Experiment No. 07 – LOC and FP Analysis

**Shaikh Zaid Muddaris Husain 23dco06 batch 03**

**Aim:** To estimate the cost of project using software metrics (LOC and FP analysis).

**Theory:** A software metric is a measure of software characteristics which are measurable or countable. Software metrics are valuable for many reasons, including measuring software performance, planning work items, measuring productivity, and many other uses. Within the software development process, many metrics are that are all connected. Software metrics are similar to the four functions of management: Planning, Organization, Control, or Improvement.

LOC Metrics: It is one of the earliest and simpler metrics for calculating the size of the computer program. It is generally used in calculating and comparing the productivity of programmers. These metrics are derived by normalizing the quality and productivity measures by considering the size of the product as a metric.

Following are the points regarding LOC measures:

* In size-oriented metrics, LOC is considered to be the normalization value.
* It is an older method that was developed when FORTRAN and COBOL programming were very popular.
* Productivity is defined as KLOC / EFFORT, where effort is measured in person-months. Size-oriented metrics depend on the programming language used.
* As productivity depends on KLOC, so assembly language code will have more productivity.
* LOC measure requires a level of detail which may not be practically achievable.
* The more expressive is the programming language, the lower is the productivity.
* LOC method of measurement does not apply to projects that deal with visual (GUI-based) programming.
* As already explained, Graphical User Interfaces (GUIs) use forms basically.
* LOC metric is not applicable here.
* It requires that all organizations must use the same method for counting LOC.
* This is so because some organizations use only executable statements, some useful comments, and some do not.
* Thus, the standard needs to be established.
* These metrics are not universally accepted.

Advantages of LOC

* Simple to measure

Disadvantage of LOC

* It is defined on the code. For example, it cannot measure the size of the specification.
* It characterizes only one specific view of size, namely length, it takes no account of functionality or complexity
* Bad software design may cause an excessive line of code
* It is language dependent
* Users cannot easily understand it

Functional Point (FP) Analysis: Allan J. Albrecht initially developed function Point Analysis in 1979 at IBM and it has been further modified by the International Function Point Users Group (IFPUG). FPA is used to make estimate of the software project, including its testing in terms of functionality or function size of the software product. The functional size of the product is measured in terms of the function point, which is a standard of measurement to measure the software application. The basic and primary purpose of the functional point analysis is to measure and provide the software application functional size to the client, customer, and the stakeholder on their request. Further, it is used to measure the software project development along with its maintenance, consistently throughout the project irrespective of the tools and the technologies. Following are the points regarding FPs 1. FPs of an application is found out by counting the number and types of functions used in the applications. Various functions used in an application can be put under five types, as shown: Measurements Parameters Examples 1. Number of External Inputs (EI) Input screen and tables 2. Number of External Output (EO) Output screens and reports 3. Number of external inquiries (EQ) Prompts and interrupts. 4. Number of internal files (ILF) Databases and directories 5. Number of external interfaces (EIF) Shared databases and shared routines. All these parameters are then individually assessed for complexity.

**Procedure:**

1. Find the cost of project using LOC.

2. Find the size of project using FP method.

**Output:**

**LOC (Lines of Code) Analysis:**

* Frontend (HTML, CSS, JavaScript):
  + Total LOC: 1,500 (HTML: 500, CSS: 400, JavaScript: 600)
* Backend (Java/Python):
  + Total LOC: 3,200 (Java/Python code for handling requests, task management, user roles)
* Database (SQL queries):
  + Total LOC: 800 (SQL for creating and managing project-related tables like tasks, users, roles)

**Total LOC: 5,500 lines of code**

**FP (Function Point) Analysis**

* External Inputs (EI):
  + Add new project (Complexity: Average)
  + Assign team members (Complexity: Average)
  + Define task deadlines (Complexity: Average)
  + Number of Inputs: 3
* External Outputs (EO):
  + Generate project report (Complexity: High)
  + Task completion report (Complexity: High)
  + Project Gantt chart (Complexity: High)
  + Number of Outputs: 3
* Internal Logical Files (ILF):
  + Project details (Complexity: High)
  + User details (Complexity: Average)
  + Task information (Complexity: High)
  + Number of Files: 3
* External Interface Files (EIF):
  + Integrating with external time tracking system (Complexity: Low)
  + External calendar API (Complexity: Average)
  + Number of Interface Files: 2
* User Inquiries (EQ):
  + Search for projects by date (Complexity: Low)
  + View team member details (Complexity: Low)
  + Number of Inquiries: 2

##### **Function Points Calculation:**

* Total EI: 3 × 4 = 12 FP (Average Complexity)
* Total EO: 3 × 7 = 21 FP (High Complexity)
* Total ILF: 3 × 7 = 21 FP (High Complexity)
* Total EIF: 2 × 3 = 6 FP (Low to Average Complexity)
* Total EQ: 2 × 3 = 6 FP (Low Complexity)

**Total Function Points (FP):**12 + 21 + 21 + 6 + 6 = 66 FP

### **Conclusion:**

* **LOC Analysis:** Provides a rough estimate of 5,500 lines of code for the Project Management System, showing the project's size in terms of code but not focusing on its complexity or functionality.
* **FP Analysis:** Based on functionality, the system has around 66 function points, offering a detailed view of the system’s complexity and the user-interaction elements. FP analysis helps better estimate effort based on what the system actually does, independent of the programming language or coding style.