**MODULE 1**

**Software testing assignment**

|  |  |
| --- | --- |
| **Q 1** | **What is SDLC?** |
| **Answer** | SDLC: Software development lifecycle   * It is an abstract, high level representation of software development process. * SDLC is essentially a series of steps or phases, that provide a model for the development and lifecycle management of an application software or piece of software. * It is a model that defines how different development phases and types of activities performed within this process relate to each other, both logically and chronologically.   There are six phases of SDLC:   1. Requirement gathering: Establish a customer needs 2. Analysis: Model and specify requirement - “What” 3. Design: model and specify a solution – “Why” 4. Implementation: Construct a solution in software 5. Testing: validate the solution against the requirements 6. Maintenance: Repair defect and adapt the solution to the new requirement |
| **Q 2** | **What is Software Testing?** |
| **Answer** | Software testing is the set of activities to discover defects and evaluate quality of software work products.  Two types of Testing: Static and dynamic   1. Static testing:  * Includes reviews and static analysis. * It does not involve execution of software.  1. Dynamic testing:  * Involves execution of software * It uses different types of test techniques and test approaches to derive test cases. |
| **Q 3** | **What is agile methodology?** |
| **Answer** | * Agile methodology is a combination of iterative and incremental process. * It divides software into small incremental builds, this build is provided in iterations, that means the big projects are divided into small chunks. * Each iteration last about two to four weeks. * Each iteration involves all the team members working simultaneously on areas like planning, requirement 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 stackholder and its released in the market. * After the release we check for the feedback of the deployed software. * If any enhancement is needed in the project then it’s done and it’s released.   Advantages:   * Frequent delivery * Face to fac communication with the customer * Less time * Adaptability   Disadvantage:   * Less documentation * Maintenance problem |
| **Q 4** | **What is SRS?** |
| **Answer** | SRS: Software requirement specification  SRS is the complete description of the behaviour of the system (application) to be developed.  It includes a set of use cases that describe all of the interactions that the users will have with the software.  SRS contains both functional requirements (use cases) and non-functional requirements. |
| **Q 5** | **What is OOPS?** |
| **Answer** | OOPS: Object oriented programming  It is the way of writing the programs in the organised way.  Objects are black box where data are hidden.  Objects of the program interact by sending messages to each other. |
| **Q 6** | **Write basic concepts of OOPS** |
| **Answer** | * Object * Class * Inheritance * Polymorphism * Encapsulation * Abstraction |
| **Q 7** | **What is object?** |
| **Answer** | Objects give the permission to access the functionality of the class.  It is the basic unit of the Object-oriented programming. |
| **Q 8** | **What is class?** |
| **Answer** | Class is the collection of data members and member functions.  It represents an abstraction of the object and abstracts the properties and behaviour of that object. |
| **Q 9** | **What is Encapsulation?** |
| **Answer** | The process wrapping the data in a single unit to secure the data from outside world.  Encapsulations enable the data hiding, hiding irrelevant information from the users of the class and exposing only the relevant details required by the users. |
| **Q 10** | **What is inheritance?** |
| **Answer** | Inheritance means that one class inherits the characteristics of another class.  In other word, making a class from an existing class, deriving the attributes of some other class. |
| **Q 11** | **What is polymorphism?** |
| **Answer** | Polymorphism means “Having many forms”  It allows different objects to respond to the same message in the different ways, the response specific to the type of the object. |
| **Q 12** | **Draw Usecase on Online book shopping** |
| **Answer** |  |
| **Q 13** | **Draw Usecase on Online bil payment system (paytm)** |
| **Answer** |  |
| **Q 14** | **Write SDLC phases with basic introduction** |
| **Answer** | SDLC has 6 phases:   |  |  | | --- | --- | | Requirement Collection/ gathering | Establish customer needs | | Analysis | Model and specify requirements-‘what’ | | Design | Model and specify the 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 Gathering:**  * Requirements definitions usually consist of natural language, supplemented by UML, Diagrams and tables. * There are two types of requirements:   1. Functional Requirements: Describes system services or functions   2. Non-Functional requirements: are constraints on the systems or the development process * **The deliverable** result at the end of this phase is **Customer requirement document.**  1. **Analysis:**  * The analysis phase defines the requirements of the system, independent of how these requirements will be accomplished. * This phase defines the problem that customer is trying to solve * which states in a clear and precise fashion what is to be built. * **The deliverable** result at the end of this phase is **requirement document**, which states in a clear and precise fashion what is to be built. This analysis represents the ‘What’ phase.   **How:**   * This phase starts with the requirement document delivered by the requirement phase and maps the requirements into architecture. * The architecture defines the components, their interfaces and behaviour. * The **deliverable design document** is the **architecture**. * This phase represents the ‘How’ phase. * Details on computer programming languages and the environments, machines, packages, application architecture, distributed architecture layering, memory size, platform, algorithms, data structures, global type definitions, interfaces, and many other engineering details are established.  1. **Design**  * 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 the test plan.  1. **Implementation:**    * In the implementation phase, the team build up the components either from the 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.    * The end deliverables is the product itself. 2. **Testing:**  * The testing phase is a separate phase which is performed by a different team after the implementation is completed. * Testing ensures the quality of the product.  1. **Maintenance:**  * Software maintenance is one of the activities in the software engineering, and is the process of enhancing and optimizing deployed software, as well as fixing defects. * The maintenance phase comes after deployment of the software into the field. |
| **Q 15** | **Explain phases of the waterfall model** |
| **Answer** | The waterfall models having 6 phases:   |  |  | | --- | --- | | Requirement Collection/ gathering | Establish customer needs  Deliverable: CRS (Customer requirement specification) | | Analysis | Model and specify requirements-‘what’  Deliverable: SRS (Software requirement specification) | | Design | Model and specify the solution-‘Why’  Deliverable: Design specification | | Implementation | Construct a solution in software  Deliverable: build | | Testing | Validate the solution against the requirements  Deliverable: release | | Maintenance | Repair defects and adapt the solution to the new requirements  Deliverable: Bug fixing after deployment, new feature added | |
| **Q 16** | **Write phases of spiral model** |
| **Answer** | Spiral model has 4 phases:   |  |  | | --- | --- | | Planning | Determination of the objectives, alternatives and constraints | | Risk analysis | Analysis of alternatives and identification/resolutions of risks | | Engineering | Development of the ‘next level’ product | | Customer evaluation | Assessment of the result of engineering | |
| **Q 17** | **Write agile manifesto principles** |
| **Answer** | * **Individuals and interactions:** In the agile development, self-organization and motivation are important, as are interactions like co-location and pair programming * **Working software:** Demo working software is considered the best means of communication with the customer to understand their requirement, instead just depending on documentation * **Customer collaboration:** As the requirements cannot be gathered completely in the beginning of the project due to various factors, continuous customer interaction is very important to get proper product requirement. * **Responding to change:** Agile development is focused on quick responses to change and continuous development. |
| **Q 17** | **Explain working methology of agile model and also write pros and cons** |
| **Answer** | Working methodology:   * Agile methodology is a combination of iterative and incremental process. * It divides software into small incremental builds, this build is provided in iterations, that means the big projects are divided into small chunks. * Each iteration last about two to four weeks. * Each iteration involves all the team members working simultaneously on areas like planning, requirement 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 its released in the market. * After the release we check for the feedback of the deployed software. * If any enhancement is needed in the project, then it’s done and it’s released.   **Pros:**   * It is 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 and changing requiremnts * Delivers early partial working solutions * Good model for environments that change steadily * Minimal rules, documentation easily employed * Enables concurrent development and delivery within an overall planned context. * Little or no planning required * Easy to manage * Gives flexibility to developers   **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 interactions, so if customer is not clear, team can be driven in wrong direction * There is very high individual dependency, since there is minimum documentation generated. * Transfer of technology to new team members may be quite challenging |
| **Q 17** | **Draw usecase on Online shopping product using COD** |
| **Answer** |  |
| **Q 17** | **Draw usecase on online shopping product using payment gateway.** |
| **Answer** |  |