Software Project Estimation

Estimation of various project parameters is an important project planning activity. The different parameters of a project that need to be estimated include:

1) Estimate the size of the development product. This generally ends up in either Lines of Code (LOC) or Function Points (FP), but there are other possible units of measure.

2) Estimate the effort in person-months or person-hours.

3) Estimate the schedule in calendar months.

4) Estimate the project cost in dollars (or local currency).

Software Sizing

Size refers to a quantifiable outcome of the software project. If a direct approach is taken, size can be measured in lines of code (LOC). If an indirect approach is chosen, size is represented as function points (FP).

Regardless of the estimation variable that is used, the project planner begins by estimating a range of values for each function or information domain value. Using historical data or (when all else fails) intuition, estimate an optimistic, most likely, and pessimistic size value for each function or count for each information domain value.

A three-point or expected value can then be computed. The expected value for the estimation variable (size) S can be computed as a weighted average of the optimistic (S_{opt}), most likely (S_m), and pessimistic (S_{pess}) estimates.

$$\boldsymbol{S = (S_{opt} + 4S_m + S_{pess}) / 6} \quad \ldots\ldots..(5\text{-}1)$$

- **Lines of Code (LOC)**

It's a software metric used to measure the size of a computer program by counting the number of lines in the text of the program's source code. LOC is typically used to predict the amount of effort that will be required to develop a program, as well as to estimate programming productivity or maintainability once the software is produced.

There are two major types of LOC measures: physical LOC and logical LOC. Specific definitions of these two measures vary, but the most common definition of physical LOC is a count of lines in the text of the program's source code excluding comment lines and blank lines.

Logical LOC attempts to measure the number of executable "statements", but their specific definitions are tied to specific computer languages. physical LOC measures are more sensitive to logically irrelevant formatting and style conventions than logical LOC.

Consider this snippet of C code as an example when determining LOC:

```
for (i = 0; i < 100; i++)  printf ("hello"); /* How many lines of code is this? */
```

In this code we have:

- 1 physical line of code (LOC),
- 2 logical lines of code (for statement and printf statement).

Depending on the programmer and coding standards, the above "line" of code could be written on many separate lines:

```
/* Now how many lines of code is this? */
for (i = 0; i < 100; i++)
{
   printf("hello");
}
```

In this code we have:

- 4 physical lines of code,
- 2 logical lines of code.

An Example of LOC-Based Estimation

CAD software:

consider a software package to be developed for a computer-aided design application for mechanical components. The software is to execute on a notebook computer. A preliminary statement of software scope can be developed:

The mechanical CAD software will accept two- and three-dimensional geometric data from a designer. The designer will interact and control the CAD system through a user interface that will exhibit characteristics of good human/machine interface design. All geometric data and other supporting information will be maintained in a CAD database. Design analysis modules will be developed to produce the required output, which will be displayed on a variety of devices. The software will be designed to control and interact with peripheral devices that include a touchpad, scanner, laser printer, and large-bed digital plotter.

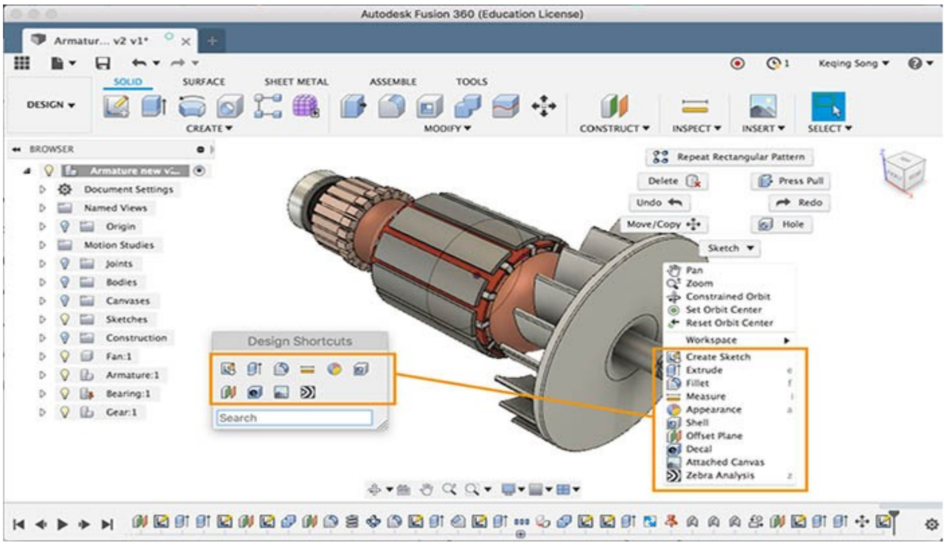

Figure 5.1: Autodesk software

The following major software functions are identified:

- User interface and control facilities (UICF)
- Two-dimensional geometric analysis (2DGA)
- Three-dimensional geometric analysis (3DGA)
- Database management (DBM)
- Computer graphics display facilities (CGDF)
- Peripheral control function (PCF)
- Design analysis modules (DAM)

An estimation table, shown in table 1, is developed. A range of LOC estimates is developed for each function. For example, the range of LOC estimates for the 3D geometric analysis function is optimistic—4600 LOC, most likely—6900 LOC, and pessimistic—8600 LOC.

Table 1: Estimation table for CAD software using LOC method

Function	Estimated LOC
User interface and control facilities (UICF)	2,300
Two-dimensional geometric analysis (2DGA)	5,300
Three-dimensional geometric analysis (3DGA)	6,800
Database management (DBM)	3,350
Computer graphics display facilities (CGDF)	4,950
Peripheral control function (PCF)	2,100
Design analysis modules (DAM)	8,400
Estimated lines of code	*33,200*

Applying equation (5-1), the expected value for the 3D geometric analysis function is 6800 LOC. Other estimates are derived in a similar fashion. By summing vertically in the estimated LOC column, an estimate of 33,200 lines of code is established for the CAD system.

System productivity describes the ratio between output and input.

Productivity = Output / Input

Productivity=Size / Effort …….. (5-1)

A review of historical data indicates that the organizational average productivity for systems of this type is 620 LOC/pm. Based on a burdened labor rate of $8000 per month, the cost per line of code is approximately $13. Based on the LOC estimate and the historical productivity data, the total estimated project cost is $431,000 and the estimated effort is 54 person-months.

Notes:

620 LOC/pm =8000$ per month

1 LOC/pm= X$

8000/620=12.9 $

Estimated project cost= $13*33200=431600$

Estimated effort=Size/ Produvtivity=33200/620=53.5

- **<u>Function Point (FP)</u>**

The function point (FP) metric can be used for measuring the functionality delivered by a system. Rather than focusing on function, each of the information domain* characteristics as well as the 14 complexity adjustment values are estimated.

Note **Requirements models represent customer requirements by describing the software in three different domains: the information domain, the functional domain, and the behavioural domain.*

Information domain values are defined in the following manner:

- *<u>Number of external inputs (EIs)</u>*

Each user input that provides data to the software is counted. Inputs are often used to update internal logical files (ILFs). Inputs should be distinguished from inquiries, which are counted separately.

- *Number of external outputs (EOs).*

Each output is data that provides information to the user. external output refers to reports, screens, error messages, etc.

- *Number of external inquiries (EQs).*

An external inquiry is defined as an online input that results in the generation of some immediate software response in the form of an online output.

- *Number of internal logical files (ILFs).*

Each internal logical file is a logical grouping of data that resides within the application's boundary and is maintained via external inputs.

- *Number of external interface files (EIFs).*

Each external interface file is a logical grouping of data that resides external to the application but provides information that may be of use to the application.

An Example of FP-Based Estimation

Referring to Table 2, inputs, outputs, inquiries, files, and external interfaces were estimated for the CAD software. To compute the count total needed in the FP equation:

$$\mathbf{FP_{estimated} = count\ total * [0.65 + 0.01 * \sum (F_i)]}$$

Table 2: Estimating information domain values (count total)

Information domain value	Opt.	Likely	Pess.	Est. count	Weight	FP count
Number of external inputs	20	24	30	24	4	96 (24 x 4 = 96)
Number of external outputs	12	14	22	14	5	70 (14 x 5 = 70)
Number of external inquiries	16	20	28	20	5	100 (20 x 5 = 100)
Number of internal logical files	4	4	5	4	10	40 (4 x 10 = 40)
Number of external interface files	2	2	3	2	7	14 (2 x 7 = 14)
Count total						*320*

Count total=320

The F_i (i = 1 to 14) are "complexity adjustment values" based on responses to the following questions:

1. Does the system require reliable backup and recovery?
2. Are data communications required?
3. Are there distributed processing functions?
4. Is performance critical?
5. Will the system run in an existing, heavily utilized operational environment?
6. Does the system require on-line data entry?
7. Does the on-line data entry require the input transaction to be built over multiple screens or operations?
8. Are the master files updated on-line?
9. Are the inputs, outputs, files, or inquiries complex?
10. Is the internal processing complex?
11. Is the code designed to be reusable?
12. Are conversion and installation included in the design?

13. Is the system designed for multiple installations in different organizations?

14. Is the application designed to facilitate change and ease of use by the user?

To compute a value for $\Sigma(F_i)$, each of the 14 complexity weighting factors listed in Table 3 is scored with a value between 0 (not important) and 5 (very important). The sum of these ratings for the complexity factors $\Sigma(Fi)$ is 52.

Table 3: complexity weighting factors

Complexity Factor	Value
Backup and recovery	4
Data communications	2
Distributed processing	0
Performance critical	4
Existing operating environment	3
Online data entry	4
Input transaction over multiple screens	5
Master files updated online	3
Information domain values complex	5
Internal processing complex	5
Code designed for reuse	4
Conversion/installation in design	3
Multiple installations	5
Application designed for change	5

Finally, the estimated number of FP is derived:

$$\mathbf{FP_{estimation} = 320 * (0.65 + 0.01 * 52) = 375}$$

The organizational average productivity for systems of this type is 6.5 FP/pm. Based on a burdened labor rate of $8,000 per month, the cost per FP is approximately $1,230. Based on the FP estimate and the historical productivity data, the total estimated project cost is $461,000 and the estimated effort is 58 person-months.

Notes:

6.5 FP/pm =8000$ per month

1 FP/pm= X$

8000/6.5=1230 $

Estimated project cost= $1230*375=461000$

Estimated effort=375/ 6.5=33200/620=58