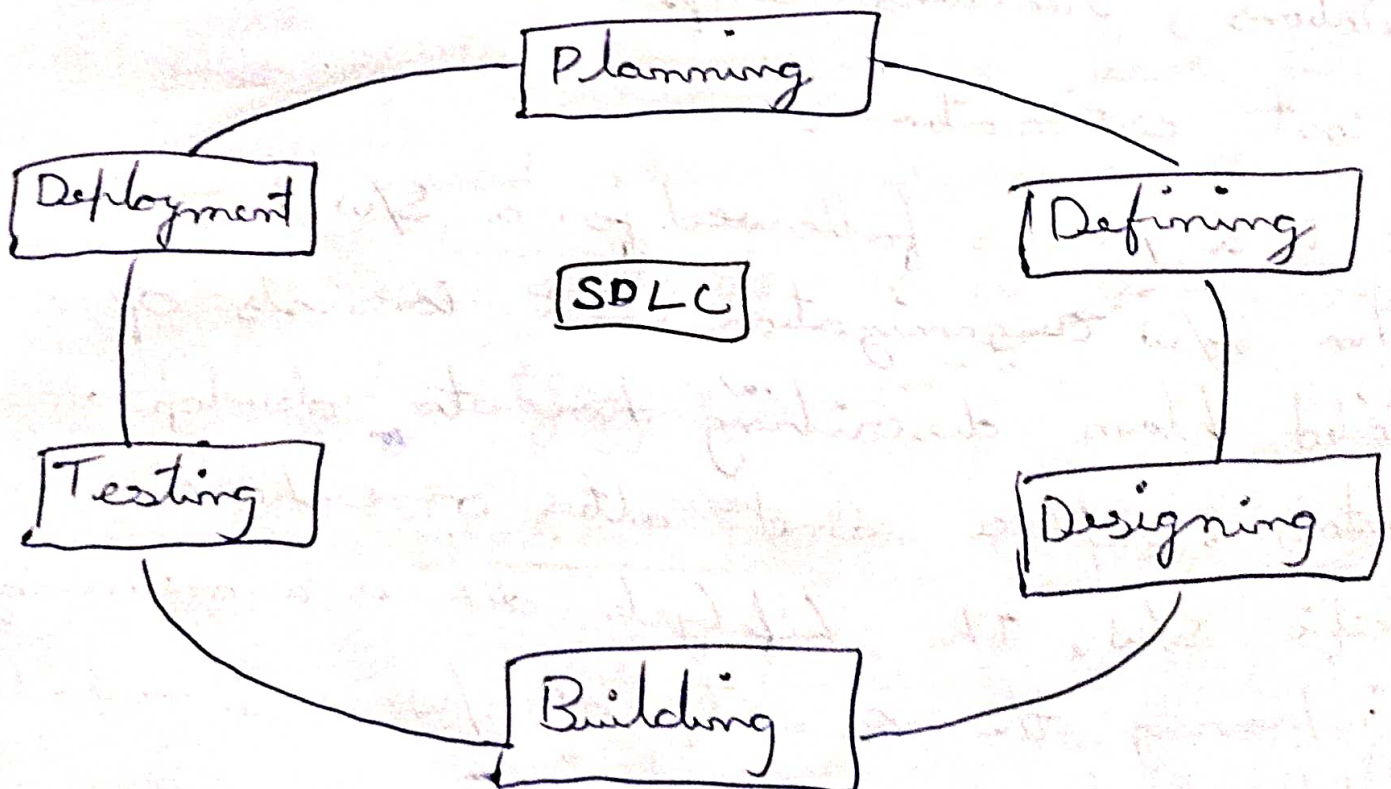


## SDLC Models

There are various Software development Lifecycle Models defined and designed which are followed during Software development process. Each process model follows a series of steps unique to its type to ensure in process of Software Development.

- ① Waterfall Model - For small size projects only
- ② Prototype Model
- ③ Spiral Model
- ④ Evolutionary Development Model
- ⑤ Iterative Enhancement Model



① Waterfall Model (Aka Classical Lifecycle Model)

② Waterfall Model was the first process model to be introduced.

③ It is also referred to as a "Linear-Sequential Lifecycle Model"

④ It is very simple to understand.

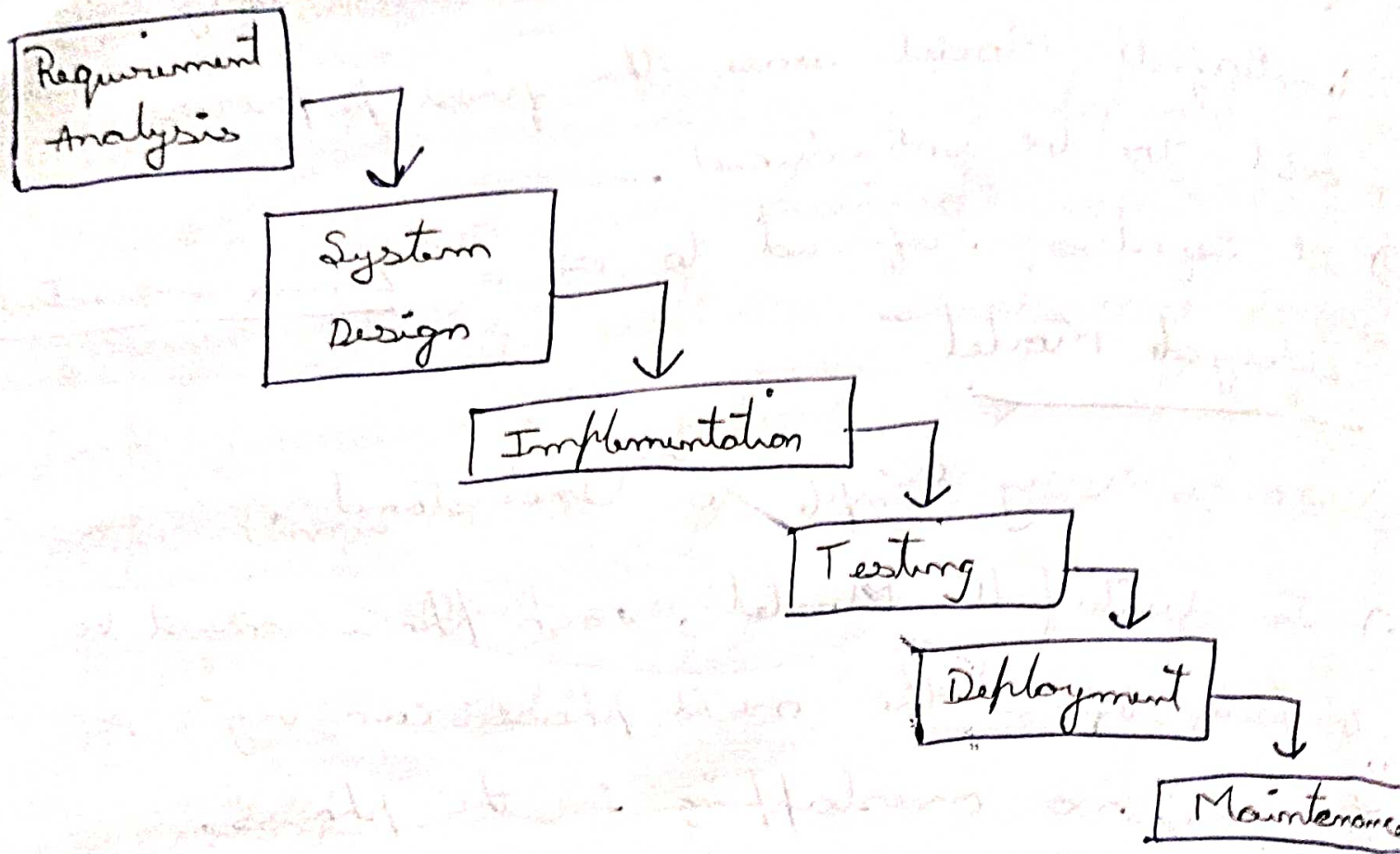
⑤ In Waterfall Model, each phase must be completed before the next phase can begin & there is no overlapping in the phases.

⑥ It is the earliest SDLC approach that was used for Software Development.

⑦ In this model, typically the outcome of one phase acts as the input for the next phase sequentially.

⑧ The Waterfall model illustrates the Software Development model.



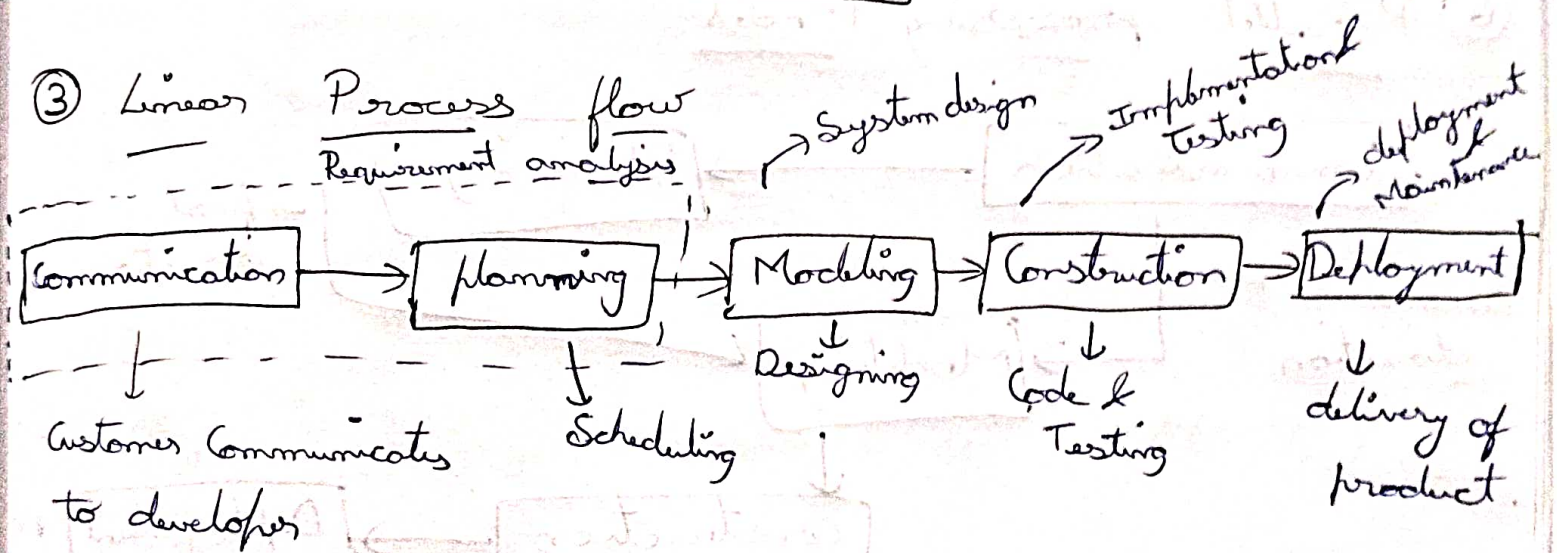
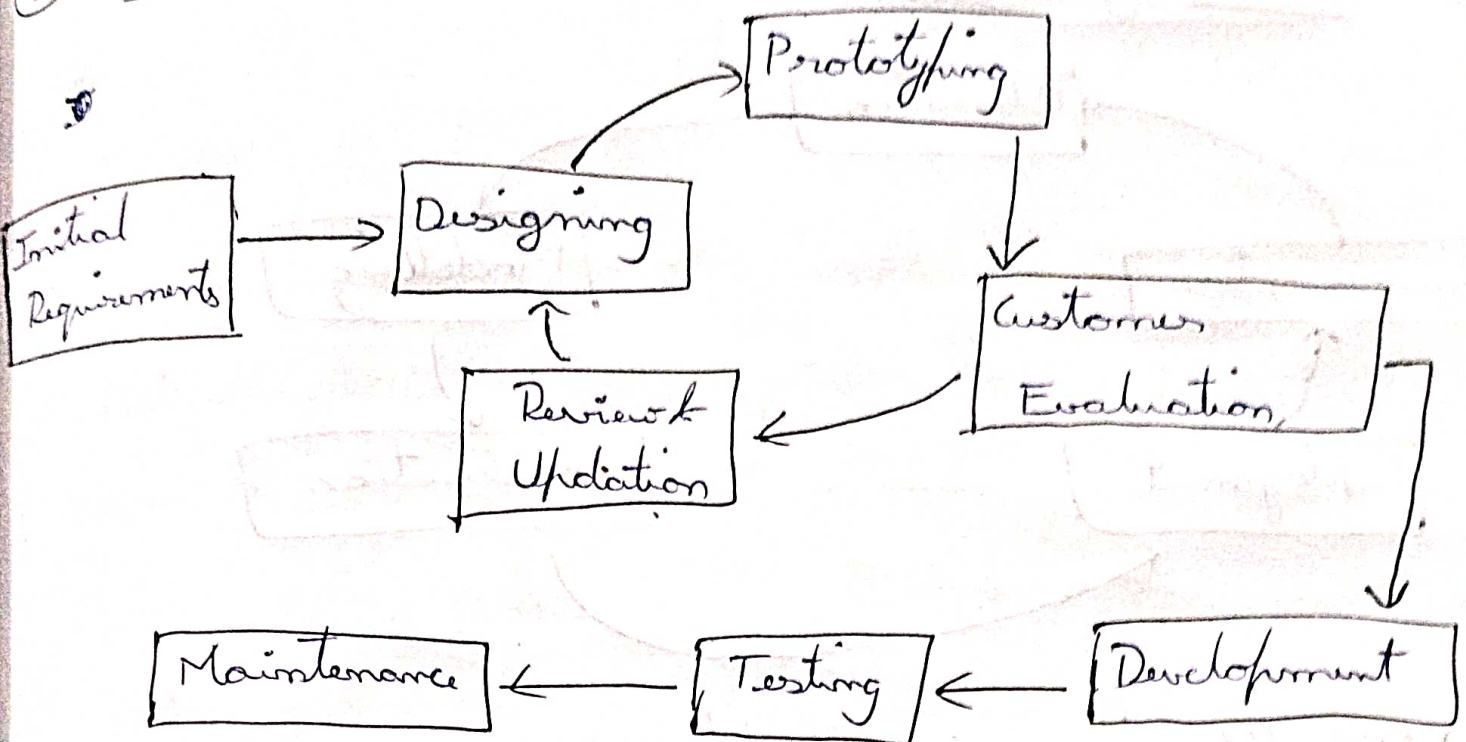


↓  
We don't go back (or) previous stage in Waterfall model

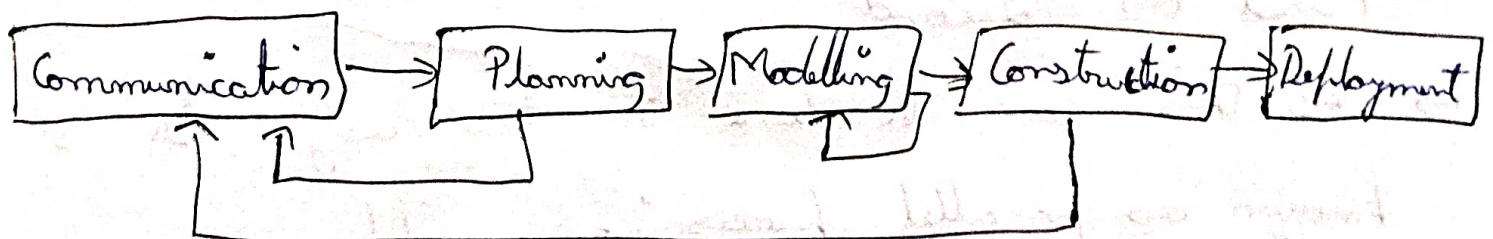
SRS - Software Requirement Specification

Unit - Sub-program (or) small part of a program

## ② Prototyping Model

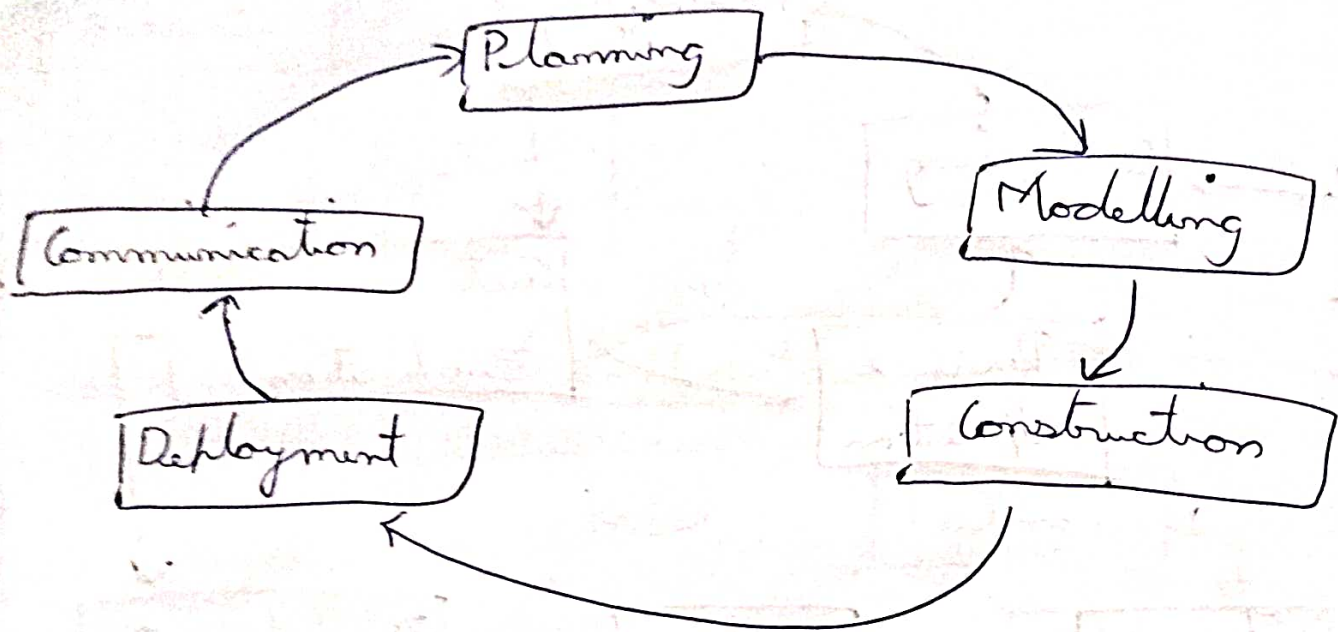


## ④ Iterative Process flow:

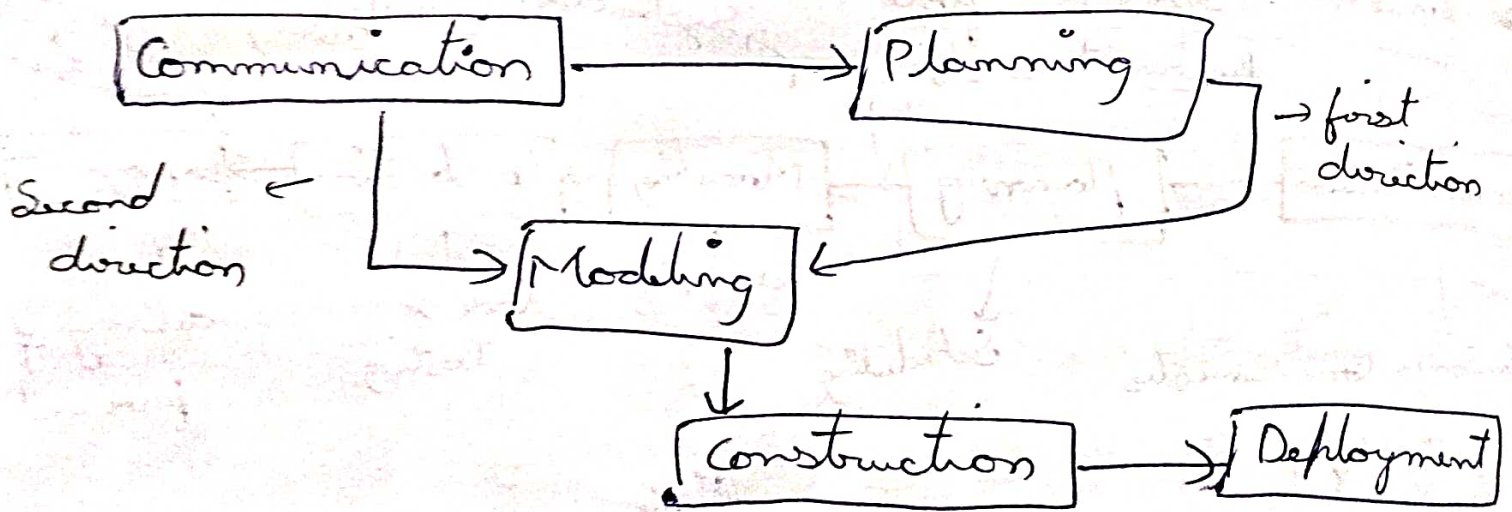




### ⑤ Evolutionary process



### ⑥ Parallel processing Model



→ This model/process can be done in either way:  
First or Second

→ As this can be done in two ways, it is known as parallel processing model.

## Evolutionary Model

- It is a combination iterative & Incremental model of SDLC.
- Incremental model implements a few basic features in the beginning, to customers, Then builds the next part & deliver it again and repeat this step until the desired system is fully achieved
- No long term plans are made.
- Iterative model's advantage is its feedback process in every phase.
- aka 'design a little, build a little, test a little, deploy a little' model



# ⑦ Spiral Model

1st Spiral

Conceptual Development

2nd Spiral

System Development

3rd Spiral

System Enhancement

4th Spiral

System Maintenance

Planning

- estimation
- Scheduling
- Risk analysis

Modelling

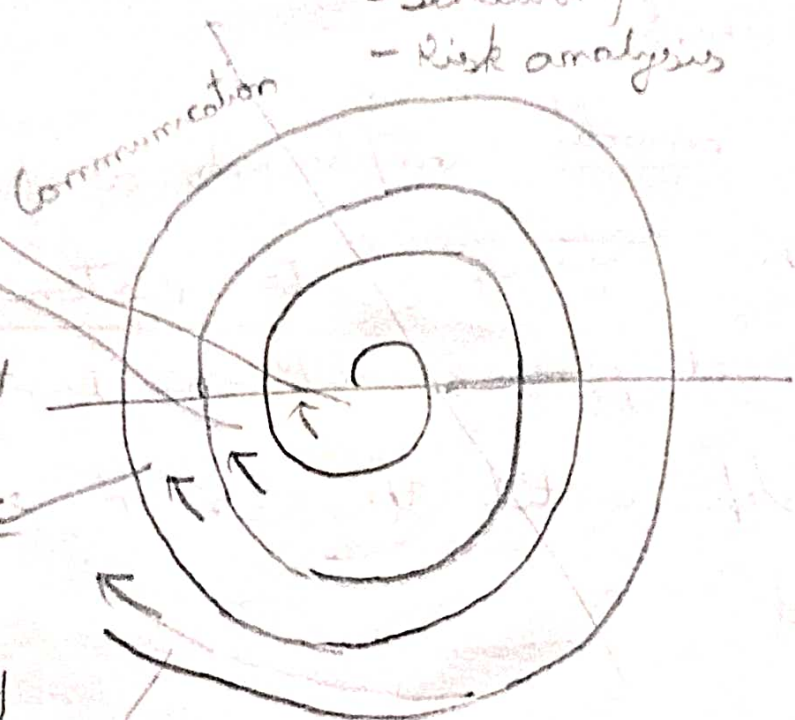
- Analysis
- Design

Construction

- Code
- Test

Deployment

- Delivery
- Support
- Feedback



# Introduction to UML

The Unified Modeling Language (UML) is a standard library language for writing software ~~but~~ blueprint of the system.

The UML may be used to visualize, specify, construct and document artifact of a S/W insert to system.

→ UML is a appropriate modelling system ranking from Enterprise Information System to distributed web base application and even to hard real time system.

→ UML effectively starts with Conceptual Modelling of the language.

## Main functions of the UML

1) Visualizing

2) Specifying

3) Constructing

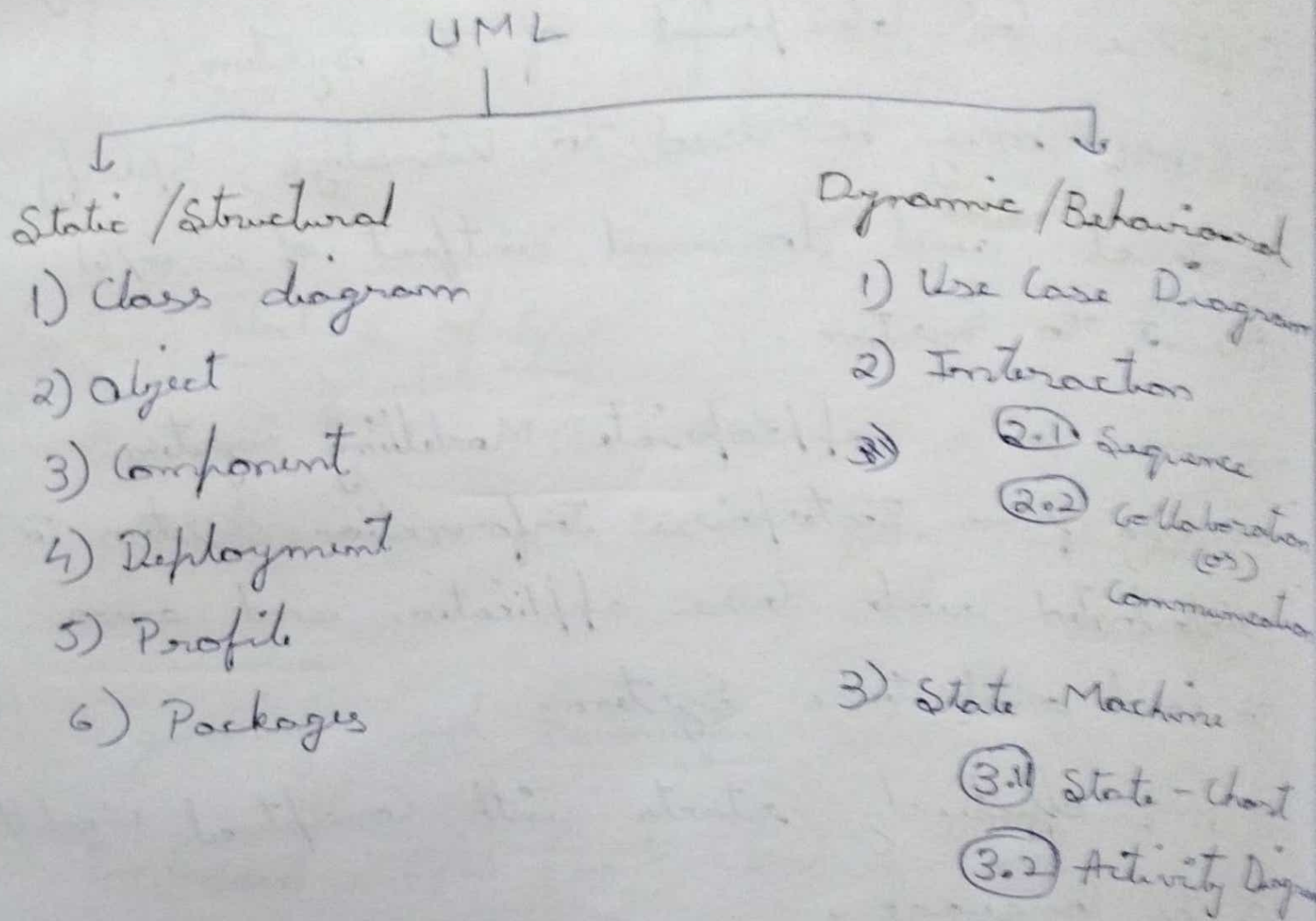
4) Documentation



# Main Building Blocks of the UML

Thinking, Relationships, Diagrams

→ There are 2 types of diagrams



## Applications of UML

UML is intended primarily for S/W Intensive Systems. It has been used effectively for such domain as

- ① Enterprise Information System..
- ② Banking financial
- ③ Tele Communication

- ④ Transportation
- ⑤ Defence & aerospace
- ⑥ Retail
- ⑦ Medical, Electronics
- ⑧ Scientific
- ⑨ Distributed web base services

### Thinkings:-

It is the most Important building blocks of UML. It can be as Structural, behavioural, ~~and~~ Grouping & Annotational.

① Structural Thinking: There defined a static part of a Model.

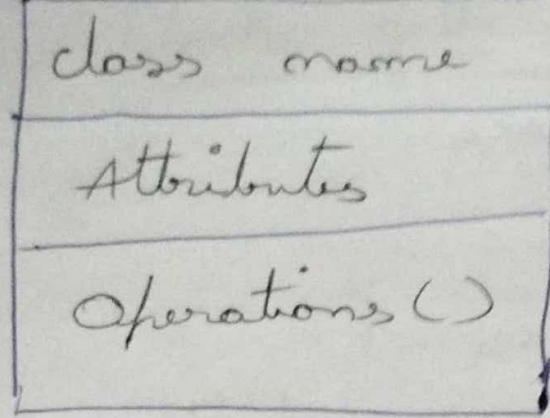
- There represents physical & Conceptual elements.

The following structural thing

⊙ Class Diagram: It describes the set of objects that shares the same attributes.

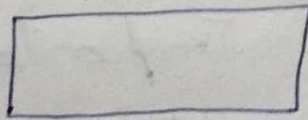
⊗ Operations, relations, Semantics





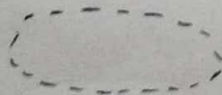
Object:

It is a collection of operations that specify a service of a class or a compound



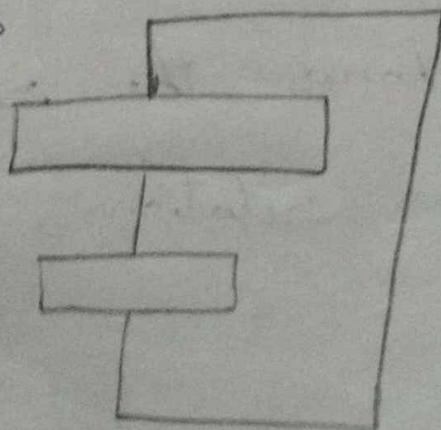
Collaboration:

It defines Interaction between elements.



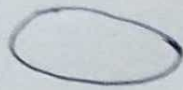
Component:

It is a physical and replaceable part that conforms to and provides realization of set of interfaces

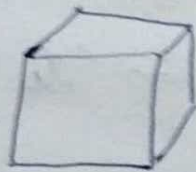


Use Case Diagrams: These are used to identify different use case components of a particular software project.

⊖ It is used to model the operation



Node: A physical resource that exist in runtime under represent a computational resource



Actor:

The Outside entity that communicates with a System, typically a person playing a role on an external device



Behavioral Thinking

There consists of dynamic parts of UML

Model.



The following are behavioral thinking:

① Interaction: It is defined as a behavioral that consists of a group of messages <sup>or</sup> exchanged among elements to accomplish a specific task.

message →

(i) State Machine: It is useful when this states of an object in its life cycle.

→ It defines the sequence of states and object goes to in response to events.

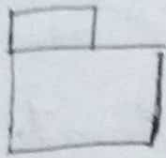
600



Grouping things:

These can be defined as a mechanism to group elements of UML model together. There is only one grouping thing available i.e.,

Package: It is used for gathering structural & behavioral thinking.



## Annotational Thinking:

These can be defined as Mechanism to capture remarks description & comments of UML Model elements

→ There is only one annotational thinking available i.e., used to render comments constraints and so on of a UML element.

## Relationship

It is another most important building block of UML.

→ These show how elements are associated to each other & their association describe the functionality of application.

There are 5 types of Relations:

### ① Dependency:



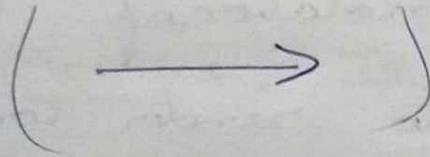
It is relationship between 2 things in which change in 1 element also affects another.



(ii) Generalization: It can be defined as relationship which connects a specialized elements with a generalized element.

→ It basically describes inheritance relationship in the object.

→ It is "is a" relationship



(iii) Realization: It can be defined as a ~~realization~~ relationship in which 2 elements are connected. One element describes some responsibility which is not implemented but other one implements.

→ The relationship exists in the case of ~~real~~ interfaces that represents



(iv) Association: It is a set of links that connects elements of UML model

→ 2 types

① Uni-Directional

② Bi-Directional

1) Aggregation:

It is a "has a" relationship.

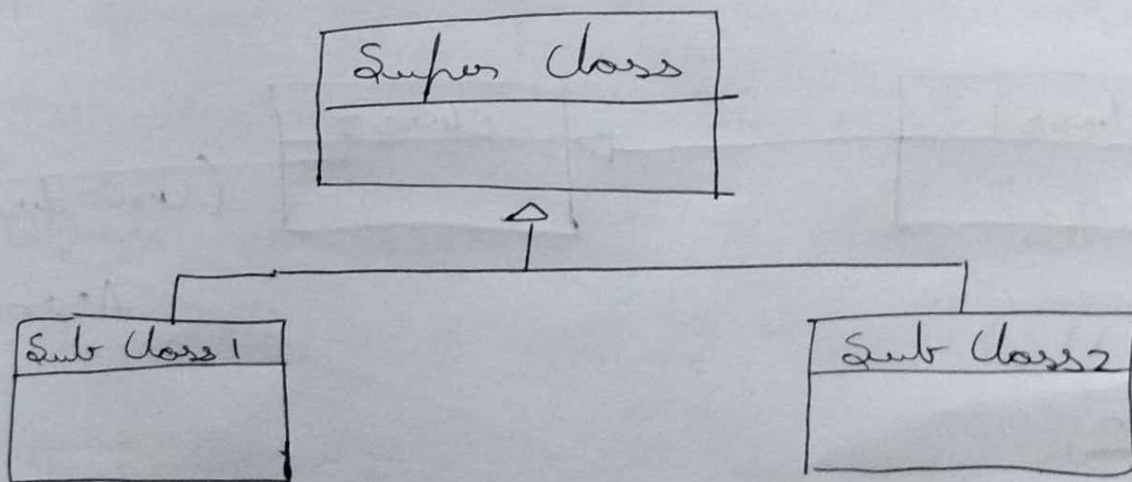
It is 2 types

① Simple Aggregation

② Composite Aggregation

Class Relationships

Inheritance (Generalization)



- Represents N is a relationship

- Sub Class 1 & Sub Class 2 are Specialisation



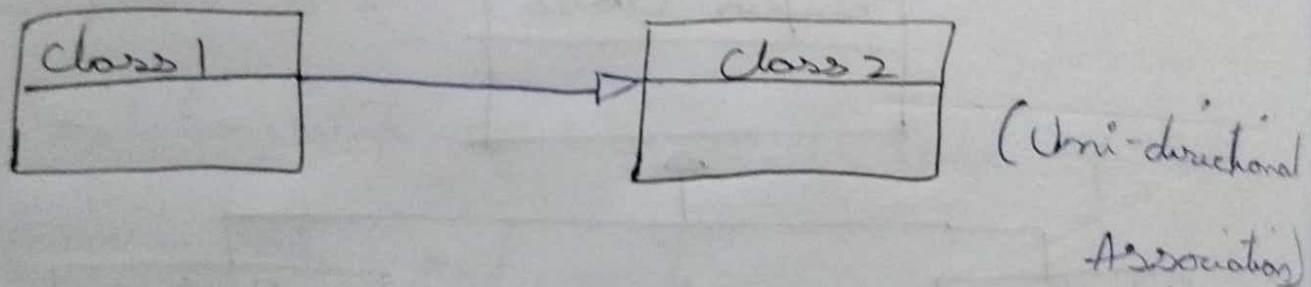
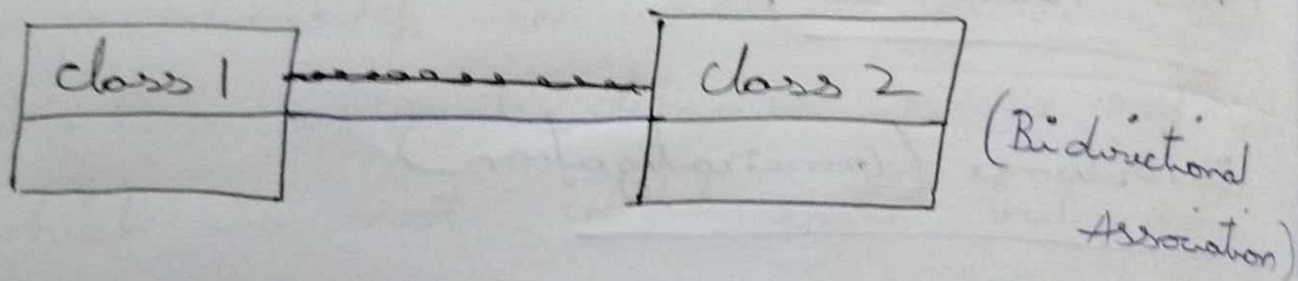
of Super Class -

- A Solid line with a hollow arrow head that points from the child to the parent class as shown in the above diagram.

~~Simple Aggrega~~

### Simple Association

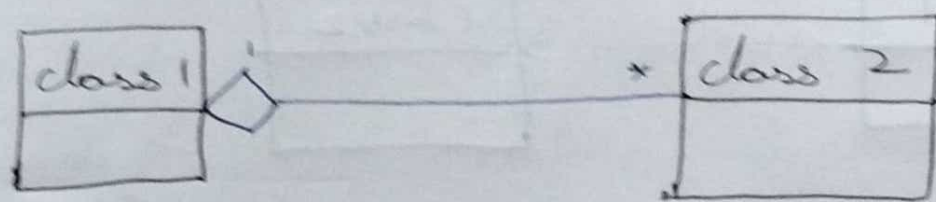
- A Structural link between 2 peer classes
- There is an association b/w Class 1 & 2
- A Solid line connecting 2 classes



### Aggregation

- A Special type of Association, it represents a part of (\*Consists of\*) relationship.

- class 2 is a part of class 1
- Many instances (denoted by \*) of class 2 can be associated with class 1.   
 (Note: An arrow points from the asterisk to the word "many" written above it.)
- Objects of class 1 & class 2 have separate lifetimes
- A solid line with an unfilled diamond at the association end connected to the class of composite



### Composition

- A Special type of Aggregation, where parts are destroyed when the whole is destroyed.
- Objects of class 2 live & die with class 1.
- class 2 cannot stand by itself.
- A solid line with a filled diamond at the association connected to the class of composite.





## Dependency

- Exists between 2 classes if we change to the definition of one many class, change to the other (but not the other way around).
- Class 1 depends on Class 2
- A dashed line with an open arrow.



(+) = public  
 (-) = private  
 (#) = protected  
 (~) = package

## Multiplicity

- How many objects of each class take part in the relationships and Multiplicity can be expressed as

Exactly one = 1

Zero or one = 0..1

Many = \* (or) 0..\*

One or More = 1 - - - - -

More or one = \* - - - - - 1

## Problem Statement

Unified Library Management System : It is a System that issues books and magazine to registered students only. The student have to login after registering. The borrower of the book can perform various functions such as Searching for designed book. Library is used to store books, but requires to navigate to a specific book. A library database System is an Infrastructure that allows users to search books and book content add or remove and download the selected books. The problem faced is library users require an efficient method to find a specific book or keywords with given a continuously ~~extant~~ expanding library. Efficiency requires a processing time should stay relatively the same even as the library contents increases. The Objectives of the project