**Project size Estimation Techniques**

Estimation of the size of software is an essential part of Software Project Management. It helps the project manager to further predict the effort and time which will be needed to build the project. Various measures are used in project size estimation. Some of these are:

* Lines of Code
* Number of entities in ER diagram
* Total number of processes in detailed data flow diagram
* Function points

**1. Lines of Code (LOC):** As the name suggest, LOC count the total number of lines of source code in a project. The units of LOC are:

* KLOC- Thousand lines of code
* NLOC- Non comment lines of code
* KDSI- Thousands of delivered source instruction

The size is estimated by comparing it with the existing systems of same kind. The experts use it to predict the required size of various components of software and then add them to get the total size.

**Advantages:**

* Universally accepted and is used in many models like COCOMO.
* Estimation is closer to developer’s perspective.
* Simple to use.

**Disadvantages:**

* Different programming languages contains different number of lines.
* No proper industry standard exist for this technique.
* It is difficult to estimate the size using this technique in early stages of project.

**2. Number of entities in ER diagram:**[ER model](https://www.geeksforgeeks.org/database-management-system-er-model/) provides a static view of the project. It describes the entities and its relationships. The number of entities in ER model can be used to measure the estimation of size of project. Number of entities depends on the size of the project. This is because more entities needed more classes/structures thus leading to more coding.

**Advantages:**

* Size estimation can be done during initial stages of planning.
* Number of entities is independent of programming technologies used.

**Disadvantages:**

* No fixed standards exist. Some entities contribute more project size than others.
* Just like FPA, it is less used in cost estimation model. Hence, it must be converted to LOC.

**3. Total number of processes in detailed data flow diagram:** Data Flow Diagram(DFD) represents the functional view of a software. The model depicts the main processes/functions involved in software and flow of data between them. Utilization of number of functions in DFD to predict software size. Already existing processes of similar type are studied and used to estimate the size of the process. Sum of the estimated size of each process gives the final estimated size.

**Advantages:**

* It is independent of programming language.
* Each major processes can be decomposed into smaller processes. This will increase the accuracy of estimation

**Disadvantages:**

* Studying similar kind of processes to estimate size takes additional time and effort.
* All software projects are not required to construction of DFD.

**4. Function Point Analysis:** In this method, the number and type of functions supported by the software are utilized to find FPC(function point count). The steps in function point analysis are:

* Count the number of functions of each proposed type.
* Compute the Unadjusted Function Points(UFP).
* Find Total Degree of Influence(TDI).
* Compute Value Adjustment Factor(VAF).
* Find the Function Point Count(FPC).

The explanation of above points given below:

* **Count the number of functions of each proposed type:** Find the number of functions belonging to the following types:
  + External Inputs: Functions related to data entering the system.
  + External outputs:Functions related to data exiting the system.
  + External Inquiries: They leads to data retrieval from system but don’t change the system.
  + Internal Files: Logical files maintained within the system. Log files are not included here.
  + External interface Files: These are logical files for other applications which are used by our system.
* **Compute the Unadjusted Function Points(UFP):** Categorise each of the five function types as simple, average or complex based on their complexity. Multiply count of each function type with its weighting factor and find the weighted sum. The weighting factors for each type based on their complexity are as follows:

| **FUNCTION TYPE** | **SIMPLE** | **AVERAGE** | **COMPLEX** |
| --- | --- | --- | --- |
| External Inputs | 3 | 4 | 6 |
| External Output | 4 | 5 | 7 |
| External Inquiries | 3 | 4 | 6 |
| Internal Logical Files | 7 | 10 | 15 |
| External Interface Files | 5 | 7 | 10 |

* **Find Total Degree of Influence:** Use the ’14 general characteristics’ of a system to find the degree of influence of each of them. The sum of all 14 degrees of influences will give the TDI. The range of TDI is 0 to 70. The 14 general characteristics are: Data Communications, Distributed Data Processing, Performance, Heavily Used Configuration, Transaction Rate, On-Line Data Entry, End-user Efficiency, Online Update, Complex Processing Reusability, Installation Ease, Operational Ease, Multiple Sites and Facilitate Change.  
  Each of above characteristics is evaluated on a scale of 0-5.
* **Compute Value Adjustment Factor(VAF):** Use the following formula to calculate VAF  
  VAF = (TDI \* 0.01) + 0.65
* **Find the Function Point Count:** Use the following formula to calculate FPC  
  FPC = UFP \* VAF

**Advantages:**

* It can be easily used in the early stages of project planning.
* It is independing on the programming language.
* It can be used to compare different projects even if they use different technologies(database, language etc).

**Disadvantages:**

* It is not good for real time systems and embedded systems.
* Many cost estimation models like COCOMO uses LOC and hence FPC must be converted to LOC.