**SOFTWARE TESTING ASSIGNMENT # 1**

**Module – 1(Fundamental)**



**SUBMITED BY:**

CHIRAG MODHA

**SUBMITED TO:**

RAHUL SIR

**DEPARTMENT OF COMPUTER SCIENCE**

**TOPS TECHNOLOGY OF SG-HIGHWAY BRANCH**

* **What is SDLC**?
* SDLC stand for software development life cycle.
* SDLC is step-by-step process to development of a software product, that define the process for planning, implementation, testing, documentation, deployment, and ongoing maintenance and support.

* **What is software testing?**
* Software testing is a process, used to identify correctness, completeness, and quality of the development computer software.
* **What is agile methodology?**
* Agile Methodology meaning a practice that promotes **continuous iteration** of development and continuous testing throughout the software development lifecycle of the project.
* **What is SRS?**

* A software requirement specification (SRS) is complete description of the behaviour of the system to be developed.
* **What is oops?**
* Oops is stands for object-oriented programming system.
* Oops is used to create object and Re-use of object throughout the program and manipulate these objects to get result.
* **Write basic concept of oops**
* Objects
* Class
* Encapsulation
* Inheritance
* Polymorphism
* Abstraction
* **What is object?**
* An object is an instance of class, whenever class is defined, no memory is allocated but when we initialized the object then class is allocated the memory.
* **What is class?**
* Class is user define datatype, which holds its own data member and member function in other word we can say class is collection of data member and member function. Which can access and use by creating object of that class.
* When you define a class, you define a blueprint for an object.
* **What is encapsulation?**
* Encapsulation is the process of wrapping data (properties) and behaviour (method) of an object into single unit and the unit hear is class (or interface).
* Encapsulation is the process of including in object everything needs hidden from other objects, the internal state is usually not accessible by other objects.
* Encapsulation enables data hiding, hiding irrelevant information from the user of a class and exposing only relevant details required by the users.
* **What is inheritance?**
* Inheritance means one class inherits characteristics of another class.
* This is very useful concept of oops since this feature reduce the code size.
* Code reusability can be achieved through this concept.
* **What is polymorphism?**
* Polymorphism means having many forms.
* It allows different objects to respond to the same message in different way response specific to the type of the object.
* The ability to change form is known as polymorphism.
* In their two types 1) overloading, and 2) overriding.
* **Draw use case on online book shopping**

**Open app**

**login**

**Search book**

**Registered customer**

**Add to cart**

**Confirm order**

**Make payment**

**Review and confirm order**

* **Draw usecase on online bill payment system (Paytm)**

**Recharge & bill payment**

**Paytm**

**Electricity bill**

**Consumer name**

**Get receipt**

**Confirm payment**

**Fill amount**

**Confirm details**

**Consumer num**

**select electricity board**

**Select state**

* **Write SDLC phases with basic introduction**

|  |  |
| --- | --- |
| **REQUIRMENTS**  **COLLECTION/GEATHERING** | **Establish customer needs** |
| **ANALYSIS** | **Model and specify the requirements “what”** |
| **DESIGN** | **Model and specify a solution “why”** |
| **IMPLIMENTATION** | **Construct a solution in software** |
| **TESTING** | **Validate the solution against the requirements** |
| **MAINTENANCE** | **Repair defect and adapt the solution to the new requirements** |

* **Explain phases of the waterfall model**

1. **Requirement gathering stage:**

* During this phase all possible requirement of the system to be developed are captured in this phase and documented in a requirement specification document.

1. **Design stage:**

* In this stage plan the programming language or more high-level technical details of the project.

1. **Built stage:**

* In this stage developer can do the coding which is suitable for gathering documents or design.

1. **Deployment phase:**

* Deploy the system in the respective environment.

1. **Maintenance phase:**

* Once your system is ready to use, you may later require change the code as per customer request.
* **Explain phases of spiral model**

1. **Planning:**

* It includes estimating the coast, schedule, and resources for the iteration it also involves understanding the system requirements for continuous communication between the system analyst and the customer.

1. **Risk analysis:**

* Identification of potential risk is done while risk mitigation strategy is planned and finalized.

1. **Engineering:**

* It includes testing coding and deploying software at the customer side.

1. **Evaluation:**

* Evaluation of software by customer also includes identifying and monitoring risk such as schedule slippage and cost overrun
* **Write agile manifesto principal**

* Individual interaction
* Working software
* Customer collaboration
* Responding to change
* **Explain working methodology of agile model and write pros and cons**
* Agile SDLC model is combination of iterative and incremental process model which focus on process adaptability and customer satisfaction by rapid delivery of working software product.
* Agile method breaks the product in to small incremental build, this build is provided in iteration.
* Each iteration typically lasts from about one to three weeks.
* At end of the iteration a working product is displayed to the customer and important stakeholders.
* Iterative approach is taken and working software build is delivered after each iteration each build is incremental in terms of feature. The final build holds all feature required by the customer.

**Proc of agile model:**

* Is a very realistic approach to software development.
* Promote teamwork and cross training.
* Functionality can be developed rapidly and demonstrated.
* Resource requirement are minimum.
* Suitable for fixed or changing requirements.
* 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 developer.

**Cons of agile model:**

* 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.
* Depend heavily on customer interaction, so if customer is not clear, team can be driven in the wrong direction.
* There is very high individual dependency, since there is minimum documentation generated.
* **Draw usecase on online shopping product using cod.**

**OPEN APP**

**LOGIN**

**SEARCH PRODUCT**

**login**

**PLACE ORDER**

**SELECT COD**

**VARIFY ADDRESS**

**BUY NOW**

**ADD TO CART**

* **Draw usecase on online shopping product using payment gateway.**

**OPEN APP**

**SEARCH PRODUCT**

**LOGIN**

**CUSTOMER**

**SELECT PAYMENT GATWAY**

**PLACE ORDER**

**VARIFY ADDRESS**

**BUY NOW**

**ADD TO CART**