2. PROJECT MANAGEMENT

2.1 PROJECT PLANNING

2.1.1 Project Development Approach and Justification

The application have been developed based on "Incremental Model" approach as there is provision to add some advanced functionalities. Some detail about incremental model is shown in the subsequent section.

Incremental Model

In incremental model the whole requirement is divided into various builds. Multiple development cycles take place here, making the life cycle a "multi-waterfall" cycle. Cycles are divided up into smaller, more easily managed modules. Each module passes through the requirements, design, implementation and testing phases. A working version of software is produced during the first module, so you have working software early on during the software life cycle. Each subsequent release of the module adds function to the previous release. The process continues till the complete system is achieved.

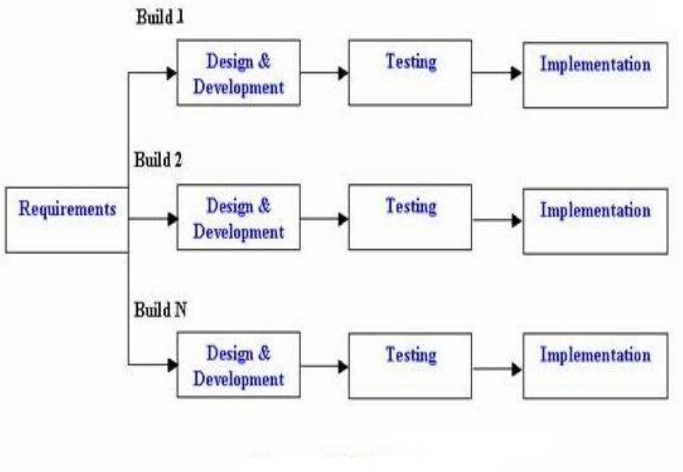


Fig: 2.1 Life Cycle of Incremental Model

Advantages of Incremental model:

- Generates working software quickly and early during the software life cycle.
- This model is more flexible – less costly to change scope and requirements.
- It is easier to test and debug during a smaller iteration.
- In this model customer can respond to each built.
- Lowers initial delivery cost.
- Easier to manage risk because risky pieces are identified and handled during it'd iteration.

Disadvantages of Incremental model:

- Needs good planning and design.
- Needs a clear and complete definition of the whole system before it can be broken down and built incrementally.
- Cost is higher than waterfall model.

We have used incremental model because we can enhance this application in future if project manager requires other extra features or if organization needs extra functionalities. In future we can add some features like implementing the other departments of organization like Energy and Engineering (Mechanical). Mainly we can test this application time by time so that bugs can be easily solved.

2.1.2 Project Effort and Time, Cost Estimation

Estimation technique introduction:

The most fundamental calculation in the COCOMO model is the use of the Effort Equation to estimate the number of Person-Months required to develop a project. Most of the other COCOMO results, including the estimates for Requirements and Maintenance, are derived from this quantity.

Techniques of COCOMO model:

- Line of code.
- Function point calculation.

I used Line of code method because the number of lines of program code is metric. It's so easy to measure and almost impossible to interpret. It can be used as a measure of complexity or productivity.

The COCOMO calculations are based on your estimates of a project's size in Source Lines of Code (LOC). SLOC is defined such that:

- Only Source lines that are DELIVERED, as part of the product is included -- test drivers and other support software is excluded.
- The project staff creates SOURCE lines -- code created by applications generators is excluded.
- One SLOC is one logical line of code.
- Declarations are counted as LOC.
- Comments are not counted as LOC.

KLOC can be used to measure thousands of LOC.

Cost Estimation:

Estimation about cost of the project by using COCOMO model is as follows:

Line of code = Number of files * average number of line

= 40 * 80

` = 3200 (KLOC)

Since project is **organic** so A1=2.4 and A2=1.05

Efforts = A1 * [(KLOC) ^ A2] Person-Months.

= 2.4 * [(KLOC) ^ 1.05] Person-Months.

= 2.4 * [(3.2) ^ 1.05]

= 8.13

= 8 Person-Months.

Estimation of development time

Development Time = 2.5 * [(Efforts) ^ 0.38] Months.

= 2.5 * [(8) ^ 0.38]

= 5.50

= 5.5 Months.

2.1.3 Roles And Responsibilities

The expert team at TecSo Global Projects Ltd. provided us their valuable guidance throughout the project. The experienced leaders helped us solved the most complex queries with their shrewd insights.

We were assigned roles on the basis of our individual talents and understandings that proved to be the most efficient way in developing the application

The team structure is as follows:

Project Leader : Ms. Komal Pathak

Project Team : Viral Joshi

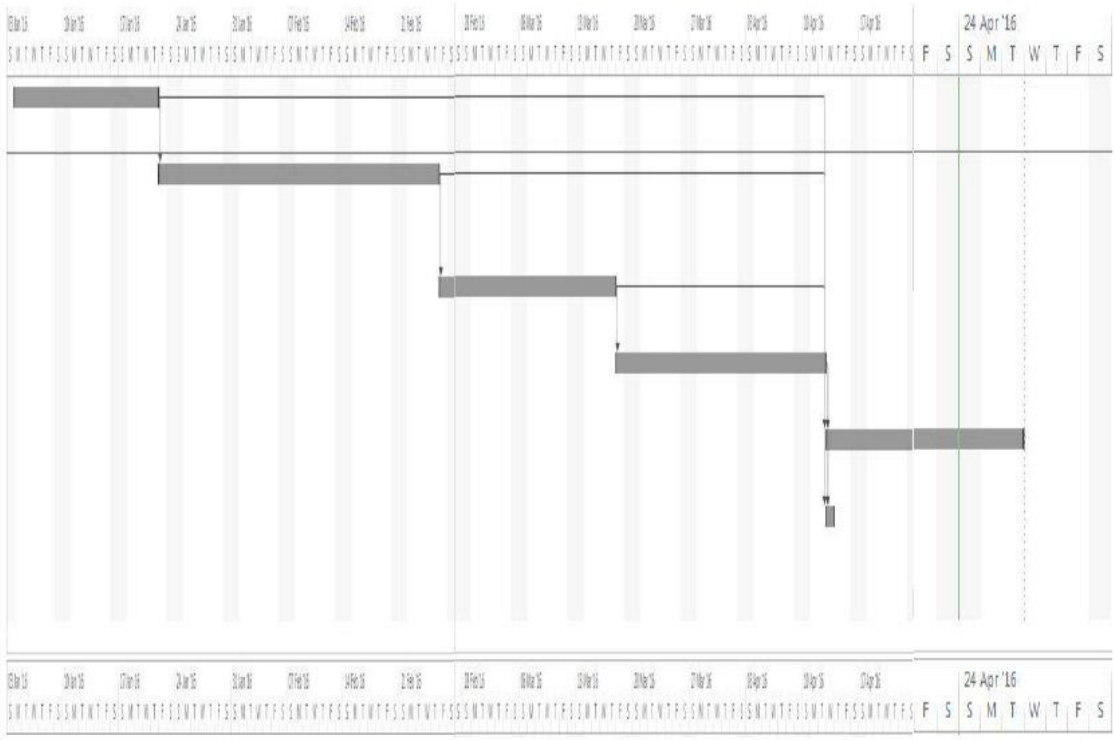
Particulars	**Name**
Analysis	Viral
Design: Database	
Design:UI (appearance)	
Implementation: Algorithm	
Implementation: Database	
Implementation: Coding	
Testing	
Documentation	

Table 2.1 Roles and Responsibilities

2.2 PROJECT WORK SCHEDULING (GANTT CHART)

GANTT CHART

	Task Mode	Task Name	Duration	Start	Finish	Predecessors
1		Domain Understanding	14 days	Thu 31-12-15	Tue 19-01-16	
2		SRS, Designing and Overview of diagrams	25 days	Fri 22-01-16	Thu 25-02-16	1
3		Screen Designing and Coding	15 days	Fri 26-02-16	Thu 17-03-16	2
4		Database Implementation	18 days	Fri 18-03-16	Tue 12-04-16	3
5		Testing and Solving Issues	10 days	Wed 13-04-16	Tue 26-04-16	4,3
6		Final Documentation	1 day	Wed 13-04-16	Wed 13-04-16	4,3,2,1


24 Apr '16
S M T W T F S
24 Apr '16
S M T W T F S