Practical 1:

Aim: Study of various Problem domains and Identify Problem definition:

Practical 1.1: Problem Definition

1. **Problem definition**

   To Run a census management system is not easy task for anyone. It includes number of tasks like managing all the details of person. It might not be easy for a person to handle it alone. It may required number of staff to handle each and every task and remember all the details so, it will take a lot of time and manpower.

   The manager may face some difficulties like lack of person, service. So, to overcome all these problems there is a software solution. Also with the help of software we can decrease the cost and manpower. It will be really easy to handle all the details.

   'Census management system.'

   Census management system is a software application that will reduce the time and manpower required for management and maintenance of a different task. It will also reduce the paper work in the existing system and hence it will be efficient.

   This system is very secure, user-friendly and reliable. The owner can use this system himself and can also give the access to his staff members. It is easy to store and retrieve the data from the system and so one can keeps all the records.

   Conclusively, the proposed system can help a user with a lot of things and can save the time and manpower.

2. **Project Title : Census management system.**

Practical 1.2: SDLC

Project: "Census Management System"

SDLC model selected for project: "Incremental Process Model"

Definition: Incremental Model is a process of software development where requirements are broken down into multiple standalone modules of software development cycle. Incremental development is done in steps from analysis design, implementation, testing/verification, maintenance.

Diagram:

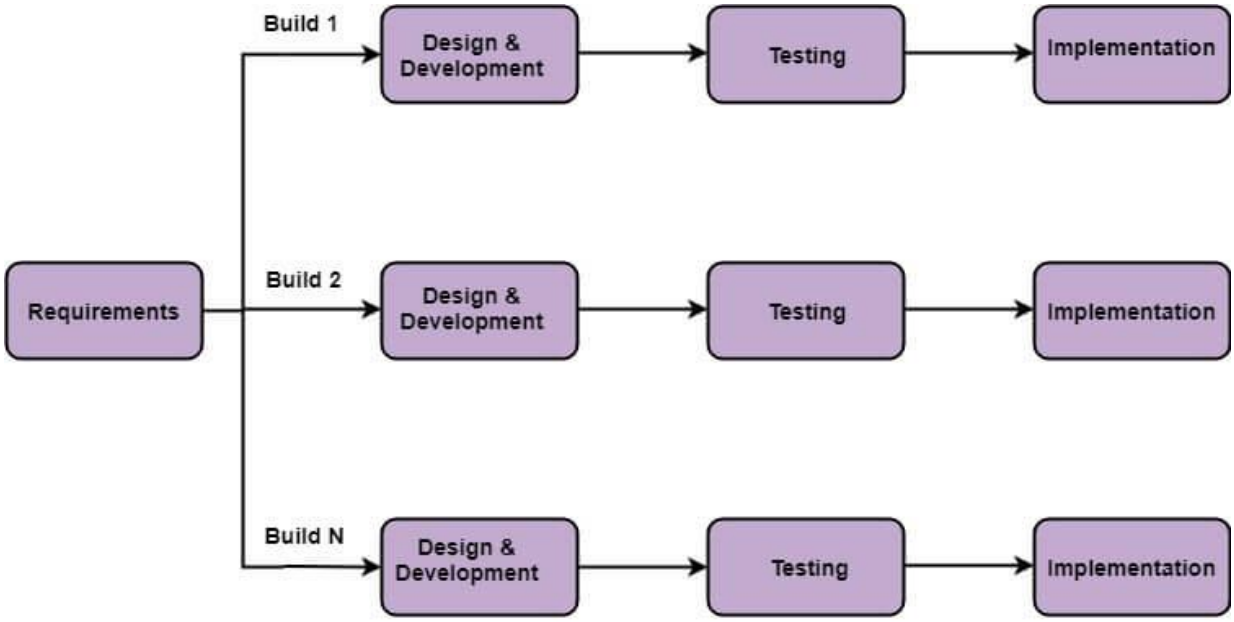

Fig: Incremental Model

Each iteration passes through the **requirements, design, coding and testing phases.** And each subsequent release of the system adds function to the previous release until all designed functionality has been implemented.

The system is put into production when the first increment is delivered. The first increment is often a core product where the basic requirements are addressed, and supplementary features are added in the next increments. Once the core product is analyzed by the client, there is plan development for the next increment.

The various phases of incremental model are as follows:

1. Requirement analysis: In the first phase of the incremental model, the product analysis expertise identifies the requirements. And the system functional requirements are understood by the requirement analysis team. To develop the software under the incremental model, this phase performs a crucial role.

2. Design & Development: In this phase of the Incremental model of SDLC, the design of the system functionality and the development method are finished with success. When software develops new practicality, the incremental model uses style and development phase.

3. Testing: In the incremental model, the testing phase checks the performance of each existing function as well as additional functionality. In the testing phase, the various methods are used to test the behavior of each task.

4. Implementation: Implementation phase enables the coding phase of the development system. It involves the final coding that design in the designing and development phase and tests the functionality in the testing phase. After completion of this phase, the number of the product working is enhanced and upgraded up to the final system product.

Advantage of Incremental Model:

- Errors are easy to be recognized.
- Easier to test and debug
- More flexible.
- Simple to manage risk because it handled during its iteration.
- The Client gets important functionality early.
- Throughout the development stages changes can be done.

Disadvantage of Incremental Model:

- Need for good planning
- Total Cost is high.
- Well defined module interfaces are needed.
- Problems might cause due to system architecture as such not all requirements collected up front for the entire software lifecycle.