Cost Estimation

Unit 4

Estimation of Cost

Estimation involves answering the following questions:

1.How much effort is required to complete each activity?

2.How much calendar time is needed to complete each activity?

3.What is the total cost of each activity?

Software productivity estimation

- Size-related software metrics. These metrics are related to the size of software. The most frequently used size-related metric is lines of delivered source code.

- Function-related software metrics. These are related to the overall functionality of the software. For example, function points and object points are metrics of this type.

Total number of Function Point in it

- external inputs and outputs,
- user interactions,
- external interfaces,
- files used by the system.

UFC=∑(number of a given feature)×(weight for the given feature)
Unadjusted Function Count

How Much Does Software Development Cost?

Elements that Affect Cost

- Human Resource
- Schedule
- Scope

Development cost estimation techniques

- Parkinson's Law
- Estimation by previous projects
- Algorithmic cost modelling
- Pricing to win

Function point Analysis

FPA gives a dimensionless number defined in function points which we have found to be an effective relative measure of function value delivered to our customer.

- The Function Point Analysis technique is used to analyse the functionality delivered by software and *Unadjusted Function Point (UFP)* is the unit of measurement.

Objectives of FPA:

- The objective of FPA is to measure functionality that the user requests and receives.
- The objective of FPA is to measure software development and maintenance independently of technology used for implementation.
- It should be simple enough to minimize the overhead of the measurement process.
- It should be a consistent measure among various projects and organizations.

Types of FPA:

- **Transactional Functional Type –**
  - **(i) External Input (EI):** EI processes data or control information that comes from outside the application's boundary. The EI is an elementary process.
  - **(ii) External Output (EO):** EO is an elementary process that generates data or control information sent outside the application's boundary.
  - **(iii) External Inquiries (EQ):** EQ is an elementary process made up of an input-output combination that results in data retrieval.

Weight of FP Attributes

Measurement Parameter	Low	Average	High
1. Number of external inputs (EI)	7	10	15
2. Number of external outputs (EO)	5	7	10
3. Number of external inquiries (EQ)	3	4	6
4. Number of internal files (ILF)	4	5	7
5. Number of external interfaces (EIF)	3	4	6

- The Function Point (FP) is thus calculated with the following formula.
- **FP = Count-total * [0.65 + 0.01 * $\sum(f_i)$]**
  = Count-total * CAF
- **CAF = [0.65 + 0.01 *$\sum(f_i)$]**
- **CAF- Complexity Adjustment Factor**

COCOMO

- Cocomo (Constructive Cost Model) is a regression model based on LOC, i.e **number of Lines of Code**.
- It is a procedural cost estimate model for software projects and often used as a process of reliably predicting the various parameters associated with making a project such as size, effort, cost, time and quality.
- It was proposed by Barry Boehm in 1970 and is based on the study of 63 projects, which make it one of the best-documented models.

Key Parameters

- The key parameters which define the quality of any software products, which are also an outcome of the Cocomo are primarily Effort & Schedule:
- **Effort:** Amount of labor that will be required to complete a task. It is measured in person-months units.
- **Schedule:** Simply means the amount of time required for the completion of the job, which is, of course, proportional to the effort put. It is measured in the units of time such as weeks, months.

Organic, Semidetached, and Embedded Systems

Boehm's definition of organic, semidetached, and embedded systems:

- **Organic –** A software project is said to be an organic type if the team size required is adequately small, the problem is well understood and has been solved in the past and also the team members have a nominal experience regarding the problem.
- **Semi-detached –** A software project is said to be a Semi-detached type if the vital characteristics such as team-size, experience, knowledge of the various programming environment lie in between that of organic and Embedded. The projects classified as Semi-Detached are comparatively less familiar and difficult to develop compared to the organic ones and require more experience and better guidance and creativity. E.g.: Compilers or different Embedded Systems can be considered of Semi-Detached type.
- **Embedded –** A software project with requiring the highest level of complexity, creativity, and experience requirement fall under this category. Such software requires a larger team size than the other two models and also the developers need to be sufficiently experienced and creative to develop such complex models.