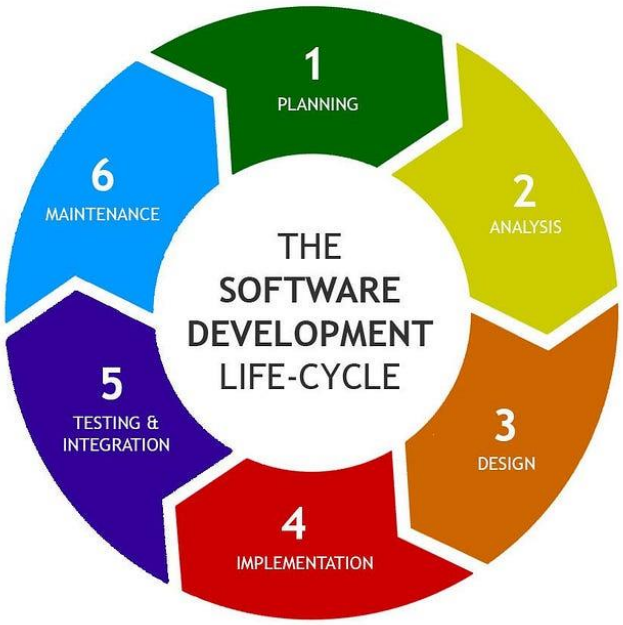
Name: Solanki Jinesh J.

Course: Diploma in Software Testing And Automation

Assignment: Module 1 Software Testing Fundamentals

1. What is SDLC?

SDLC is a step-by-step approach to develop any software/ product with high quality, within the time and within the cost.



2. What is software testing?

Software testing is a process to verify that the requirements are fulfilled or not.

OR

Software testing is a process which is used to identify the correctness, completeness and quality of the developed software.

3. What is agile methodology?

Agile model believes that every project needs to be handled differently and the existing methods divided into small time frames to deliver specific feature of a release.

Agile methodology is a structured approach into manageable phases, focusing on continuous improvement. It is an iterative process that involves planning, execution, and evaluation. Each iteration typically lasts from about one to three weeks.

Agile Methods break the product into small incremental builds.

4. What is SRS?

Software Requirement Specification (SRS) is a document that describes what a software/ product will do and how it will perform. It includes a set of use cases that describe all of the interactions that the users will have with the software.

5. What is oops?

Object-oriented programming (OOP) is a computer programming model that organizes software design around objects, rather than functions and logic.

6. Write Basic Concepts of oops.

- Class
- Object
- Encapsulation
- Inheritance
- Polymorphism
  - Overloading
  - Overriding
- Abstraction

7. What is object?

An object is a fundamental unit that represents a specific instance of a class. Object will give the memory to the class.

8. What is class?

A class is a blueprint for creating objects. It defines a type of object by specifying its data members and member functions.

9. What is encapsulation?

Encapsulation means wrapping up of data into a single unit. It is also called Data hiding.

10. What is inheritance?

Inheritance describes as one class to inherit attributes and methods from another class. Inheritance provides code re-usability. In place of writing the same code, again and again, we can simply inherit the properties of one class into the other class.

❖ **Types of Inheritance:**

- Single Inheritance.
- Multilevel Inheritance.
- Multiple Inheritance.
- Hierarchical Inheritance.
- Hybrid Inheritance.

11. What is polymorphism?

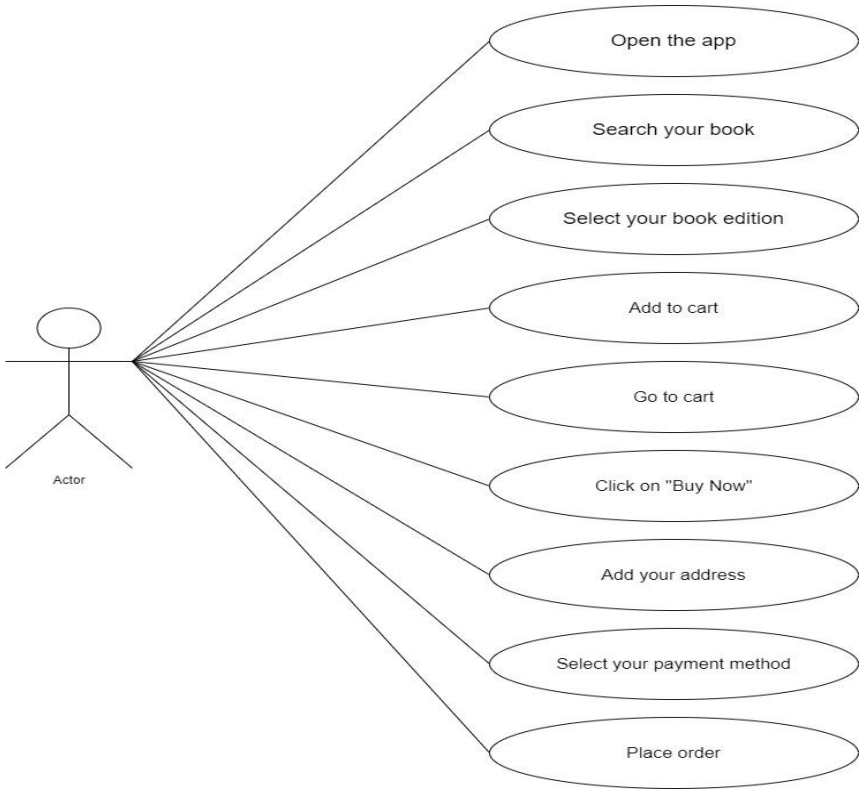
Polymorphism in oops, it is easy to represent one single form into various forms.

It has two types:

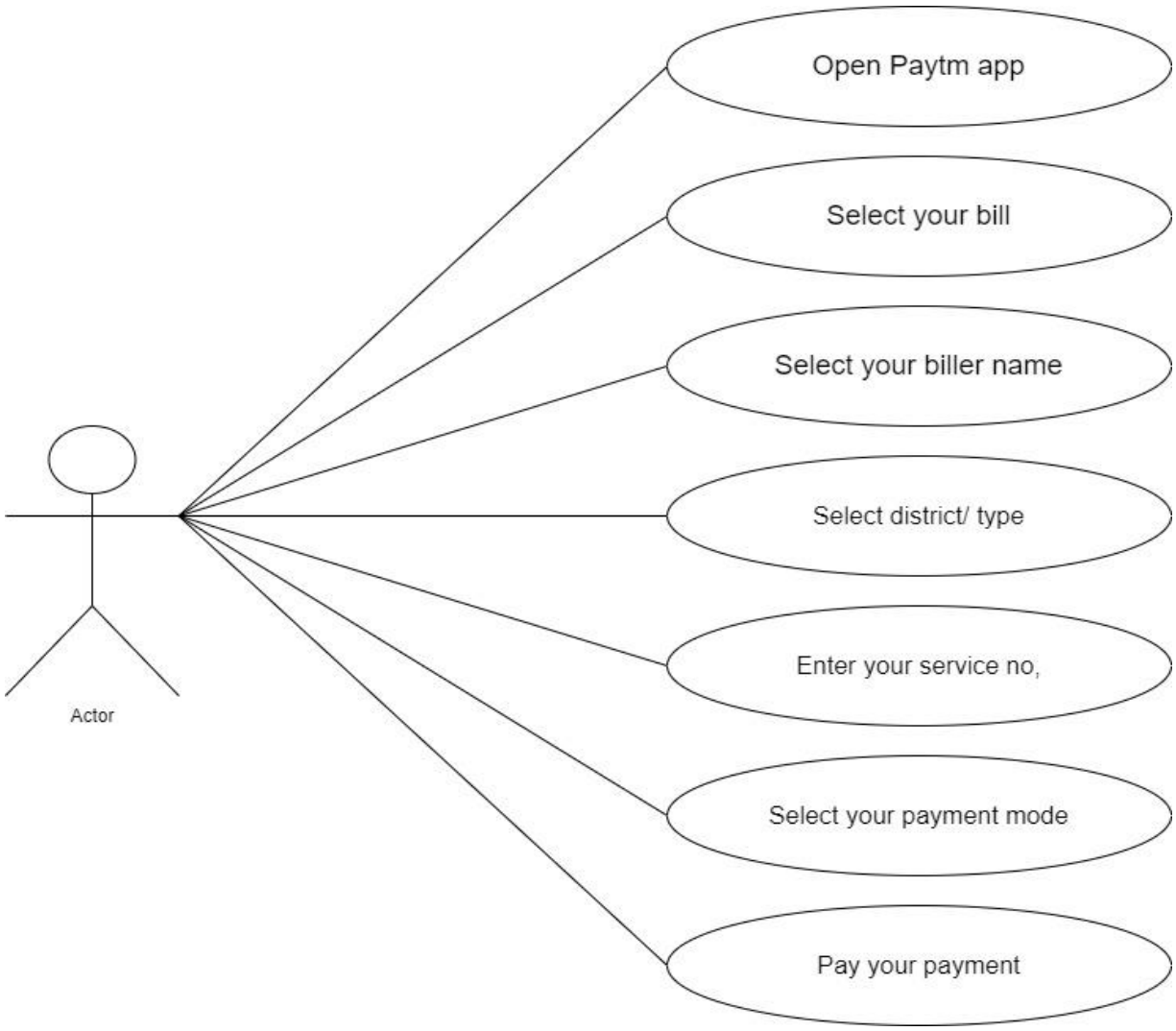
Overloading: It allows a class to have multiple methods with the same name but different parameter lists.

Overriding: It describes as, when a subclass provides a specific implementation for a method that is already defined in its superclass.

12. Draw Usecase on Online book shopping.



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



14. Write SDLC phases with basic introduction.

SDLC is a step-by-step approach to develop any software/ product with high quality, within the time and within the cost.

There are Six phases of it are as under.

- **Planning/ Requirement gathering (What)**

  Problem can be raised while gathering the requirements.

  - Lack of clarity
  - Requirement confusion (Functional & Non- functional)
  - Requirement amalgamation (group)

❖ **Analysis (How)**

The analysis phase defines the requirements of the system, on dependent of how these requirements will be accomplished.

Ideally, this document states in a clear and precise fashion what is to be built.

❖ **Design**

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

The architecture team also converts the typical scenarios into a test plan.

e.g.

DFD : Data Flow Diagram
ER : Entity Relationship

❖ **Implementation**

In the implementation phase, the team builds the components either from scratch or by composition.
Software can be implementation by the technology (java, python, PHP).

The implementation phase deals with issues of quality, performance, baselines, libraries, and debugging.

❖ **Testing**

The testing phase is a separate phase which is performed by a different team after the implementation is completed.

Testing is used to verify and validate the software.

❖ **Maintenance**

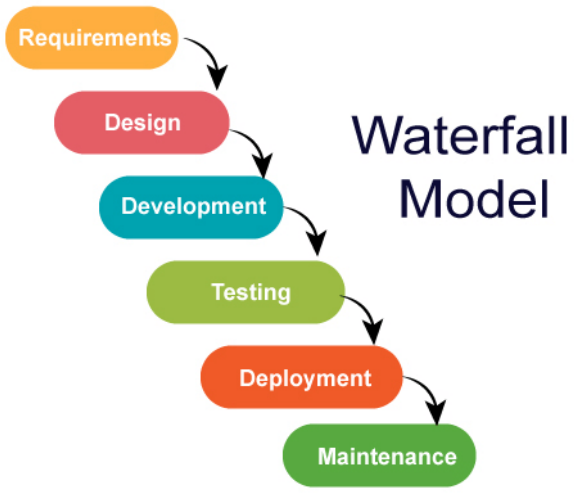
Software maintenance is one of the activities in software engineering, and is the process of enhancing and optimizing deployed software (software release), as well as fixing defects.

There are three types of maintenance:

Corrective maintenance	:	Identifying and repairing defects
Adaptive maintenance	:	Adapting the existing solution to the new platforms.
Perfective Maintenance	:	Implementing the new requirements

15. Explain Phases of the waterfall model.

The Waterfall model is a traditional and linear approach to software development, where each phase of the project is completed before the next phase begins.



It is a classical software lifecycle models like step by step “waterfall”, so it is called waterfall model.

Requirement gathering :- it describes as gathering information , total cost and time to require to develop any product/software.

Analysis :- It is short document that define entire Life cycle of project.

Design :- It is a visualization of software of product by designing.

Coding :- It is a programming code for software.

Testing :- It is performing if there is no bug in the software or verify the quality, completeness and correctness of software.

Deployment :- In simple terms it is sales after service, fixing bugs.

❖ Applications (When to use?)

- Requirements are very well documented, clear and fixed.
- Product definition is stable.
- Technology is understood and is not dynamic.
- There are no ambiguous requirements.
- Ample resources with required expertise are available to support the product.
- The project is short.

❖ Pros (Why Waterfall Model)

- Simple and easy to understand and use
- Easy to manage due to the rigidity of the model. Each phase has specific deliverables and a review process.
- Phases are processed and completed one at a time.
- Works well for smaller projects where requirements are very well understood.
- Clearly defined stages.
- Well understood milestones.
- Easy to arrange tasks.
- Process and results are well documented.

❖ Cons (Why not Waterfall Model)

- No working software is produced until late during the life cycle.
- High amounts of risk and uncertainty.
- Not a good model for complex and object-oriented projects.
- Poor model for long and ongoing projects.
- Not suitable for the projects where requirements are at a moderate to high risk of changing. So risk and uncertainty is high with this process model.
- It is difficult to measure progress within stages.
- Cannot accommodate changing requirements.
- No working software is produced until late in the life cycle.
- Adjusting scope during the life cycle can end a project.

16. Write phases of spiral model.

The Spiral Model is a Software Development Life Cycle (SDLC) model that provides a systematic and iterative approach to software development. In its diagrammatic representation, looks like a spiral with many loops. The exact number of loops of the spiral is unknown and can vary from project to project. Each loop of the spiral is called a phase of the software development process.

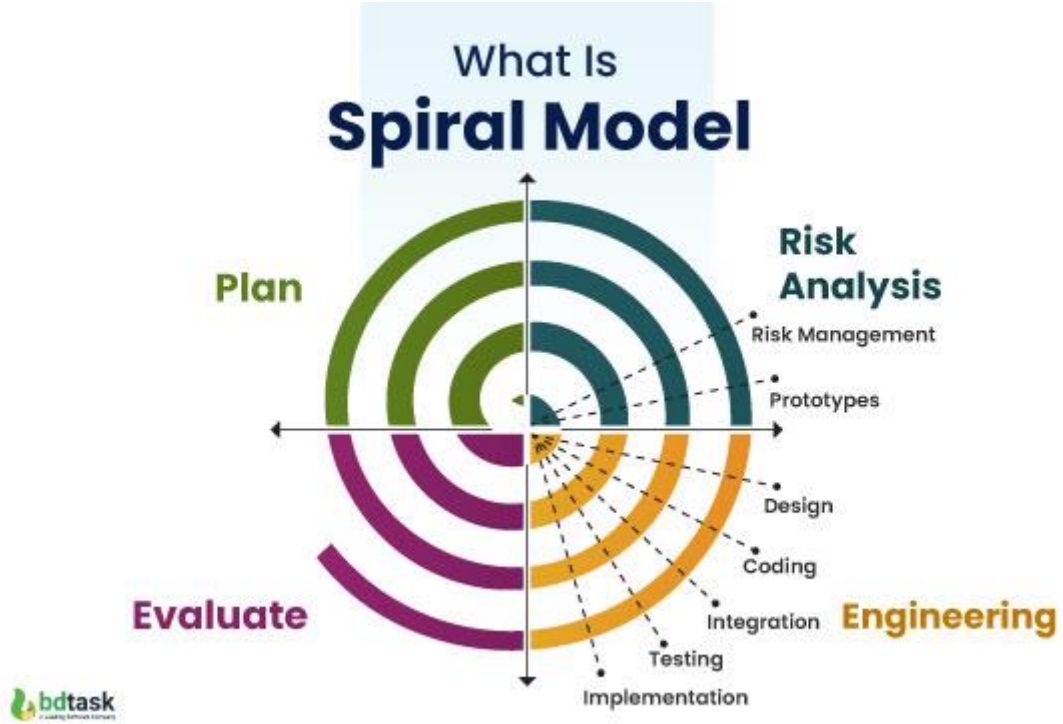
❖ Phases of the Spiral Model

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

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

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

4. **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.



17. Write agile manifesto principles.

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

- 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. • 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.

- 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.

Agile development model is also a type of Iterative Incremental model. Software is developed in incremental, rapid cycles. This results in small incremental releases with each release building on previous functionality. Each release is thoroughly tested to ensure software quality is maintained. It is used for time critical applications.

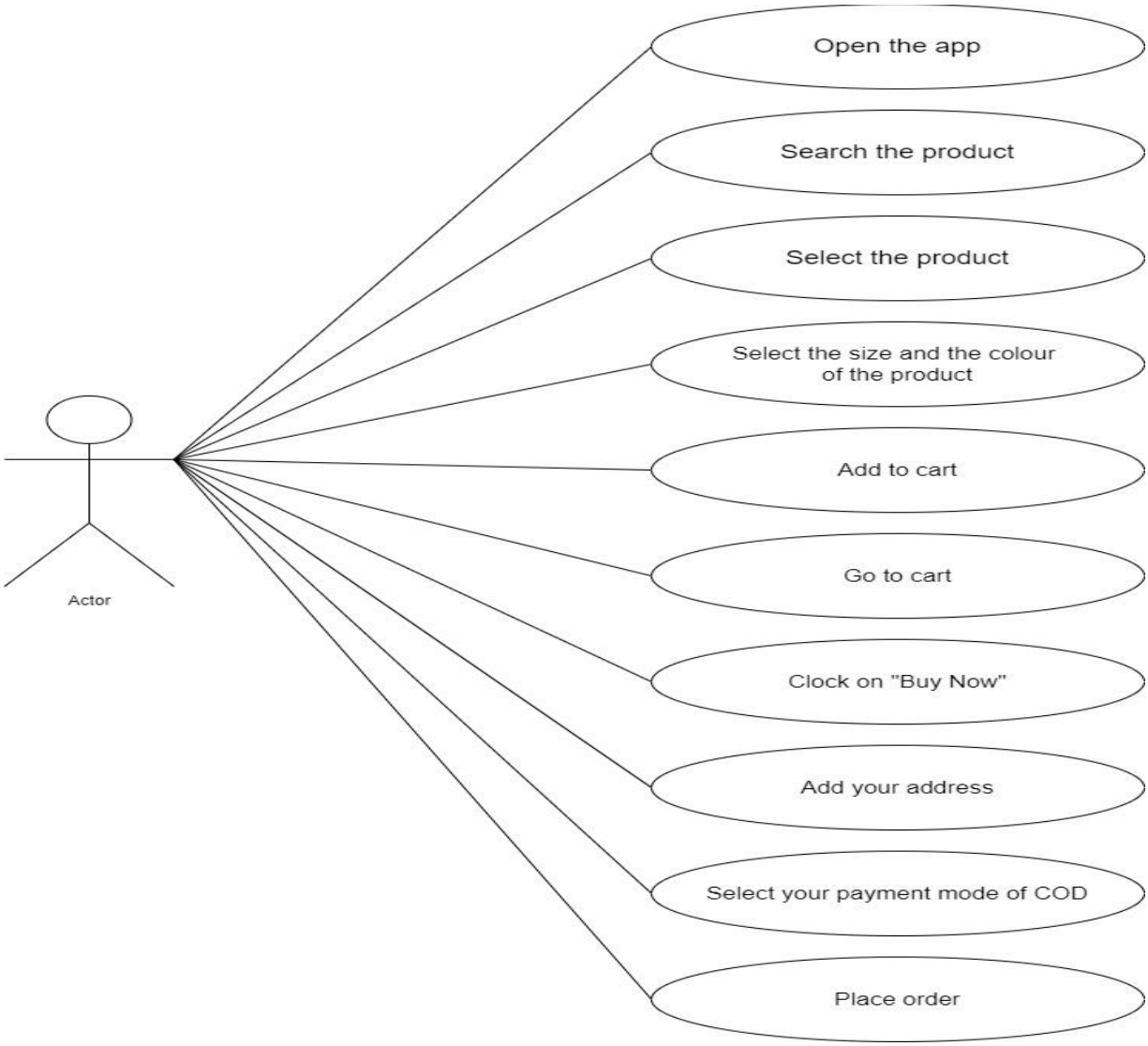
❖ Pros

- It 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.
- There is very high individual dependency, since there is minimum documentation generated.
- Depends heavily on customer interaction, so if customer is not clear, team can be driven in the wrong direction.
- Transfer of technology to new team members may be quite challenging due to lack of documentation.

19. Draw usecase on Online shopping product using COD.



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