➢ Advantage (Pros) of Iterative Model:

1. Testing and debugging during smaller iteration is easy.
2. A Parallel development can plan.
3. It is easily acceptable to ever-changing needs of the project.
4. Risks are identified and resolved during iteration.
5. Limited time spent on documentation and extra time on designing.

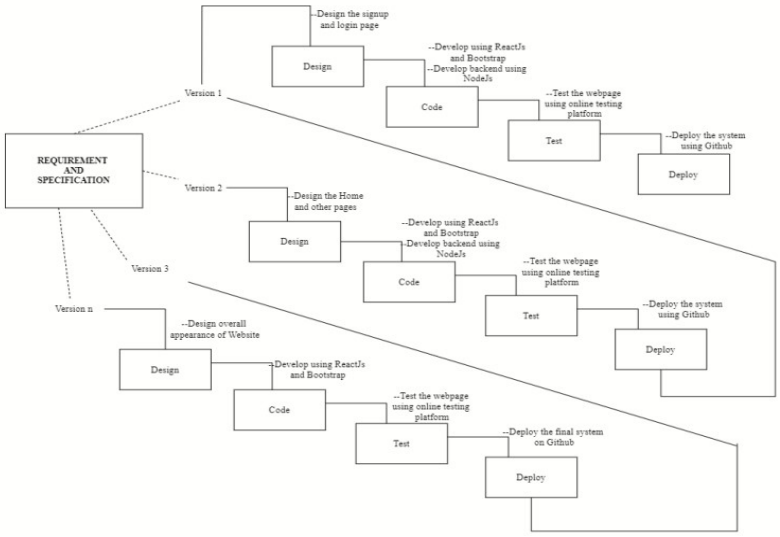
➢ Disadvantage (Cons) of Iterative Model:

1. It is not suitable for smaller projects.
2. More Resources may be required.
3. Design can be changed again and again because of imperfect requirements.
4. Requirement changes can cause over budget.
5. Project completion date not confirmed because of changing requirements.

➢ JUSTIFICATION: THAT WHY YOU CHOOSE THIS MODEL ONLY:

- Some working functionality can be developed and early in the software development life cycle (SDLC).
- It is easily adaptable to the ever-changing needs of the project as well as the client.
- It is best suited for agile organizations.
- It is more cost effective to change the scope or requirements in Iterative model.
- Parallel development can be planned.
- Testing and debugging during smaller iteration is easy.

➢ DIAGRAM OF MODEL ACCORDING TO YOUR DEFINATION:



Practical 4

AIM:- Develop Software project management planning (SPMP) for the specified module.

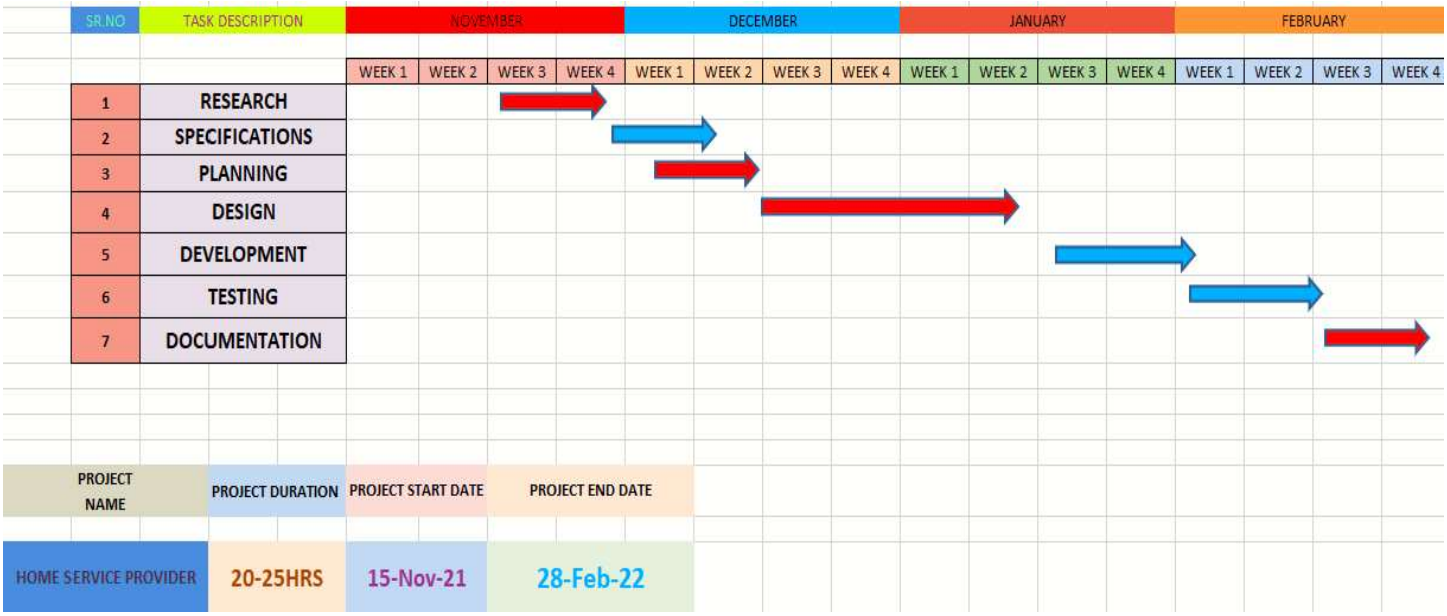
Theory:

- Once project designing is complete, project managers document their plans during a software package Project Management set up (SPMP) document. The SPMP document ought to discuss an inventory of various things that are mentioned below.
- This list will be used as a doable organization of the SPMP document. Organization of the software package Project Management set up (SPMP) document.
- **Introduction:-**
  - Objectives
  - Major Functions
  - Performance Issues
  - Management and Technical Constraints
- **Project Estimates:**
  - Historical Data Used
  - Estimation Techniques Used
  - Effort, Resource, Cost, and Project Duration Estimates
- **Schedule:**
  - Work Breakdown Structure
  - Task Network Representation
  - Gantt Chart Representation
  - PERT Chart Representation
- **Project Resources:**
  - People
  - Hardware and Software
  - Special Resources
- **Staff Organization:**
  - Team Structure
  - Management Reporting

- **Risk Management Plan:**
  - Risk Analysis
  - Risk Identification
  - Risk Estimation
  - Risk Abatement Procedures
- **Project Tracking and Control Plan**
- **Miscellaneous Plans:**
  - Process Tailoring
  - Quality Assurance Plan
  - Configuration Management Plan
  - Validation and Verification
  - System Testing Plan
  - Delivery, Installation, and Maintenance Plan

What Is a Gantt Chart?

A Gantt chart is a bar chart that provides a visual view of project tasks scheduled over time. A Gantt chart is used for project planning: it's a useful way of showing what work is scheduled to be done on specific days. It helps project managers and team members view the start dates, end dates and milestones of a project schedule in one simple stacked bar chart.



Practical 5

AIM:- Do Cost and Effort Estimation using different Software Cost Estimation models.

THEORY:

5.1 Cocomo model:

Cocomo (Constructive Cost Model) is a regression model based on LOC, i.e. number of Lines of Code. It is a procedural cost estimate model for software projects and often used as a process of reliably predicting the various parameters associated with making a project such as size, effort, cost, time and quality.

❖ **The Modes**

- **Organic :**
  2-50 KLOC, small, stable, little innovation
- **Semi-detached :**
  50-300 KLOC, medium-sized, average abilities, medium time-constraints
- **Embedded :**
  > 300 KLOC, large project team, complex, innovative, severe constraints

The constants cocomo model:

Mode	a	b
Organic	2.4	1.05
Semi-detached	3.0	1.12
Embedded	3.6	1.20

5.1.1 These are types of COCOMO model:

1. **Basic COCOMO Model**
2. **Intermediate COCOMO Model**
3. **Detailed COCOMO Model**

1.Basic cocomo model.

Equation of basic cocomo:

$$E = a(KLOC)^b$$

Where,

- E = the Effort in staff months.
- a & b = coefficients to be determined.
- KLOC = thousands of lines of code.

Effort:

KLOC = 3, a = 2.4, b = 1.05

$E=2.4(5)^{1.05}$

E=13.005 staff-month

Development time(Project Duration):

$$TDEV=c(E)^d$$

Where,

- TDEV is time for development
- c and d are constants to be determined
- E is the effort

Constants for TDEV,

Mode	c	d
Organic	2.5	0.38
Semi-detached	2.5	0.35
Embedded	2.5	0.32

$$TDEV=c(E)^d$$

$TDEV=2.5(13.005)^{0.38}$

TDEV=6.626 month

Average Staff Size:

$$\text{Staff size}=\frac{E}{TDEV}$$

$$\text{Staff size}=\frac{13.005}{6.626}\frac{staff-\cancel{mont}}{\cancel{month}}$$

Staff size=1.962 staff

Productivity:

$$\textbf{Productivity}= \frac{Size}{E}$$

Where,

- LOC = line of code
- E = effort

$$\text{Productivity}= \frac{5000}{13.005} \quad \frac{LOC}{Staff-month}$$

3) Detailed cocomo model

Detailed COCOMO incorporates all characteristics of the intermediate version with an assessment of the cost driver's impact on each step of the software engineering process. The detailed model uses different effort multipliers for each cost driver attribute. In detailed cocomo, the whole software is divided into different modules and then we apply COCOMO in different modules to estimate effort and then sum the effort.

The Six phases of detailed COCOMO are:

• Planning and requirements
• System design
• Detailed design
• Module code and test
• Integration and test
• Cost Constructive model

The effort is calculated as a function of program size and a set of cost drivers are given according to each phase of the software lifecycle.