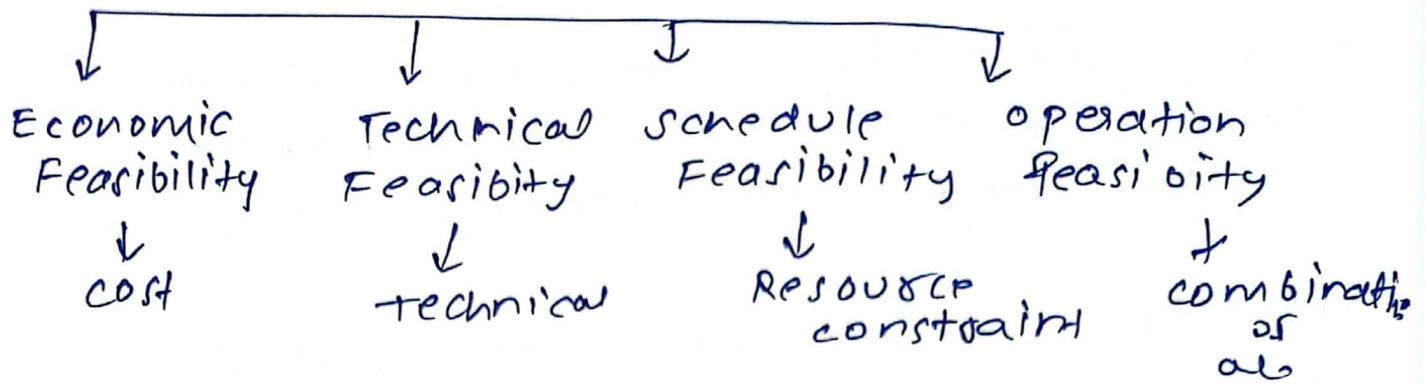
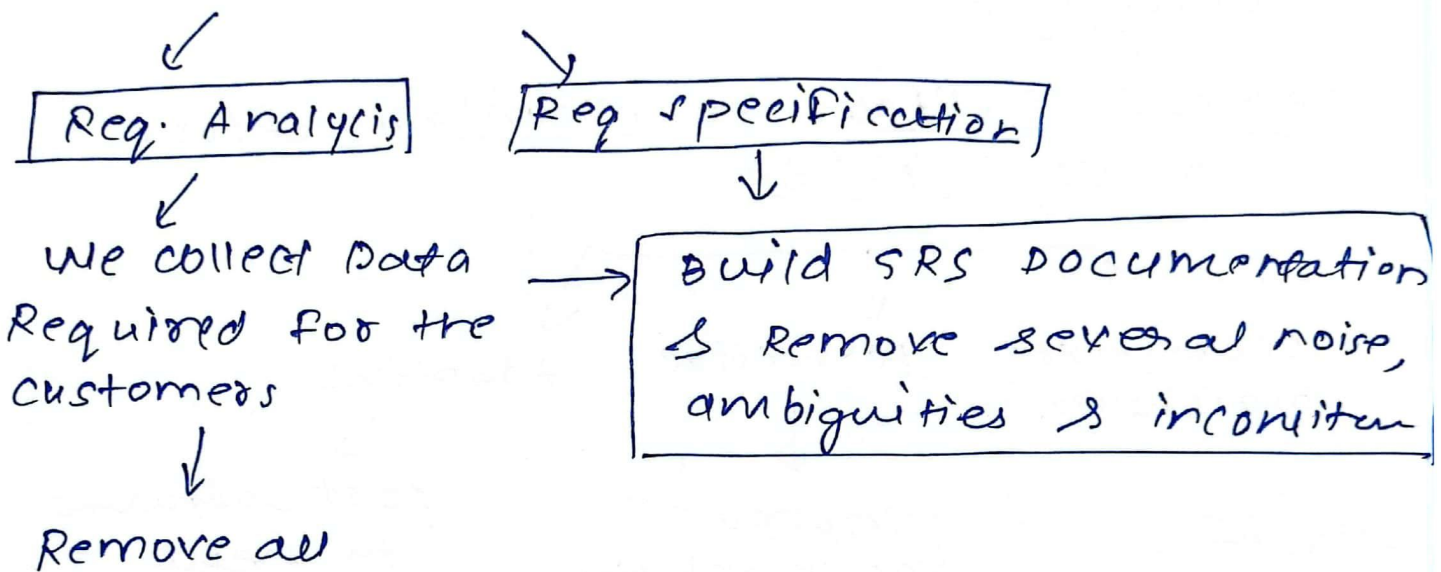


## Feasible study



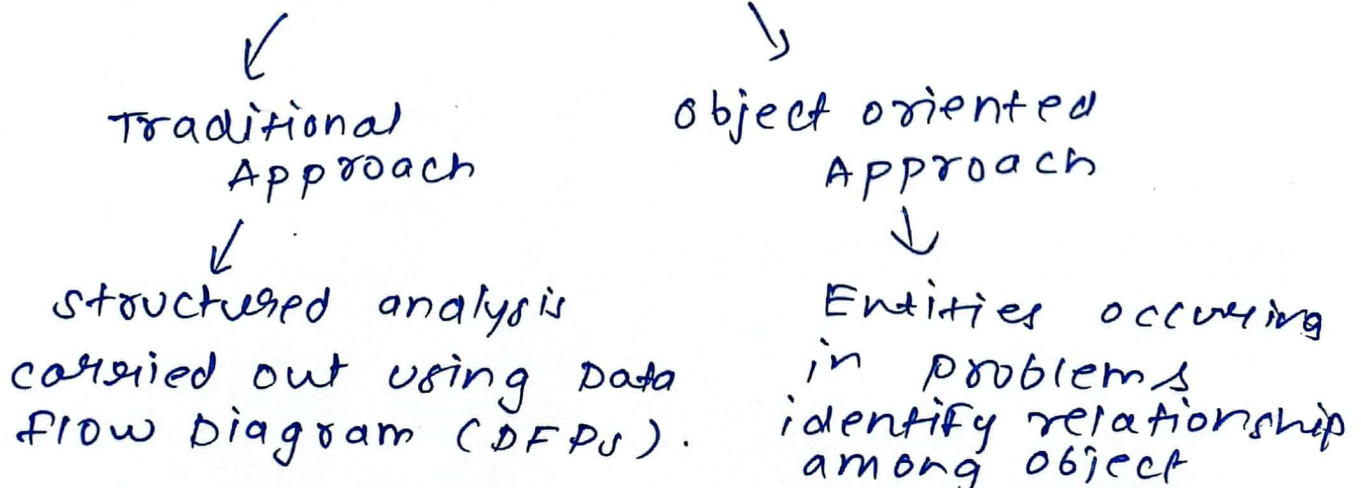
## Req. Analysis

customer needs → Document



## Design phase

During Design phase some software Architecture is involved



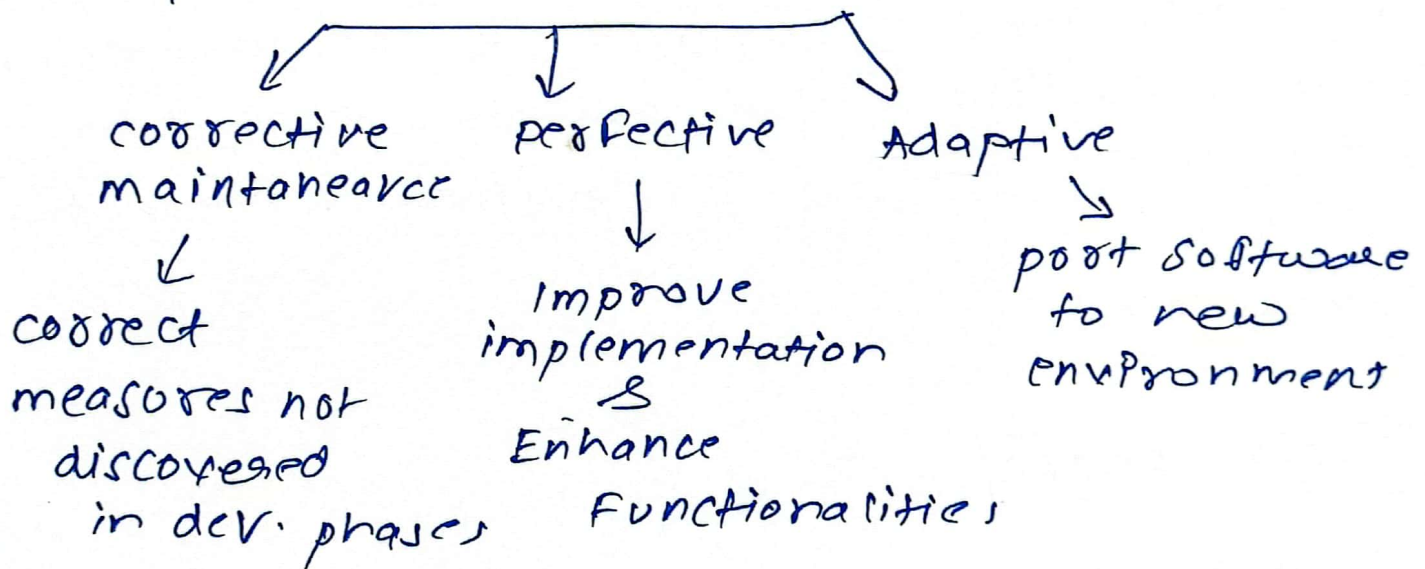
## coding and Testing

Each phase module is implemented into some programming Language and as per the design implemented earlier.

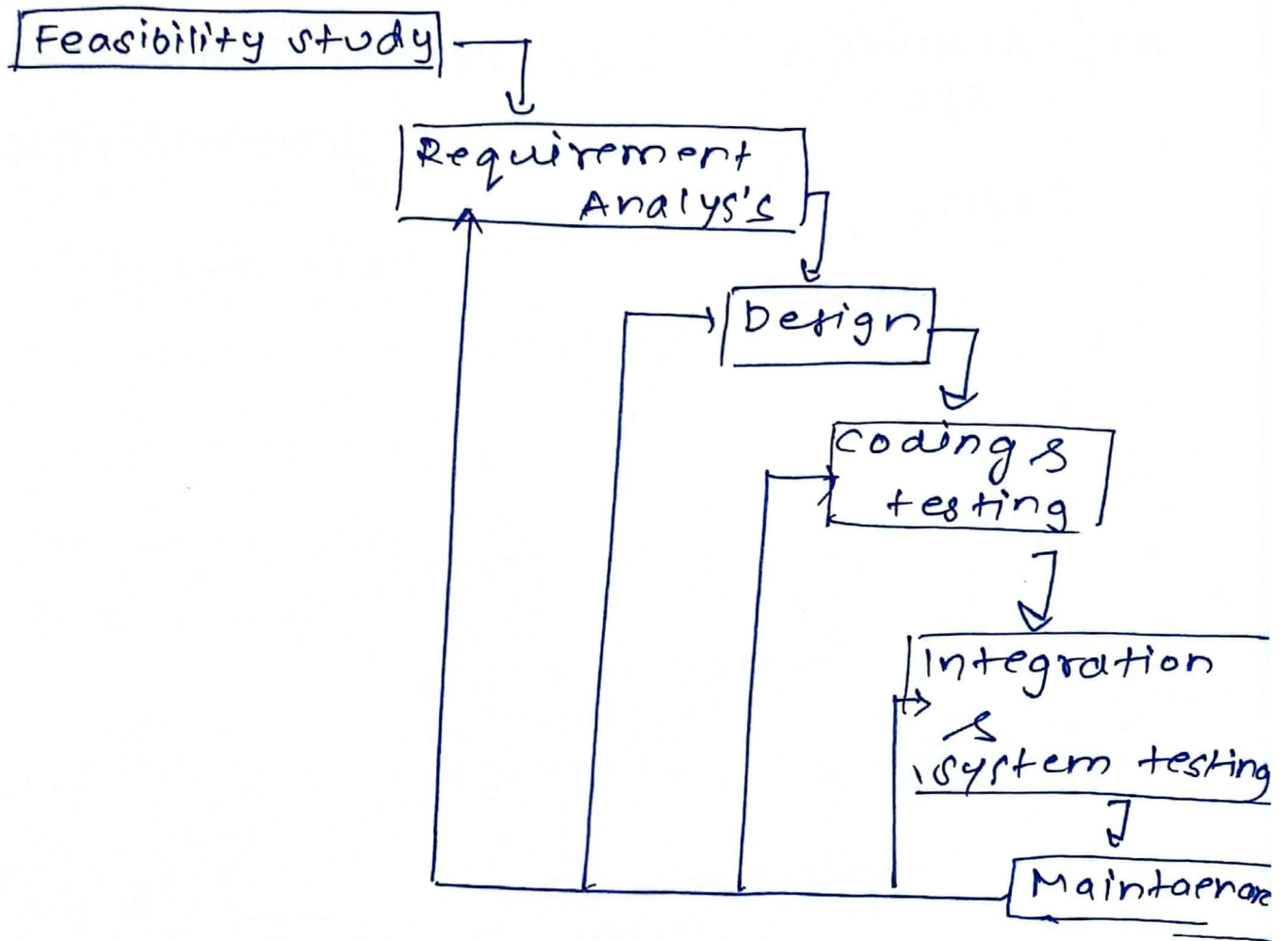
Goal of Testing is to ensure that the developed system functions according to requirements of SRS

## Maintenance

development effort to maintenance effort is typically - 40:60



software process model  
↓  
classical waterfall | Iterative waterfall model

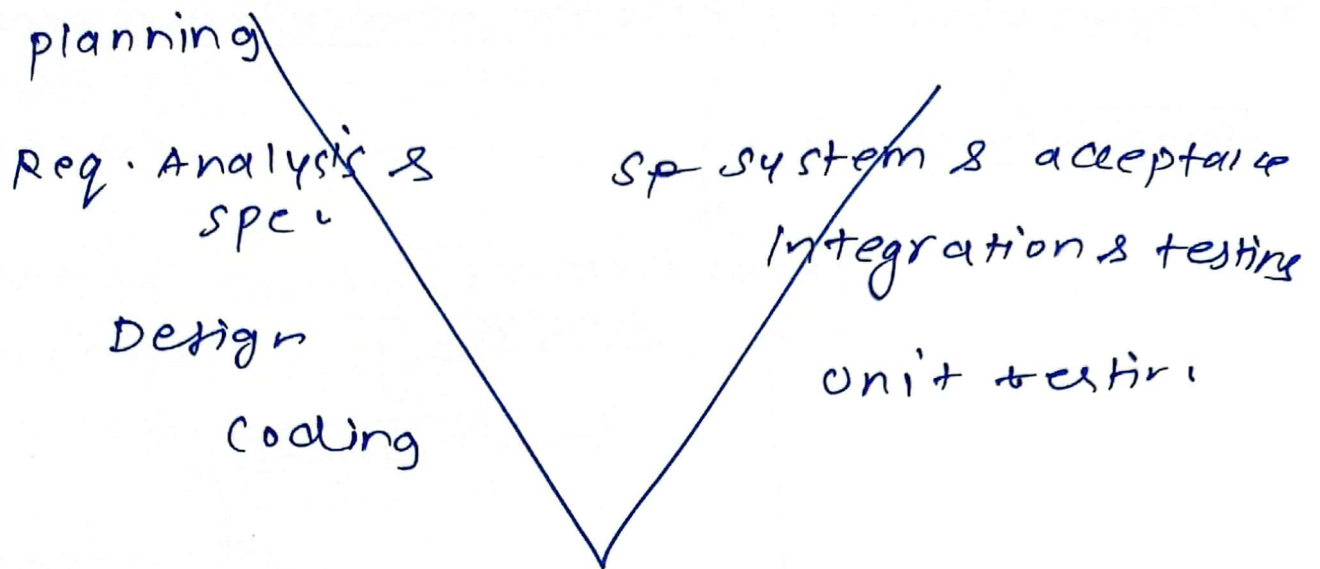


The principle of detecting errors as close to their points of occurrence is known as phase containment of errors

V-Model

it is a variant of waterfall with more focus on verification & validation

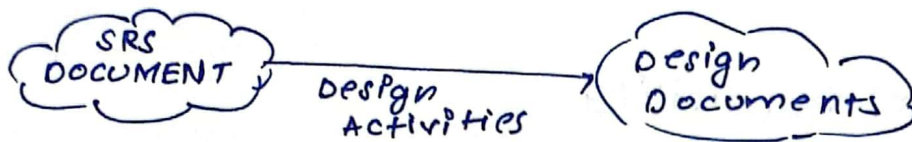
## V model steps





## Software - Design

Transform SRS Document to Design Document

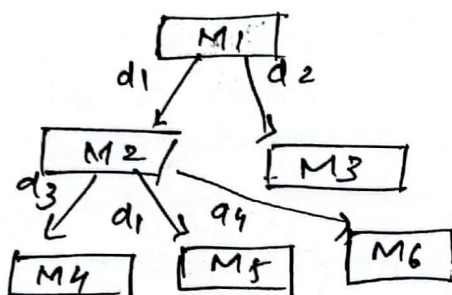


- Items Implemented during design Phase
- Module structure
  - Control relationship among the modules
  - Interface among different modules
  - Data-structures of individual modules

### stages in Design

preliminary Design  
(High-level)

detailed Design



structured chart  
other notation

Jackson Diagram or  
Warnier-orz diagram  
can also be used.

Each Module, design  
data-structure  
algorithms

outcome of detailed  
design

Module  
Specification

## Good software Design

- should implement all functionalities of the system correctly
- should be easily understandable
- should be efficient
- should be easily amenable to change  
i.e easily maintainable

## understandability of Design

- A Design that is easy to understand is also
- easy to maintain and change.

### How to Improve understandability

- use consistent and meaningful names
  - For various design components
- Design solution should consist of
  - A set of well decomposed modularity
- different modules should be neatly arranged in a hierarchy
  - (1) In a tree-like diagram called layering

# Software - Requirements

## Requirement Analysis AND Specifications

Requirement engineering (RE) refers to the process of defining, documenting and maintaining requirements in engineering process.

### Four steps

- Feasibility study
- Requirement Analysis
- Software Requirement specification
- Software Requirement validation

### Tools support for Requirements Engineering

- Observation users reports
- questionnaires
- use cases
- Requirement workshops
- Mind - mapping
- Role playing
- prototyping

### FUNCTIONAL V/s NON-FUNCTIONAL REQUIREMENTS

Requirements, which are related to functional/working aspect of software falls in category

Non Functional Requirements are expected characteristics of target software. (security, ~~stor~~ storage, configuration, performance, cost etc..)



## SOFTWARE REQUIREMENTS SPECIFICATION (SRS)

SRS is a description of a software system to be developed

Lays out functional and non functional requirements of the software to be developed

It may include a set of usecases

### USER REQUIREMENTS

- Easy & simple to operate
- Quick Response
- Effectively Handling operational errors
- Customer support

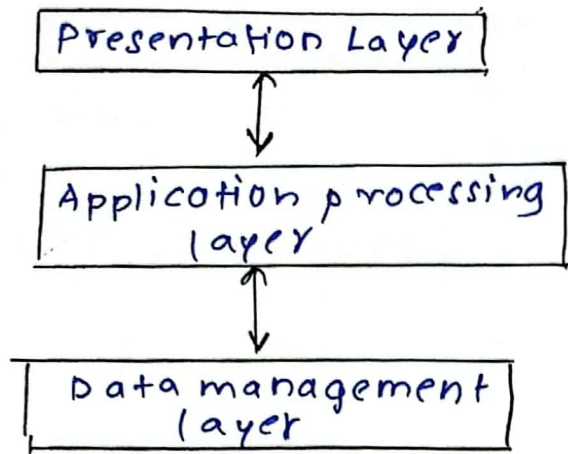
### USER REQUIREMENT SPECIFICATION

The user requirement(s) document (URD) or user requirement specification (URS) is a document that specify what users expects the software to do

Its contractual agreement.

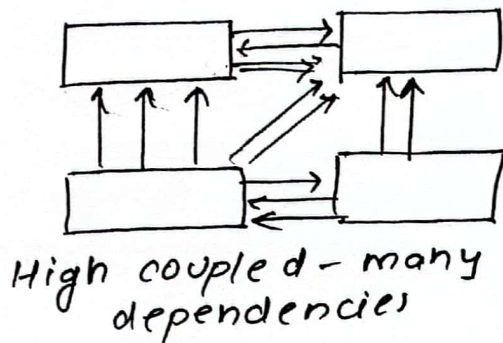
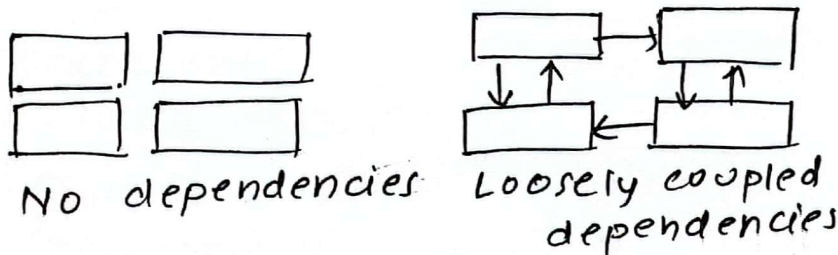


### 3-Layer architecture of application



- Modularity should display
- High cohesion
  - Low coupling

coupling: degree of dependence among component

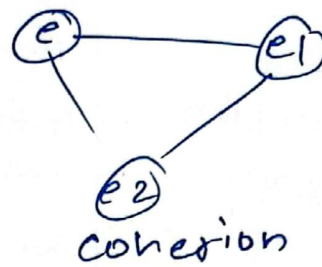
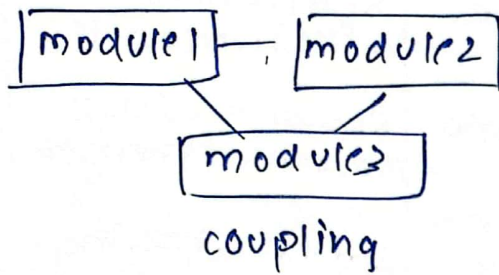


High coupling makes modifying parts of the system difficult. e.g. modifying a component to which the component is connected.

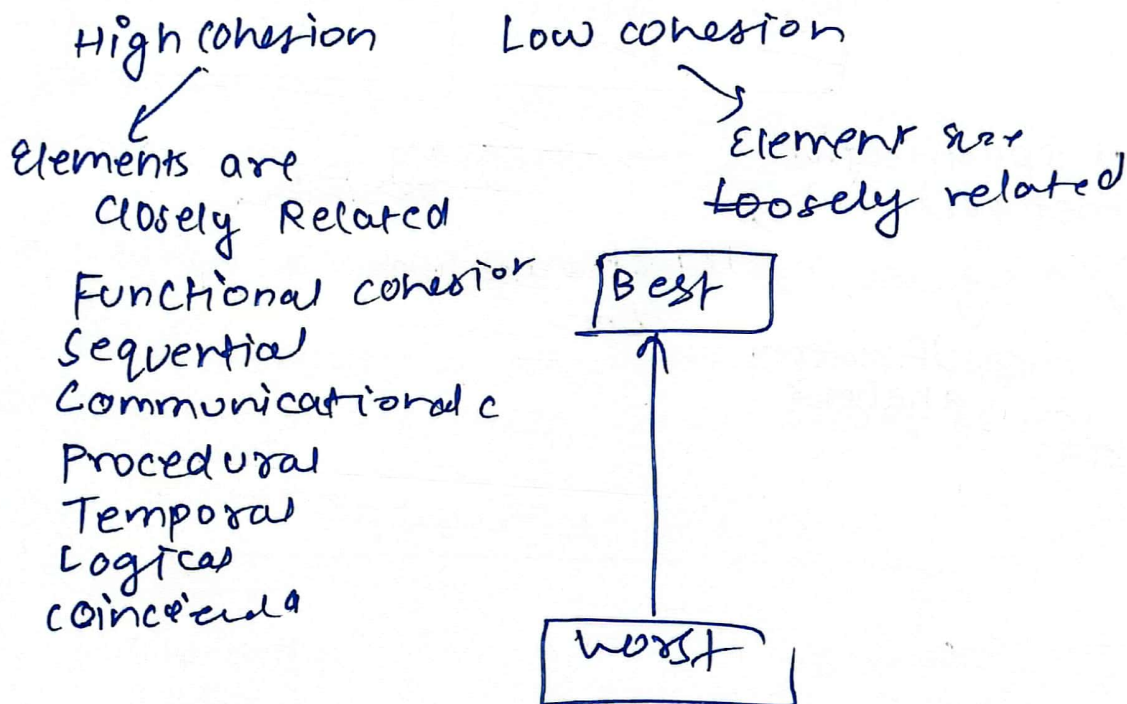
cohesion	coupling.
cohesion is a measure of functional strength of a module 1) <u>A cohesive module perform a single task or function.</u>	A measure of the <u>degree of interdependence or interaction</u> between the two module.

coupling and cohesion.

coupling and cohesion are two key ~~used to~~ concepts in software engineering used to measure quality of software



cohesion refers to degree to which elements within a module work together to fulfill a single well defined purpose



coupling refers to the interdependence between software modules

High coupling means the modules are closely connected to each other changes in one module can affect each other

MS  
NDSCEE

No direct  
Data coupling  
stamp com.  
control  
External  
common  
control

coupling	cohesion
coupling is also called inter-module binding	Cohesion is also called intra-module binding
Relationships b/w modules	Relationship within module
Independence b/w module	Shows module's relative strength
While creating aim for low coupling	While creating aim for high cohesion
