Module-1
Fundamental

(*) what is SDLC:-
⇒ Software development life cycle.
⇒ SDLC 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.
⇒ SDLC is essentially a series of steps, or phases, that Provide a model for the development and lifecycle management of an application.

(*) What is software Testing?
⇒ Software Testing is a Process used to identify the correctness, completeness, and quailty of developed Computer software.

(*) What is Agile Methodology?
⇒ Agile SDLC model is a Combination of iterative and incremental Process models which Foces on Process adaptability and customer

satisfaction by ~~ka~~ rapid delivery of working software Product.

⇒ In ~~[illegible]~~ Agile the tasks are divided to time boxes to deliver specific features for a release.

(*) What is SRS -
⇒ software requirement specification is a complete description of the behavior of the system to be developed.

(*) What is Oops -
⇒ Identifying objects and assigning responsibilities to the objects.
⇒ objects communicate to other objects by sending messages.

(*) what is object?

⇒ An object represents an individual, identifiable item, Unit or entity, either role or abstract, with a well defined role in the Problem domain.

⇒ An object is Anything to which a concept applies.

(*) what is class?

⇒ class is nothing but the blueprint for an object.

A class represents an abstraction of the object and abstracts the Properties and behavior of that Project.

SDLC Phases:-

(i) Requirment gathering ⇒ find customer needs
(ii) Analysis ⇒ Model & specify the requirement
(iii) Design ⇒ Model & specify the Solution.
(iv) Implementation ⇒ Construct the Solution in software
(v) Testing ⇒ Validate the Solution
(vi) maintenance. ⇒ Repair defects

Water fall Model.

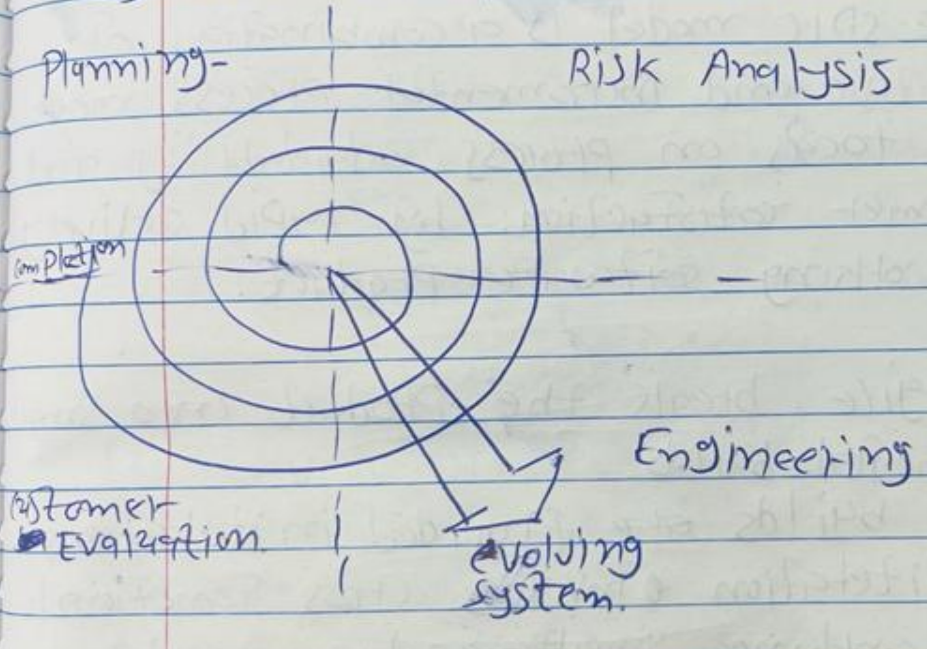
⇒ this is a classical software lifecycle model.
⇒ In this requirements must be Frozen.
⇒ requirements are validated too late.

When to use:-
⇒ requirements are very well documented, clear & Fixed.
⇒ Product definition is stable.
⇒ short Project.

Pros
⇒ Simple & easy to understand and use.
⇒ Easy to manage

(*) Cons :-
=> High amount of risk & Uncertainity.
=> Poor model for long & ongoing Project.
=> Hard to measure the Progress within stages.

(*) Bohem's spiral model :-



(i) Planning => determination of objectives, alternatives,
(ii) Risk Analysis => Analysis of alternatives & identification / resolution of Risks.

(iii) Enginreening - Development of the Next level Prodect.

(iv) Customer evaluation → Assessment of the results of enginaering.

(*) <u>Agile model</u>

⇒ 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.

⇒ In Agile, breaks the product into small incremental builds.

⇒ These builds are provided in iterations.

⇒ Each iteration involves cross functional team working simultaneously on varies areas.

⇒ At the end of the iteration a working Product is displayed to the customer and important Stakeholders.

Pros :-

⇒ very realistic approach to software development.
⇒ Promotes teamwork & cross training.
→ Resource requirements are minimum.
⇒ suitable for fixed & changing reqiments.

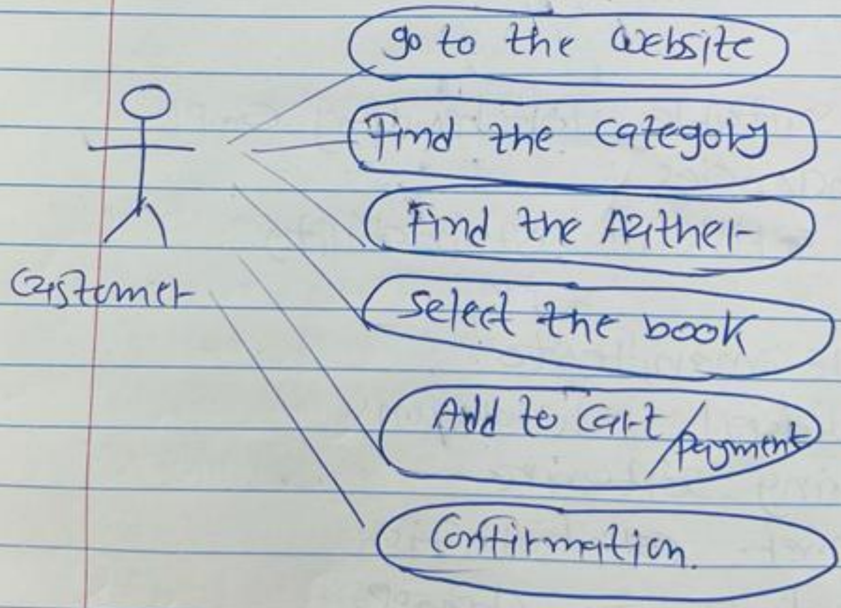
Cons :-
⇒ Not suitable for handling complex dependencies.
→ More risk of sustainability.

(*) Agile manifesto.
(1) Individual inter-actions
(2) working software
(3) customer collaboration
(4) responding to change.

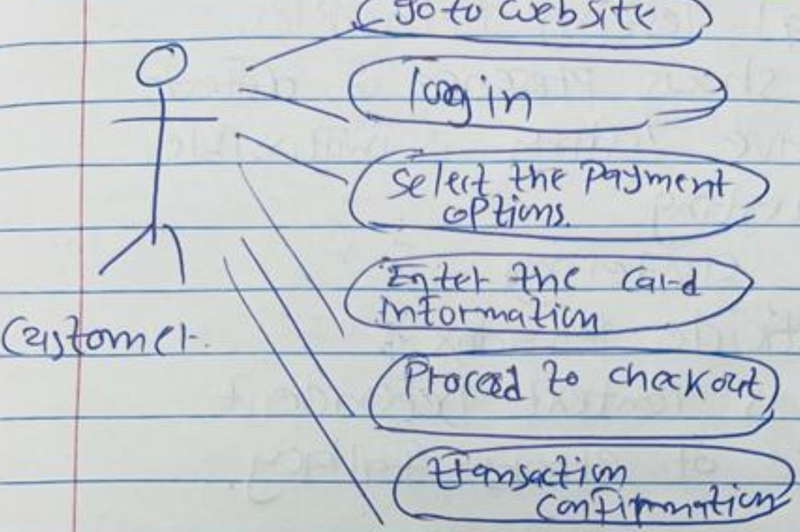
(*) <u>Use case:-</u>

⇒ A Use Case is the specification of a sequence of actions, including variants, that a system can perform, interacting with actors of the system.

(*) Use case for online book shopping.



(*) Usecase for online bill Payment.



(*) Polymorphism:-

⇒ Polymorphism means having many forms.

⇒ It allows different objects to respond to the same message in different ways, the response specific to the type of the object.

⇒ The ability to change form is known as polymorphism.

(*) Inheritance:-

⇒ Inheritance means that one class inherits the characteristics of another class. this is also called a "is a" relationship.

(*) General Testing Principles.

1. Testing shows presence of defects
2. Exhaustive Testing is impossible.
3. early testing
4. Defect clustering
5. the pesticide paradox
6. Testing is context Dependent
7. Absence of errors Fallacy.

(*) <u>Encapsulation</u> :-

⇒ 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.

(*) <u>Abstraction</u> :-

⇒ Abstraction is the representation of the essential features of an object. these are encapsulated into an abstract data type.