CMM:

① SEICMM stands for "Software engineering institute Capability maturity model.

It is a process or methodology used to develop and refine an organijation's Software development process.

2| The CMM provides the different levels based on the standards a company aquires so, if an IT company is a new company then the CMM will provide "Level 1" like this if the company starts developing and following the guidlines provided by CMM to increase the maturity level.

* The CMM provides total 5 levels:-

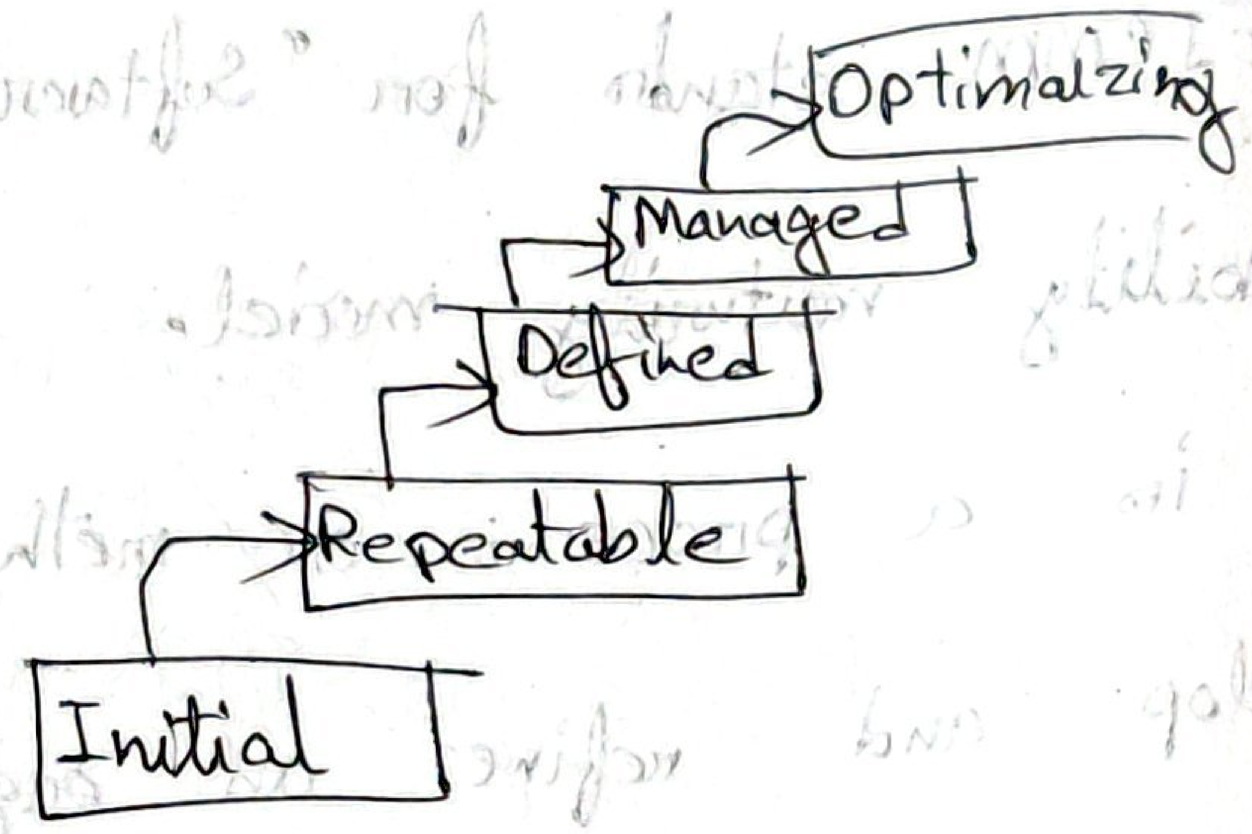
1| initial

2| Repeatable

3| Defined

4| Managed

5| Optimaizing



3| The CMM is developed by 'Softaware Engineering Institute' for an IT organization just to check/verify the maturity level of the organization.

COCOMO

* COCOMO Stands for

Level 1:

Initial:

No KPA's

Level 2:

Reppeatable

- Project ~~plannno~~ planning
- Configuration Management
- Requirements "
- Subcontract "
- Software Quality Assurance

Level 3:

Defined

- Peer Reviewes
- Intergroup coordination
- Organization Process Definition

- Organization Process Focus
- Training Program

Level - 4:

Managed -

- Software Quality Management
- Quantitative Management

Level - 5

Optimizing

- Process change management
- Technology change "
- Defect Prevention

COCOMO Model

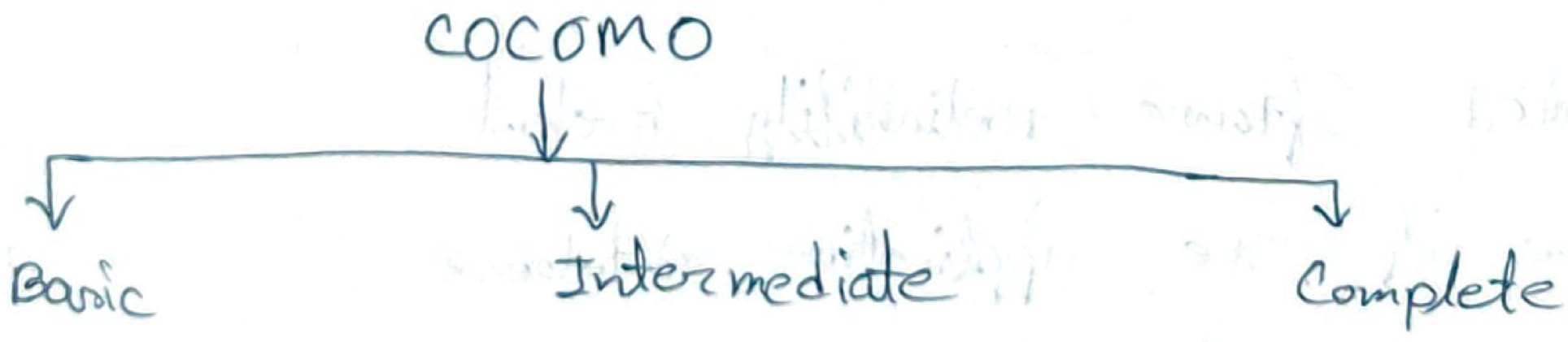
* COCOMO stands for "Constructive Cost model".

It is one of the very famous model which is used to estimate the of cost of the project.

* COCOMO model was proposed by Boehm in the year 1981

* Using this model we can estimate the time (month) and no. of people (members) needed to develop a project.

* According to Boehm ~~projo~~



① Basic COCOMO -

It pridict the effort and cost of project.

$E = a(KLOC)^b$

$time = c(Effort)^d$

Person required = Effort / time

② Intermediate COCOMO :-

This is an extension of the basic COCOMO model. The intermediate COCOMO model uses 15 additional predictors, Considering its environment to estimate a value or cost.

- Required software reliability extent
- Size of the application database
- The complexity of the product
- Run-time performance constraints

- Memory constraints
- The volatility of the virtual machine environment
- Required turnabout time
- Analyst capability
- Software engineering capability
- Applications experience
- Virtual machine experience
- Programming language experience
- Use of software tools
- Application of software engineering methods
- Required development schedule

(iii) Complete cocomo:

The complete COCOMO model is an extension of the intermediate COCOMO model. This model is phase sensitive. It does not depend on any phase, it is used to calculate the amount of effort required to complete each phase.

Six phases

1| Planning and requirements

2| System design

3| Detailed "

4| Module code and test

5| Integration and test

6| Cost Constructive model

Software design

① S/W design is a very important phase in the SDLC. It is a technique through which (Software Development life cycle) we can design a meaningful representation of something that we want to built.

② It is a process by which the requirement (SRS) are translated into blueprint for creating a S/W.

③ The blueprint gives us the complete details of working s/w.

④ The design must meet all the user requirement

⑤ The entire S/W must be breakdown

into no of modules known as "top level design"

6| It reduce effort and ocnoi

7| The required algorithm and data structure is used to implement a particula module known as detail design,

8| Sw designing concept → two part

- Top on high level desigh
- Internal design,

Software design process.

1| The architecture of the system: The set of components ~~interact~~. that make up the program.

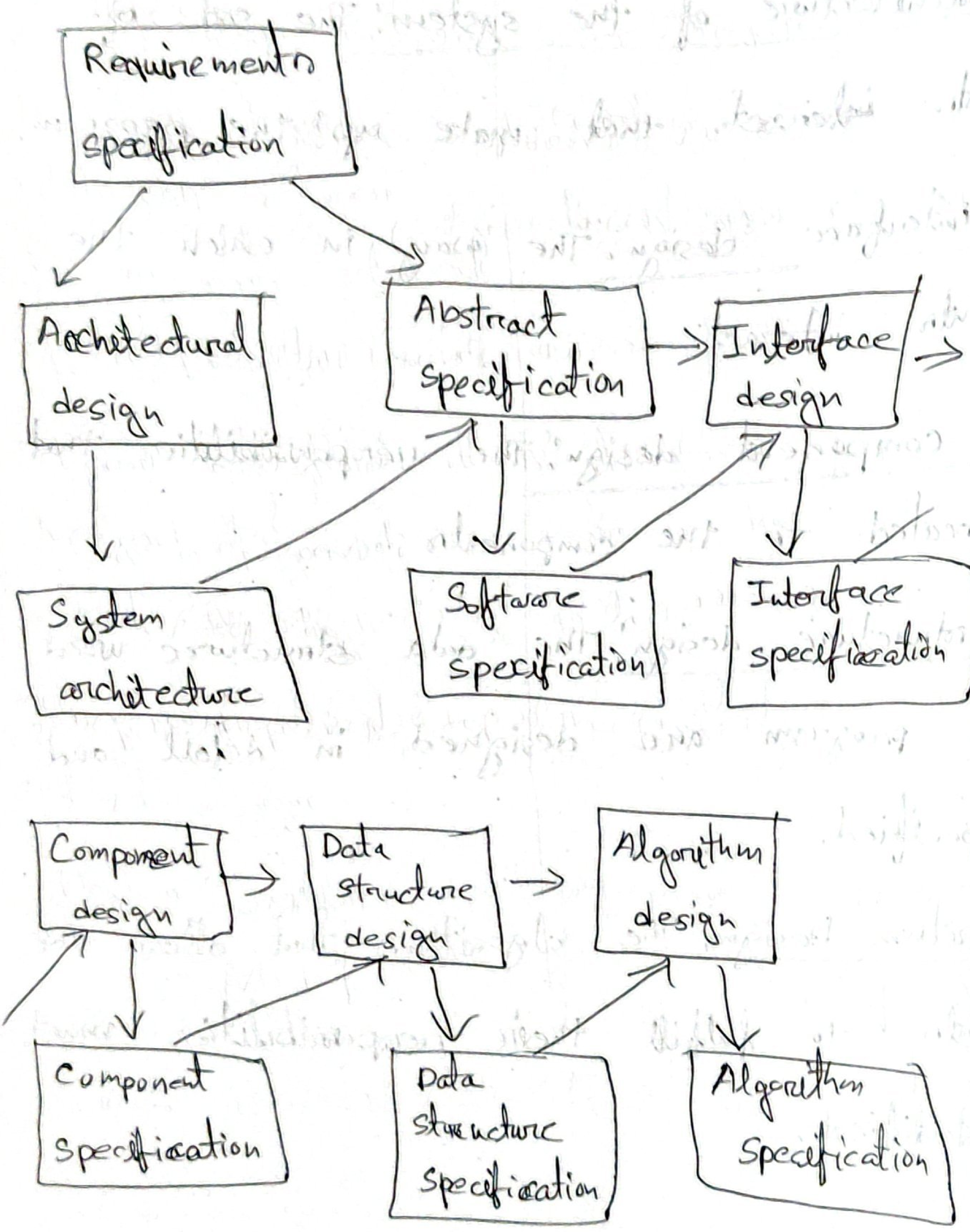
2| The interface design: The way in which the components interact.

3| The component design: The responsibilities that are allocated to the components

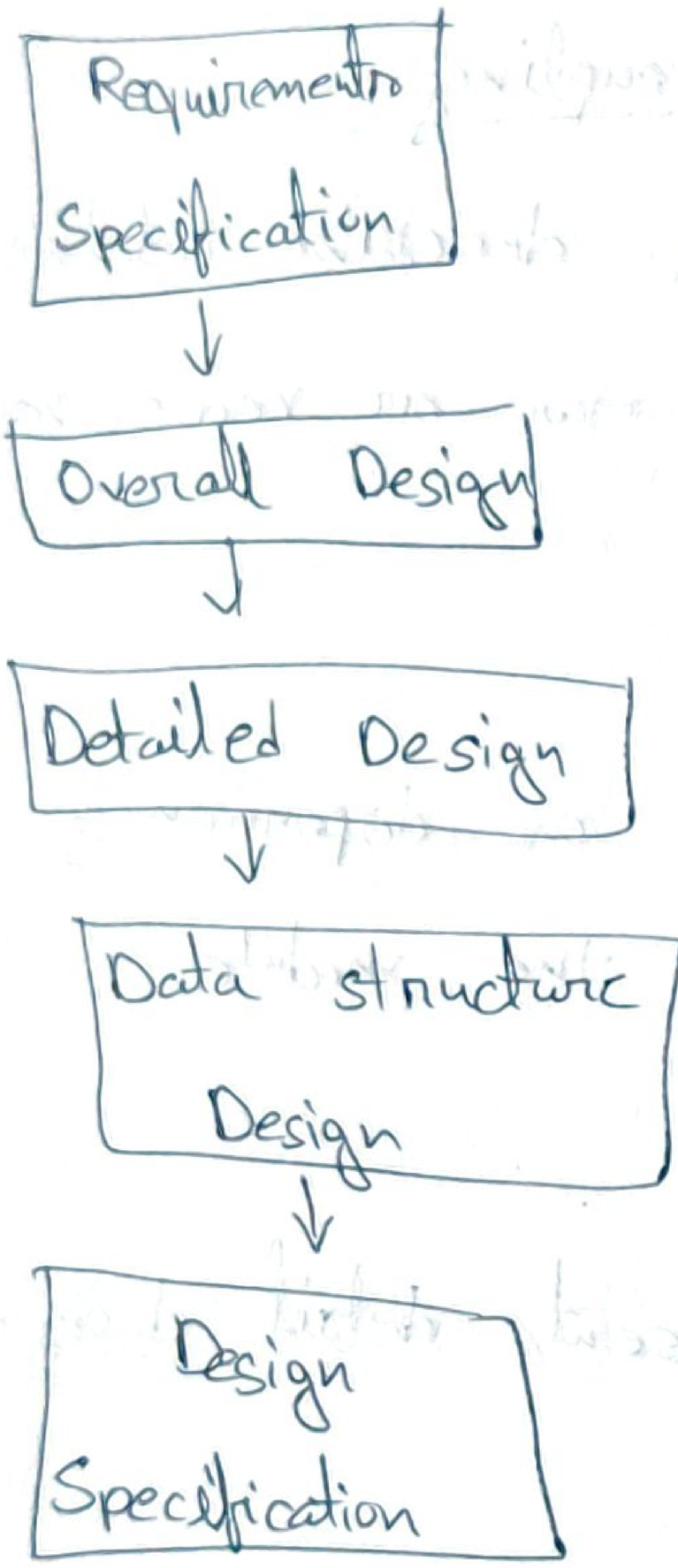
4| Data structure design: The data structure used in the program are designed in detail and and specified.

5| Algorithm Design: The algorithms that allow the components to fulfill their responsibilities must be identified.

A general model of software design process



A simple Design process:

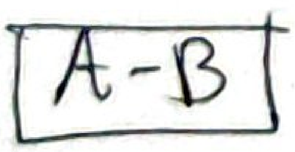
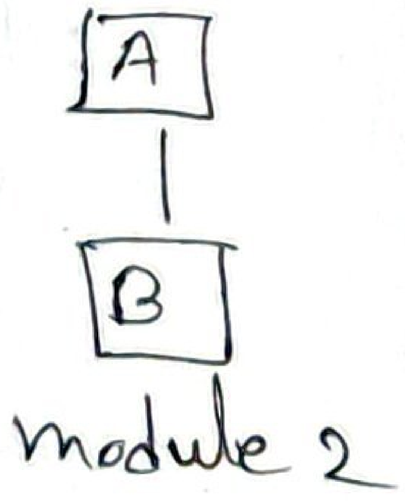


Overall Design: set the structure or shape of the software. It identifies the set of modules

Detailed Design: defines the processing that is carried out in the ~~it~~ modules.

Data Structure Design: chooses the data structures to be used.

*

Cohesion	Coupling
1| Cohesion describe relationship between two or more elements of a particular module	① Coupling describe relationship between two or more modules.
2| focus on functional strength of module	2| focus on dependency between the module
3| should be high	3| low
4| ~~repor~~ represent top - level design	4| represent detail design
5| Diagram - module 	5| Diagram - module1  module 2

Black Box testing
White Box testing

Black Box testing	White Box testing
1| In this testing method, the programming code of the software in hidden and nothing in known about it.	1| In this testing method, the programmer or tester must have knowledge about the internal programming code of the software.
2| Mostly done by software testers.	2| software developers.
3| can be referred as external software testing	3| internal software testing
4| It is less time-consuming	4| more
5| also called ~~cat~~ closed testing	5| also called as clear box testing.

6| No knowledge of programming in required	6| It in mandatory to have knowledge of programming
7| No knowledge of implementation in needed.	7| knowledge of implementation is required.
8| It is not well suitable for algorithm testing	8| It is well suitable and recommended for algorithm testing.
9| applicable to the higher levels of testing of software	9| lowe levels
10| types: funtional and Non funtional testing.	10| type: Path testing loop testing etc.
11| advantage: • Efficient for large segments • Separation between user's and developer's perspectives	11| advantage: • Efficient in finding error, hidden error and problems. • Help optimizing the code.

12| Disadvantage:

- limited coverage
- Inefficient testing

12| Dis:

- Might not find missing features.
- Requires code access.

Object oriented design (OOD)

* diffirent approaches of S|W design

1| Function oriented design (FOD)

2| Object oriented design (OOD)

Object oriented design (OOD):

- In ~~ob~~ OOD instead of concentrating on function we concentrate upon objects, an object

constains both data and functions which are used to create application.

- In OOP the entire system is viewed as an object.
- Each object has its own state and behavior.

*In object-oriented systems, there are the following additional dependencies:

- Class to class dependencies
- Class to method "
- Class to message "
- Class to variable "
- Method " "
- " " massage "
- " " method "

Issues in testing classes.

1) Fault Based testing:

- Depend on tester completely
- Good results with less efforts
- It misses Incorrect special specification

2) Scenario - Based Testing:

- User creates scenario
- It cover incorrect specifications
- Focuses on what user does, not what the Product does.

3) Random testing

4) Partition testing

5) Class testing Based on Method testing

Purpose of OOP:

1) Object Interaction validation

2) Determining Design Error

3) Finding integration Problems

4) Assessment of Reusable code

5) ~~Verifiat~~ Verification of Handling Exceptions

6) Verification of Uniformity