ASSIGNMENT 01

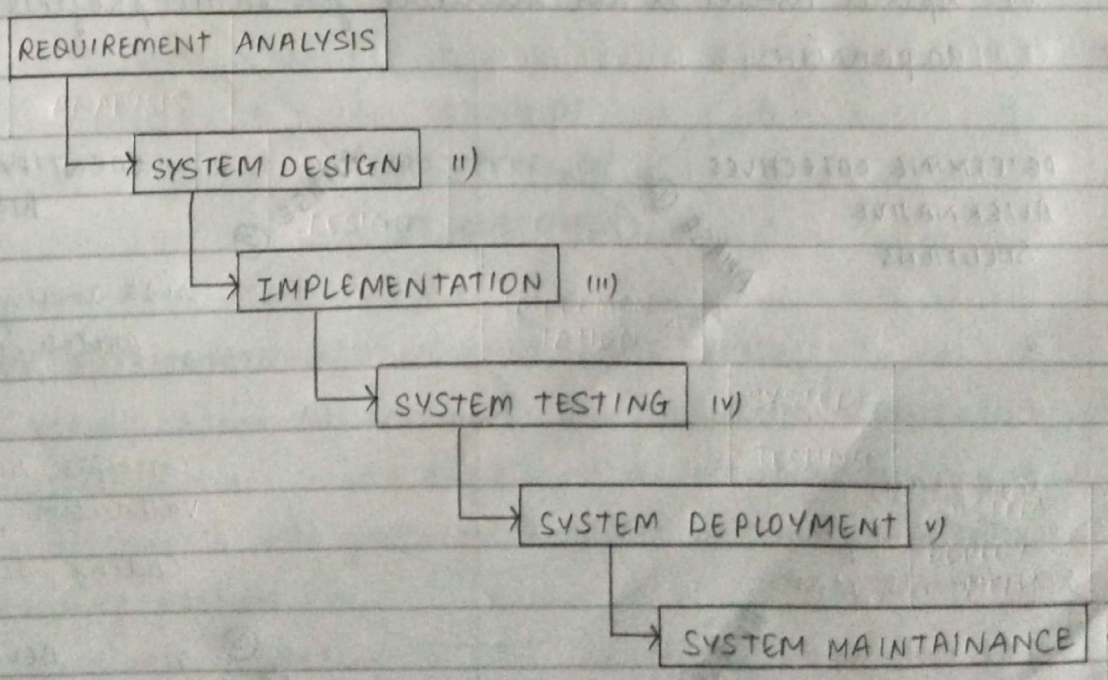
Q:1 Give detail explanation of all Process Models with advantages & disadvantages and suitable diagrams.

Ans: there are various types of Process Models:

i) **WATERFALL MODEL** : or LINEAR SEQUENTIAL MODEL

This is the most basic software development life cycle process which is followed broadly in the industry. Here the developers follow a sequence of processes where the processes flow progressively downwards towards the ultimate goal. It is like a waterfall where there are a number of phases.

- **Advantages** :
  a) Easy to implement and maintain.
  b) the requirement of resources is minimal and Testing is done after completion of each phase.
- **Disdvantages** :
  a) It is not possible to alter or update requirements.
  b) Cannot start the next phase until the previous phase is completed.



ii) **SPIRAL MODEL :**

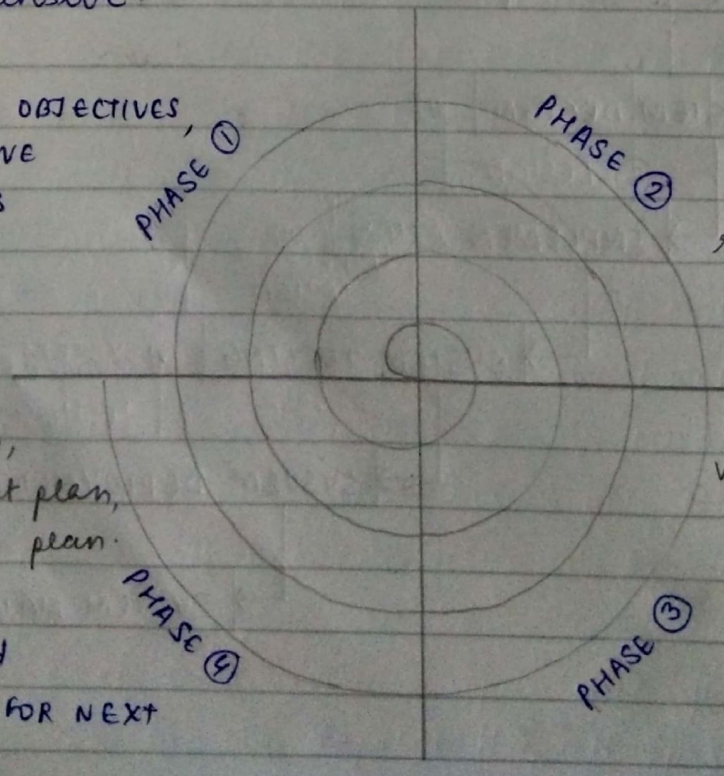
Its main emphasis is on RISK ANALYSIS and is based on RISK DRIVEN APPROACH. It has four phases : & its 4 quadrants are :

a) Objectives determination and identify alternative solutions.
b) Identify and resolve risks.
c) Develop next version of the product.
d) Review and plan for the next Phase.

* Four phases are the main base for the spiral model. or the baseline for the spiral model.
Each loop of the spiral is called a Phase of the software development process.

- **Advantages :**
  a) Spiral Model is the best development model to follow due to the risk analysis and risk handling at every phase.
  b) It is recommended to use the spiral Model in large and complex projects.

- **Disadvantages :**
  a) The spiral Model is much more complex than other SDLC models.
  b) Spiral Model is not suitable for small projects as it is expensive.



iii) **PROTOTYPING MODEL :**

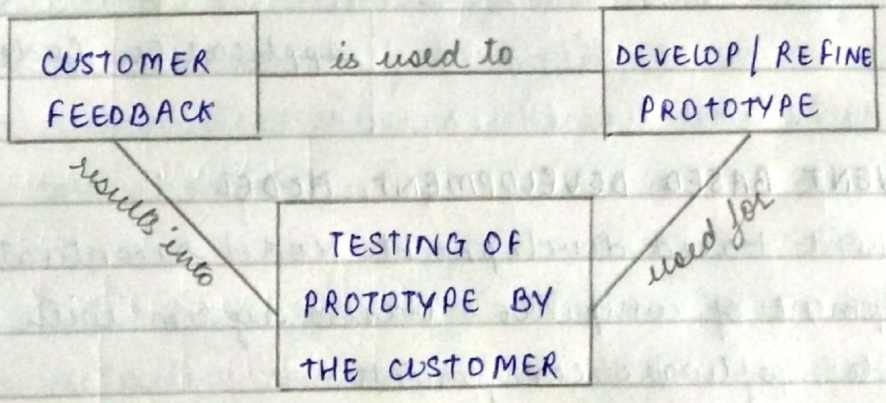
The prototyping model is a systems development method in which a prototype is built, tested and then reworks as necessary until an acceptable outcome is achieved from which the complete system or product can be developed.

There are 2 approaches for this model :

a) Rapid throwaway Prototyping.

b) Evolutionary Prototyping.

- Advantages :
  a) Missing functionalities can be easily figured out.
  b) New requirement can be easily accomodated as there is scope for refinement.
- Disadvantages :
  a) To much variations in requirements each time the prototype is evaluated by the customer.
  b) Poor documentation due to continously changing customer requirements.



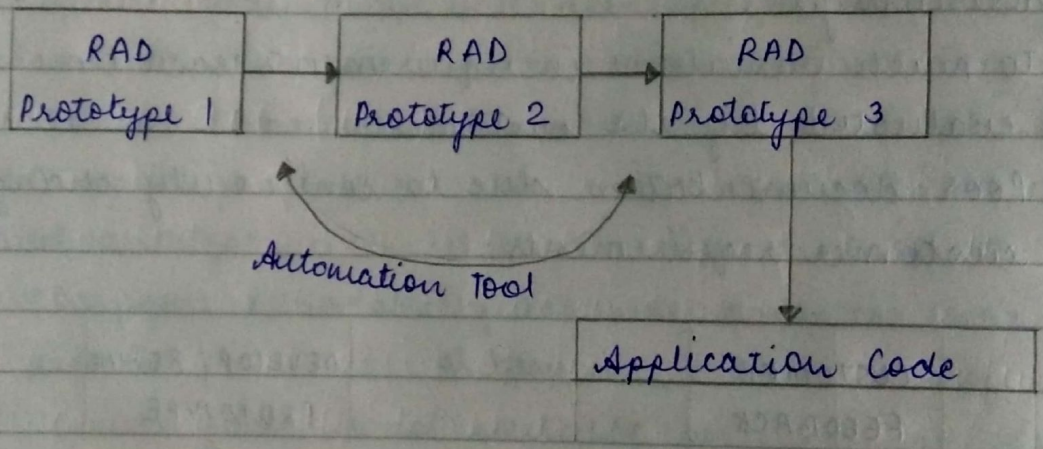
iv) **RAPID APPLICATION DEVELOPMENT (RAD) MODEL :**

Rapid application Development process is an adoption of the waterfall model ; it targets at developing software in a short span of time. RAD follows the iterative SDLC. RAD model has following phases :

- Business Modeling
- Data Modeling
- Process Modeling
- Application Generation
- Testing and turnover.

It focuses on input - output source and destination of the information.
The main features of RAD model are that it focuses on the reuse of templates, tools, processes, and code.

- **Advantages :**
  a) Flexible and adaptable to changes.
  b) Due to prototyping in nature, there is a possibility of lesser defects.
- **Disadvantages :**
  a) It can't be used for smaller projects.
  b) When technical risks are high, it is not suitable.
  c) Required highly skilled designers or developers.



VI) COMPONENT BASED DEVELOPMENT MODEL :

Component based development Model accentuates the design and development of computer - based system with the help of reusable software components.
There are multiple models under this :

* the Y Model
* the V Model
* the X Model
* CBSD Dual life cycle Model
* the Knot Model
* W Model

This Model is developed because of getting rid of making models continuously. It is based fully on Object Oriented technology.

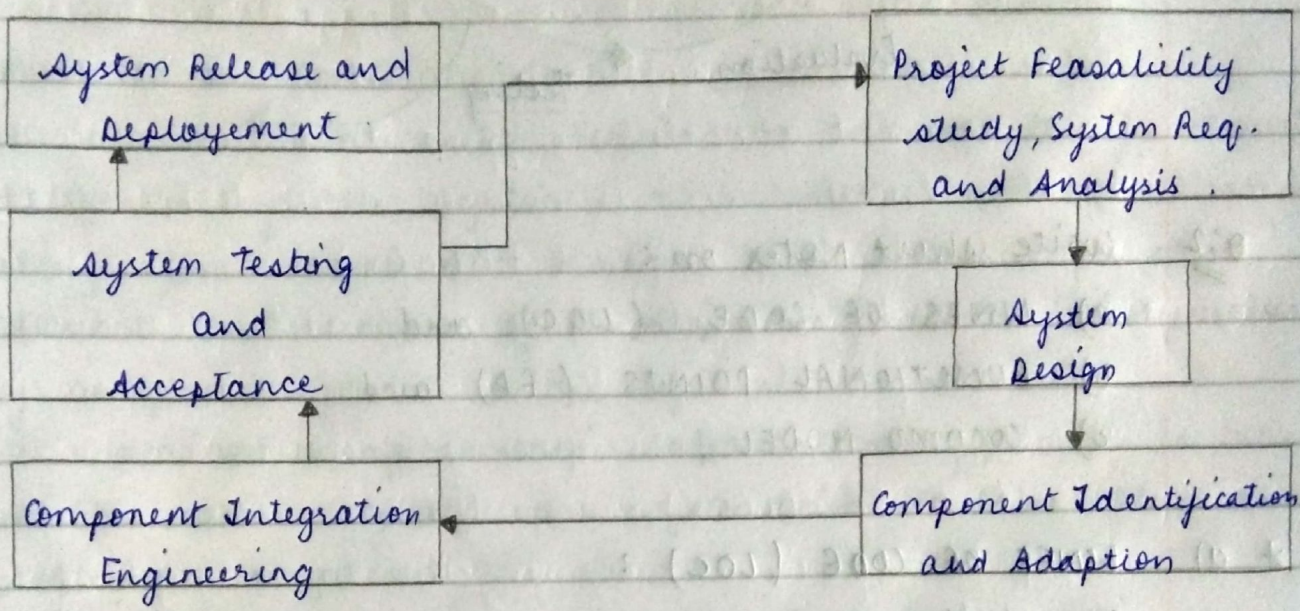
- **Advantages :**
  a) it is easy to manage & update large and complex software
  b) Components can be easily added, removed or updated and

are highly scalable.

- **Disadvantages :**

a) Components are heterogeneous in terms of programming lang., platform, data structures, naming conventions etc.

b) Expensive and inadequate Component-testing.



VI) **INCREMENTAL MODEL :**

This model combines elements of the linear sequential model with the iterative nature of prototyping.

When a incremental model is used, the first increment is often a core product, in which basic requirements are addressed, but many supplementary features remain undelivered. The core product is evaluated by the user or undergoes detailed review, as a result, a plan is developed for the next increment.

- **Advantages :**

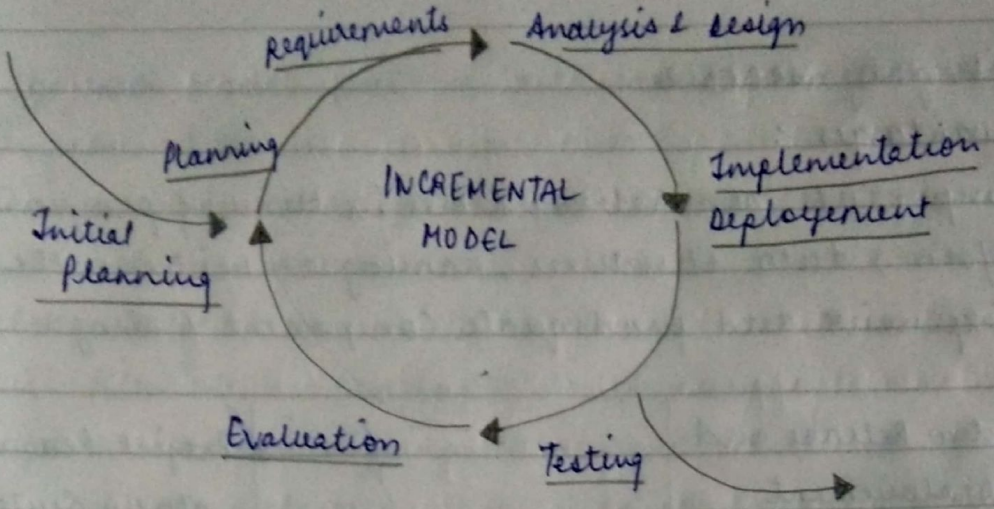
a) generates working software quickly and early during the software life cycle.

b) Easier to test and debug during a smaller iteration.

- **DisAdvantages :**

a) Each pase of an iteration is rigid and not overlap each other

b) Not all requirements are gathered up front for the entire software life cycle.



Q:2 Write short Notes on :

a) LINES OF CODE (LOC) and

b) FUNCTIONAL POINTS (FP) and

c) COCOMO MODEL.

* a) **LINES OF CODE (LOC) :**

One of the most commonly used software size metric is the line of code, which is highly dependent on the programming language. LOC can be defined as :

the number of delivered lines of code in software excluding comments and the blank lines.

LOC depends on the programming language chosen for the project.

However, the exact no. of lines of code can only be determined after the project is complete since less information about the project is available at the early stages of development.

Using data, project planner estimates three values for each size determination :

i) S_{opt} (optimistic),

ii) S_m (most likely),

iii) S_p (pessimistic)

An expected value S can be estimated or computed by the following equation :

$$S = (S_{opt} + 4S_m + S_p) / 6$$

- Certain guidelines are followed for determining LOC :
  * One line of code is for one logical line of code.
  * Declarations in the code program are counded as lines of code.
  * Comments in the programs are not counted as lines of code.

  .etc.

*b) **FUNCTIONAL POINT :**

or function Points metric is used to measure the functionality delivered by the system. FP estimates can help in estimating effort required to design, code & test software, predict the number of errors, and forecast the number of components used in the system.

FP is derived using an empirical relationship, which is based on the measure of software information domain value and software complexity.

Information Domain Value can be defined through :

i) Number of external Inputs (EI),
ii) Number of external Output (EO),
iii) Number of external inquires (EO),
iv) Number of internal logical files (ILF),
v) Number of external interface files (EIF)

Once all the information regarding information domain value is collected, complexity value is for each count is determined.

Organisations using FP method develops criteria, which helps in determining whether a particular entry is simple, average or complex.

A weighting factor in terms of numeric value is assigned for each level of complexity.

*c) **COCOMO Model :** (detailed explanation)

Constructive Cost Model, developed by Barry Boehm in the early 80's to estimate the total effort required to develop a software project.

COCOMO Model is commonly used as it is based on the study of already developed software projects. While estimating total effort for a software project, cost of development, management and other support tasks are included.

In this model, size is measured in terms of thousand of delivered lines of Code (KDLOCs).

- In order to estimate effort accurately, COCOMO model divides projects into three categories :

i) Organic Projects : these projects are small in size ($\leq$ 50 KDLOCs). In these projects, small teams with prior experience work together to accomplish user requirements, which are less demanding. ex: Library Management System.

ii) Embedded Projects : these projects are complex in nature ($\geq$ 300 KDLOCs). and the org. have less experience in developing such type of projects. Developed under tight constraints (hardware, software and people). ex: software systems used in avionics & military hardware.

iii) Semi-detached Projects : these projects are less complex ($\leq$ 300 KDLOCs) as the user requirements are less stringent compared to embedded projects. ex : operating system, compiler design, and database design.

* Advantages *

a) easy to verify the working involved in it.
b) Cost drivers are useful in effort estimation.
c) efficient and good for sensitivity analysis.
d) Can be easily adjusted according to the org. needs and environment.

* Disadvantages *

a) Difficult to accurately estimate size, in the early phase of the project.
b) Vulnerable to misclassification of the project type.

- Constructive cost model is based on the hierarchy of 3 models namely:

I) BASIC MODEL: In basic Model, only the size is measured in terms of KDLOC. only the size of project is considered while calculating effort. The calculation of effort is by using the following equation:

$$E = A \times (size)^B$$

∴ Here A and B are constants and E is efforts. the unit of E is Person Months (PM).

II) INTERMEDIATE MODEL: In this Model, parameters like software reliability and software complexity are also considered along with the size, while estimating effort. The Cost drivers are identified for each project:

a) Reliability
b) Complexity
c) Application experience
d) Programmer Capability.

The calculation of effort is by using the following equation:

$$\text{Total Effort} = EAF \times E_i$$

III) ADVANCED MODEL: In advanced Model, effort is calculated as a function of program size and a set of cost drivers for each phase of software engineering. this model incorporates all characteristics of the intermediate model & provides procedures for adjusting the phase wise distribution of the development schedule.

There are four phases in advanced COCOMO Model which are named as:

* requirements planning and Product design (RPD),
* Detailed Design (DD),
* Code and Unit Test (CUT). and
* Integration test (IT).

ACAP (analyst capability) Cost driver is found Through the Multiplying factors and then advanced Model effort is calculated.

ASSIGNMENT 02

Q: Describe SOFTWARE ENGINEERING and its characteristics with layered technology approach.

Ans: Software engineering is the establishment and use of sound engineering principles in order to obtain economically software that is reliable and works efficiently on real machines

Characteristics of software are:

I) Functionality : It refers to the degree of performance of the software against its intended purpose.

II) Reliability : A set of attribute that bear on capability of software to maintain its level of performance under the given condition.

III) Efficiency : It refers to the ability of the software to use system resources in the most efficient and effective manner.

IV) Usability :

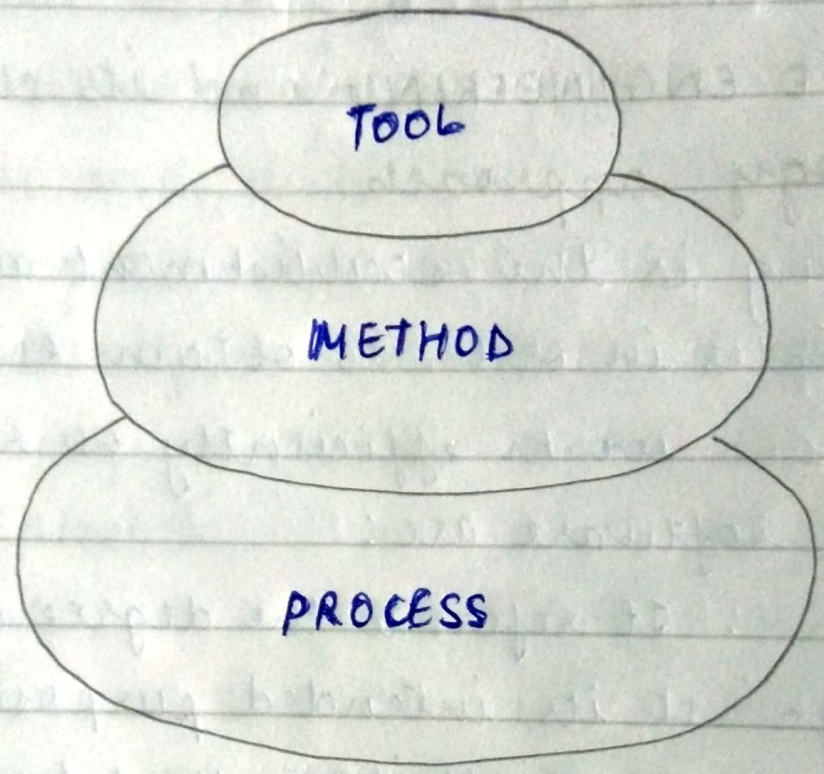
V) Maintainability , Portability.

* LAYERED TECHNOLOGY APPROACH :

- The PROCESS LAYER is an adhesive that enables rational and timely development of computer software. Process defines an outline for a set of key process areas that must be done for effective delivery of software engineering technology.

- The METHOD LAYER provides technical knowledge for developing software. this layer covers a broad array of tasks that include requirements, analysis, design, program construction Testing and support phases of the software development.

- the TOOLS LAYER provides computerized or semi-computerised support for the process and method layer. Sometimes tools are integrated in such a way that other tools can use information created by one tool. The multi-usage is known as Computer Aided Software Engineering (CASE).



Layers of software Engineering

ASSIGNMENT 03

Q:1 Case study and detailed description of higher education online library system with diagram.

Ans: As the name suggests, online library system is related to the storage of information regarding the library. It is place from where the students are the faculties issue the books for their reference purposes. But the maintenance of keeping the records of issuing and borrowing is difficult. To make this task easier, online library system will be very useful. It helps in maintaining the info. regarding the issuing and borrowing of books by the students and faculties.

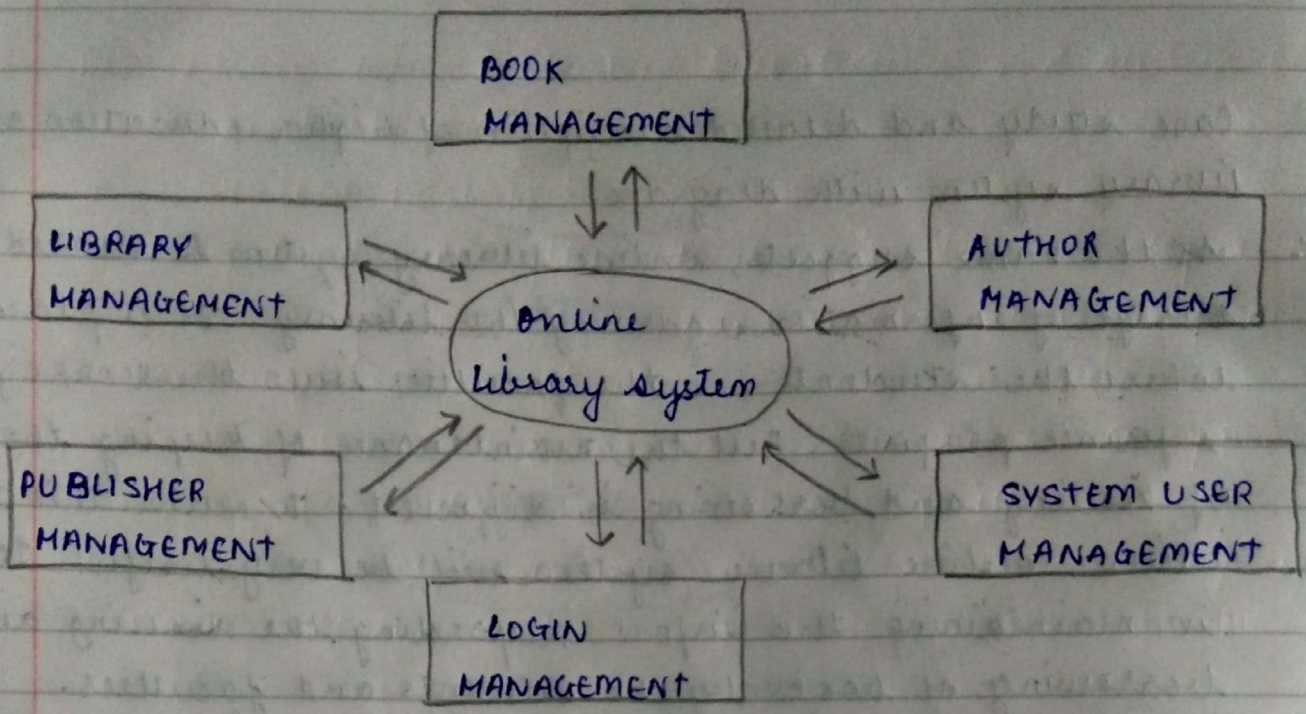
* PURPOSE :

1) to provide a friendly environment to maintain the details of books and library members.
2) Improved user service through greater access to accurate information.
3) due to computerized information it reduces the risk of paper work such as file loss, file damaged and time consuming. It can help user to manage the transaction or record more efficiently and timesaving.

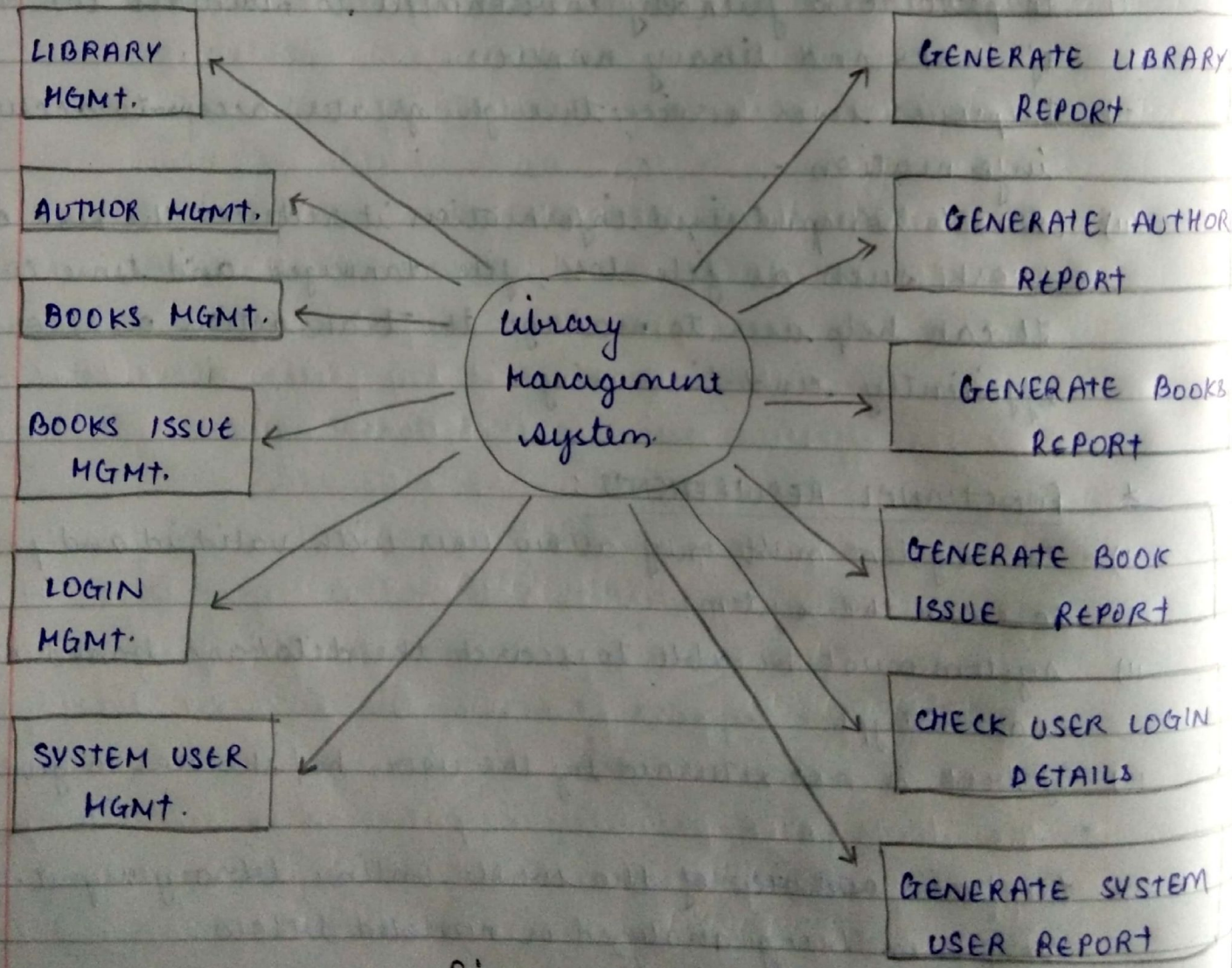
* FUNCTIONAL REQUIREMENTS :

1) The system must only allow user with valid id and password to enter the system.
2) System must be able to search the database based on slect search type.
3) if book is not returned by the user, he/she has to give fine.

* the basic overview of the whole online library mgmt. system or process being analyzed or modeled below:

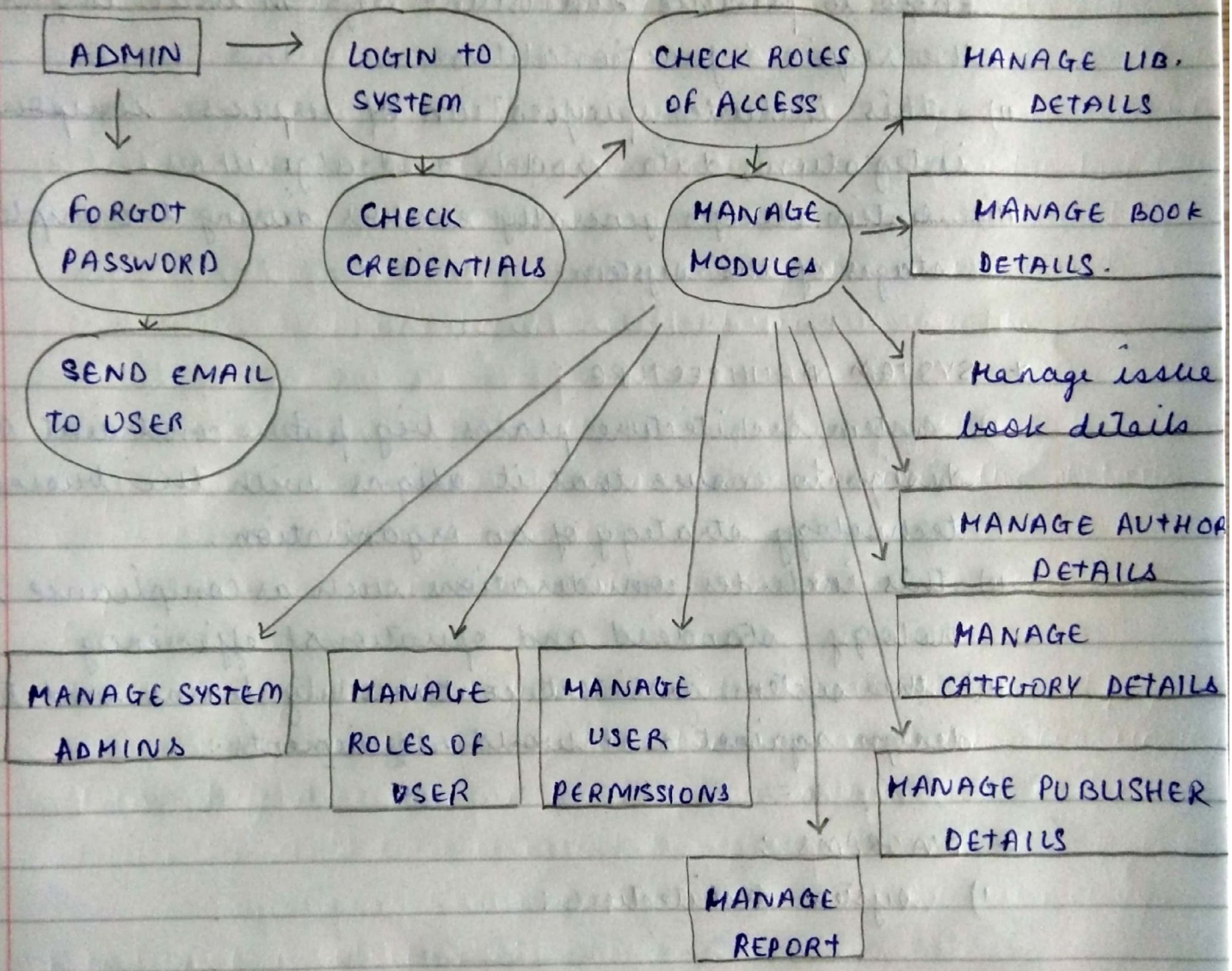


Zero level DFD



First Level DFD

It shows the system is divided into sub-systems each of which deals with one or more of the data flows to or form from an external agent and which together provides all of the functionality of the online library system as whole.



second level DFD

This level goes deeper into parts of level1 of library Management. It requires more functionalities of library management to reach the necessary level of detail about the library Management functioning.

Q:2 Differentiate between System Architecture and system design with suitable diagrams.

Ans: * SYSTEM DESIGN :

i) It provides everything that a system developer need to know to produce consistant system that implement the required functionalities.

ii) this includes specification of services, components, integration, data models and algorithms.

iii) System design generally evolves during the implementation stages of the system.

* SYSTEM ARCHITECTURE :

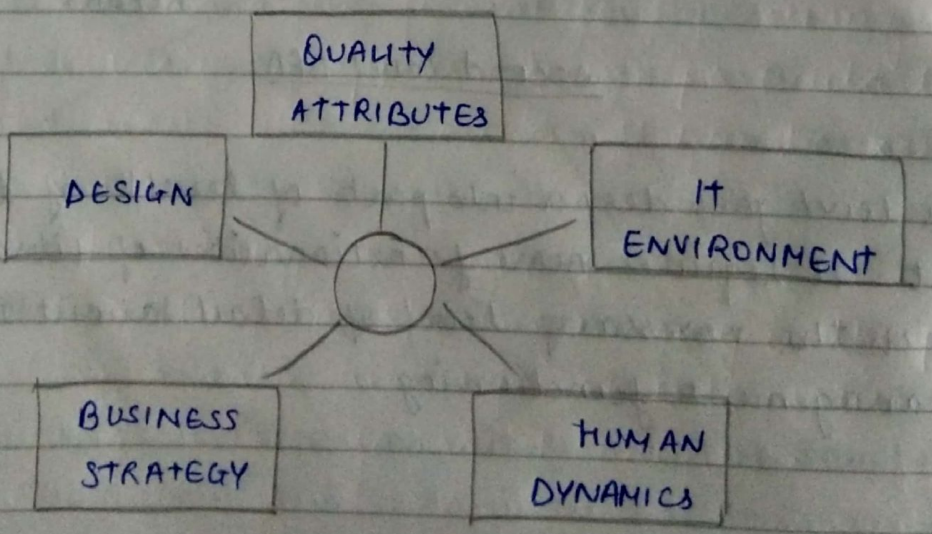
i) System Architecture places big picture constraint on the design to ensure that it aligns with the business and technology strategy of an organisation.

ii) This includes considerations such as compliance, technology, standard and operational efficiency.

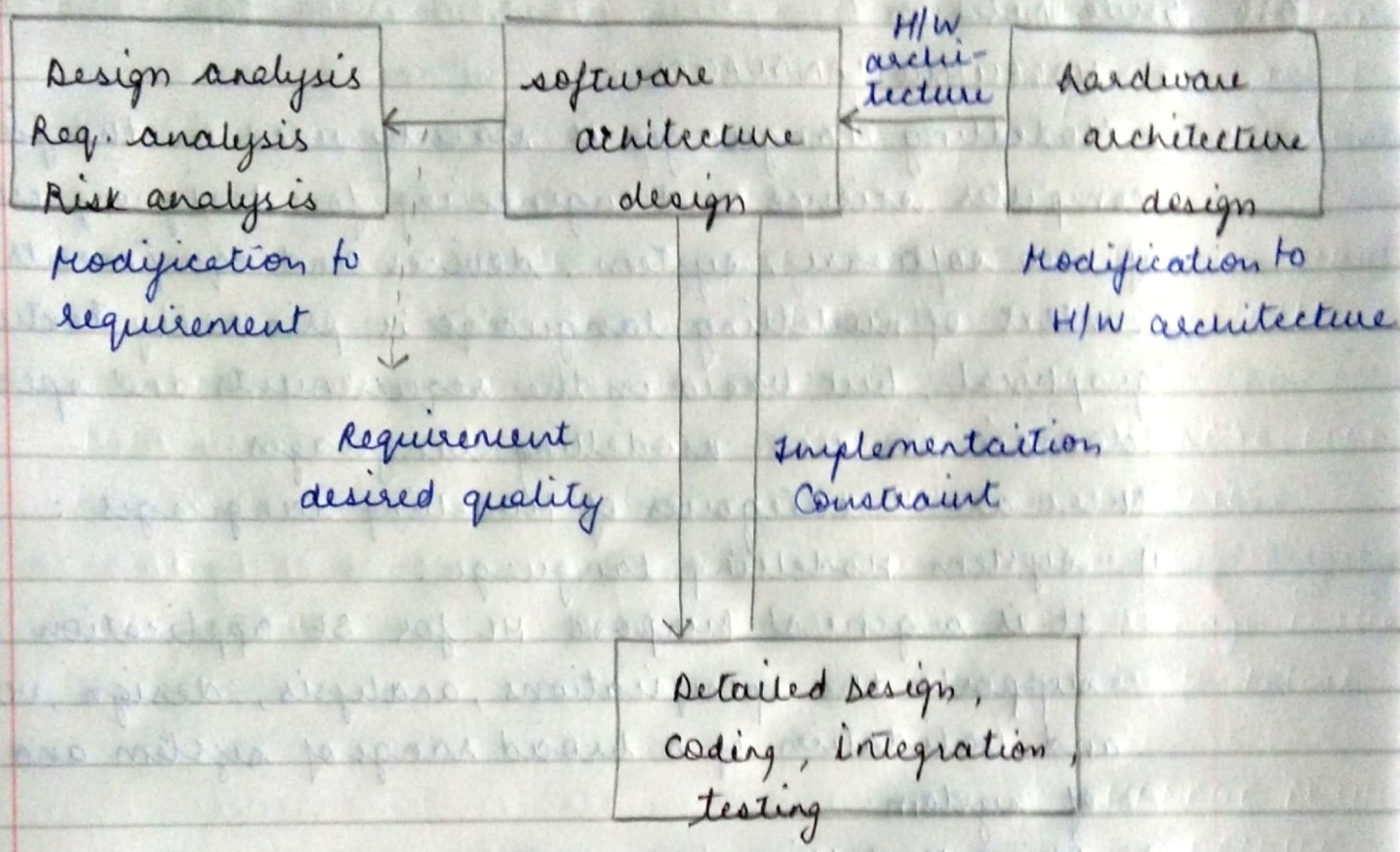
iii) the system architect continuously learns and tests the design against real world requirements.

* DIAGRAMS :

1) System Architecture :



II) System Design :



ASSIGNMENT 04

8:1 Write Notes on :

a) MODELLING LANGUAGES :

Modelling languages is mainly used in the field of computer science and engineering for designing models of new software, system, devices and equipments. The context of modelling language is primarily textual and graphical, but based on the requirements and specific domain in use, modelling languages.

there are 4 categories of Modelling languages :

I) System Modelling language :

It is a general Purpose ML for SE application. It supports the specifications, analysis, design, verification and validation of a broad range of system and system of system.

II) Object Modelling language :

It is a standardized set of symbols used to model a S/W system using OO framework. the symbol can be either informal or formal ranging from predefined graphical templates to formal object models defined by grammer and specifications.

III) Virtual Reality Modelling language :

It is standard file format for representing 3-D interactive vector graphics, design particularly with the world wide web in mind. It has be superseded by X3D.

IV) Data Modelling language :

Data Modelling in S/W engineering is the process of creating a data model for an information system by applying certain formal Techniques.

b) Models FOR SOFTWARE DEVELOPMENT :

the software development Models are the various processes or methodologies that are being selected for the development of the project depending on the projects goals and aims.

The models specify the various stages of the process and the order in which they are carried out.

The selection of models has very high impact on the testing that is carried out.

It will define the what, where, and when of our planned testing, influence regression testing and largely determines which test techniques to use.

There are various software development models such as

- Waterfall Model
- V Model
- Incremental Model
- RAD Model
- Agile Model
- Spiral Model
- Prototype Model.

ASSIGNMENT 05

Q:1 Give detailed description of TESTING with types and suitable diagram and provide a case study.

Ans: Software testing is a process, which is used to identify the correctness, completeness and quality of software.
Software testing is often used in association with the terms verification and validation.
Verification refers to checking or testing of items, including software for conformance and consistency with an associated specification.
Validation refers to the process of checking that the developed software is according to the requirements specified by the user.

CASE STUDY :

Objective :

- to remove errors, which prevent software from producing outputs according to the user.
- to remove error that leads to software failure.
- to improve the quality of software by removing max. possible error from it.

Casestudy on an online BOOK App. That is WATTPAD.

Wattpad is an online BOOK store app that as multiple and maximum number of books available for downloading for free. But before releasing this app, it has to undergo various types of testings :

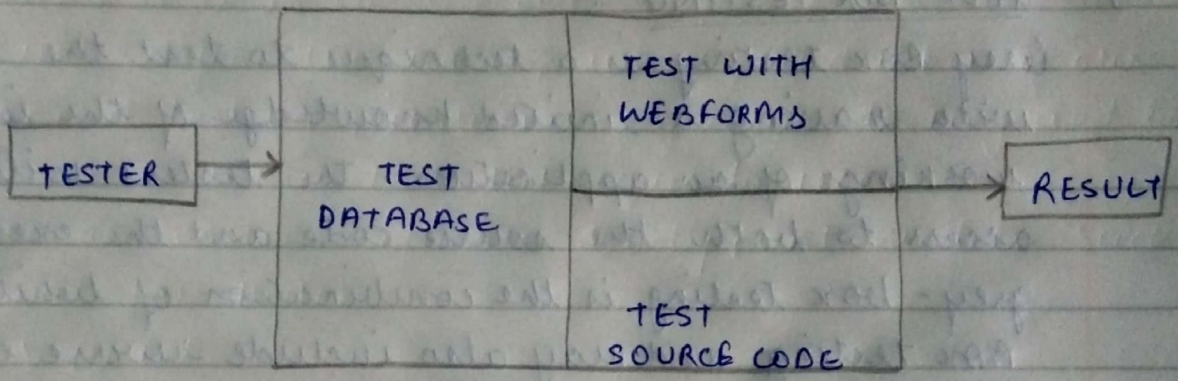
1) WHITE BOX TESTING :

White box Testing a.k.a structural testing, transparent box Testing etc) verifies the internal structure or workings of a program as opposed to the functionality exposed

to the end user. In white box testing, an internal perspective of the system as well as programming skills are used to design test cases.

working process -

- Input - Requirement, functional specification, source code
- Processing : Performing risk analysis for guiding
- Proper-test Planning : Designing test cases.
- Output : Preparing final report of the entire testing process



ii) BLACK BOX TESTING :

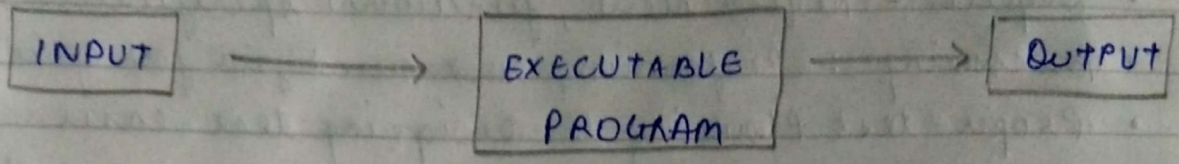
Black Box Testing (a.k.a. functional testing) is a S/W testing technique in which functionality of a software is tested without looking at the internal code structure, implementation details and knowledge of internal paths of the system software. This testing is based on the software requirement and specifications. In Black-Box testing we just focus on input and output of the s/w system without bothering about internal knowledge of the software program.

It can be done in following way:

- Syntax Driven Testing : This type of testing is applied to the system that can be represented through syntax by some language.

- Equivalence Partitioning : It is often seen that many

types of inputs work similarly so instead of giving all of them separately we can group them together & test only one input of each group.



(iii) GREY BOX TESTING :

Grey Box testing is a technique to test the application with having a limited knowledge of the internal workings of an application. The testers will often have access to both the source code and the executable binary. grey-box testing is the combination of white and Black Box testing and may also include reverse engineering to determine, for instance boundary values or error messages.

Benefits :

- It provides combined benefits of both white box and black box Testing. It can handle complex design test scenarios more intelligently.
- Drawbacks :

- Complete white-box testing cannot be done due to inaccessible source code / binaries.

It is difficult to associate defects when we perform grey-box testing for a distributed system.