**software Testing Assinment module:-1**

Q.1 What is SDLC

Ans. Software Development Life Cycle 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 models.

Q.2 What is software testing?

Ans. Software Testing is a process used to identify the correctness, completeness, and quality of developed computer software.

Q.3 What is agile methodology?

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.

Q.4 what is SRS?

Ans. A software requirements specification (SRS) is a complete description of the behavior of the system to be developed.

Q.5What is oops

Ans. Identifying objects and assigning responsibilities to these objects.

Q.6 Write Basic Concepts of oops

Ans. 1) Object

1. Class
2. Encapsulation
3. Inheritance
4. Polymorphism
5. Abstraction

Q.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.

Q.8What is class

Ans. A class represents an abstraction of the object and abstracts the properties and behavior of that object.

* 1. 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.

* 1. What is inheritance

Ans. Inheritance means that one class inherits the characteristics of another class. This is also called a is a relationship.

* 1. What is polymorphism

Ans. Poly refers to many. That is a single function or an operator functioning in many ways different upon the usage is called polymorphism.

* 1. Write SDLC phases with basic introduction?

Ans. 1)Requirements Collection/Gathering

2)Analysis

3)Design

4)Implementation

5)Testing

6)Maintenance

1)Requirements collection/Gathering:-

Requirements definitions usually consist of natural language, supplemented by (e.g., UML) diagrams and tables.

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.

2)Analysis:-

This analysis represents the “what” phase.

This phase represents the “how” phase.

The requirement documentaries to capture the requirements from the customer's perspective by defining goals.

3)Design:-

The Design team can now expand upon the information established in the requirement document.

The requirement document must guide this decision process.

Critical Priority Analysis Performance

Analysis Test

Plan.

4)Implementation:- coding developer

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:- tester

A customer satisfied with the quality of a product will remain loyal and wait for new functionality in the next version.

market production Regression

Testing

Internal Testing

Unit Testing

Application Testing

Stress Testing

6)Maintenance:-

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 the delivery and deployment of the first prototype can be considered “maintenance”.

Q.13 Explain Phases of the waterfall model?

Ans. 1)Waterfall Model/Methodology (Classical

Software Cycle)

2)Iterative & Incremental Model/Methodology

3)Spiral Model/Methodology

4)Agile Model/Methodology

1)Waterfall Model/Methodology:-

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

Requirements are validated too late

Use:-

The project is short.

Technology is understood and is not dynamic.

There are no ambiguous requirements.

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.

Clearly defined stages.

Well understood milestones.

Easy to arrange tasks.

Process and results are well documented

Cons:-

High amounts of risk and uncertainty.

Not a good model for complex and object-oriented projects.

Poor model for long and ongoing projects.

It is difficult to measure progress within stages.

Cannot accommodate changing requirements.

No working software is produced until late in the life cycle. Adjusting scope during the life cycle can end a project. Integration is done as a "big-bang. at the very end, which doesn't allow identifying any technological or business bottleneck or challenges early.

Q.14 Write phases of spiral model?

Use:-

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.

Pros:-

Changing requirements can be accommodated. Allows for extensive use of prototypes Requirements can be captured more accurately.

Users see the system early. cons:-

Management is more complex.

End of project may not be known early.

Not suitable for small or low risk projects and could be expensive for small

projects.

Process is complex

Spiral may go indefinitely.

Q.15 Write agile manifesto principles?

Individuals and interactions - in agile development, self-organization and motivation are important, as are interactions like co-location and pair programming. 1) Working software - Demo working software is considered the best means of communication with the customer to understand their requirement, instead of just depending on documentation.

1. 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 requirements..
2. Responding to change - agile development is focused on quick responses to change and continuous development.

Q.16 Explain wroking methodology of agile model and also write pros and cons? Pros:-

1. Is a very realistic approach to software development Promotes

teamwork and cross training.

1. Functionality can be developed rapidly and demonstrated.

Resource requirements are minimum.

1. Suitable for fixed or changing requirements Delivers early partial

working solutions.

4)Good model for environments that change steadily.

5) Minimal rules, documentation easily employed.

6)Enables concurrent development and delivery within an overall planned context.

1. Little or no planning required
2. Easy to manage
3. Gives flexibility to developers

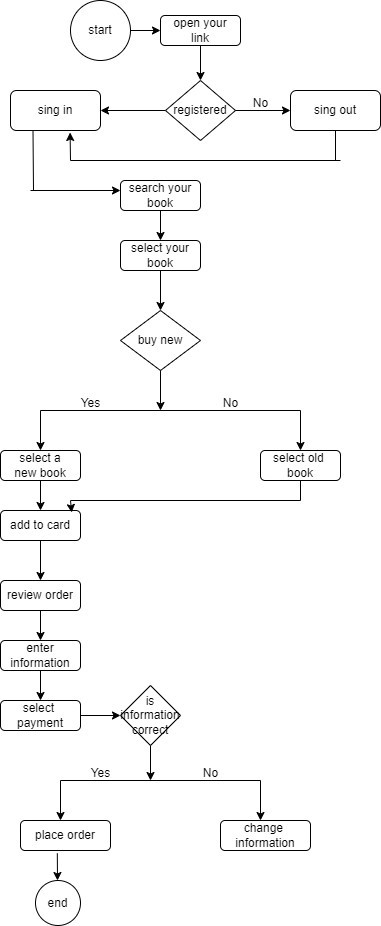
Cons:-

1. Not suitable for handling complex dependencies.
2. More risk of sustainability, maintainability and extensibility.
3. An overall plan, an agile leader and agile PM practice is a must without which it will not work.
4. Strict delivery management dictates the scope, functionality to be delivered, and adjustments to meet the deadlines.
5. Depends heavily on customer interaction, so if customer is not clear, team can be driven in the

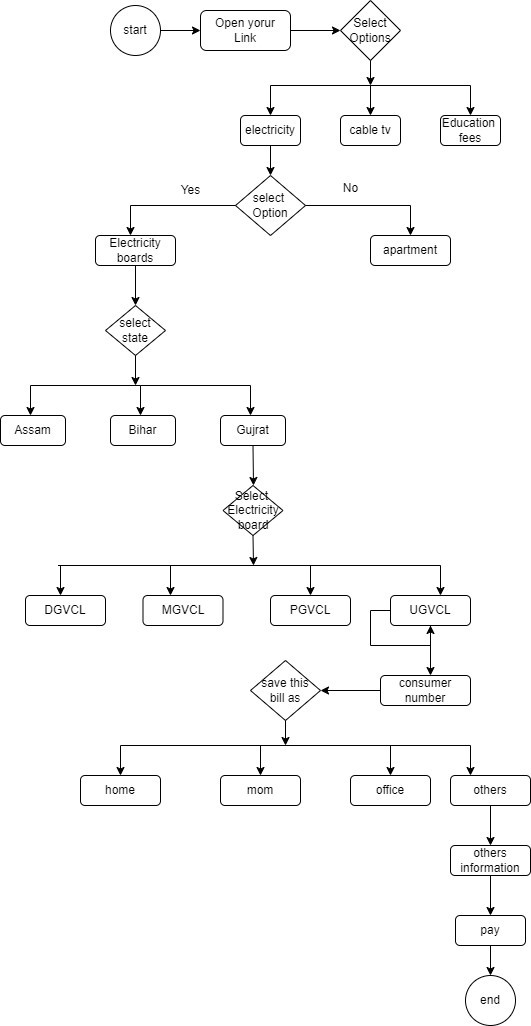
wrong direction.

1. There is very high individual dependency, since there is minimum documentation generated.
2. Transfer of technology to new team members may be quite challenging due to lack of

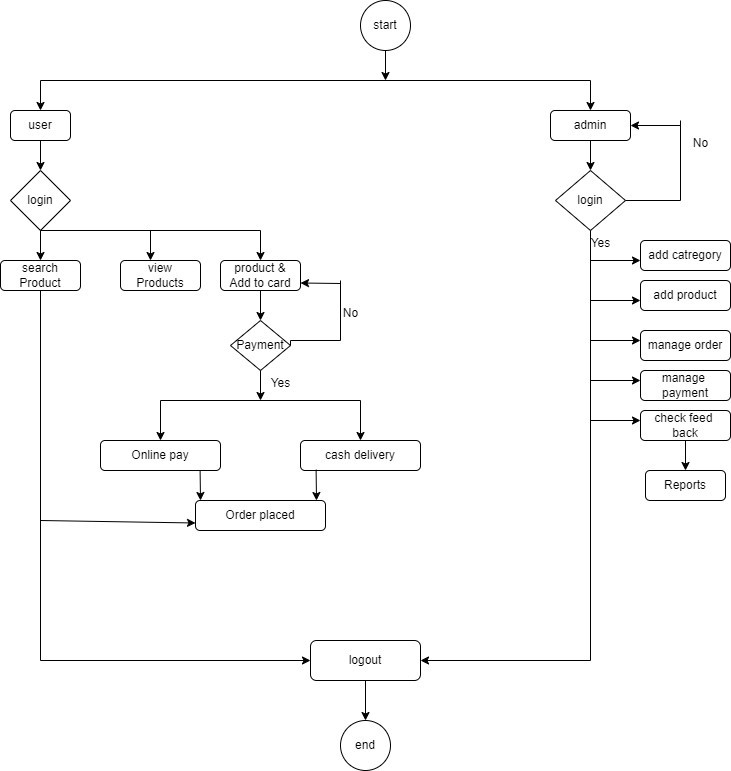
* 1. Draw Usecase on Online book shopping



* 1. Draw Usecase on online bill payment system (paytm)



* 1. Draw usecase on Online shopping product using COD.



* 1. Draw usecase on Online shopping product using payment gateway

