Assignment no.1

**Que.1:** What is software testing?

**Answer:** Software testing is the process used to identify the correctness, completeness

And quality of developed computer software.

**Que.2:** What is SDLC?

**Answer:** 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.

**Que.3:** Write SDLC phases with basic introduction?

**Answer:** There are 6 SDLC phases,

|  |  |
| --- | --- |
| Requirements collection/Gathering | Establish Customer Needs |
| Analysis | Model And Specify the requirements “ What ” |
| Design | Model And Specify a solution – “Why” |
| Implementation | Construct a solution in Software |
| Testing | Validate the solution against the requirements |
| Maintenance | Repair defects and adapt the solution to the new requirements |

1. Requirement Collection/ Gathering:-

* Features
* Usage scenarios
* Although requirements may be documented in written form, they may be incomplete or incorrect.
* Requirements will change.
* Inadequately captured or expressed in the first place
* User and business needs change during the project.
* Build constant feedback into the project plan.
* Plan for change
* Functional and Non-Functional

**Three types of problems can arise:**

* **Lack of clarity:** It is hard to write documents that are both precise and easy to read.
* **Requirements confusion:** Functional and Non-Functional requirements tend to be intertwined.
* **Requirements Amalgamation:** Several different requirements may be expressed together.

**Types of requirements:**

* Functional Requirements: - Describe system services or functions.
* Compute sales tax on a purchase
* Update the database on the server
* Non-Functional Requirements:-Non-functional requirements are constraints on the system.

2. Analysis phase:

* The analysis phase defines the requirements of the system, independent of how these requirements will be accomplished.
* This phase represents the “how” phase.
* 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.
* Ideally, this document states in a clear and precise fashion what is to be built.
* This analysis represents the “what” phase

3. Design phase:

* Design Architecture Document
* Implementation plan
* Critical priority Analysis
* Performance analysis
* Test Plan
* The requirement document must guide this decision process.

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, through there is still room for innovation and flexibility.
* The implementation phase deals with issues of quality, performance, baselines, libraries and debugging.
* The end deliverable is the product itself.

5. Testing Phase:

* Simply stated, quality is very important. Many companies have not quality is important and deliver more claimed functionality but at a lower quality level.
* Regression Testing
* Internal Testing
* Unit Testing
* Application Testing
* Stress testing
* Configuration and version management
* Reengineering
* Updating all analysis, design and user documentation.

6. Maintenance Phase:

* **Corrective Maintenance:** Identifying and repairing defects
* **Adaptive Maintenance:** Adapting the existing solution to the new platforms.
* **Perfective Maintenance:** Implementing the new requirements in a spiral lifecycle, everything after delivery and deployment of the first prototype can be considered “maintenance”.

**Que.4:** What is agile methodology?

**Answer:**

1. Agile model believes that every project needs to be handled differently and the existing methods need to be tailored to best suit the project requirements.
2. In agile the tasks are divided to time boxes to deliver specific features for a release.
3. 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.

**Que.5:** Explain Phases of the waterfall model?

**Answer:** The classical software lifecycle models the software development as a step by step

Waterfall between the various development phases.

1. Requirement Collection/ Gathering:-

1. Features
2. Usage scenarios
3. Requirements will change.
4. Plan for change
5. Functional and Non-Functional

**Three types of problems can arise:**

* **Lack of clarity**: It is hard to write documents that are both precise and easy to read.
* **Requirements confusion**: Functional and Non-Functional requirements tend to be intertwined.
* **Requirements Amalgamation**: Several different requirements may be expressed together.

**Types of requirements;**

1. Functional Requirements: - Describe system services or functions.

* Compute sales tax on a purchase
* Update the database on the server

1. Non-Functional Requirements:-Non-functional requirements are constraints on the system.

2. Analysis phase:

1. The analysis phase defines the requirements of the system, independent of how these requirements will be accomplished.

2. This phase represents the “how” phase.

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

4. The deliverable result at the end of this phase is a requirement document.

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

6. This analysis represents the “what” phase.

3. Design phase:

1. Design Architecture Document
2. Implementation plan
3. Critical priority Analysis
4. Performance analysis
5. Test Plan
6. The requirement document must guide this decision process.

4. Implementation Phase:

1. In the implementation phase, the team builds the components either from scratch or by composition.
2. 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, through there is still room for innovation and flexibility.
3. The implementation phase deals with issues of quality, performance, baselines, libraries and debugging.
4. The end deliverable is the product itself.

5. Testing Phase:

1. Simply stated, quality is very important. Many companies have not quality is important and deliver more claimed functionality but at a lower quality level.
2. Regression Testing
3. Internal Testing
4. Unit Testing
5. Application Testing
6. Stress testing
7. Configuration and version management
8. Reengineering
9. Updating all analysis, design and user documentation.
10. Maintenance Phase:
11. Corrective Maintenance : Identifying and repairing defects
12. Adaptive Maintenance: Adapting the existing solution to the new platforms.
13. Perfective Maintenance: Implementing the new requirements in a spiral lifecycle, everything after delivery and deployment of the first prototype can be considered “maintenance”.

**Que.6:** What is SRS?

**Answer:** A software requirement 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.

**Que.7:**  What is OOPS?

**Answer:** Identifying objects and assigning responsibilities to these objects. Object communicate to other object by sending massages. An object is like a black box.

**Que.8:** Write basic concepts of OOPS?

**Answer:** There are some basic concepts of OOPS

* Object- Instances of a class. To create memory for that class to access the whole properties on an class except private
* Class- Is an collection of data member (variable) and member function (process, methods) with its behavior
* Encapsulation-Private your data member and member function.
* Inheritance-Properties of parent class extends into child class.
* Polymorphism-Ability to take one name having many forms , multiple , different forms
* Abstraction-Data hiding in extended level.

**Que.9:**  What is object?

**Answer:** Instances of a class. To create memory for that class to access the whole properties on a class except private

**Que.10:**  What is Class?

**Answer:** Is a collection of data member (variable) and member function (process, methods) with its behavior

**Que.11:** What is Encapsulation?

**Answer:** 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.

**Que.12:** What is Inheritance?

**Answer:** Inheritance means that one class inherits the characteristics of another class. This is also called a “is a” relationship.

**Que.13:** What is Polymorphism?

**Answer:** Polymorphism means “having many forms”. It allows different objects to respond to the same message in different ways, the response specific to the object.

**Que.14:** Write phases of spiral model?

**Answer:**

* Planning-Determination of objectives, alternatives and constraints.
* Risk Analysis-Analysis of alternatives and identification/resolution of risks.
* Customer Evaluation-Assessment of the results of engineering.
* Engineering-Development of the “next level” product.

**Que.15:** Write agile manifesto principles?

**Answer:**

* Individual interaction
* Working Software
* Customer Collaboration
* Respond into change

**Que.16:** Explain working methodology of agile model and also write pros and cons?

**Answer:** 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. This builds are provided in iterations. Customer and important stakeholders.

**Pros:**

* Promotes teamwork and cross training.
* Resource requirement 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.
* 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 if documentation.

**Que.17:** Draw Use Case on online book shopping

**Answer:**

Log In

Search Product

Add to Cart

Select credit card

Purchase Book

Review order details

Customer Select Address

Pay Payment

**Que.18: Draw Use-Case on online bill payment (Paytm)**

**Answer:**

Open Paytm Application

Select Electricity Bill Option

Select Electricity Board

Select State

Select Sub-division

Type Customer No details

Customer Select City

Pay Bill Payment

**Que.19: Draw Use-Case on online shopping product using COD**

**Answer:**

Log In

Search Product

Add to Cart

Select Address

Select Payment method

Select COD

Customer Place order

Deliver product on COD

**Que.20: Draw Use-Case on online shopping product using payment gateway.**

**Answer:**

Log In

Search Product

Add to Cart

Select Address

Select Payment method

Select Gateway

Customer Select Payment Option

Make Payment

Deliver Product