***SOFTWARE TESTING ASSIGNMENT***

**MODULE -1**

**Q1.What is SDLC?**

**Ans. Software development life cycle is used to development of a software product that defines the process for planning, implementation, testing, documentation, deployment, and ongoing maintance and support.**

**Q2.What is software testing?**

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

**Q3.What is agile methodology?**

**Ans. Agile methodology is a combination of iterative and incremental process model with focus on process adaptability and customer satisfaction by rapid delivery of working software product.**

**Q4.What is SRS?**

**Ans. A software requirements specification(SRS) is a complete description of behavior of the system to be developed. It includes a set of use cases that describes all of the interactions that the users will have with the software. Use cases are also known as functional requirements. In addition to usecases, the SRS also contains nonfunctional (or Supplementary) requirements.**

**Q5.Whats is oops?**

**Ans. Object oriented programming is used identifying objects and assigning responsibilities to these objects. It communicate to other by sending messages. An object is like a black box. The internal details are hidden. Object-oriented programming has a web of interactingobjects , each house-keeping its own state.**

**Q6.Write basic concepts of oops?**

**Ans. There are six basic concepts.**

* **Object.**
* **Class.**
* **Encapsulation.**
* **Inheritance.**
* **Polymorphism.**
* **Abstraction.**

**Q7.What is Object?  
Ans. It a instances of an class.**

* **To access the whole properties of an class except private.**
* **Object = Data + Methods.**

**Q8.What is class?**

**Ans. It is an collection of data member (variable) and member function (Method, Process) with the behavior.**

* **It defines blue print of an object.**
* **In the case of a car or laptop , there will be a blueprint or design created first and then the actual car or laptop, there will be built based on that.**

**Q9.What is Encapsulation?**

**Ans. Data hide – Wrapping up of data into single unit.**

* **Example: private your data member and member fun.**
* **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 not object-oriented programming provides you framework to place the data and the relevant functions together in the same object.**
* **The internal state is usually not accessible by other objects.**

**Q10.What is inheritance?**

**Ans. Properties of parent class extends into the child class.**

* **Properties of super class extends into subclass.**
* **Main purpose Is Reusability, Extendsibility.**
* **There are mainly 5 types.**
* **Single.**
* **Multilevel.**
* **Hierarchical.**
* **Multi: Java does not support directly.**
* **Hybrid: Java does not support directly.**
* **Inheritance means that one class inherits the characteristics of another class. This is also called a “is a” relationship.**

**Q11.What is Polymorphism?**

**Ans. Ability to take one name having many forms.**

* **There are mainly two types.**
* **Compile time.**
* **Run time.**
* **It allows different objects to respond to the same message in different ways, the responses specific to the type of the object.**
* **The most important aspect of an object is its behavior.**
* **A behavior is initiated by sending message to the object.**

**Q12.Write SDLC phases with basic introduction?**

**Ans. There are mainly six types of phases.**

* **Requirements Collection/Gathering.**
* **Analysis.**
* **Design.**
* **Implementation.**
* **Testing.**
* **Maintenance.**

**! REQUIREMENTS GATHERIN.!**

**1.Features.**

**2.User scenario.**

**3.Requirements will change!**

**4. Plan for change.**

**\*Three types of problems can arise:**

**(Lack of clarity, Requirements confusion , Requirements Amalgamation).**

**!Analysis Phase!**

* **The analysis Phase defines the requirement of the system, independent of how these requirements will be accomplished.**
* **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.**

**! Design Phase!**

* **Design architecture document.**
* **Implementation plan.**
* **Performance Analysis.**
* **Test plan.**

**! Implementation phase!**

* **In the implementation phase, the team builds the components either from scratch or by composition. Phase and the requirement document from the analysis phase, the team should build exactly what has been requested, though there is still room for innovationand flexibility.**
* **For example, a component may be narrowly designed for this particularsystem, or the component may be made more general to satisfy a reusabilityguideline .**
* **Implementation – Cod.**

**! Testing phases!**

* **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 l**evel.
* **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.**

**! Maintance !**

* **Maintenance is the process of changing a system after it has been deployed.**
* **CORRECTIVE MAINTENANCE: Identifying and repairing defects.**
* **ADAPTIVE MAINTENANCE : Adapting the existing solution to the new platforms.**
* **PERFECTIVE MAINTENANCE : Implementing the new on decides the utility and value of the software at a particular level of quality outweighs the impact of the know defect and deficiencies.**

**Q13. Explain phases of Waterfall model?**

**Ans. The classical software lifecycle and the software development as a step by step “waterfall” between the various development phases.**

* **Requirement must be frozen to early in the life product. Project must be short .**
  + - ***Pros***
* Simple and easy to understand and use
* Easy to manage due to the rigidity of the model.
* Each phase has specific deliverables and a review process.
* Phases are processed and completed one at a time.
* Works well for smaller projects where requirements are very well understood.
* Clearly defined stages.
* Well understood milestones.
* ***Cons***
* No working software is produced until late during the life cycle.
* High amounts of risk and uncertainty.
* Not a good model for complex and object-oriented projects.
* Poor model for long and ongoing projects.

**Q14. Write the phases of the 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 customeras 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.
* **Intial requirements.**
* **Go, no- go decision.**
* **First prototype.**
* **Alpha deo.**
* **Pros.**
* Changing requirements can be accommodated.
* Allows for extensive use of prototypes.
* Requirements can be captured more accurately.
* **Cons.**
* Management is more complex.
* End of project may not be known early.

**Q15. Write agile manifesto principles.**

**Ans.** 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. In agile the tasks are divided to time boxes (small time frames) to deliver specific features for a release.

* Iterative approach is taken and working software build is delivered after each iteration. Each build is incremental in terms of features; the final build holds all the features required by the customer.
* Agile thought process had started early in the software development and started becoming popular with time due to its flexibility and adaptability.
* *Pros*:
* Is a very realistic approach to software development.
* Promotes team work and cross training?
* Functionality can be developed rapidly and demonstrated.
* Resource requirements are minimum.
* Delivers early partial working solutions.
* *Cons:*
* Not suitable for handling complex dependencies.
* More risk of sustainability, maintainability and extensibility.
* Depends 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.

**Q16.Explain working methodology of agile model and also write pros and cons.**

**Ans.** Requirements of the complete system are clearly defined and understood.

* Major requirements must be defined; however, some functionalities or requested enhancements may evolve with time.
* There is a time to the market constraint.
* A new technology is being used and is being learnt by the development team while working on the project.
* Resources with needed skill set are not available and are planned to be used on contract basis for specific iterations.
* There are some high risk features and goals which may change in the future.
* **PROS**
* Results are obtained early and periodically.
* Parallel development can be planned
* Progress can be measured.
* Less costly to change the scope/requirements.
* Risk analysis is better.
* Initial Operating time is less.
* Better suited for large and mission-critical projects.
* **CONS**
* More resources may be required.
* Although cost of change is lesser but it is not very suitable for changing requirements.
* More management attention is required.
* Not suitable for smaller projects.
* Management complexity is more.
* End of project may not be known which a risk is.
* Highly skilled resources are required for risk analysis.