

Unit \Rightarrow 6

Software Process Model

6.1 Software Project Concept

A project is a well-defined task which is a collection of several operations done in order to achieve a goal. A software project is a complete procedure of software development from requirement gathering to testing and maintenance, in a specific period of time to achieve desired software product.

6.2. Concept of software development process

Software development is a process programmers use to create computer programs. This process is also known as software development life cycle (SDLC). Its aim is to create an effective products within defined budget and time. The softwares are developed for the variety of purposes:

- ① To solve a real world problem.
- ② For personal use.
- ③ To meet the specific need of client or business.

6.3 Concept of SDLC

Software is not a small program, they are the combination of several programs integrated together. These softwares can be developed by using the sequence of steps which are solved under software development life cycle.

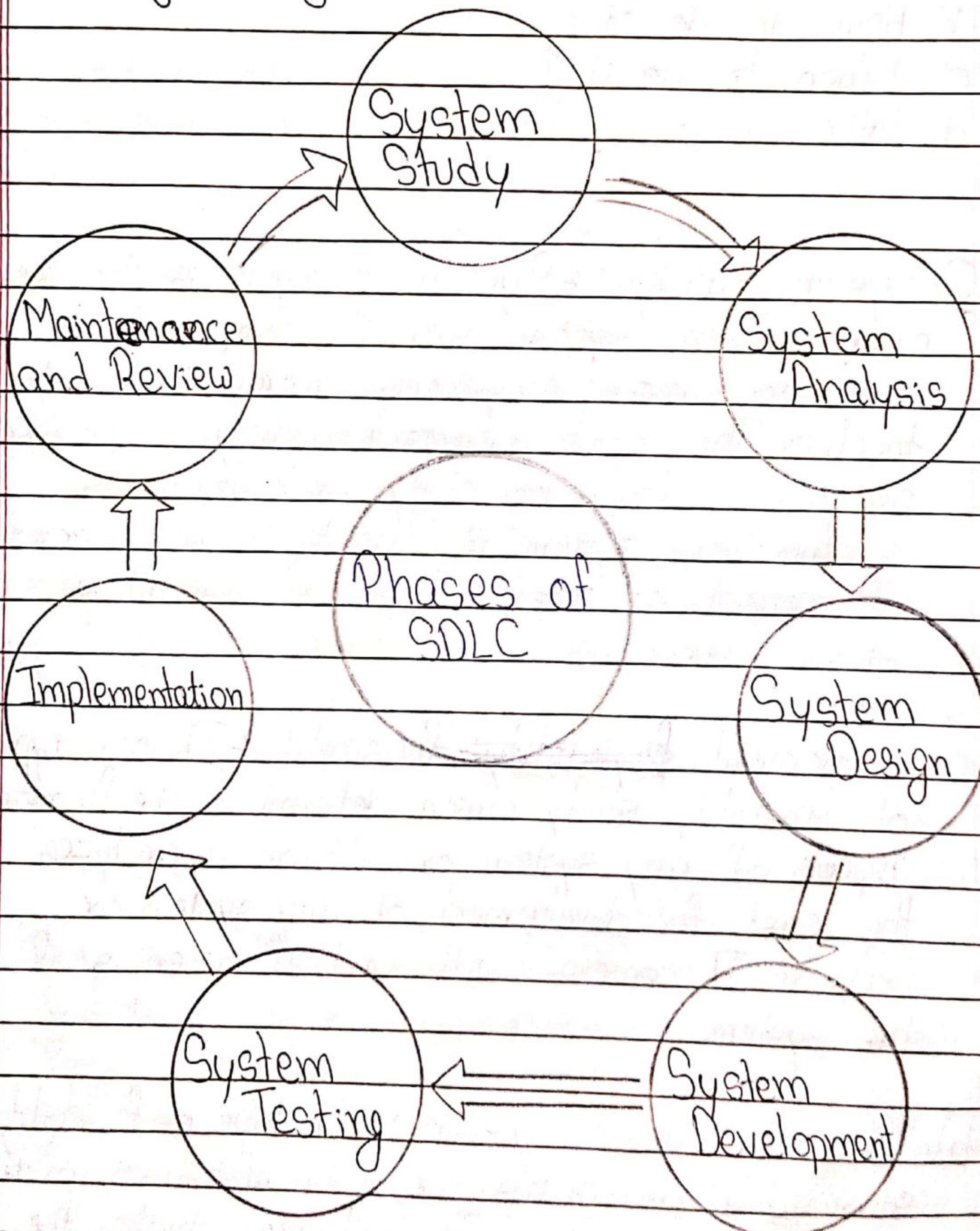
SDLC is a step-by-step process where each stage is a building block of an effective and efficient program.

Importance of SDLC

- ① It is a base for project planning, scheduling and estimating.
- ② It provides framework for a standard set of activities.
- ③ It increases and enhances development process.
- ④ It improves client relations.
- ⑤ It helps to decrease project work.

Phases of SDLC

As the computing power increases, it places a higher demand on softwares and developers, companies must reduce cost, deliver software faster and meet the customer needs. For this SDLC plays a vital role in achieving the goals.



① **System study:** This is the first stage in SDLC process in which the needs or requirements are collected. This stage gives a clear picture and scope of the entire project. System study studies the following questions:

- a) What is to be done in future?
- b) How to do it?
- c) When to do it?
- d) Who is to do it?

② **System analysis:** System analysis is the dissection of the system into its components to study how those components interact and work together. The most important activity in the system analysis is feasibility study which determines whether the system is feasible to design or not. It consists of several types of feasibility studies such as:

a) **Economical feasibility:** It is a type of feasibility study which determines the financial aspects of any system or software. It analyzes the cost for development of any system or software. It concerns with cost effectiveness of the system.

b) **Technical feasibility:** It is a type of feasibility study which determines the technical aspects of the any system or software. It analyzes whether the technical parts of any system or software are

feasible or not.

c) Operation feasibility:- It is a type of feasibility study which determines the operational aspects of any system or software. It analyzes the cost for development of any system or software.

d) Time / schedule feasibility:- It is a type of feasibility study which specially focuses the time scheduling for the development of any system or software. It analyzes what time will be the overall development of any software or system take to be completed. In this process project is splitted into small tasks and estimate time and required to complete each task.

e) Legal feasibility:- It is a type of feasibility study which deals or analyzes the legal procedures for the development of any software or system. It considers copyright law, labour law, foreign trade, tax, etc.

③ System design:- The next step is to design develop a logical design (framework) of the system. There are several tools used for system design such as algorithm, flowchart, pseudo code, ER design, DFD, etc. In this phase the overall logical solution of the problem or project is determined.

④ System development:- In this the logical structure is converted into program with the help of several programming languages such as C, C++, Java, Python, QBASIC, FORTAN, etc. In system development following tasks need to be done:

- i) Convert logical structures into program.
- ii) Create database
- iii) Train users
- iv) Prepare documentation

⑤ System testing:- It is an investigation done to check the quality of the product or services. In this the programs are executed with the intention of finding software bugs. Software testing is also called the process of validating and verifying the software product. Some of the testing methods are white-box testing (testing internal data structures including codes) and black-box testing (functional testing). Similarly, system testing consists of three components (phases). i.e.

- i) Unit testing
- ii) System testing
- iii) Integration testing

⑥ System implementation:- In this phase, the system is installed in the client's computer to support the desired business function. The implementation operation can be divided into four major categories.

- i) Direct implementation (conversion)
- ii) Parallel implementation (conversion)
- iii) Phased implementation (conversion)
- iv) Pilot implementation (conversion)

⑦ Maintenance and feedback:- This step involves making changes to hardware, software and documentation to support the system performance, enhance security and read customer's / user's feedback. It helps to redesign (modification/update) the system in the next version of this system in upcoming life cycle.

6.4 System analyst vs Software engineer

System analyst is a key person in system development process who analyzes the existing system and implement the new one that tries to overcome the limitations of existing system. A system analyst constantly interacts with the top level management, users, competitors and other external entities.

He/she is responsible for the overall progress of a project for quality control to end-user perspective.

Software engineer has an extensive knowledge of programming language, software development and computer operating system. Software engineer applies engineering principles to software creations. She must have in-depth computer programming knowledge.

Characteristics of system analyst

The characteristics of system analyst are listed below:

- Knowledge of organization.
- Technical knowledge
- Good communication skills
- Problem solving skills
- Good inter-personal relationship.
- Good character and ethics.

Characteristics of software engineer

Characteristics of software engineer are listed below:

- In-depth of programming language.
- Creative and logical mind
- Extensive knowledge of operating system.
- Must know about the nuts and bolts of the computer system.
- Able to create documentation.

Roles of system analyst

The various roles of system analyst are listed below:

- She must collect information about current system.
- She tries to overcome the limitations of existing system.
- She guides the overall system by giving proper directions.
- She must examine and analyse existing IT systems and business model and identify requirements.

Roles of software engineer

- She designs, develops and maintains computer software.
- She investigates new technologies.
- She ensures all systems are stable and operate efficiently.
- She responds to any software-related issues.
- She executes full life cycle software development.

6.5 Requirement collection methods

System analyst must collect information about the current system and how to improve its functionality with new system. During this phase data are collected from various sources such as

- i) Interview
- ii) Survey
- iii) Observation
- iv) Focus group
- v) Prototyping
- vi) Document analysis
- vii) Workshops and seminars (also webinars)

These sources are explained below:

i) Interview:- This source of data/requirements collection is all about taking interviews with the professionals of particular system and get more ideas and requirements of particular system.

ii) Survey:- We can perform several surveys about such system in order to know about the scope and

challenges of that system.

iii) Observation:- We can observe competitor system so that we can get ^{more} ideas about design and operation of such system.

iv) Focus group:- A focus group is a gathering of people who are representative of the users or customers of a product to get feedback. This gives us feedbacks about, needs, opportunities and problems to identify requirements; etc.

v) Prototyping:- It means developing a small or prototype model same as required and showing it to the client so that we can get additional requirements.

vi) Document analysis:- Reviewing the documentation of an existing system is known as document analysis. We should analyse documentations which can introduce us to the requirements that drove creation of the existing system.

vii) Workshops and Seminars:- Workshops and Seminars can be very effective for gathering requirements as the session involves several professionals and competitors. Idea sharing is the most done activity in the workshops and seminars.

6.6 System Design Tools

There are various system design tools available. Some of them are given below:

1) Algorithm : Algorithm is one of the most basic tools used for problem solving. It is defined as the finite sequence of instructions for solving a problem.

2) Flowchart : Flowchart is a diagrammatic or pictorial or symbolic representation of an algorithm written in order to solve a particular problem. It shows the order of instructions and relationships between them. Flowchart is of two types: System and program

3) DFD (Data Flow Design) : It is a logical diagram that describes flow of data (information) inside the components of the system. The main components inside DFD are :

- a. Process
- b. Data store
- c. Data flows
- d. External entities

There are several layers in DFD according to the system complexity.

4) Entity Relationship Diagram (ERD): This is a diagrammatic representation of several real world entities along with its attributes. The ER diagram is an overall logical structure of a database that can be shown in graphically or in pictorial form. Some of the basic components of ER-diagram are given below:

- a) Entity
- b) Attribute
- c) Relationship
- d) Lines

5) Pseudo code: The literal meaning of pseudo is fake or false. So, pseudo code means fake code which looks like program code but actually not. It is a kind of algorithm for solving a problem.

6) Decision table: It allows us to identify the different course of action according to the given conditions. It generally consists of two parts i.e.

- a) Action
- b) Condition

7) Decision tree: Decision tree is similar as decision table except that it follow the tree like structure.

6.7 Software and quality

The quality of software can be defined as the ability of the software to function according to the given specifications. The software can be considered as the good software if it contains following qualities:

- i) Good design:- It's always important to have a good and aesthetic to please users. Good design pushes users to go through the software and test it.
- ii) Good functionality:- It is important to have good and smooth functionality of the software so that users can enjoy it without any issues.
- iii) Durable and reliable:- In this context, durability means the ability of the software to work without any issues for a long period of time.
- iv) Portable:- The software should be portable, means software can be used in different operating System and computer system so that more users can use it.
- v) Efficient:- Here, efficiency means the accuracy of the software on the given instructions of the user. To meet the users need, the software needs to be efficient.

vi) Value for money:- The software should meet value for money so that customer and companies who make this app should feel the money spent on this app has not gone to waste.

vii) Good after sale service:- The software should meet the customer needs and after sale service should also be good to make the users satisfied.

viii) Flexibility and maintainability:- The software should be flexible and maintainable so that bugs associated should be able to capture and fix quickly and new tasks and enhancement must be added without trouble.

6.8 System Development Model

1. Waterfall model

It is the systematic and sequential model to develop software that begins with system requirements and progress through planning, modeling, construction. In this phase, once the development is completed in each stage the turn goes to next step/stage and there is no any turning back. It works as downhill structure. The diagram below shows waterfall model with its stages:

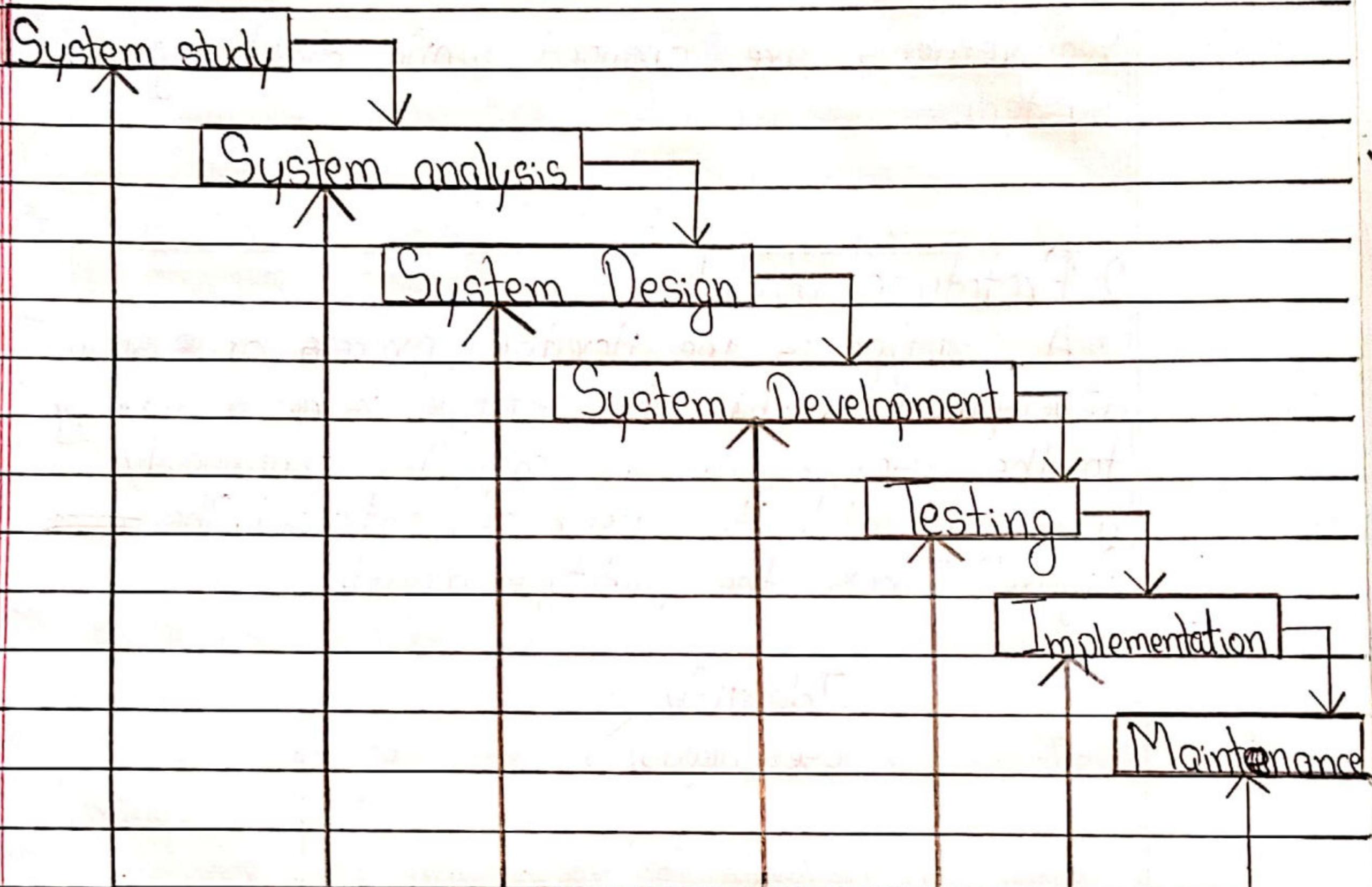


fig: Waterfall model

Advantages

- i) It is simple and easy to use.
- ii) It does not overlap any stages.
- iii) It is suitable for static projects.
- iv)

Disadvantages of waterfall model

- i) It is not suitable for dynamic projects.
- ii) It does not have back-track mechanism.
- iii) It may be costly and time consuming if the

requirements are changed while developing the system.

2. Prototype model

A prototype is the iterative process of system development. In this, the prototype is build according to the user requirements and then continuously modified until the user is satisfied. The ~~above~~ figure ^{below} shows the prototype model.

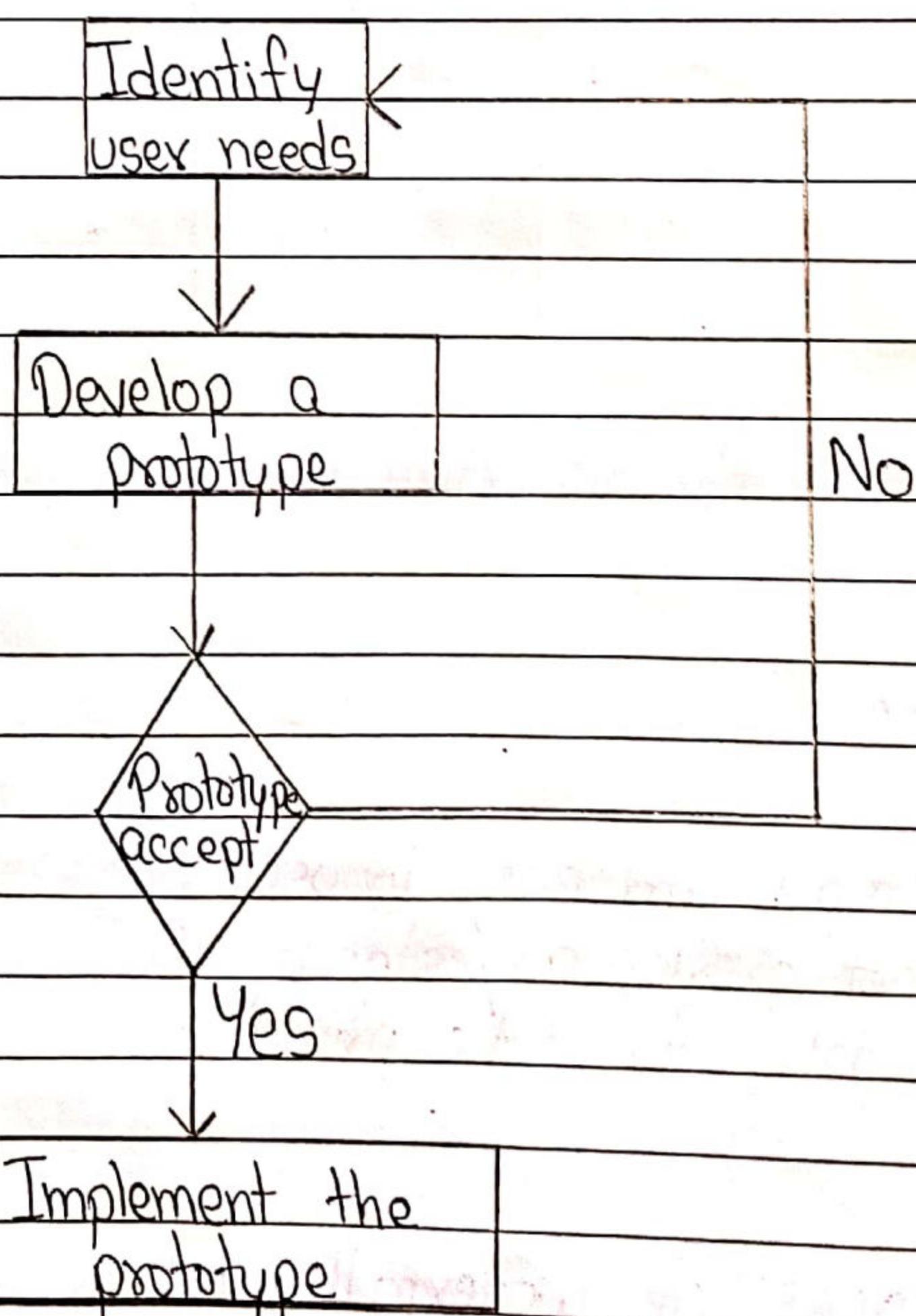


fig: Prototype mode

Advantages of prototype model

The advantages of prototype model are mentioned below:

- ① This model is suitable for dynamic system.
- ② Errors can be detected much earlier as the system is made side by side.
- ③ It gives user a better understanding of the project.

Disadvantages of prototype model

The disadvantages of prototype model are listed below:

- ① It is not suitable for complex systems in which user requirements are changing rapidly.
- ② It may increase time and cost depending on the further updates.
- ③ Sometimes user may never get satisfied and hence this model can be time consuming.

3. Spiral model

This is a software development process combining the features of waterfall and prototype mode. It is a recent model used for large, expensive and complicated project.

This model consists of four quadrants as shown in figure below.

Advantages of spiral model

The advantages of spiral model are as follow:

- ① It estimates budget and time in a more realistic manner.
- ② This model is suitable for complex and sophisticated projects.
- ③ This is the only model with risk management techniques.

Disadvantages of spiral model

The disadvantages of spiral model can be as follows:

- ① This needs highly technical manpower resources.
- ② This may be costlier than other models.
- ③ Sometimes the spiral model may go on and on.

4. Agile model

The agile model is mainly intended for helping developers build a project that can be easily adapted to the changing requests quickly. It helps to make easy and rapid projects. In this the people and users are given more importance than process and tools.

Advantages of Agile method

- ① It quickly responds to changes.
- ② High customer collaboration.
- ③ People and team interactions are given importance than process and tools.
- ④ Promotes teamwork and cross training.

Disadvantages of Agile model

- ① It is costlier than other methods.
- ② With continuous interaction, the project may delay.
- ③ It is not suitable for static project.
- ④ It has more risk of sustainability, maintainability, and extensibility.

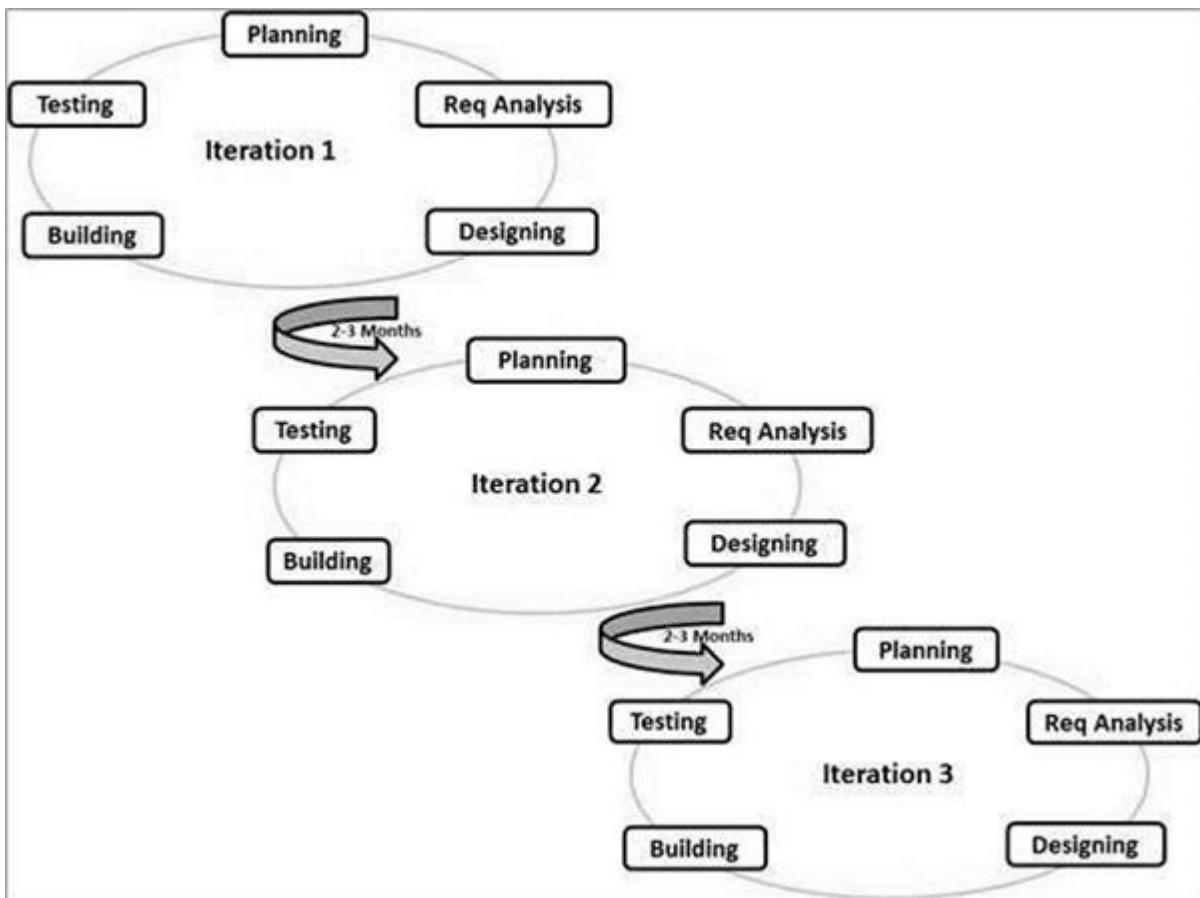


FIG: Agile model