MODULE - 1

(1 ) What is SDLC

Ans - SDLC is a structure imposed on the development

of a software product that defines the process for planning,

implementation, testing, documentation,deployment,

and ongoing maintenance and support.

- There are a number of different development model.

- A Software Development Life Cycle is essentially a series of steps, or

phases, that provide a model for the development and lifecycle

management of a application or piece of software.

(2) What is software testing?

Ans - Testing is the process of evaluating a system or its

components with the intent to find that whether it satisfies

the specified requirements or not.

- This activity results in the actual, expected and difference between

their result.

- simple words testing is executing a system in order to identify

any gaps, errors or missing requirements in contrary to the

actual desire or requirements.

(3) What is agile methodology?

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.

- iteration involves cross functional teams working

simultaneously on various areas like planning,

requirements analysis, design, coding, unit testing,

and acceptance testing.

(4) What is SRS

Ans - A software requirements specification (SRS)

is a complete description of the behavior of the system

to be developed.

- It includes a set of use cases that describe all of the interactions that

the users will have with the software.

- Use cases are also known as functional requirements. In

addition to use cases, the SRS also contains nonfunctional

or supplementary requirement.

- This standard describes possible structures, desirable contents, and

qualitie of a software requirements specification.

(5) What is oops

Ans - Identifying objects and assigning responsibilities to these objects.

- Objects communicate to other objects by sending messages.

- Messages are received by the method of an object

- An object is like a black box.

- The internal details are hidden.

- Object is derived from abstract data type

- Object-oriented programming has a web of intracting

objects, each house-keeping its own state.

(6) Write Basic Concepts of oops

Ans - Object,Class,Encapsulation,Inheritance,Polymorphism,Overriding,

Overloading,Abstraction these all are basic concepts of oops

(7) What is object

Ans - An object represents an individual, identifiable item, unit, or entity,

either real or abstract, with a well-defined role in the problem domain.

- An "object" is anything to which aconcept applies.

- This is the basic unit of object oriented programming.

- That is both data and function that operate on data are bundled as a

unit called as object.

(8) What is class

Ans - When you define a class, you define a blueprint for an object.

- This doesn't actually define any data, but it does define what

the class name means, that is, what an object of the class will

consist of and what operations can be performed on such an object.

- A class represents an abstraction of the object and the

properties and behavior of that object.

(9) What is encapsulation

Ans - Encapsulation is the practice of including in an object everything

it needs hidden from other objects. The internal state is usually

not accessible by other objects.

- Encapsulation is placing the data and the functions that work on that

data in the same place. While working with procedural languages, it is

not always clear which functions work on which variables

but object orient programming provides you framework to

place the data and the relevant functions together in the same object.

(10) What is inheritance

Ans - Inheritance means that one class inherits the characteristics of

another class. This is also called a “is a” relationship

- One of the most useful aspects of object-oriented programming

is code reusability. As the name suggested Inheritance is the

process of forming a new class from an existing class that is from

the existing class called as base class, new class is formed called

as derived class.

(11) What is polymorphism

Ans - Polymorphism means “having many forms”.

It's allows different objects to respond to the same message

in different ways, the response specifice to the type of the object.

- The ability to use an operator or function in different ways in

other words giving different meaning or functions to the

operators or functions is called polymorphism.

(12) Draw Usecase on Online book shopping

Ans - <https://drive.google.com/file/d/1zh-6KuT-HvgZzdWFLmnsoVEr33XcF3n0/view>

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

Ans -

<https://drive.google.com/file/d/1UP6r7_M0cBN4OIQRW36lMmN7ekmb1udN/view>

(14) Write SDLC phases with basic introduction

Ans -

(1) Requirement Gathering

- Although requirements may be documented in written form,

they may be incomplete, unambiguous, or even incorrect.

- Requirements will Change

- Validation is needed throughout the software lifecycle, not only when

the “final system” is delivered.

(2) Analysis Phase

- The analysis phase defines the requirements of the system,

independent of how these requirements will be accomplished.

- This phase defines the problem that the customer is trying to solve.

- The deliverable result at the end of this phase is a

requirement document.

(3) Design Phase

- The Design team can now expand upon the information established in

the requirement document.

- The requirement document must guide this decision process.

Analyzing the trade-offs of necessary complexity allows for many

things to remain simple which, in turn, will eventually lead to a higher

quality product.

- The architecture team also converts the typical scenario into a test

plan.

(4) Implementation Phase

-In the implementation phase, the team builds the components either

from scratch or by composition.

-Given the architecture 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 and flexibility.

-For example, a component may be narrowly designed for

this particular system, or the component may be made more

general to satisfy a reusability guideline.

(5) Testing Phase

-Simply stated, quality is very important. Many companies have not

learned that quality is important and deliver claimed

functionality but at a lower quality level.

-It is much easier to explain to a customer why there is a

missing feature than to explain to a customer why the product

lacks quality.

- A customer satisfied with the quality of a product will remain loyal and

wait for new functionality in the next version.

-Quality is a distinguishing attribute of a system indicating the

degree of excellence.

(6) Maintenance Phase

-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.

-Software maintenance is also one of the phases in the System

Development Life Cycle , as it applies to software development.

-The maintenance phase is the phase which comes after deployment of

the software into the field.

(15) Explain Phases of the waterfall model

Ans - The waterfall is unrealistic for many reasons,

especially:

-Requirements must be “frozen” to early in the life cycle

-Requirements are validated too late

- Applications(When to use?)

-Requirement 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.

(16) Write phases of spiral model

Ans -

-Spiral Model is very widely used in the software industry as it is

in synch with the natural development process of any product

i.e. learning with maturity and also involves minimum risk for

the customer as well as the development firms. Following are

the typical uses of Spiral model:

When costs there are a budget constraint and risk evaluation is

important.

- For medium to high risk projects.

-Long-term project commitment because of potential changes to

economic priorities as the requirements change with time.

-Customer is not sure of their requirements which are usually the case.

-Requirements are complex and need evaluation to get clarity.

-New product line which should be released in phases to get enough

customer feedback.

-Significant changes are expected in the product during the

development cycle.

(17) Write agile manifesto principles

Ans- 1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.

2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's advantage.

3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference for a shorter timescale.

4. Business people and developers must work together daily throughout the project.

5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.

6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.

7. Working software is the primary of progress.

8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

9. Continuous attention to technical excellence and good design enhances agility.

10. Simplicity the art of maximizing the amount of work not done — is essential.

11. The best architecture, requirements, and designs emerge from self-organizing teams.

12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts behavior accordingly.

(18) 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 last 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 training.

-Functionality can be developed rapidly and demonstrated.

Resource requirements are minimum.

-Suitable for fixed or changing requirements

Delivers early partial working solutions.

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.

(19) Draw usecase on Online shopping product using COD.

Ans - <https://drive.google.com/file/d/1713szUPqP0LVJbp43ZWT7k0m8LkP_DsJ/view>

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

Ans - <https://drive.google.com/file/d/1NnmQ3trIBLfhmUYqG1qT9eIk7scO6DAB/view>