Module–1 Fundamental of software Testing

1. What is software testing?

Ans. Software Testing is a process to check whether the actual results is match the expected results. Ensure that developed system or software is defect free.

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

2. What Is SDLC?

Ans. SDLC (Software Development Life Cycle) is a structured process that is used to design, develop and test high quality of software.

The SDLC aims to produce a high-quality software that meets customer expectations, reaches completion within times and cost estimates.

3. Write SDLC phases with basic introduction.

Ans. SDLC Phases

PHASES	DESCRIPTION
Requirements gathering	Gathering business requirements from your client or stakeholders
Analysis	"What do we want?" the team determines the cost and resources required for implementing the analysed requirements.
Design	In this phase, The software design is created.
Implementation	This phase involves building the system, including coding.
Testing	The software is thoroughly tested to ensure that it meets the requirements and works correctly.
Maintenance	This phase includes ongoing support, bug fixes, and updates to the software.

4. Explain Phases of the waterfall model.

Ans. Waterfall model was the first model which was widely used in the software industry.

It is divided into phases and the output of one phase becomes the input of the next phase.

In short, There is no overlapping in the Waterfall model.

In waterfall, the development of one phase starts only when the previous phase is complete.

It is also referred to as a linear-sequential life cycle model.

4 What is agile methodology?

Ans. Agile methodology is a combination of iterative and incremental model.

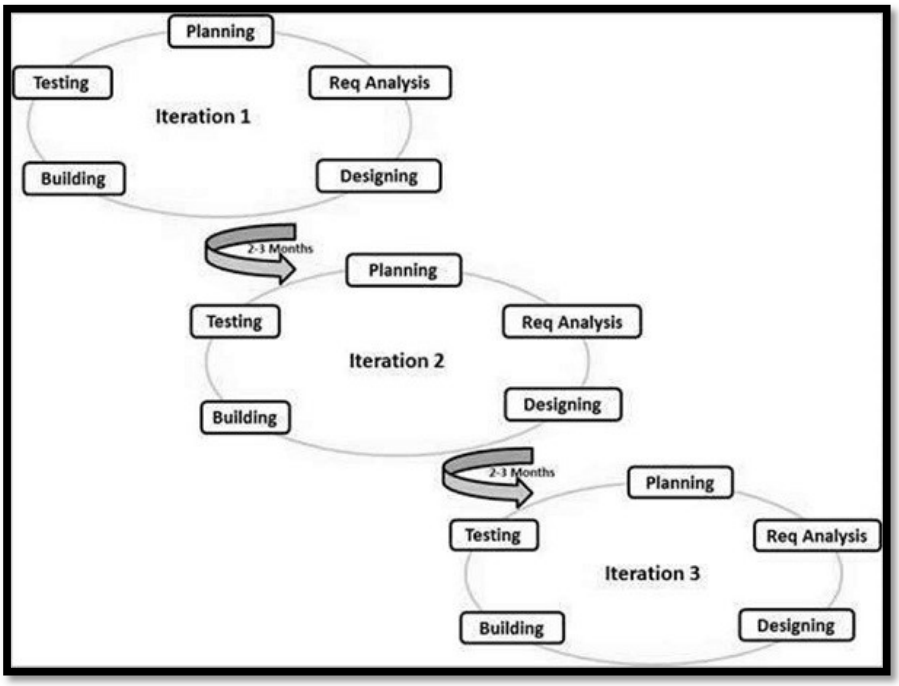
it's known for breaking a project into small incremental builds.

These builds are provided in iterations, that means the big projects are divided into small chunks (iterations).

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

Ans. It is the combination of both iterative and incremental process models.

- Agile Methods break the product into small incremental builds.
- These builds are provided in iterations, that means the big projects are divided into small chunks. (iterations)
- Each iteration typically lasts from about two to four weeks.
- Each iteration involves all the team members working simultaneously on areas like planning, requirements analysis, design, coding, unit testing and acceptance testing.
- At the end of the iteration the working product is displayed to the customer or the important stakeholder and it is released in the market.
- After the release we check for feedback of the deployed software.
- If any enhancement is needed in the project, then it's done and it's re-released.



Advantages of Agile Model

- Project is divided into short and transparent iterations.
- It has a flexible change process.
- It minimizes the risk of software development.
- Quick release of the first product version.
- The correctness of functional requirement is implemented into the development process.
- Customer can see the result and understand whether he/she is satisfied with it or not.

Disadvantages of Agile Model

- The development team should be highly professional and client-oriented.
- With further correction and change, there may be chances that the project will cross the expected time.
- There may be difficult to estimate the final coast of the project due to constant iteration.
- A defined requirement is absent.
- Less documentation.

6. Write Agile manifesto principles

Ans. The Twelve Principle of Agile Manifesto.

1. Customer Satisfaction: Manifesto provides high priority to satisfy the costumer's requirements. This is done through early and continuous delivery of valuable software.

2. Welcome Change: Making changes during software development is common and inevitable. Every changing requirement should be welcome, even in the late development phase. Agile process works to increase the customers' competitive advantage.

3. Deliver the Working Software: Deliver the working software frequently, ranging from a few weeks to a few months with considering the shortest time period.

4. Collaboration: Business people (Scrum Master and Project Owner) and developers must work together during the entire life of a project development phase.

5. Motivation: Projects should be build around motivated team members. Provide such environment that supports individual team members and trust them. It makes them feel responsible for getting the job done thoroughly.

6. Face-to-face Conversation: Face-to-face conversation between Scrum Master and development team and between the Scrum Master and customers for the most efficient and effective method of conveying information to and within a development team.

7. Measure the Progress as per the Working Software: The working software is the key and primary measure of the progress.

8. Maintain Constant Pace: The aim of agile development is sustainable development. All the businesses and users should be able to maintain a constant pace with the project.

9. Monitoring: Pay regular attention to technical excellence and good design to maximize agility.

10. Simplicity: Keep things simple and use simple terms to measure the work that is not completed.

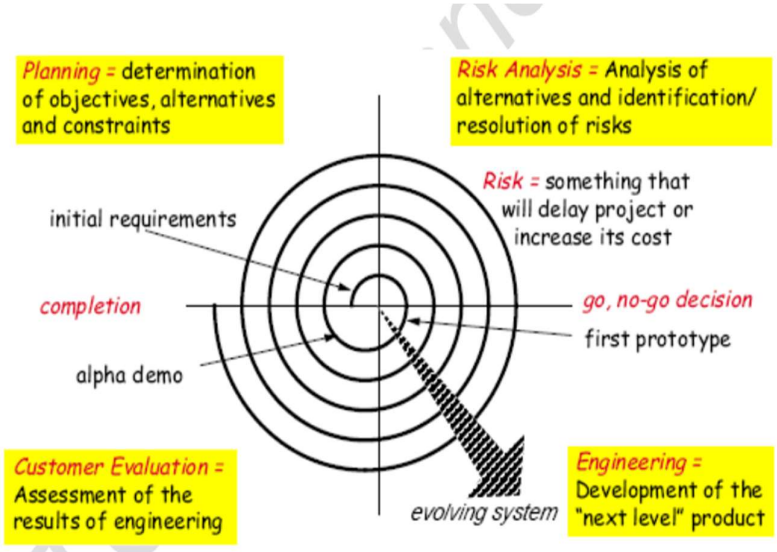
11. Self-organized Teams: The Agile team should be self-organized. They should not be depending heavily on other teams because the best architectures, requirements, and designs emerge from self- organized teams.

12. Review the Work Regularly: The work should be reviewed at regular intervals, so that the team can reflect on how to become more productive and adjust its behaviour accordingly.

7. Write phases of spiral model.

Ans. The spiral model is a SDLC method used for risk management that combines the iterative development process model with elements of the Waterfall model. The spiral model is used by software engineers and is use for large, expensive and complicated projects.

Phases of Spiral Model:



8. What is SRS?

Ans. SRS stands for Software Requirements Specification.
A SRS is complete description of the requirements of the system to be developed.
These requirements can be functional as well as non-functional depending upon the type of requirement.

9. What is OOPS?

Ans. Object Oriented Programming is way off writing the programs in organised way.
Objects are like a black box where data are hidden.
It is based on objects.
It is based on real word.
It provides reusability feature.

10. Write Basic Concepts of oops.

Ans. Basic Concepts of oops are:

- Class
- Object
- Inheritance
- Polymorphism
- Encapsulation
- Abstraction

11. What is object?

Ans. An object is real word entity which have properties and functionality.
Object is also called an instance of class.
Object takes some space in memory.
For example, Fruit is class and its objects are mango, apple, banana.
--OR--
Object gives the permission to the access functionality of class.

12. What is class?

Ans. Class is a collection of data member and member function.
--OR—
Class is a collection of objects. It is don't consume any space in memory.
For Example, Imagine you have a coffee maker with:
- Water inside (private)
- Coffee beans inside (private)
- A button to brew coffee (public)

13. What is encapsulation?

Ans. The process wrapping the data in a single unit. To secure the data from outside the word.
For Example, Imagine you have a coffee maker with:
- Water inside (private)
- Coffee beans inside (private)
- A button to brew coffee (public)

You can press the button to brew coffee, but you can't directly access the water or coffee beans inside the machine. The coffee maker encapsulates the ingredients and the process of brewing coffee, only exposing the button as a public interface.

14. What is inheritance?

Ans. Making a class from an existing class. Deriving the attributes of some other class.

15. What is polymorphism?

Ans. One name multiple form.
For example, a female at the same time a mother, a daughter, an employee.

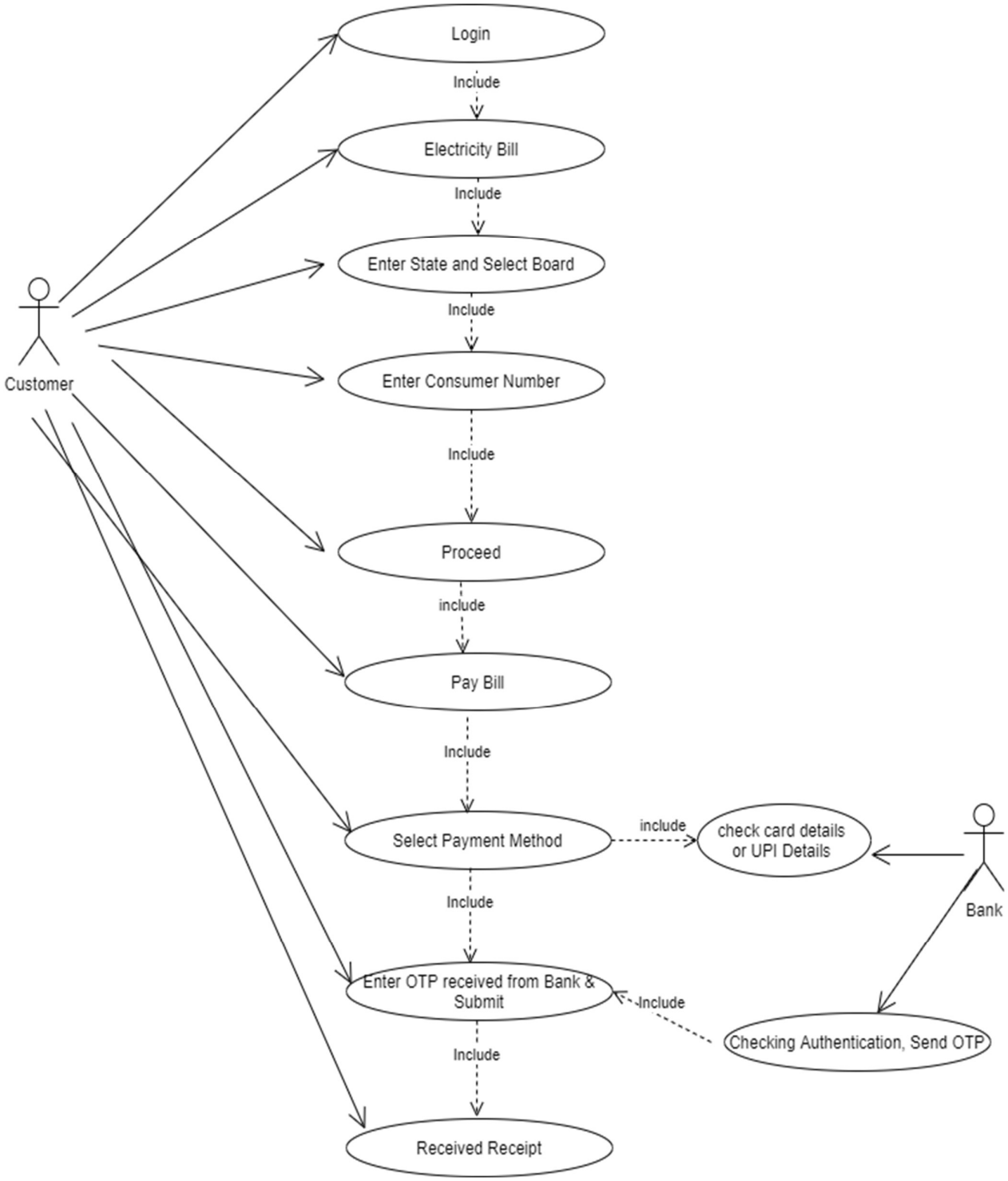
16. What is an abstraction?

Ans. Showing essential information and hiding details.
Shows only necessary information outside.
Lets you use something without knowing how it works inside.
For example, Think of a smartphone

- You use it without knowing how camera, processor and memory work.
- You only care about taking pictures, browsing and using apps.
- The phone hides the complex technology inside, making it easy to use.

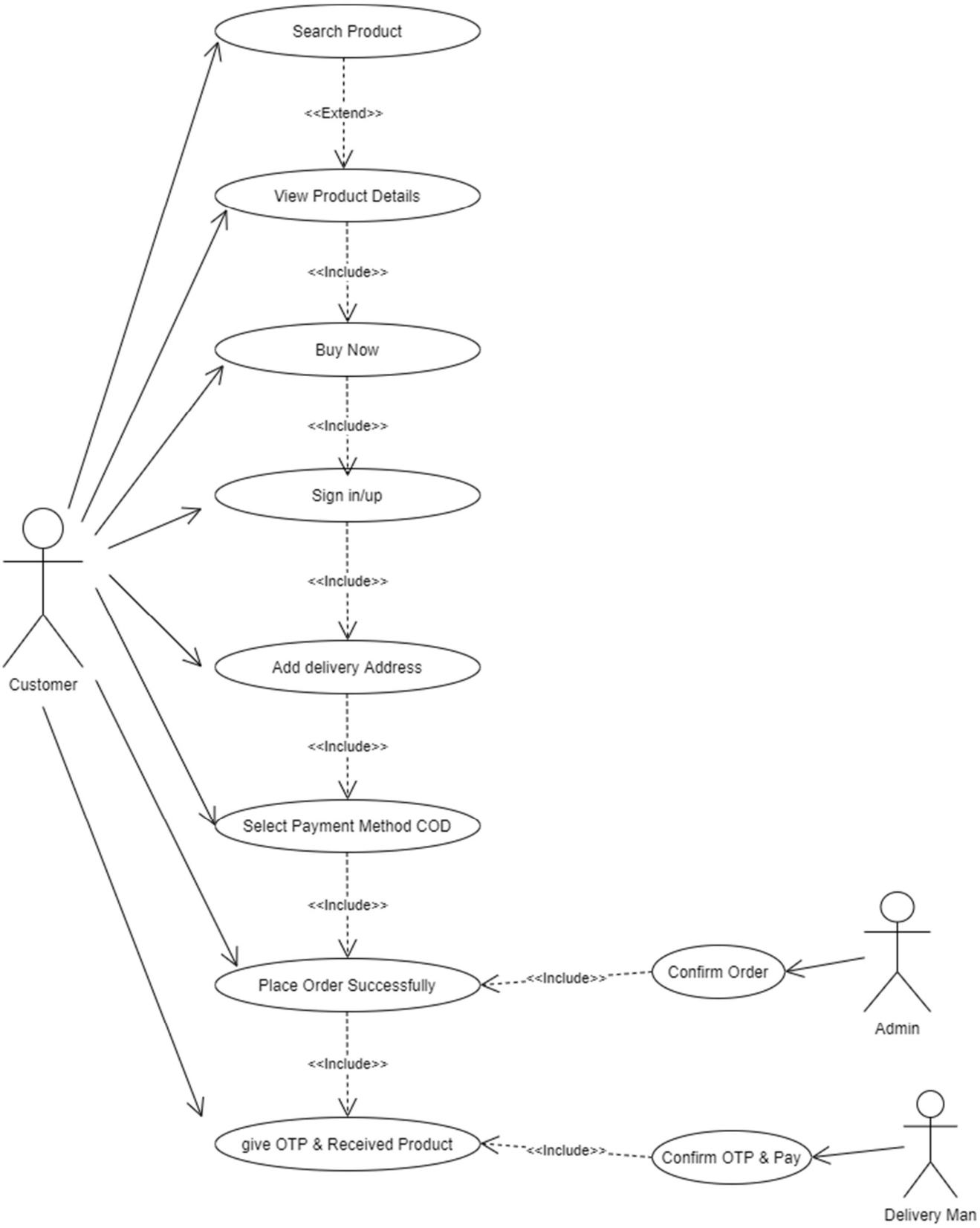
17. Draw Use case on On line book shopping.

18 Draw Use case on online bill payment system (paytm).
Ans



19. Draw usecase on Online shopping product using COD.

Ans.



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

Ans.