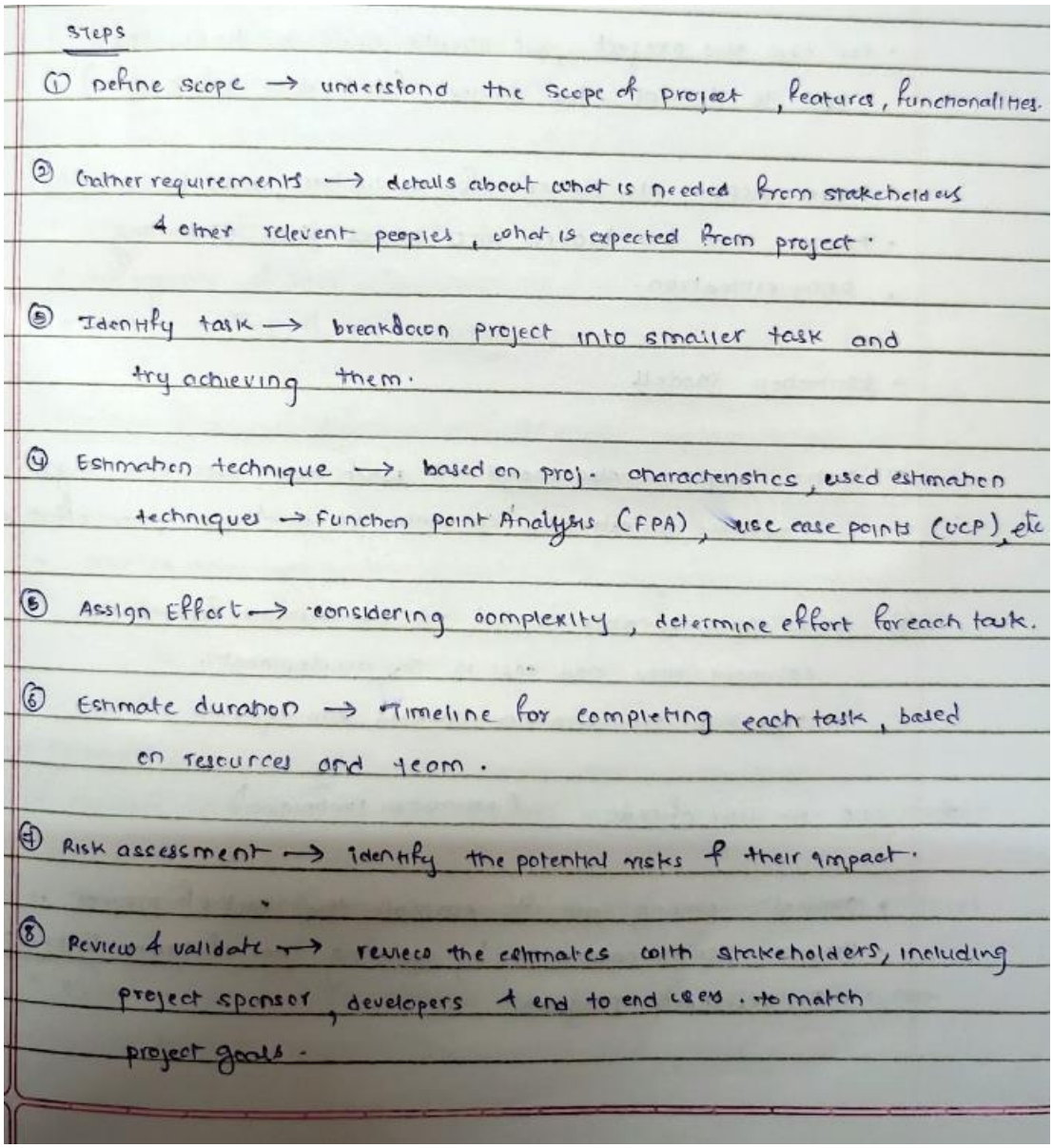
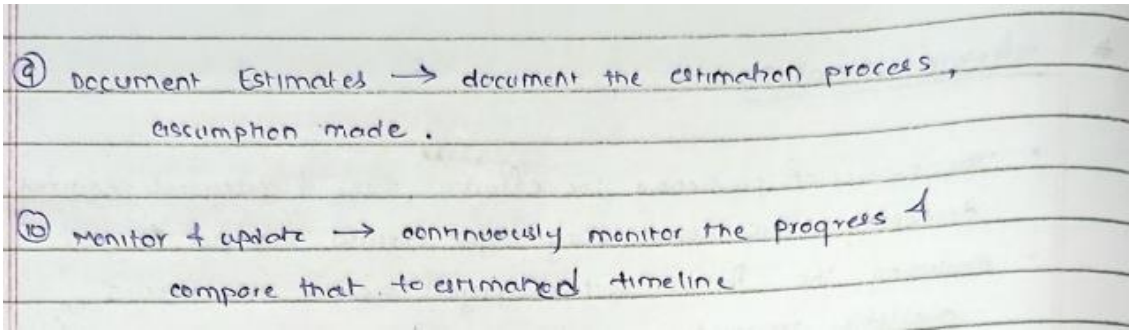
OOSE MOD-2

2.1)

➢ Software Estimation

- Software estimation involves predicting the effort, time, and resources required to develop a software project. It helps in budgeting, scheduling, and resource allocation.
- The process of predicting the effort ,time & resources required for developing a software project using object oriented principles.
- Analysing the factors such as project scope,requirements, available resources, capabilities to provide estimate for project planning.





➢ LOC

- LOC stands for "Lines of Code" in software engineering. It is a metric used to measure the size or complexity of a software project by counting the number of lines in the source code.
- ***Measurement:*** LOC measures the physical size of the codebase by counting the number of lines in the source code files. It includes both code lines (executable statements) and non-code lines (comments, whitespace, and blank lines).
- ***Estimation:*** LOC can be used for estimating the effort required to develop or maintain a software project. It's often used as a basis for estimating cost, schedule, and resource requirements.
- 1. Optimistic 2. Most likely 3. Pessimistic

  For example, following formula

  $$S = [S_{opt} + 4 * S_m + S_{pess}]/6$$

Advantages of LOC:

- Simple & straight forward metric.
- Basic project size can be known.

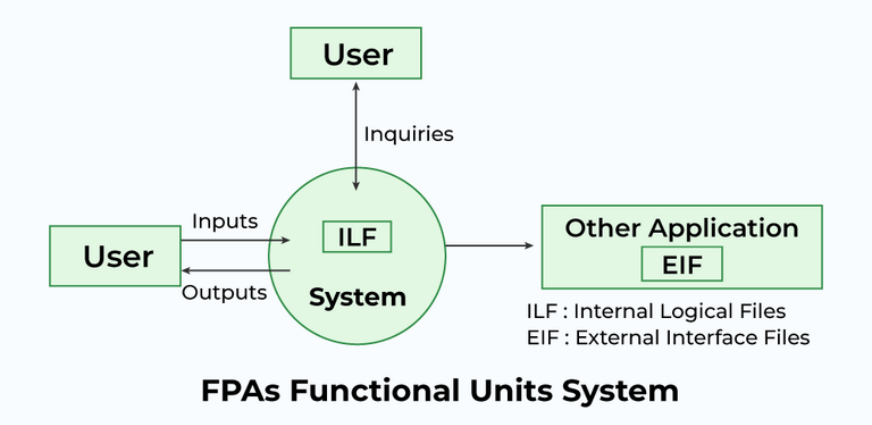
Disadvantages of LOC:

- Varies significantly based on language used.
- Complexity of project is ignored.
- Only quantity & no quality.

➢ FP(Function Point)

- In software engineering, function points (FP) are a unit of measurement used to estimate the size and complexity of software projects based on the functionality delivered to users.
- **Key Components:**

- ***External Inputs (EI):*** Data entering the system from users or external systems.
- ***External Outputs (EO):*** Data sent out of the system to users or external systems.
- ***External Inquiries (EQ):*** Questions posed to the system that require a response.
- ***Internal Logical Files (ILF):*** Data structures stored within the system.
- ***External Interface Files (EIF):*** Files exchanged with other systems.



FPAs Functional Units System

Benefits of using Function Points:

- **Objectivity:** Provides a standard measure independent of programming language or coding style.
- **Early Estimation:** Useful in early stages of project planning when detailed design is unavailable.
- **Effort Estimation:** Can be used to estimate development effort, cost, and schedule.
- **Project Comparison:** Enables comparison of different projects based on functionality.

Limitations of Function Points:

- **Complexity in Calculation:** Requires training and effort to apply accurately.
- **Subjectivity in Weighting:** Assigning weights to components can be subjective.
- **Limited Scope:** Primarily focuses on functional requirements, not non-functional aspects.

➢ Basic COCOMO Model

- The COCOMO (Constructive Cost Model) is a widely used estimation model in software engineering for estimating the cost, effort, and schedule of a software project. It was first proposed by Barry Boehm in 1981 and has since undergone several revisions. COCOMO is based on the concept that the effort required to develop software is proportional to the size and complexity of the project.

Classes of Software Projects

1) **Organic mode:** In this mode, relatively small, simple software projects with a small team are handled. Such a team should have good application experience to less rigid requirements.

2) **Semi-detached projects:** In this class an intermediate projects in which teams with mixed experience level are handled. Such projects may have mix of rigid and less than rigid requirements.

3) **Embedded projects:** In this class, projects with tight hardware, software and operational constraints are handled.

Merits:

- **Simplicity:** Basic COCOMO is straightforward and easy to understand. It requires minimal input parameters, primarily focusing on the size of the software project in lines of code (LOC).
- **Quick Estimates:** Since Basic COCOMO relies primarily on project size, it allows for rapid estimation of effort and cost early in the project lifecycle, even before detailed project requirements are known.
- **Historical Data Comparison:** Organizations can compare Basic COCOMO estimates with historical data from past projects to gauge the reasonableness of the estimates and identify potential risks early on.

Limitations:

- **Sensitivity to Size:** Basic COCOMO relies heavily on the size of the software project as measured by lines of code (LOC). However, the size estimation itself can be challenging and may vary significantly based on the programming language, coding style, and other factors.

- **Limited Factors:** It does not consider other factors that can influence project effort and cost, such as project complexity, team experience, development environment, and software reuse.
- **Lack of Flexibility:** Basic COCOMO provides a single-point estimate and does not offer flexibility for scenario analysis or sensitivity analysis.

➢ SPMP

- SPMP is a document that outlines how a software project has to be managed from start to end.
  → This discusses the approaches, strategies for managing software development project.
  → SPMP is a crucial component for projects documentation and serves the roadmap for project team.

2.2)

- ➢ Scheduling
- ➢ Work Breakdown Structure
- ➢ Gantt Chart
- ➢ Tracking the schedule

2.3)

- ➢ Risk Identification
- ➢ Risk Assessment
- ➢ Risk Projection
- ➢ RMMM Plan

2.4)

- ➢ Software Configuration Items
- ➢ SCM Process
- ➢ Identification

- **Version Control**
- **Change Control**
- **Configuration Audit**
- **Status Reporting**