

# **Terna Engineering College**

## **Computer Engineering Department**

**Program: Sem VI**

**Course: Software Engineering Lab**

**LAB Manual**

**PART A**

**(PART A: TO BE REFERRED BY STUDENTS)**

### **Experiment No.01**

#### **A.1 Aim:**

Prepare detailed problem statements (abstract) for the selected mini project and identify suitable process models for the same with justification.

#### **A.2 Prerequisite:**

1. Knowledge of different process models

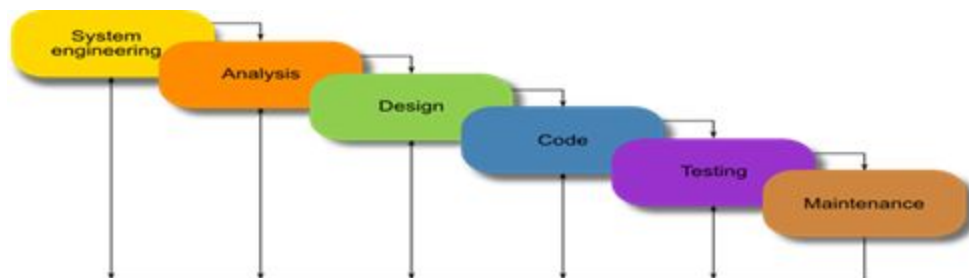
#### **A.3 Outcome:**

After successful completion of this experiment, students will be able to:

1. Refine requirements from a set of initial conditions.
2. Identify problem statements from a given case study.
3. To find a process model with justification for a problem.

#### **A.4 Theory:**

##### **Waterfall model**



The Waterfall Model was the first Process Model to be introduced. It is also referred to as a **linear-sequential life cycle model**. It is effortless to understand and use. In a waterfall model, each phase must be completed fully before the next step can begin. This type of software development model is used for a small project, and there are no uncertain requirements. At the end of each phase, a review takes place to determine if the project is on the right path and whether or not to continue or discard the project.

#### When to use the waterfall model

Requirements are very clear and fixed.

- There are no ambiguous requirements.
- Ample resources with the required expertise are available freely.
- The client has high confidence in the organization.
- The organization has experience of similar projects.
- The project is short.
- It is good to use this model when the technology is well understood

### **V Model**

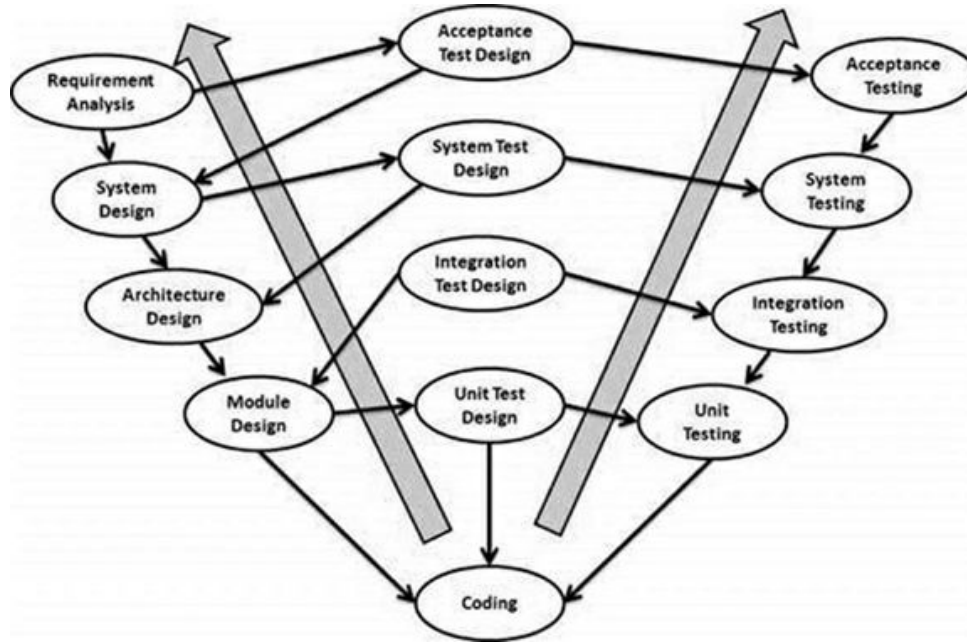
The V-model is an SDLC model where the execution of processes happens sequentially in a V-shape. It is also known as the Verification and Validation model.

The V-Model is an extension of the waterfall model and is based on the association of a testing phase for each corresponding development stage. This means that for every single phase in the development cycle, there is a directly associated testing phase. This is a highly-disciplined model and the next phase starts only after completion of the previous phase.

#### When to use the V-model:

- The V-shaped model should be used for small to medium-sized projects where requirements are clearly defined and fixed.
- The V-Shaped model should be chosen when ample technical resources are available with needed technical expertise.

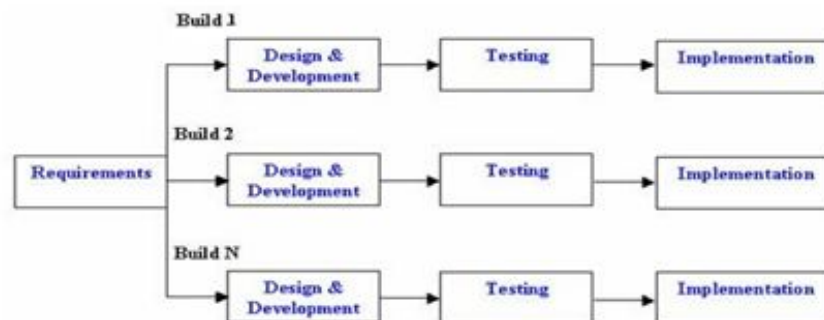
High confidence of customers is required for choosing the V-Shaped model approach. Since no prototypes are produced, there is a very high risk involved in meeting customer expectations.



## Incremental Process Model

The incremental build model is a method of software development where the product is designed, implemented, and tested incrementally (a little more is added each time) until the product is finished. It involves both development and maintenance. The product is defined as finished when it satisfies all of its requirements. This model combines the elements of the waterfall model with the iterative philosophy of prototyping.

The product is decomposed into many components, each of which is designed and built separately (termed as builds). Each component is delivered to the client when it is complete. This allows partial utilization of the product and avoids a long development time. It also avoids a large initial capital outlay and subsequent long waiting period. This model of development also helps ease the traumatic effect of introducing a completely new system all at once.



Incremental Life Cycle Model

### When to use the Incremental model:

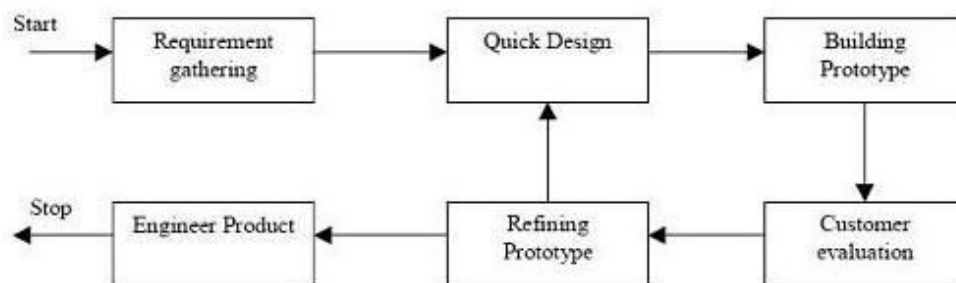
- This model can be used when the requirements of the complete system are clearly defined and understood.
- Major requirements must be defined; however, some details can evolve with time.
- There is a need to get a product to the market early.
- New technology is being used.
- Resources with the needed skill set are not available.
- There are some high-risk features and goals.

## Evolutionary Process Model

1. The prototyping model
2. The spiral model
3. Concurrent development model

### 1. The prototyping model

The basic idea in the Prototype model is that instead of freezing the requirements before a design or coding can proceed, a throwaway prototype is built on understanding the requirements. This prototype is developed based on the currently known requirements. The prototype model is a software development model. By using this prototype, the client can get an “actual feel” of the system since the interactions with the prototype can enable the client to better understand the requirements of the desired system. Prototyping is an attractive idea for complicated and large systems for which there is no manual process or existing system to help to determine the requirements. The prototype is usually not complete systems, and many of the details are not built in the prototype. The goal is to provide a system with overall functionality.



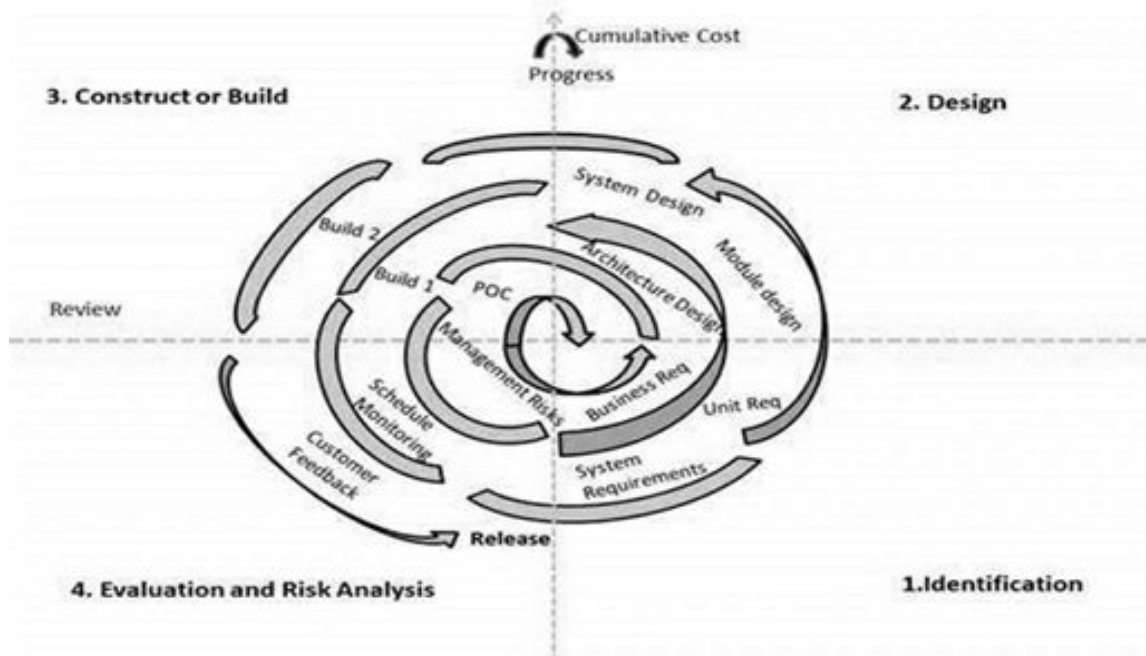
*Prototyping Model*

### When to use the Prototype model:

- The prototype model should be used when the desired system needs to have a lot of interaction with the end-users.
- Typically, online systems, web interfaces that have a very high amount of interaction with end-users, are best suited for the Prototype model. It might take a while for a system to be built that allows ease of use and needs minimal training for the end-user.
- Prototyping ensures that the end-users constantly work with the system and provide feedback that is incorporated in the prototype to result in a usable system. They are excellent for designing good human-computer interface systems.

## **2. The Spiral Model**

Spiral Model is a combination of a waterfall model and an iterative model. Each phase in the spiral model begins with a design goal and ends with the client reviewing the progress. The spiral model is similar to the incremental model, with more emphasis placed on risk analysis. The spiral model has four phases: Planning, Risk Analysis, Engineering, and Evaluation. A software project repeatedly passes through these phases in iterations (called Spirals in this model). In the baseline spiral, starting in the planning phase, requirements are gathered, and risk is assessed. Each subsequent spiral builds on the baseline spiral. It's one of the software development models like Waterfall, Agile, V-Model.



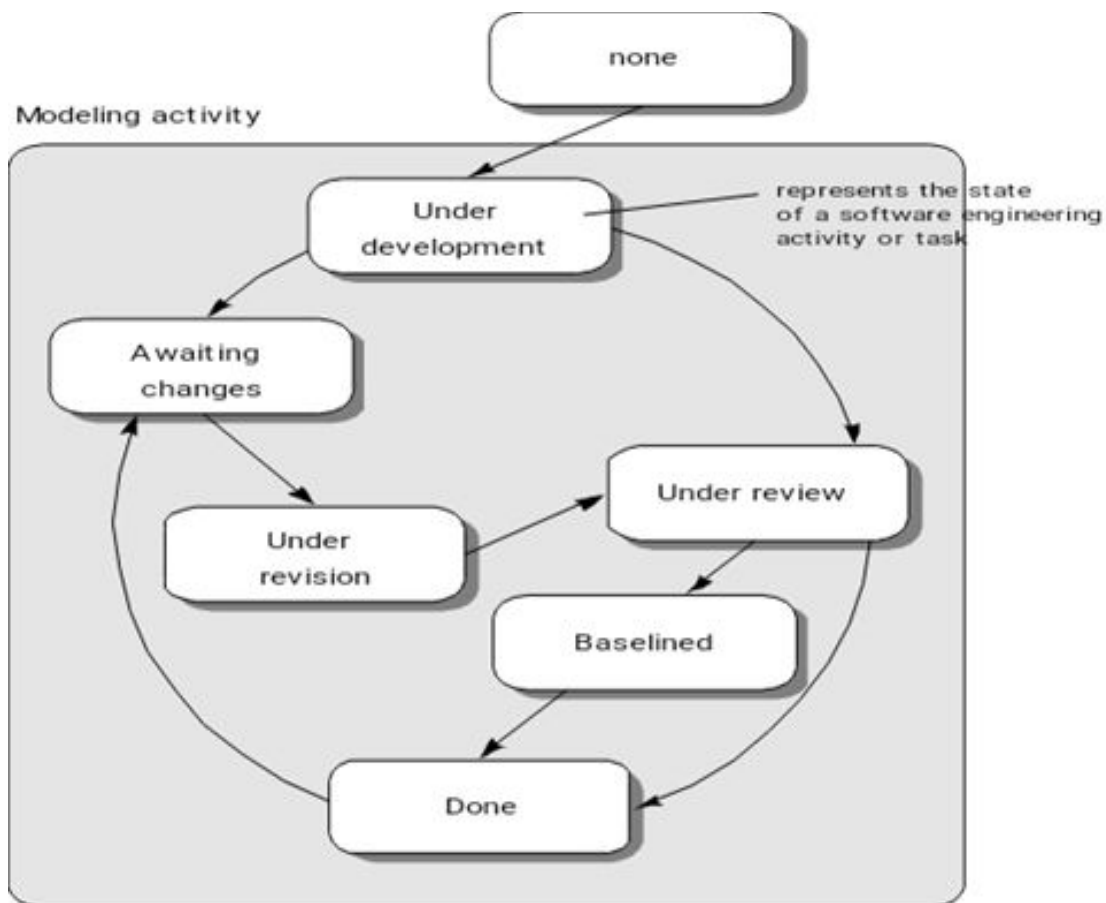
### When to use Spiral Methodology?

- When a project is large
- When releases are required to be frequent
- When the creation of a prototype is applicable
- When risk and costs evaluation is important
- For medium to high-risk projects
- When requirements are unclear and complex
- When changes may require at any time
- When long term project commitment is not feasible due to changes in economic priorities

### **3. Concurrent model**

The concurrent development model, sometimes called concurrent engineering, can be represented schematically as a series of framework activities, Software engineering actions of tasks, and their associated states.

The concurrent model is often more appropriate for system engineering projects where different engineering teams are involved.



The figure above provides a schematic representation of one Software engineering task within the modeling activity for the concurrent process model. The activity – modeling- may be in any one of the states noted at any given time.

All activities exist concurrently but reside in different states.

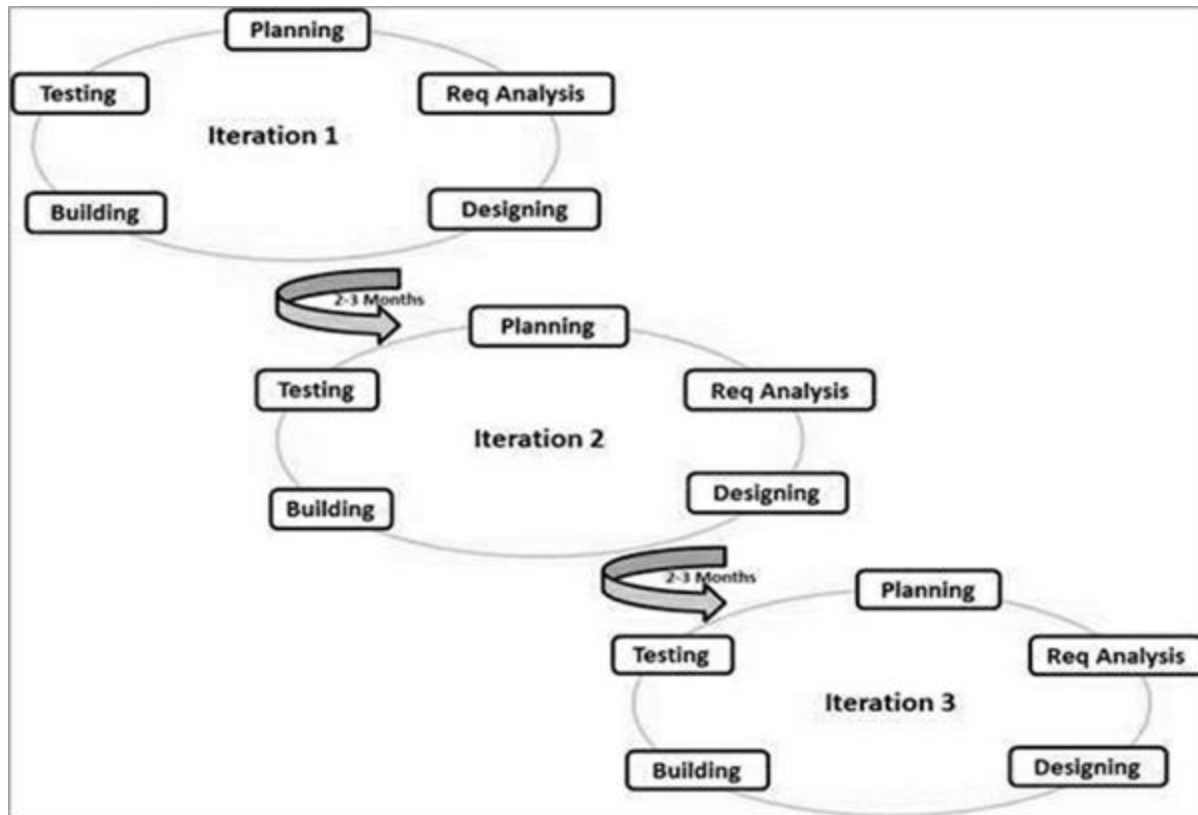
For example, early in the project, the communication activity has completed its first iteration and exists in the awaiting changes state. The modeling activity which existed in the non-state while initial communication was completed now makes a transition into an underdevelopment state.

If, however, the customer indicates the changes in requirements must be made, the modeling activity moves from the underdevelopment state into the awaiting changes state. The concurrent process model defines a series of events that will trigger transitions from state to state for each of the Software engineering activities, actions, or tasks.

## **Agile Model**

Agile SDLC model is a combination of iterative and incremental process models with a focus on process adaptability and customer satisfaction by rapid delivery of working software products. Agile Methods break the product into small incremental builds. These builds are provided in iterations. Each iteration typically lasts from about one to three weeks. The agile model believes that every project needs to be handled differently, and the existing methods need to be tailored to best suit the project requirements. In Agile, the tasks are divided into time boxes (small time frames) to deliver specific features for a release.

An iterative approach is taken, and a working software build is delivered after each iteration. Each build is incremental in terms of features; the final build holds all the features required by the customer.



#### When to use the Agile model:

- When new changes are needed to be implemented. The freedom agile gives to change is very important. New changes can be implemented at very little cost because of the frequency of new increments that are produced.
- To implement a new feature, the developers need to lose only the work of a few days, or even only hours, to roll back and implement it.
- Unlike the waterfall model in the agile model, very limited planning is required to get started with the project. Agile assumes that the end-users' needs are ever-changing in a dynamic business and IT world. Changes can be discussed, and features can be newly affected or removed based on feedback. This effectively gives the customer the finished system they want or need.
- Both system developers and stakeholders alike find they also get more freedom of time and options than if the software was developed in a more rigid sequential way. Having options gives them the ability to leave important decisions until more or better data or even entire hosting programs are available, meaning the project can continue to move forward without fear of reaching a sudden standstill.



## **CASE STUDY: School Management System**

### **Problem Statement**

In this century, the information revolution is the largest and most innovative that has marked the life of mankind. Today, the business is done online worldwide, the management of institutions is done through network technology, all the systems of information management have been digitized. All these innovations have the aim to simplify life by making a lot of things easy and in a short time.

In most schools today, all the processes are done manually, so the system has numerous drawbacks. The main drawback of the existing system is that many schools lack online payment, and also extracurricular activities are not introduced. The schools do not provide online report cards, tutorials, and practice tests. The system that we are proposing will be designed for better interaction between students, teachers, parents & management. The parents of students are very busy nowadays, so they cannot monitor their children and their activities properly and regularly. This school management system helps the parents monitor their children from anywhere. They can check their children's academic performance from a remote location. Any problems related to school or students can be communicated online with the teacher or principal. And also, any feedback, if required, can be provided.

This school management system will be providing online school notices, class timetable, student attendance, assignments, thought of the day, homework, and a list of suggested books and holidays. Also, the system will provide online leave applications, and fees can be paid online.

For designing the proposed system, the program installation needs about 200MB of disk space. The database and the management software will require nearly 10GB of hard disk space on the server. It should support all known operating systems, such as Windows, Linux. It requires a computer with specifications of 512MB RAM, monitors with a minimum resolution of 1024x768, a keyboard, and a mouse. Hard Drive should be in NTFS file-system formatted with a minimum of 10 GB of free space. A Laser printer will need to be used to print these reports and notes. The language used for coding is JAVA. The software should be designed to run on any platform above Microsoft Windows 7 (32bit). It should support Microsoft .NET Frameworks 4.0 or above and Microsoft SQL Server Management Studio Express 2008.

\*\*\*\*\*

## PART B

### (PART B: TO BE COMPLETED BY STUDENTS)

*(Students must submit the soft copy as per the following segments within two hours of the practical. The soft copy must be uploaded on the Blackboard or emailed to the concerned lab in charge faculties at the end of the practical in case there is no Blackboard access available)*

<b>Roll No.</b> 50	<b>Name:</b> AMEY THAKUR
<b>Class:</b> Comps TE B	<b>Batch:</b> B3
<b>Date of Experiment:</b> 27/01/2021	<b>Date of Submission:</b> 27/01/2021
<b>Grade:</b>	

### B.1 Observations and learning:

1. Prepare a detailed problem statement for the case study.

Ans:

#### PROBLEM STATEMENT:

The software to be designed is for a bookstore that wishes to go online. It is to be developed to improve the efficiency for the customer.

The important features to be developed include:

- The Login/Registration module requires the customer to login into the system or he can create an account if he does not yet have one.
- Order module requires a customer to enter the book details that he/she wants to buy.
- Book detail(s) module allows the system to keep book information in detailed by name, genre etc.
- Stock management will tell you about the number of books left in the store.
- Payment module allows the customer to online payment like Paytm and credit/debit cards or cash on delivery.
- Delivery and tracking module gives information about tracking and by whom it is delivered.
- User feedback module.

**ABSTRACT:**

The main objective of the project is to create an online book store that allows users to search and purchase a book online based on the title, author and subject. The selected books are displayed in a tabular format and the user can order their books online through credit card payment. Using this Website the user can purchase a book online instead of going out to a bookstore and wasting time.

There are many online bookstores like Powell's, Amazon which were designed using Html. I want to develop a similar website using .NET, SQL Server.

Online Book store is an online web application where the customer can purchase books online. Through a web browser the customers can search for a book by its title or author, later can add to the shopping cart and finally purchase using a credit card transaction. The user can log in using his account details or new customers can set up an account very quickly. They should give the details of their name, contact number and shipping address. The user can also give feedback to a book by giving ratings on a score of five. The books are divided into many categories based on the subject like Software, Database, English, Architecture etc.

The Online Book Store Website provides customers with online shopping through a web browser. A customer can create, sign in to his account, place items into a shopping cart and purchase using his credit card details.

The Administrator will have additional functionalities when compared to the common user. He can add, delete and update the book details, book categories, member information and also confirm a placed order.

This application is developed using C#, ASP.NET programming language. The Master page, data sets, data grids, user controls are used to develop the Online Book store.

**PURPOSE AND MOTIVATION:**

The main objective of the project is to create an online book store that allows users to search and purchase a book based on the title, author and subject. The selected books are displayed in a tabular format and the user can order their books online through credit card payment. The Administrator will have additional functionalities when compared to the common user.

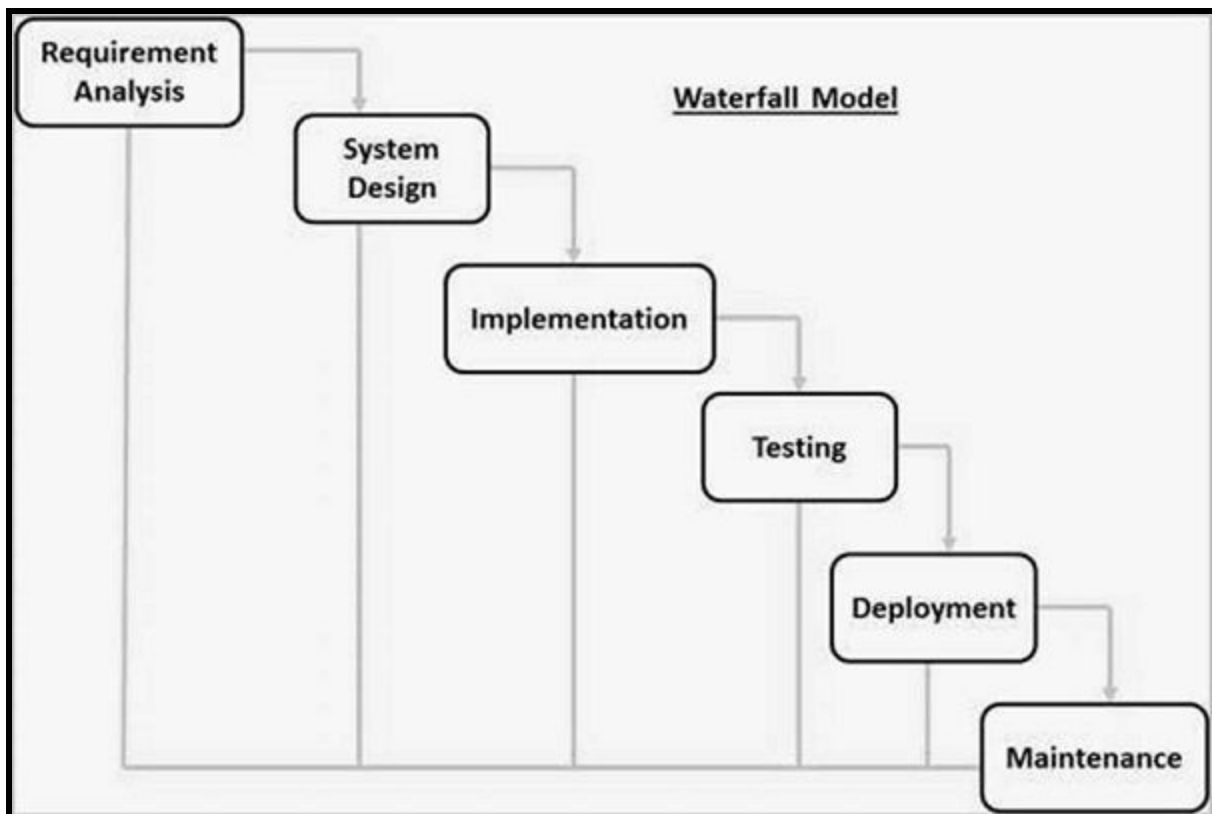
The motivation to create this project has many sources

- Interest to develop a good user-friendly website with many online transactions using a database.
- To increase my knowledge horizon in technologies like .NET, SQL, CSS, HTML.
- To gain good experience in .NET before joining in a full-time job.
- To gain expertise using Data Grid, Data Set, Data Table, Data Adapter and Data Readers.

2. Identify the process model for the case study.

Ans:

**PROCESS MODEL:**



We choose the WATERFALL MODEL due to the following reasons:

- This model is chosen because our requirements are very well known, clear and fixed.
- Product definition is stable.
- There are no ambiguous requirements in our project.

- The project is short.
- This model is simple and easy to understand and use.
- It is easy to manage due to the rigidity of the model – each phase has specific deliverables and a review process.
- In this model, phases are processed and completed one at a time. Phases do not overlap.
- Waterfall model works well for smaller projects where requirements are very well understood.

3. Provide Justification for the selected process.

Ans:

First of all the feasibility study is done. Once that part is over the requirement analysis and project planning begins. The design starts after the requirement analysis is complete and the coding begins after the design is complete. Once the programming is completed the testing is done. In this model, the sequence of activities performed in a software development project are: -

- Requirement Analysis
- Project Planning
- System design
- Detail design
- Coding
- Unit testing
- System integration & testing

Here the linear ordering of these activities is critical. End of the phase and the output of one phase is the input of another phase. The output of each phase is to be consistent with the overall requirement of the system. Some of the qualities of the spiral model are also incorporated like after the people concerned with the project review completion of each of the phases of the work done.

WATERFALL MODEL was being chosen because all requirements were known beforehand and the objective of our software development is the computerization/automation of an already existing manual working system.

## **B.2 Conclusion:**

**(Students must write the conclusion as per the attainment of individual outcome)**

It is concluded that there are many existing models for developing systems for different sizes of projects and requirements. Waterfall models and spiral models are used commonly in developing systems. Generally, most of the project models are dependent on the software development process by gathering the requirements for the design and development phases. Each model has advantages and disadvantages for the development of systems, so each model tries to eliminate the disadvantages of the previous model for future work. Suggesting a model simulates advantages that are found in different models to software process management.

\*\*\*\*\*