Practical-3

Cost and Effort Estimation for Software Development

Aim: Perform a cost and effort estimation for the selected system by understanding the scope of the software to be developed.

Cost estimation and effort analysis are essential for planning and managing software projects, helping predict the required resources, time, and budget. **Function Point Analysis (FPA)** is a widely used method to estimate software size, effort, and cost. It measures the functionality delivered to the user, independent of programming languages or technologies. Function points are calculated based on five key components: **External Inputs (EI)**, **External Outputs (EO)**, **External Inquiries (EQ)**, **Internal Logical Files (ILF)**, and **External Interface Files (EIF)**. Each component is assigned a weight based on its complexity (Low, Medium, High), and these are summed up to determine the system's size. This quantitative measure is then used with productivity rates or cost factors to estimate the total effort (in person-hours) and cost of the project, ensuring accurate and standardized planning.

3.Cost and Effort Estimation

To estimate the **cost** and effort for developing a Library Management **System** using Function Point Analysis (FPA)**,** the following steps can be utilized.

3.1 Understand the Scope of the System

The College Management System (CMS) is designed to automate and streamline various administrative, academic, and financial activities within a college. Its scope includes:

- **Student Management**: Handles admissions, profiles, and course enrollments
- **Faculty Management**: Stores faculty details, schedules, and salaries.
- **Attendance Tracking**: Monitors student and faculty attendance.
- **Examination Management:** Manages scheduling, grading, and result processing.
- **Fee Management**: Tracks payments, due fees, and financial reports.

3.2 Identify and Classify Function Types

FPA involves categorizing system components into five elements:

1. **External Inputs (EI)**: Inputs provided to the system.
2. **External Outputs (EO)**: Outputs generated by the system.
3. **External Inquiries (EQ)**: User-driven queries.
4. **Internal Logical Files (ILF)**: Logical files/data tables maintained within the system.
5. **External Interface Files (EIF)**: Files/data from external systems

Table 3.1: Functionality breakdown

Functionality	Type	Complexity	Count
User Registration	EI	Low	1
Faculty Registration	EI	Low	1
Login/Logout	EI	Low	1
Add/Update/Delete Student Record	EI	Medium	3
Add/Update/Delete Faculty Record	EI	Medium	3
Course Management	EI	Medium	3
Search for Student	EQ	Medium	2
Search for Faculty	EQ	Medium	2
Assign courses for faculty	EI	Medium	2
Issue ID card	EI	Medium	1
Generate Student Report	EO	Medium	1
Generate Faculty Report	EO	Medium	1
Student Table	ILF	Low	1
Faculty Table	ILF	Low	1
Course Table	ILF	Medium	1
Attendance Record	ILF	Medium	1

Table 3.2: Assigning weight for functions

Type	Low	Medium	High
EI	3	4	6
EO	4	5	7
EQ	3	4	6
ILF	7	10	15
EIF	5	7	10

3.3 Calculating Unadjusted Function Points (UFP)

Using the counts and weights above listed in table 3.1 and 3.2.

UFP=∑ (Function Count × Weight

Table 3.3: Function point calculation

Type	Complexity	Count	Weight	Total FP
EI	Low	1	3	3
EI	Low	1	3	3
EI	Low	1	3	3
EI	Medium	3	4	12
EI	Medium	3	4	12
EI	Medium	3	4	12
EQ	Medium	2	4	8
EQ	Medium	2	4	8
EI	Medium	2	4	8
EI	Medium	1	4	4
EO	Medium	1	5	5
EO	Medium	1	5	5
ILF	Low	1	7	7
ILF	Low	1	7	7
ILF	Medium	1	10	10
ILF	Medium	1	10	10
Total UFP				=117

3.4 Adjust for Complexity Factors

The **Value Adjustment Factor (VAF)** is a multiplier applied to the **Unadjusted Function Points (UFP)** to adjust for the specific characteristics of the software system. It is based on 14 General System Characteristics (GSCs), each rated on a scale of 0 to 5, where:

- **0** = No influence on the system.
- **5** = Strong influence on the system.

The formula for calculating the **VAF** is:

VAF=0.65+(TDI×0.01)

Adjusted Function Points (AFP):

AFP=UFP×VAF

Where **TDI** (Total Degree of Influence) is the sum of the ratings for all 14 GSCs.

Table 3.4: General System Characteristics for LMS and ratings

Characteristic	**Description**	**Rating (0–5)**
1. Data Communications	The degree to which the system interacts with other systems or devices.	**4** (e.g., student records, examination databases, online payment gateways)
2. Distributed Data Processing	The extent of data processing done across multiple locations or systems.	**3** (like online student access, faculty access)
3. Performance Requirements	The need for high-performance processing (e.g., quick searches or database operations).	**4** (e.g., Quick retrieval of student details, grading system performance)
4. Heavily Used Configuration	How extensively the system uses the hardware and software configuration (e.g., server loads).	**3** (e.g., faculty, students, administration)
5. Transaction Rate	The frequency of transactions like book issues, returns, and catalog searches.	**4** (e.g., Frequent transactions for student enrollment, attendance records, and exam grading updates)
6. On-Line Data Entry	The amount of real-time data entry required by the system.	**5** (e.g., Real-time data entry for student registration, course enrollment, attendance tracking, and faculty management.)
7. End-User Efficiency	The importance of ease of use and efficient interaction for end-users.	**4** (e.g., dashboard for academic performance tracking)
8. On-Line Update	The need for real-time updates to the data (e.g., updating book records).	**5** (e.g., Continuous real-time updates for attendance, fees, grades, and timetable changes are essential.)
9. Complex Processing	The complexity of processing logic in the system (e.g., validations, calculations).	**3** (e.g., exam results, student eligibility verification, and fee calculations.
10. Reusability	The extent to which the system’s components are reusable for other purposes or projects.	**3** (e.g., authentication and document management)
11. Installation Ease	The ease with which the system can be installed and configured.	**4** (e.g., be installed and configured easily, with predefined roles and settings.)
12. Operational Ease	The level of automation and ease of operation for system administrators.	**4** (e.g., Automated backups, attendance tracking, and user access controls simplify system administration.)

13. Multiple Sites	The degree to which the system supports multiple sites or branches.	**3** (e.gColleges with multiple campuses might need centralized data access,)
14. Facilitate Change	The ease of modifying the system to adapt to new requirements or technologies.	**3** (e.g : new grading systems)

- **TDI (Total Degree of Influence)**:

  TDI = 4+3+4+3+4+5+4+5+3+3+4+4+3+3 = 52

- **VAF**:

  VAF =0.65+ (TDI×0.01)

  VAF =0.65+ (52×0.01) =0.65+0.52= 1.17

- **AFP**

  AFP= UFP*VAF

  =117*1.17

  =136.87

3.5 Effort Estimation

Effort is estimated based on **AFP** and the **productivity rate**. The productivity rate in the context of Function Point Analysis (FPA**)** refers to the number of Function Points (FPs) that a development team can complete in one Person-Month (i.e., the amount of work one person can complete in a month).

It is an empirical measure based on:

- Team skill level
- Technology used
- Development environment
- Past project data

Typical Productivity Rates

- Simple systems**:** 25–30 FP/Person-Month
- Moderately complex systems**:** 15–20 FP/Person-Month
- Highly complex systems**:** 7–10 FP/Person-Month

Effort (Person-Months)=AFP /Productivity Rate

For the Library Management System, a **Productivity Rate of 20 FP/Person-Month** was used.

This assumes:

- A moderately experienced team.
- Standard tools and technologies.
- Moderate system complexity.

Effort=AFP/Productivity Rate

Approximately, 136/ 20 = 7 Person-Month

3.6 Cost Estimation

To estimate cost, multiply the effort by the **developer's monthly cost**. **Assuming** Developer Monthly Cost = $5,000

Cost=7 ×5000 =$35000

(Note : cost and effort values are based on assumptions)