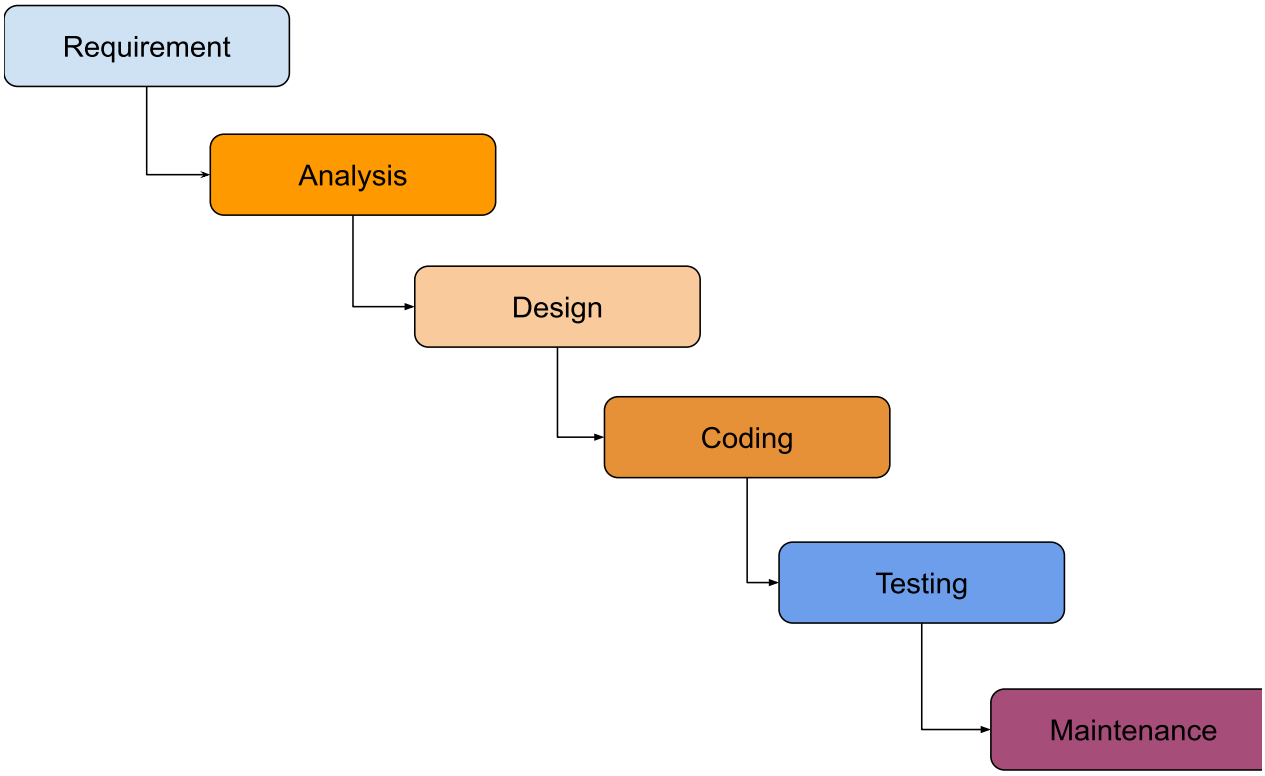
6) **Maintenance** :- Software maintenance is one of the activities in software engineering, and is the process of enhancing and optimizing deployed software (software release), as well as fixing defects. Three main category for maintenance :-

- ☐ **Corrective maintenance**: identifying and repairing defects.
- ☐ **Adaptive maintenance**: adapting the existing solution to the new platforms.
- ☐ **Perfective Maintenance**: implementing the new requirements.

Software Testing (Fundamentals)

Q-10 Explain Phases of the waterfall model?

Ans :- The waterfall model is a classical software lifecycle that models the software development as step-by-step “waterfall” between the various development phases.



1) **Requirement** :- During this phase, detailed requirements of the software system to be developed are gathered from the client. Although requirements may be documented in written form, they may be incomplete, unambiguous, or even incorrect.

2) **Analysis** :- The analysis phase defines the requirements of the system,independent of how these requirements will be accomplished.This phase defines the problem that the customer is trying to solve.The deliverable result at the end of this phase is a requirement document.This phase starts with the requirement document delivered by the requirement phase and maps the requirements into architecture.

3) **Design** :- Plan the programming language, for Example java, PHP , .net. Design Architecture Document,Implementation Plan

4) **Coding** :- After the design stage, it is nothing but coding the software.Given the architecture document from the design phase and the requirement document from the analysis phase, the team should build exactly what has been requested, though there is still room for innovation and flexibility.

5) **Testing :-** In this phase, you test the software to verify that it is built as per the specifications given by the client.All the units developed in the implementation phase are integrated into a system after testing of each unit.

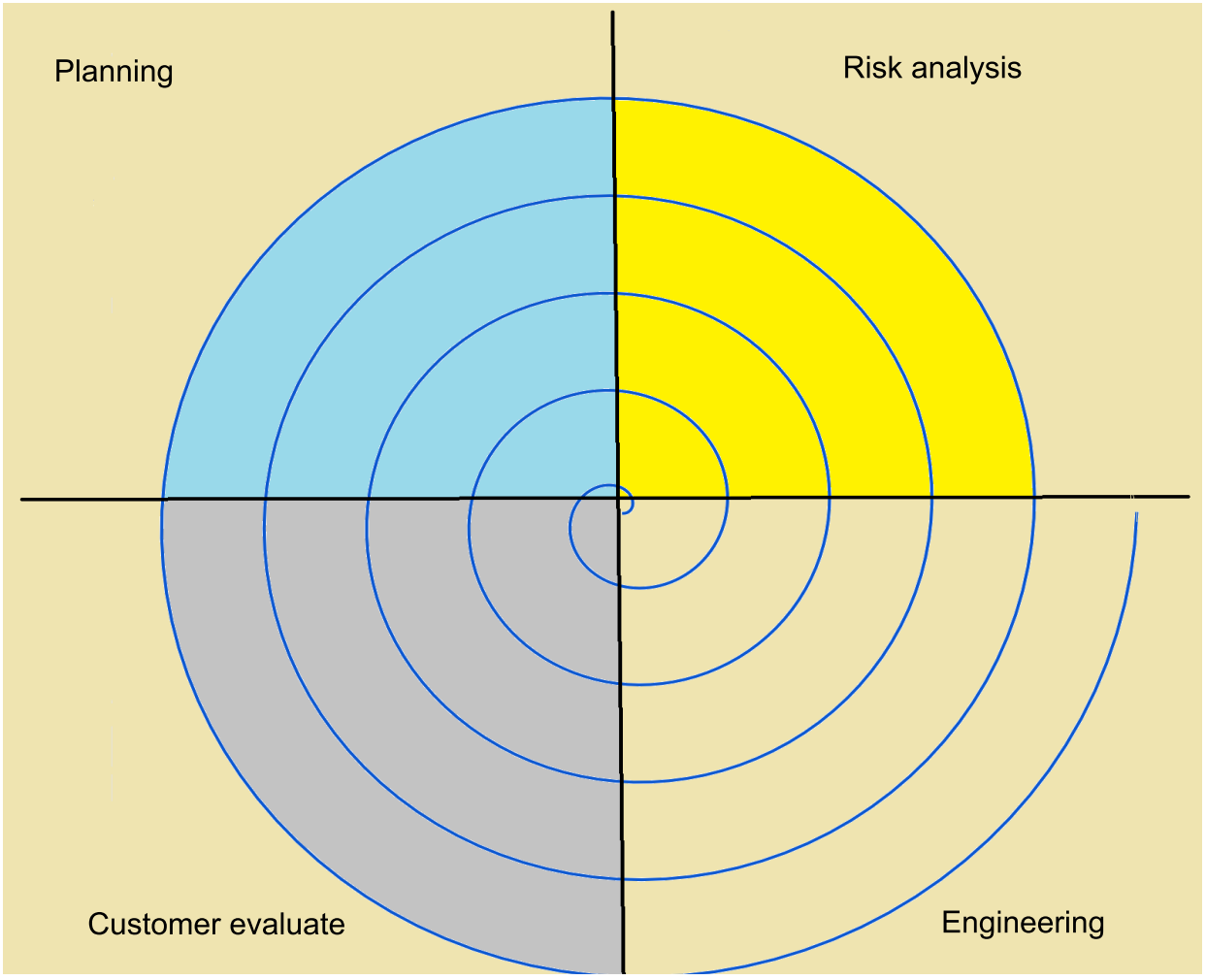
6) **Maintenance** :- Once your system is ready to use, you may later require to change the code as per customer request.

   Three mainly types for the maintenance:-

   - ☐ Corrective maintenance: identifying and repairing defects
   - ☐ Adaptive maintenance: adapting the existing solution to the new platforms.
   - ☐ Perfective Maintenance: implementing the new requirements

Q-11 Write phases of spiral model?

Ans:-



Planning and Requirements :- The initial phase involves gathering basic requirements for any website.The team analyzes any risks, such as security or scalability, and creates a small prototype.

Risk Analysis :- analysis of alternative and identification risk resolution of risks.The spiral model involves risk analysis and handling in every phase, improving security and the chances of avoiding attacks and breakages.

Engineering :- start coding for developing software and applications.

Customer evaluation :- The final phase involves full implementation, thorough testing.

Q-12 Write agile manifesto principles?
Ans :-
Individuals and interactions :- Focuses on the importance of effective communication and collaboration among team members and clients.

Working software :- Prioritizes the delivery of functional software as the primary measure of progress.

Customer collaboration :- Encourages customers and stakeholders to have active involvement throughout the development process.

Responding to Change :- On changing requirements, embracing flexibility and ability to adapt even late in the development process.
It can change requirement from side customer side responding change immediately.

Q- 13 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.

Pros:-

- ☐ Very realistic approach
- ☐ Rapid delivery.
- ☐ Functionality can be developed rapidly
- ☐ Resource requirements are minimum.
- ☐ Little or no planning required

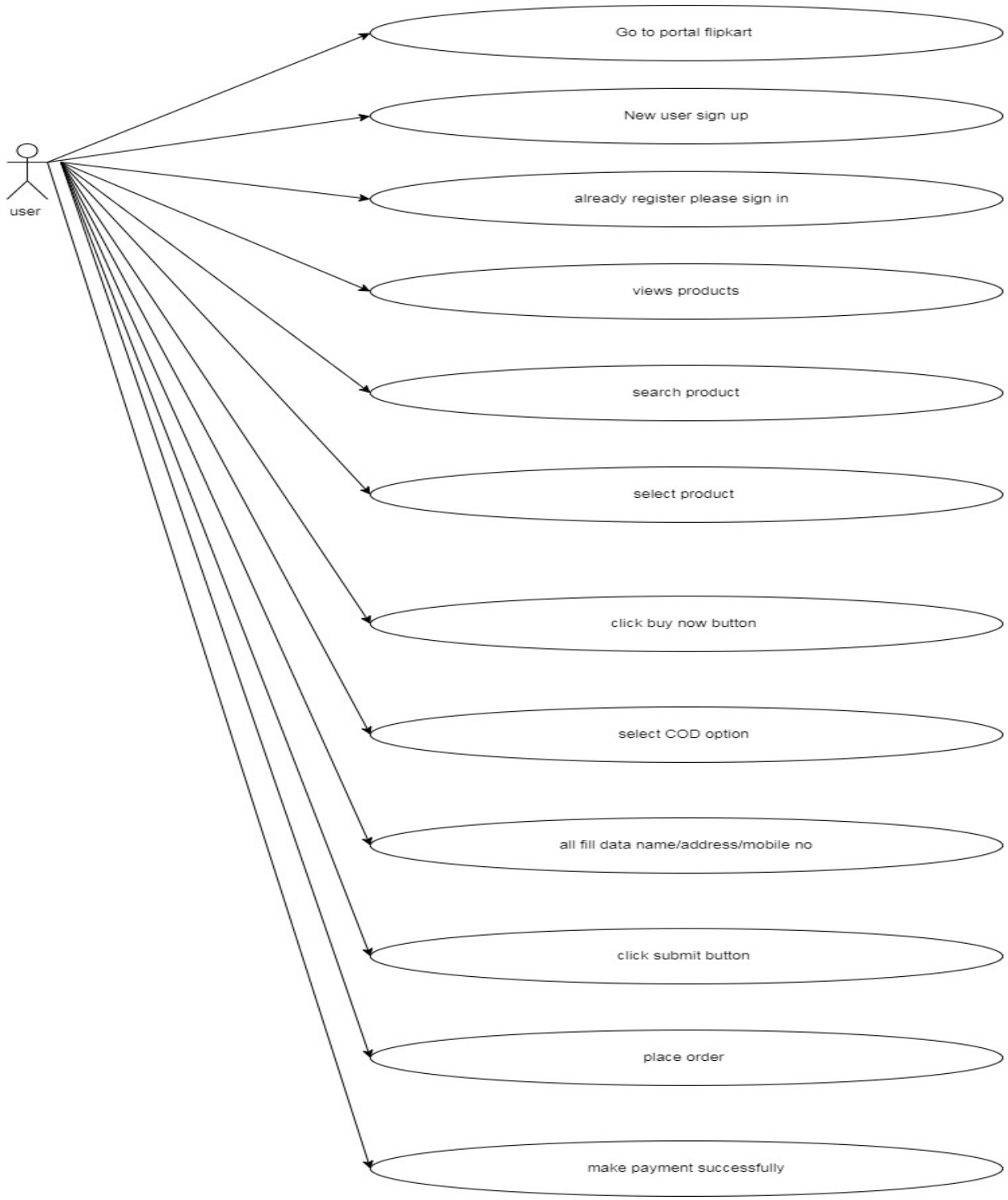
Cons:-

- ☐ More risk of sustainability, maintainability and extensibility.
- ☐ Depends heavily on customer interactions.
- ☐ Very high individual dependency.
- ☐ Minimum documentation generated.
- ☐ Not useful for small projects.
- ☐ Not suitable for handling complex dependencies.

Software Testing (Fundamentals)

Q- 14 Draw use cases on online shopping product using COD.

Ans :-



Software Testing (Fundamentals)

Q- 15 Draw use cases on online shopping product using payment gateway.

Ans :-