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 **Bus Management System** using **Function Point Analysis (FPA),** the following steps can be utilize.

3.1 Understand the Scope of the System

A Bus Management System typically includes the following modules:

1. **User Management:** Manage passengers, drivers, and administrative staff.
2. **Bus Management:** Add, update, or remove buses from the fleet.
3. **Booking Management:** Book, modify, and cancel bus ticket reservations.
4. **Payment & Billing:** Process payments and generate invoices.
5. **Reports:** Generate reports on bookings, bus usage, revenue, and maintenance.
6. **Authentication:** Secure login/logout functionality for users.
7. **Maintenance Scheduling:** Schedule and track bus maintenance.

3.2 Identify and Classify Function Types

FPA for a Bus Management System (BMS) involves categorizing system components into five elements:

1. **External Inputs (EI):** Inputs provided to the system, such as passenger registration, ticket booking details, bus schedule entries, and maintenance logs.
2. **External Outputs (EO):** Outputs generated by the system, including ticket receipts, booking confirmations, schedules, invoices, and various reports.
3. **External Inquiries (EQ):** User-driven queries like searching bus routes, checking seat availability, viewing booking history, or tracking buses.
4. **Internal Logical Files (ILF):** Logical files or data tables maintained within the system such as passenger records, bus fleet details, booking records, route info, and maintenance history.
5. **External Interface Files (EIF):** Files or data imported from external systems, such as payment gateway records, GPS tracking data, or third-party scheduling services.

Table 3.1: Functionality breakdown

Functionality	**Type**	**Complexity**	**Count**
Customer Registration	EI	Low	1
Login/Logout	EI	Low	1
Vehicle Add/Update/Delete	EI	Medium	2
Reservation Booking	EI	Medium	2
Payment Processing	EI	Medium	1
View Rental History Report	EO	Medium	1
View Revenue Report	EO	Medium	1
Customer Table	ILF	Low	1
Vehicle Table	ILF	Medium	1
Reservation Table	ILF	Medium	1
Payment Table	ILF	Low	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	2	3	6
EI	Medium	2	4	8
EQ	Medium	2	4	8
EO	Medium	2	5	10
ILF	Low	1	7	7
ILF	Medium	2	10	20
Total UFP				**59**

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.	**3** (e.g., connecting to external databases like student records)
2. Distributed Data Processing	The extent of data processing done across multiple locations or systems.	**4**(e.g., library branches syncing data)
3. Performance Requirements	The need for high-performance processing (e.g., quick searches or database operations).	**5** (e.g., fast response for book queries and checkouts)
4. Heavily Used Configuration	How extensively the system uses the hardware and software configuration (e.g., server loads).	**5** (e.g., multi-user access during peak hours)
5. Transaction Rate	The frequency of transactions like book issues, returns, and catalog searches.	**4** (e.g., frequent transactions in busy libraries)
6. On-Line Data Entry	The amount of real-time data entry required by the system.	**5** (e.g., real-time user regi., book issue, and returns)
7. End-User Efficiency	The importance of ease of use and efficient interaction for end-users.	**3** (e.g., intuitive interfaces for staff and users)
8. On-Line Update	The need for real-time updates to the data (e.g., updating book records).	**3** (e.g., real-time updates during issue/return transactions)
9. Complex Processing	The complexity of processing logic in the system (e.g., validations, calculations).	**3** (e.g., overdue fines, handling reserved books)

10. Reusability	The extent to which the system's components are reusable for other purposes or projects.	**2** (e.g., some reusable modules like user authentication)
11. Installation Ease	The ease with which the system can be installed and configured.	**4** (e.g., straightforward deployment process)
12. Operational Ease	The level of automation and ease of operation for system administrators.	**4** (e.g., automated backups, simple user management tools)
13. Multiple Sites	The degree to which the system supports multiple sites or branches.	**3** (e.g., library branches accessing a centralized system)
14. Facilitate Change	The ease of modifying the system to adapt to new requirements or technologies.	**3** (e.g., relatively adaptable system for future changes)

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

TDI = 3 + 4 + 5 + 5 + 4 + 5 + 3 + 3 + 3 + 2 + 4 + 4 + 3 + 3 = 51

- **VAF**:

VAF =0.65+ (TDI×0.01)

VAF =0.65+ (51×0.01) =0.65+0.51= 1.16

- **AFP**

AFP= UFP*VAF

=59*1.18

= 69.62

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 **Bus 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, 84/ 20 = 4 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=4 * 5000 = $20000

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