

## **Software testing assignment**

### **Module – 1 (Fundamental)**

#### **Q1) What is SDLC?**

**Ans :** Software development life cycle is essentially series of steps or management of application or piece of software that define process for planning, implementation, testing documentation, deployment and ongoing maintenance and support.

#### **Q2) What is software testing?**

**Ans:** Software testing is a process that used to identify the correctness, completeness, and quality of developed computer software.

#### **Q3) What is agile methodology?**

**Ans:** agile SDLC model is a combination of iterative and incremental process model with focus on process adaptability and customer satisfaction by rapid delivery of working software product.

- Agile method breaks the product into small incremental builds.
- These builds are provided in iterations
- Each iteration typically last form about one to three weeks.
- every iteration involves cross functional teams working simultaneously on various areas like planning requirements design, coding, units testing and acceptance testing.

#### **Q4) What is SRS?**

**Ans:** A Software Requirements Specification (SRS) is a complete description of behavior of the system to be developed.

- It includes a set of use cases that describe all of the interactions that the user will have with the software

#### **Q5) What is oops?**

**Ans:** Object oriented programming is way of writing the programs in organized way.

- Objects are like black box where data are hidden.
- Security
- Less space redundancy

□

#### **Q6) Write Basic Concepts of oops.**

**Ans:**

1. Class
2. Objects
3. Inheritance
4. Polymorphism
5. Encapsulation
6. Abstraction

#### **Q7) what is class?**

**Ans:** class is collection of data member and member function.

For example – bank contain all information of customer.

#### **Q8) what is object?**

**Ans:** Object give permission to access functionality of class.

For example – bank have all information about us and they give information regularly which we want like bank balance, withdraw money etc.

#### **Q9) what is encapsulation?**

**Ans:** The process wrapping the data in a single unit to serve the data from outside world.

For example – it is like Mobile it contain all application like call messages etc. we can use any of them with the use of mobile.

#### **Q10) What is inheritance?**

**Ans:** Making a class from an existing class deriving the attribute of some other class

For example – Android version are updated regularly but their main purpose and functionality is same in all versions. Only new feature is added.

#### **Q11) What is polymorphism?**

**Ans:** one name multiple form

For example - Consider the stadium of common wealth games. Single stadium but it perform multiple task like swimming, lawn tennis etc.

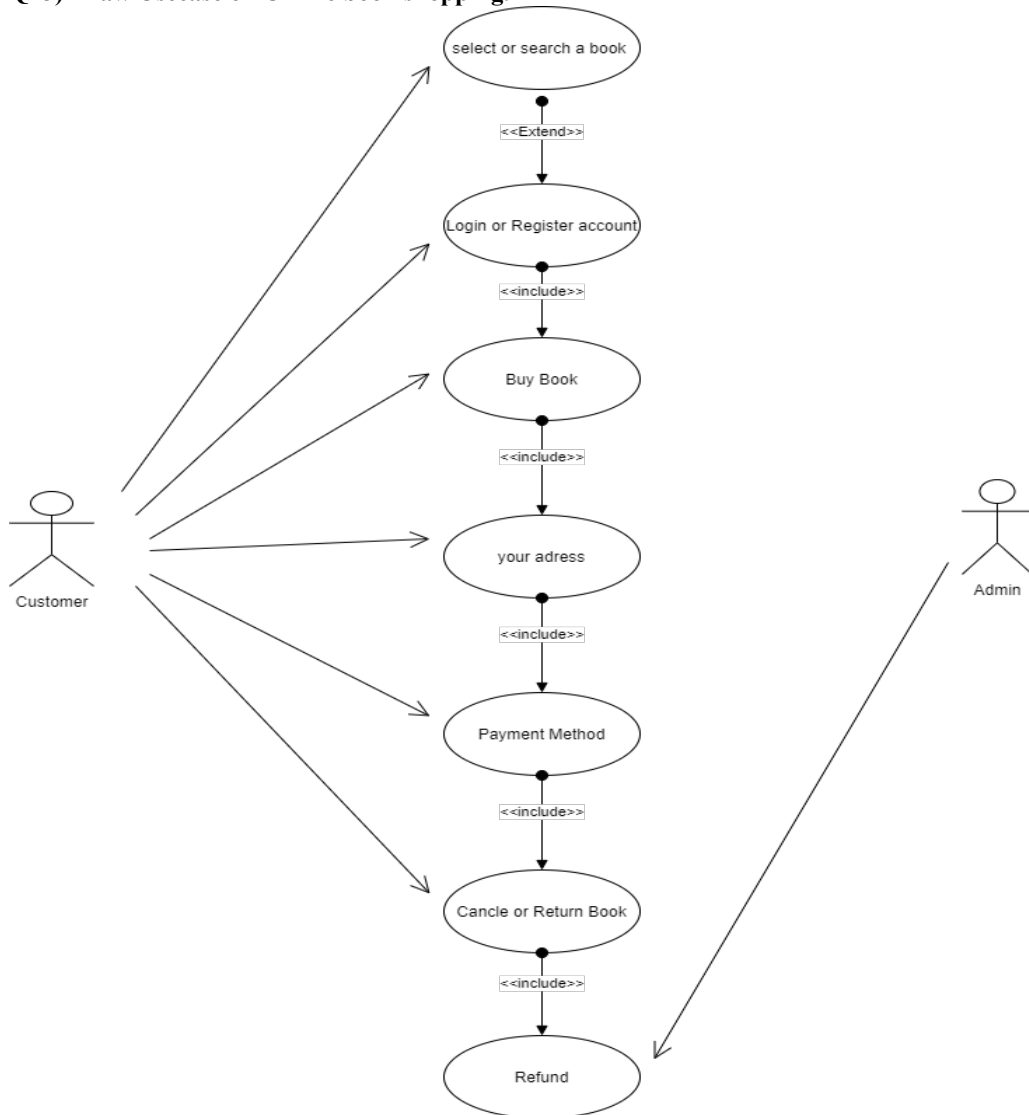
□

**Q12) What is Abstraction?**

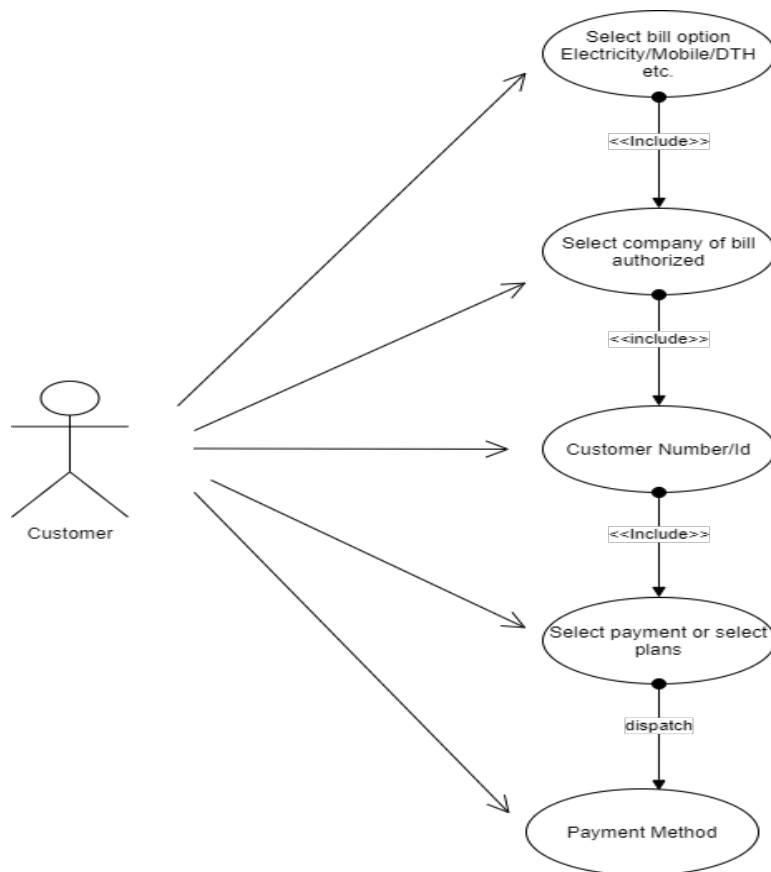
**Ans:** hiding details showing only essential information.

For example - ATM Machine: we can shows only what on the screen but inner mechanism we don't know

**Q13) Draw Usecase on Online book shopping.**



**Q14) Draw Usecase on online bill payment system (paytm).**



**Q15) Write SDLC phases with basic introduction**

**Ans:** There are 6 phase of SDLC,

- |                                     |   |
|-------------------------------------|---|
| 1 Requirements gathering/collection | Establish customer needs                                      |
| 2 Analysis                          | Model and specify the requirement “what”                      |
| 3 design                            | Model and specify the solution “why”                          |
| 4 Implementation                    | Construct the solution in software                            |
| 5 Testing                           | Validate the solution against requirements                    |
| 6 maintenance                       | Repair defects and adopt the solution to the new requirements |

➤ There are 3 major maintenance

- 1) Corrective maintenance: Identifying and repairing defects.
- 2) Adoptive maintenance: Adopt the existing solution to the new platform.
- 3) Perfective maintenance: Implementing the new requirements.

**Q16) Explain Phases of the waterfall model**

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

The waterfall model is unrealistic for many reasons especially :

- Requirements must be ‘frozen’ to early in the life cycle
- Requirements are ‘validated too late.
- There are 6 phase of SDLC,

- |                                     |   |
|-------------------------------------|---|
| 1 Requirements gathering/collection | Establish customer needs                                      |
| 2 Analysis                          | Model and specify the requirement “what”                      |
| 3 design                            | Model and specify the solution “why”                          |
| 4 Implementation                    | Construct the solution in software                            |
| 5 Testing                           | Validate the solution against requirements                    |
| 6 maintenance                       | Repair defects and adopt the solution to the new requirements |

**Q17) Write phases of spiral model**

**Ans:** There are 4 phases of spiral model

1. Planning – determination of objectives, alternatives and constraints.
2. Risk Analysis – Analysis of alternative and identification/resolution of risk.
  - a. Risk= something that will delay project or increase its cost.
3. Engineering – Development of the “next level” product
4. Customer Evaluation – Assessment of the result of engineering.

□

**Q18) Write agile manifesto principles**

**Ans:** Agile Manifest Principles are following :

1. Customer satisfaction through early and continuous software delivery – Customer are happier when they receive working software at regular intervals rather than waiting extended periods of time between release.
2. Accommodate changing requirements throughout the development process – The ability to avoid delays when a requirement or feature request changes.
3. Frequent delivery of working software – Scrum accommodates this principle since the team operates in software sprints or iterations that ensure regular delivery of working software.
4. Collaboration between the business stakeholders and developers throughout the project – Better decisions are made when the business and technical team are aligned.
5. Support, trust, and motivate the people involved – Motivated teams are more likely to deliver their best work than unhappy teams.
6. Enable face-to-face interactions – Communication is more successful when development teams are co-located
7. Working software is the primary measure of progress – Delivering functional software to the customer is the ultimate factor that measures progress.
8. Agile processes to support a consistent development pace – Teams establish a repeatable and maintainable speed at which they can deliver working software, and they repeat it with each release.
9. Attention to technical detail and design enhances agility – The right skills and good design ensures the team can maintain the pace, constantly improve the product, and sustain change.
10. Simplicity – Develop just enough to get the job done for right now.
11. Self-organizing teams encourage great architectures, requirements, and designs – Skilled and motivated team members who have decision-making power, take ownership, communicate regularly with other team members, and share ideas that deliver quality products
12. Regular reflections on how to become more effective – Self-improvement, process improvement, advancing skills, and techniques help team members work more efficiently.

□

**Q19) 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.

**Pros :**

- Is a very realistic approach to software development
- Promotes teamwork and cross training.
- Functionality can be developed rapidly and demonstrated.

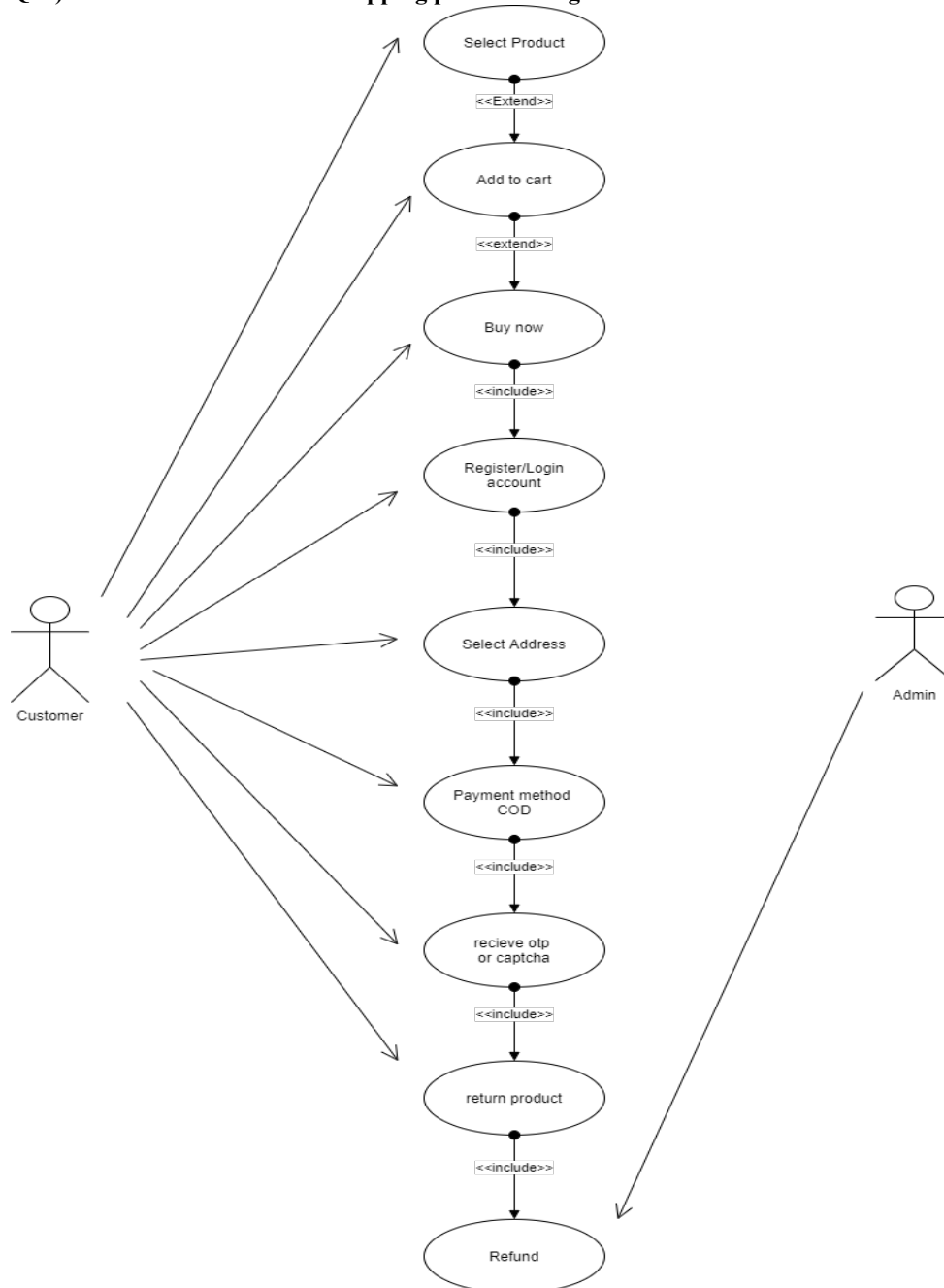
- Resource requirements are minimum.
- Suitable for fixed or changing requirements
- Little or no planning required
- Easy to manage
- Gives flexibility to developers

**Cons:**

- There is very high individual dependency, since there is minimum documentation generated.
- Transfer of technology to new team members may be quite challenging due to lack of documentation.

□

**Q20) Draw usecase on Online shopping product using COD**



**Q21) Draw usecase on Online shopping product using payment gateway**

