Module–1(Fundamental)

1. What is SDLC?

- **SDLC** is **Software Development Life Cycle.**
- SDLC is a framework defining tasks performed at each step in the software development process.
- SDLC is define the step by step development of software at each phase. It covers the detailed plan for building, deploying and maintaining the software.

2. Write SDLC phases with basic introduction

- Software Development Life Cycle, SDLC for short, is a well-defined, structured sequence of stages in software engineering to develop the intended software product.

- **Given below are the various phases:**

1. **Requirement gathering :** customer need requirements

   - Types of **Problem** :
     - Lack of clarity
     - Requirements confusion
     - Requirements Amalgamation
   - Types of **Requirements** :
     - Functional Requirements
     - Non –Functional Requirements

2. **Analysis : **Model And Specify the requirements- “What”
   Gather the requirements for the system.

3. **Design : **design the architecture model

4. **Implementation : **programming phase, documentation
   build the database & program.

5. **Testing : **validation & verification **,**write the test case condition

6. **Maintenance : **Repair defects and adapt the solution to the new requirements.

   - Corrective maintenance
   - Adaptive maintenance
   - Perfective Maintenance

3. Explain Phases of the waterfall model

- Waterfall model is one type of classical software cycle.
- The classical software lifecycle models the software development as a step-by-step waterfall between the various development phases.
- Requirements are very well documents ,clear and fixed & definition is stable.

- The project is sort.
- Waterfall model is **unrealistic** for :
  - requirements must be frozen to early in the life cycle,
  - requirements are validated too late.

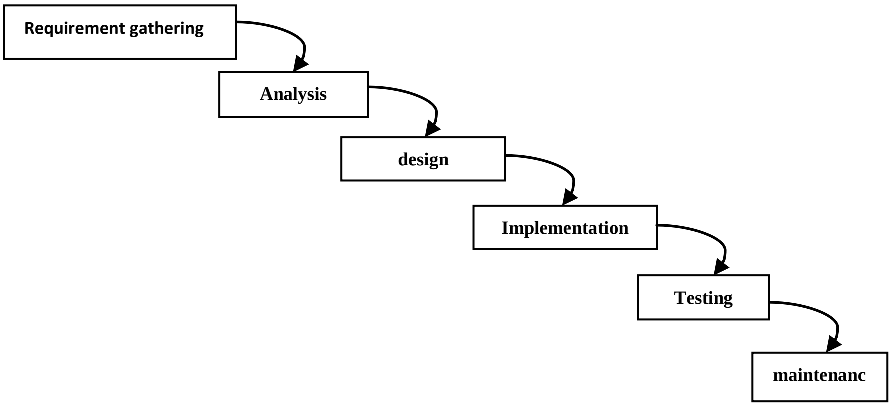


Fig : Waterfall model

- **Given below are the various phases:**

1. **Requirement gathering :** customer need requirements

   - Types of **Problem** :
     - Requirements confusion
     - Lack of clarity
     - Requirements Amalgamation
   - Types of **Requirements** :
     - Functional Requirements
     - Non –Functional Requirements

2. **Analysis :** Model And Specify the requirements- "What"
   Gather the requirements for the system.

3. **Design :** design the architecture model

4. **Implementation :** programming phase, documentation
   build the database & program.

5. **Testing :** validation & verification **,**write the test case condition

6. **Maintenance :** Repair defects and adapt the solution to the new requirements.

   - Corrective maintenance
   - Adaptive maintenance
   - Perfective Maintenance

Advantages:

- This model is simple and easy to understand and use.
- It is a easy to manage.
- Phases do not overlap.
- In this phases are processed and completed one at a time.
- Waterfall model works well for small projects and requirements are very well understood.

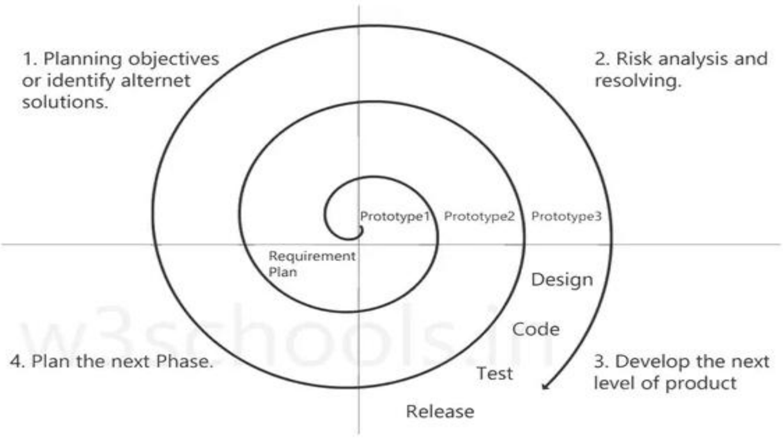
Disadvantages:

- Its difficulty application to go back.
- Not good model for complex and object-oriented projects.
- High amounts of risk and uncertainty.
- Not suitable for project requirements are changes.
- Poor model for long and ongoing projects.
- Cannot accommodate changing requirements.

4. Write phases of spiral model

- **Given below are the various phases:**

1. Planning objectives or identify alternative solutions:
2. Risk analysis and resolving:
3. Develop the next level of product:
4. Plan the next Phase:


1. Planning objectives or identify alternet solutions.
2. Risk analysis and resolving.
Prototype1
Prototype2
Prototype3
Requirement Plan
Design
Code
Test
Release
3. Develop the next level of product
4. Plan the next Phase.

5. What is agile methodology?

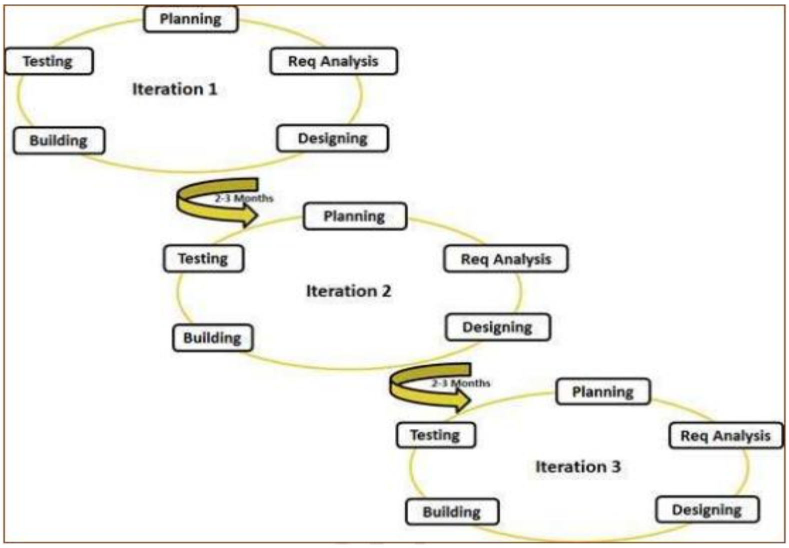
- Agile SDLC model is a combination of iterative and incremental process models with focus on process adaptability and customer satisfaction by rapid delivery of working software product.
- Its break the product into small incremental builds.

6. Write agile manifesto principles

- Individuals and interactions
- Working software
- Customer collaboration
- Responding to change

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

- The Agile methodology is a way to manage a project by breaking it up into several phases.
- It involves constant collaboration with stakeholders and continuous improvement at every stage. Once the work begins, teams cycle through a process of planning, executing, and evaluating.



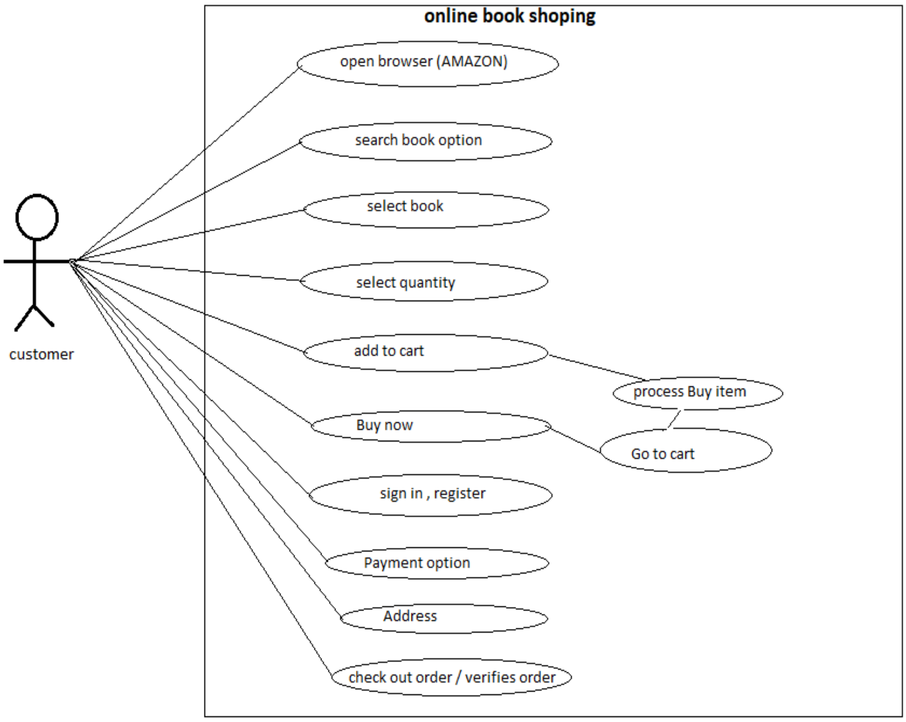
Pros :

- 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.
- Little or no planning required.
- Easy to manage.
- Gives flexibility to developers.

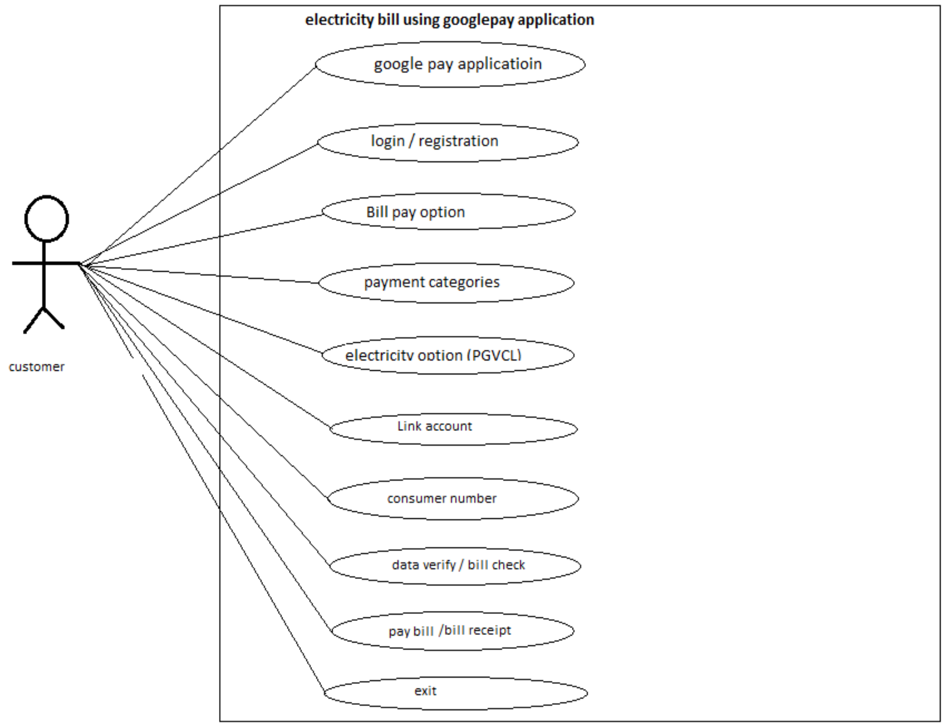
Cons:

- ☞ 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.

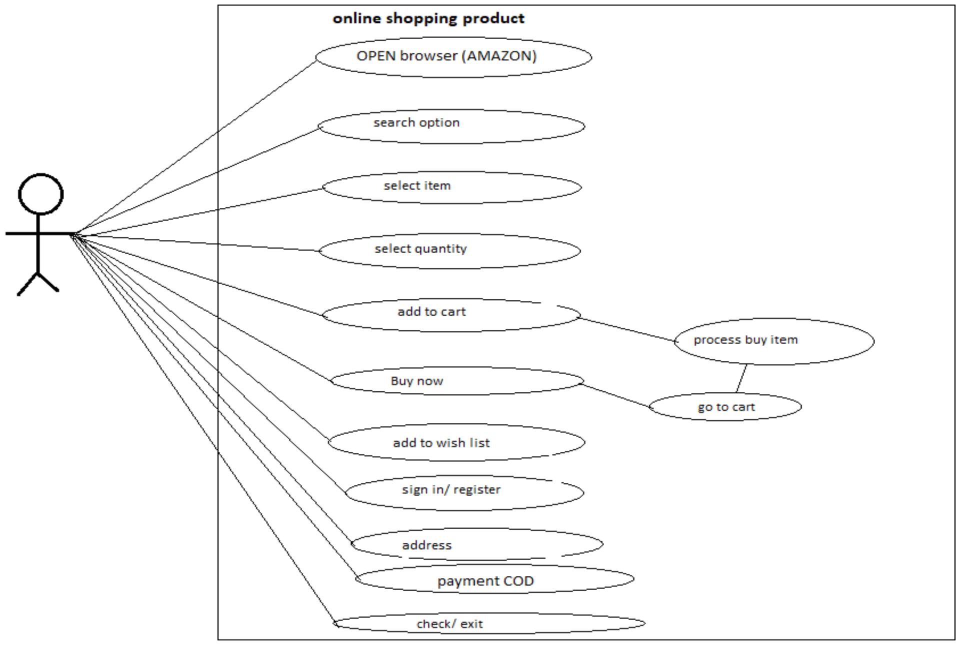
8. Draw Usecase on Online book shopping.



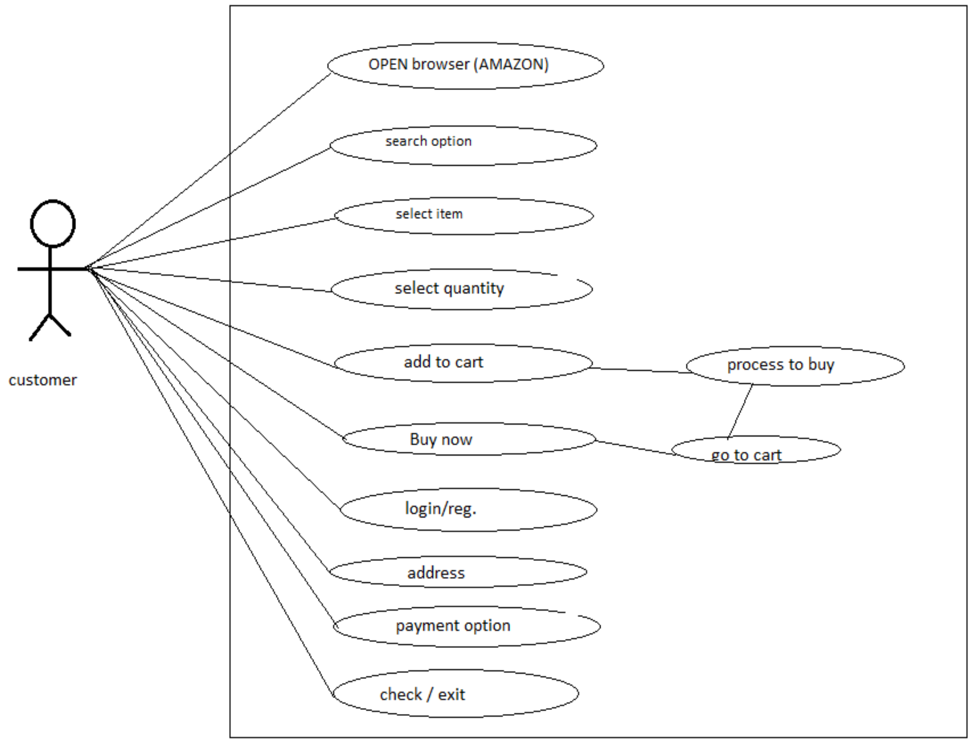
9. Draw Usecase on online bill payment system (paytm).



10. Draw usecase on Online shopping product using COD.



11. Draw usecase on Online shopping product using payment gateway.



12. What is SRS.

A software requirements specification (**SRS**) is a document that captures complete description about how the system is expected to perform.

13. What is oops?

OOPS : Object Oriented Programming Systems
An object is like a black box.
The internal details are hidden.

14. Write Basic Concepts of oops.

basic concepts of object oriented programming are –

- CLASS
- OBJECTS
- ENCAPSULATION
- POLYMORPHISM
  - Overriding
  - Overloading
- INHERITANCE
  - Single
  - Multilevel
  - Hierarchical
  - Multiple
  - Hybrid
- ABSTRACTION

15. What is object?

Instances of an class.
To create memory for the class.
To access whole the properties of a class except private.

16. What is class?

Class is a collection of data member (variable) and member function (process, method) with its behavior….

17. What is encapsulation?

Encapsulation (Data hiding) is a wrapping up of data into single unit.

Private your data member and member function.

18. What is inheritance?

- properties of parent class derived into child class
- properties of super class extends into sub class
- main purpose is : reusability , extensibility
- **there are 5 types of**
  1. single
  2. multilevel
  3. hierarchical
  4. multiple :- java does not support
  5. hybrid:- java does not support

19. What is polymorphism?

- ability to take one name having different forms
- many forms or multiple forms
- **there are 2 types**
  1. compile time (Method overloading)
  2. Run time (method overriding)

20. What is RDBMS.

RDBMS (Relational database management system) the software used to store, manage, query and retrieve data stored in a relational database is calledarelational database management system.The RDBMS provides an interface between users and applications and the database.

21. What is SQL.

SQL stand for Structure query language.
To store in permanent medium.
SQL is a standard language for storing, manipulating and retrieving data in databases.

22. Write SQL Commands.

DDL : data definition language
Create table ,create database, use ,truncate etc….
DML : data manipulate language
INSERT ,UPDATE, DELETE
DQL : data query language
SELECT
DCL/TCL : data/tran control language
COMMIT,ROLLBACK,GRANT ETC….

23. What is join?

A JOIN clause is used to combine rows from two or more tables, based on a related column between them.

24. Write type of joins.

- **INNER JOIN:** returns rows when there is a match in both tables.
- **LEFT JOIN:** returns all rows from the left table, even if there are no matches in the right table.
- **RIGHT JOIN**: returns all rows from the right table, even if there are no matches in the left table.
- **FULL JOIN:** returns rows when there is a match in one of the tables.DDL – Data Definition Language