• What is SDLC

Ans: SDCL is structure that is impose in the software development that define process of planning, implementation, testing, documentation, development, ongoing, maintenance and support there are different development models.

• What is software testing?

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

• What is agile methodology?

Ans: Agile method is a combination of iterative and incremental process model focus on process adaptability and customer satisfaction. Agile model is break the product into small incremental builds, these builds are provide iteration, every iteration cross functional team working. like planning, requirement, analysis, design, cording, unit testing and acceptance testing.

• What is SRS

Ans: Software requirement specification is a complete description of the behaviour of the system to be development.

Use case are also know functional requirement. But in additions use case of SRS also contains non-functional requirement.

Approaches for the specification of software requirements are described by IEEE-830-1998.

Non-functional requirements are requirements which impose constraints the design or implementation.

There are three types of requirements 1. Customer requirement 2.functional requirement 3. Non-functional requirement

• What is oops

Ans: Oops means object-oriented programming systems is using for identifying object and assigning responsibilities to these objects. Object communicate to others object by sending message. Message are received by the method of an object.an object is like a black box. The internal details are hidden.

• Write Basic Concepts of oops

Ans: there are six concepts of oops

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

• What is class

Ans: class is a collection of data member(variable) and member function (program, method) with it’s behaviour.

Sy:

class classmate

Data member Members function

• What is object

Ans: Object is an instance of class.

To create a memory for that class to access all the priorities of class except private.

Sy:

Classname objectname= new classname();

• What is encapsulation

Ans: encapsulation means data hiding, wrapping of data into single unit i.e.

\*Private your data member and member function\*

• What is inheritance

Ans: inheritance means properties of parent class extend into child class

~Properties of superclass extends into subclass.

~Main purpose is extensibility, reusability.

­­­~There are mainly 5 types

1. Single
2. Multilevel
3. Hierarchical
4. Multiple
5. Hybrid

• What is polymorphism

Ans: polymorphism is ability to take one name having different forms.

multiples forms, Many forms. There are two types method

1. Method overloading
2. Method overriding

• Draw Use case on Online book shopping

Online book shopping

customer

• Draw Use case on online bill payment system (Paytm)

bill payment by Paytm

customer

• Write SDLC phases with basic introduction

Ans: software development life cycle have six phases

**Software Developer Life Cycle**

|  |  |
| --- | --- |
| Req. collection | Collect information what needs to customer from client. |
| Analysis | Model and specify the requirement “what” |
| design | Make a design like a demo our project |
| Developer | Developer will make the project like as a design |
| Testing | After make a project than do testing |
| Maintenance | This apply when product will using by customer. |

**1.req. collection:** There are a two types requirement functional and non-functional requirement.

Functional Requirements: describe system services or Functions, Compute sales tax on a purchase, Update the data-base on the server.

Non-Functional Requirements: are constraints on the system or the development process, Non-functional requirements may be more critical than Functional requirements, if these are not met, the system is useless

Three types of problems have in requirement collection

1. Lack of clarity:

2. Requirements confusion

3. Requirements Amalgamation

**2. Analysis Phase:** The analysis phase defines the requirements of the system, independent of how these requirements will be accomplished.

There are some points of analysis

* + The deliverable result at the end of this phase is a requirement document.
  + Ideally, this document states in a clear and precise fashion what is to build
  + This analysis represents the “what” phase.
  + The requirement documentaries to capture the requirements from the customer's perspective by defining goals.
  + The architecture defines the components, their interfaces and Behaviour.
  + The deliverable design document is the architecture.
  + This phase represents the “how” phase.
  + The design may include the usage of existing components.

**3.Design Phase:**

* + Design Architecture Document
  + Implementation Plan
  + Critical Priority Analysis
  + Performance Analysis
  + Test Plan
  + The Design team can now expand upon the information established in the requirement document.
  + The requirement document must guide this decision process.
  + The architecture team also converts the typical scenarios into a test plan.

**4.Developer Phase:** The implementation phase, the team builds the components either 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. For example, A component may be designed for this particular system.

**5. Testing Phase:**

* Simply stated, quality is very important. Many companies have not learned that quality is important and deliver more claimed functionality but at a lower quality level
* Software maintenance is also one of the phases in the System Development Life Cycle.
* Configuration and version management.
* Updating all analysis, design and user documentation.

**6. Maintenance:** maintenance is the process of changing a system after it has been developed. There are three parts

1. corrective maintenance: identifying and reaping defects.

2. adaptive maintenance: adapting the existing solution to the new platforms

3.perfective maintenance: Implementing the new on decide the utility and value of the software at a particular level of quality outweighs the impact of the known defects and deficiencies.

• Explain Phases of the waterfall model

Ans: There are six phases in waterfall model.

* Requirement
* Analysis
* Design
* Development
* Testing
* Maintenance

The waterfall model is one way model.

Requirement are frozen early in life cycle.

Project is short.

Advantage:

* Simple and easy to use and understand.
* Easy to manage due to high rigidity of the method.
* Esay to arrange task.
* Results are well documented.

Dis-Advantage:

* High amount risk and Uncertainty.
* Model is not good long and ongoing process.
* Model is not complex and object oriented project.
* It is not flexible and not responding to change.

• Write phases of spiral model

Ans: there are a for phases in spiral model.

1. Planning
2. Risk analysis
3. Engineering
4. Customer evaluation

• Write agile manifesto principles

Ans: There are four types principles

* 1. Individual integration
  2. Working software
  3. Customer collaboration
  4. Response into change

• 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.
* At the end of the iteration a working product is displayed to the

**Advantage:**- 1. Changing requirement can be accommodated. 2. for extensive use of Prototypes. 3.Requirements can be captured more properly. 4.Users see the system early. 5.Development can be divided into smaller part and risky part can be developed earlier which helps better risk management.

**Disadvantage:-** 1.Managment is more complex 2.End of project may not be known early. 3.not suitable for small and low risk project. 4. Process is complex. 5. Spiral may go indefinitely. 6. Large number of intermediate stages requires excessive documentation.

• Draw use case on Online shopping product using COD.

Online COD shopping

customer

• Draw use case on Online shopping product using payment gateway

shopping by payment gateway

customer