

TOPS TECHNOLOGIES

SOFTWARE TESTING ASSIGNMENT

SHREEYA

2023

RAJKOT

1. What is SDLC?

- SDLC is Software Developer Life Cycle
- A Software Development Life Cycle is essentially a series of steps, or phases, that provide a model for the development and lifecycle management of an application or piece of software.

2. What is software testing?

- Software Testing is a process used to identify the correctness, completeness, and quality of developed computer software.
- Testing is the process of evaluating a system or its component(s) with the intent to find that whether it satisfies the specified requirements or not.
- The process of validating and verifying that a software program or application or product:

3. What is SRS?

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

4. What is OOP?

- Object Oriented Programming
- An object-based programming language is one which easily supports object-orientation.
- Identifying objects and assigning responsibilities to these objects.
- An object is like a black box. The internal details are hidden.
- Object-oriented programming has a web of interacting objects, each house-keeping its own state.

5. Write Basic Concepts of oops?

- Object
- Class
- Encapsulation
- Inheritance
- Polymorphism
 - Overriding
 - Overloading
- Abstraction

6. What is object?

- Any Entity which has own state and behavior that is called an object.

7. What is class?

- Collection of objects that is called class.

8. What is encapsulation?

- Wrapping up of data or binding of data that is called encapsulation.

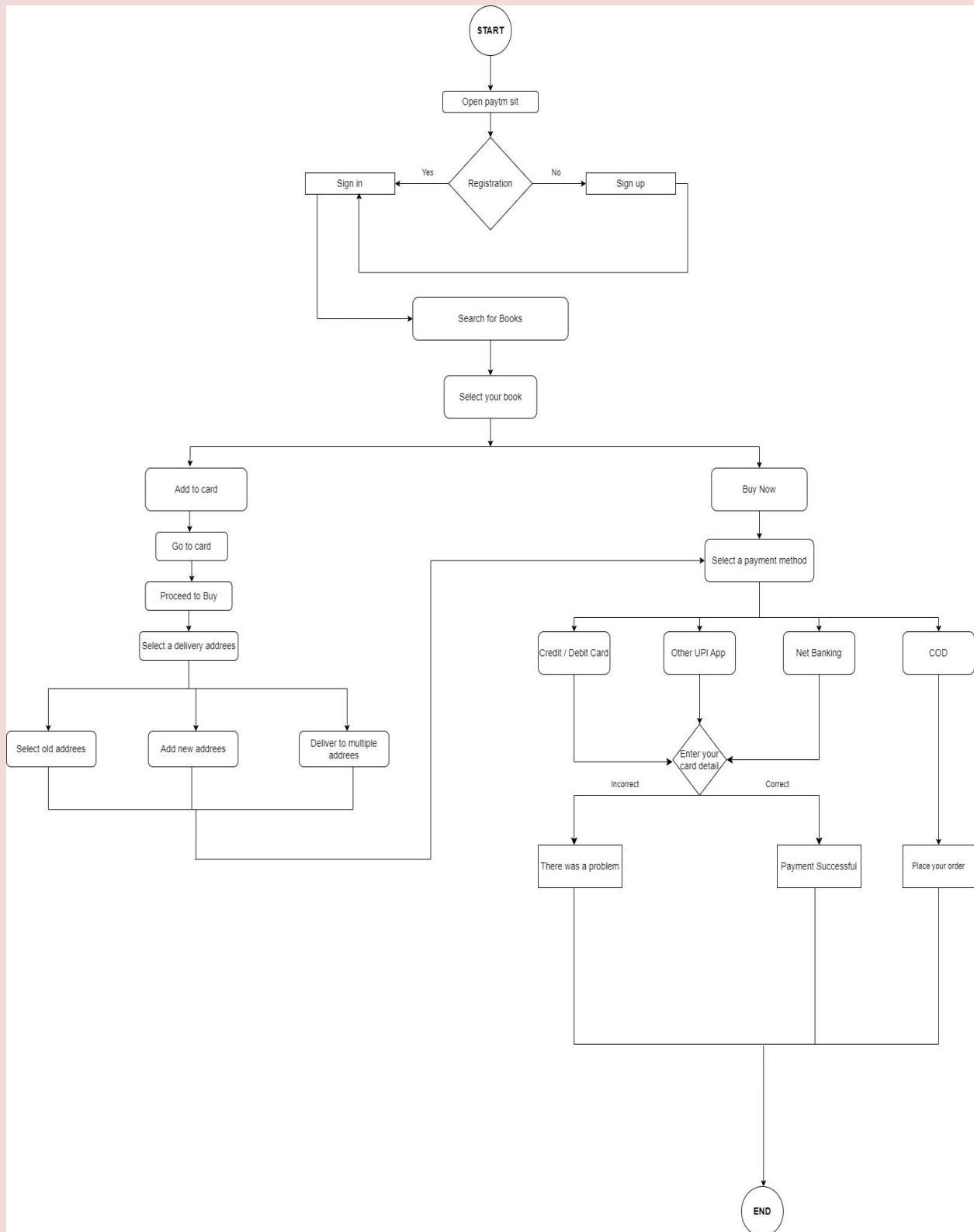
9. What is inheritance?

- When one object acquires all the properties and behavior of parent class.

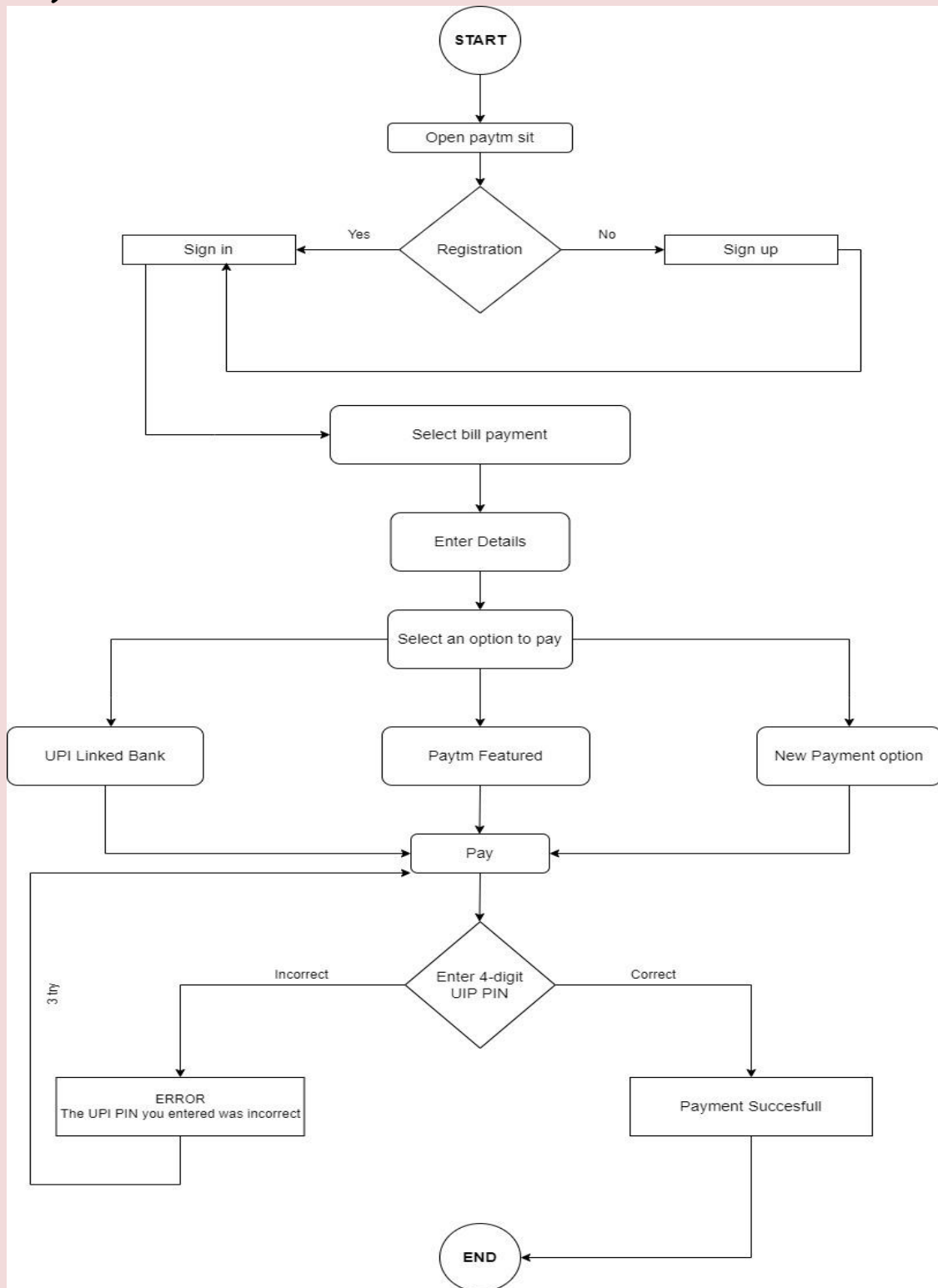
10. What is polymorphism?

- Many ways to perform anything.

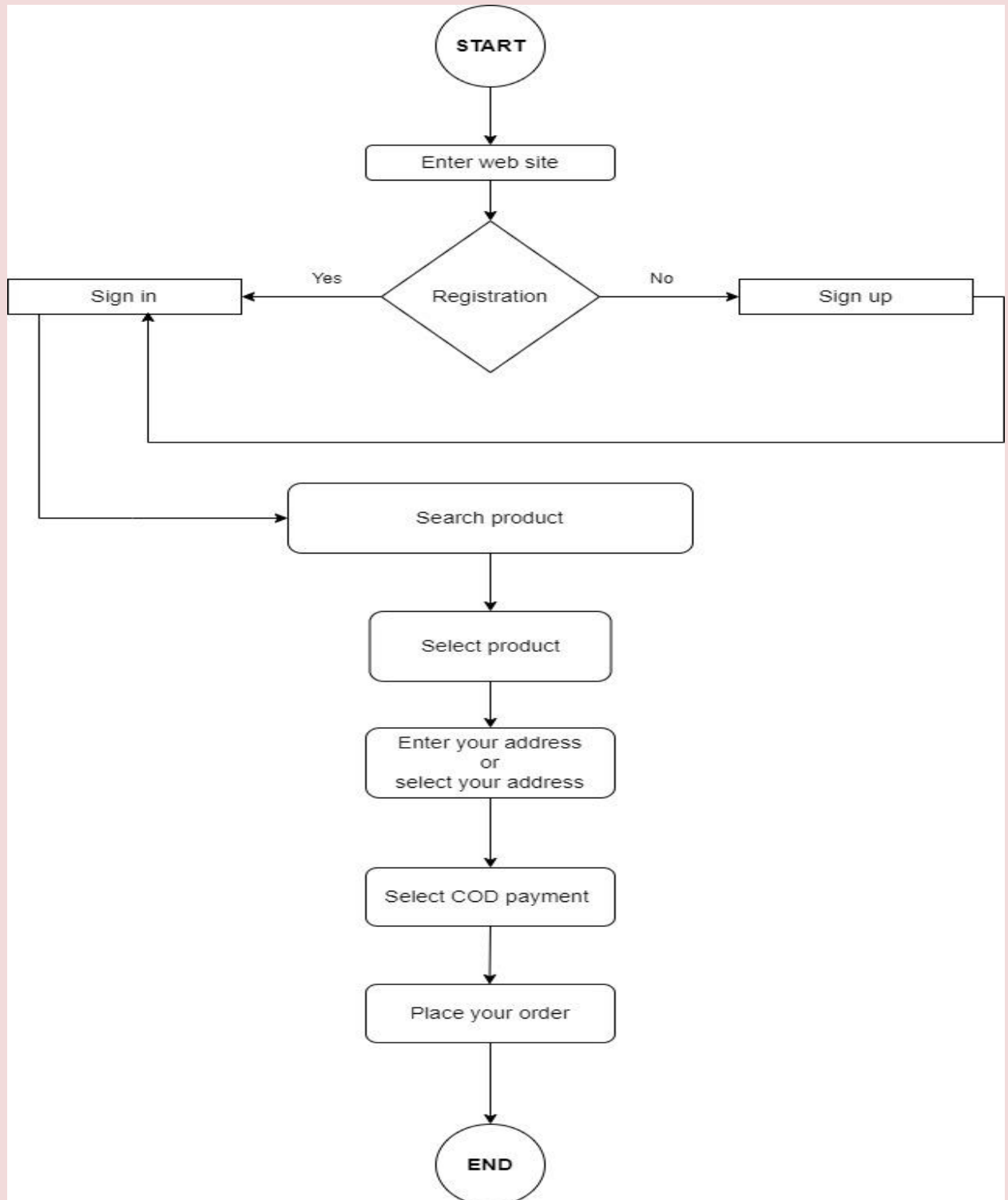
11. Draw Use case on online book shopping



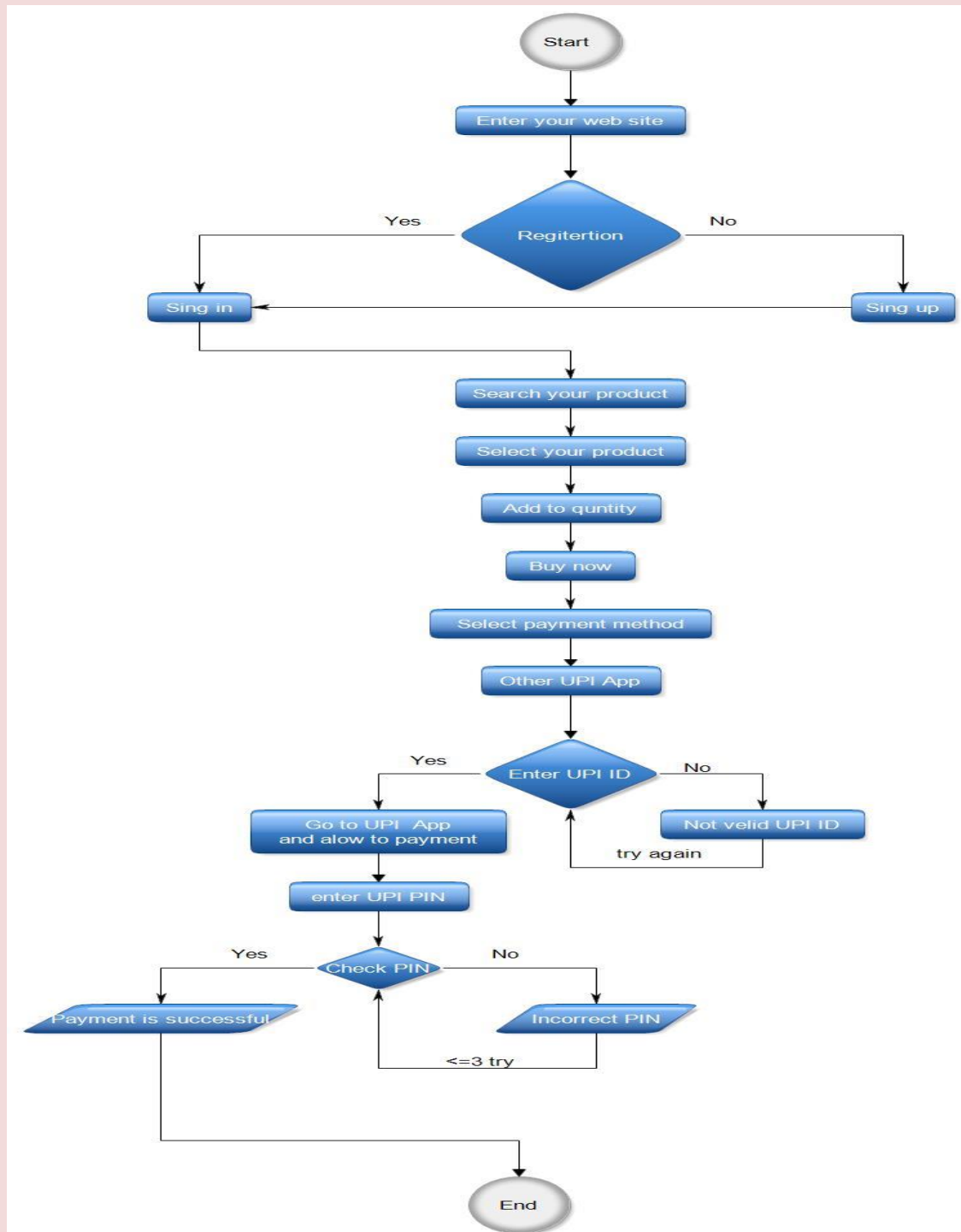
12. Draw Use case on online bill payment system (paytm)



13. Draw use case on Online shopping product using COD.



14. Draw use case on Online shopping product using payment gateway.



15. Write SDLC phases with basic introduction

➤ Requirement Gathering

- Although requirements may be documented in written form, they may be incomplete, • unambiguous, or even incorrect.
- User and business needs change during the project
- Requirements definitions usually consist of natural language, supplemented by (e.g., UML) diagrams and tables.
- Types of Requirements:
 1. Functional Requirements: Describe system services or functions.
 - ✓ Compute sales tax on a purchase
 - ✓ Update the database on the server
 2. Non-Functional Requirements: are constraints on the system or the development process.
- Non-functional requirements may be more critical than functional requirements

16. What is agile methodology?

- 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.
- These builds are provided in iterations.
- Each iteration typically lasts from about one to three weeks.

- Every iteration involves cross functional teams working simultaneously on various areas like planning, requirements analysis, design, coding, unit testing, and acceptance testing.
- At the end of the iteration a working product is displayed to the customer and important stakeholders.

17. 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.
- 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.
- 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.
- Responding to change - agile development is focused on quick responses to change and continuous development.

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

- 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 build

- These builds are provided in iterations.
- Each iteration typically lasts from about one to three week.
- At the end of the iteration a working product is displayed to the customer and important and important stakeholders.

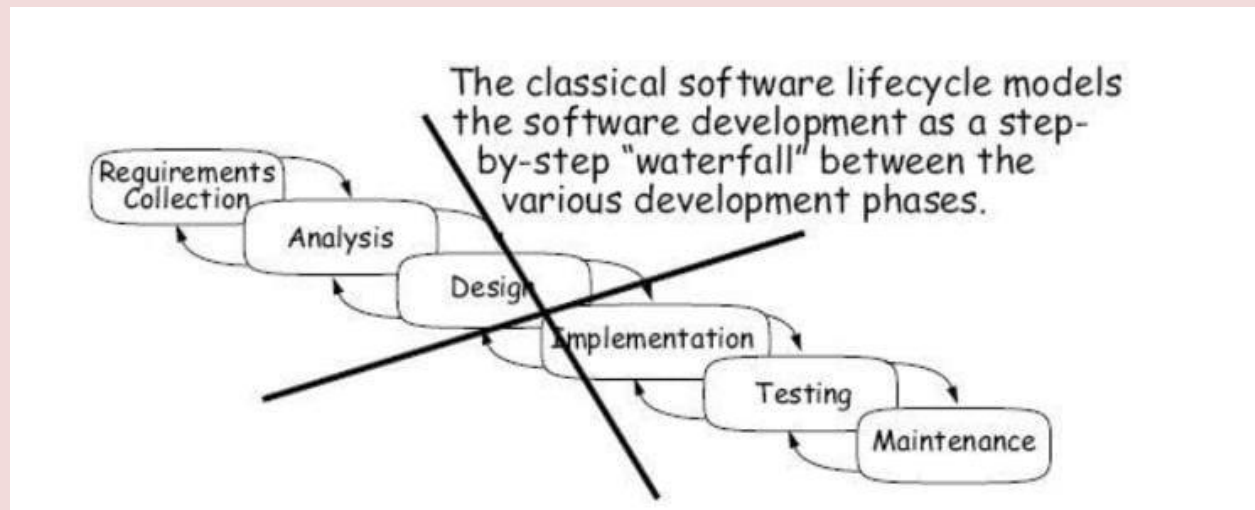
➤ Pros

- Is a 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 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 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 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 generate
- Transfer of technology to new team members may be quite challenging due to lack of

19. Explain Phases of the waterfall model



- The waterfall is unrealistic for many reasons, especially:
 - Requirements must be "frozen" too early in the life cycle
 - Requirements are validated too late

- Applications (When to use?)
 - The project is short.
 - There are no ambiguous requirements.
 - Technology is understood and is not dynamic.
 - Product definition is stable.
 - Requirements are very well documented, clear and fixed.
 - Simple and easy to understand and use.

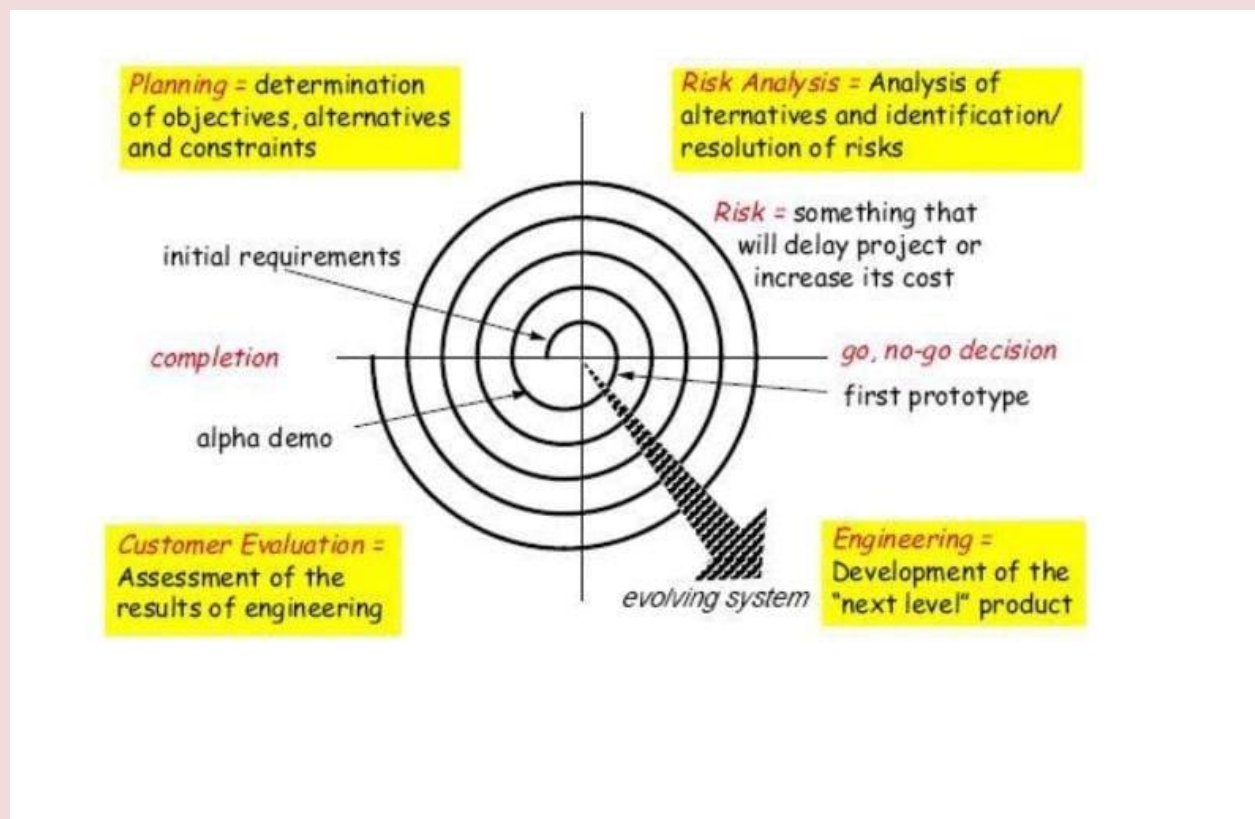
- Pros :-
 - Simple and easy to understand and use
 - Easy to manage due to the rigidity of the model.
 - Phases are processed and completed one at a time.
 - Process and results are well documented.
 - Easy to arrange tasks.
 - Well understood milestones.

- Clearly defined stages.

➤ Cons:-

- Adjusting scope during the life cycle can end a project
- It is difficult to measure progress within stages.
- Poor model for long and ongoing projects.
- Not a good model for complex and object-oriented projects.
- High amounts of risk and uncertainty.
- No working software is produced until late during the life cycle.

20. Write phases of spiral model



➤ Application :-

- Spiral Model is very widely used in the software industry
- When costs there are a budget constraint and risk evaluation is important.
- For medium to high-risk project
- 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 cost
- Requirements are complex and need evaluation to get clarity.
- Significant changes are expected in the product during the development cycle.

➤ Pros:-

- Allows for extensive use of prototypes
- Changing requirements can be accommodated.
- Development can be divided into smaller parts and more risky parts
- Users see the system early.
- Requirements can be captured more accurately.
- Can be developed earlier which helps better risk management.

➤ 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

- Large number of intermediate stages requires excessive documentation.