Estimation Based on Function Points

Based on the

- Number of input items (I): those items provided by the user that describe distinct application-oriented data (e.g., file names)
- Number of output items (O): those items provided to the user that generate distinct application-oriented data (e.g., reports)
- Number of user inquires (Q): interactive inputs requiring a response
- Number of files (F): master files in the system
- Number of external interfaces (E): interfaces to other systems

Estimation Based on Function Points

The FP components are categories based on the project complexity and each is given a different multiplier value:

Components	Simple	Average	Complex
Input item	3	4	6
Output item	4	5	7
Inquiry	3	4	6
File	7	10	15
Interface	5	7	10

Technical Factors for Function Points

Technical factors for function points:

1.	Data communication
2.	Distributed data processing
3.	Performance criteria
4.	Heavily utilized software
5.	High transaction rate
6.	Online data entry
7.	End-user efficiency
8.	Online updating
9.	Complex computations
10.	Reusability
11.	Ease of installation
12.	Ease of operation
13.	Portability
14.	Maintainability

Calculating Function Points

1. Calculate unjustified function points (UFP) as sum of function points for each component

   $FP = (n \times I) + (n \times O) + (n \times Q) + (n \times F) + (n \times E)$

2. Compute technical complexity factor (TCF)

   Based on degree of influence (DI) of 14 technical factors (e.g., portabilty). Each factor may have a value between 0 (no influence) to 5 (strong influence):

   $TCF = DI \times .01 + 0.65$

   Possible value of $DI = 0.0$ thru $DI = 5 \times 14 = 70$

   $TCF = (0.0..0.70) + 0.65$ = a value in range 0.65 .. 1.35

3. Compute function points: $FP = UFP \times TCF$

How to Use FPs

- FPs form the basis for effort estimation
- Suppose a new project has 506 FPs
- If historical project measurements reveal that it takes an average of 3 person-days to implement a function point, the total efforts for the new project will be $3 \times 506 = 1,518$ person-days

The COCOMO Model

- COnstructive COst Model (COCOMO)
- Combines statistical figures, mathematical equations, and expert judgement
- Widely used
- Three levels based on the level of details taken into account
  - Basic: development effort is estimated as a function of program size
  - Intermediate: cost drivers are considered
  - Detailed: cost drivers impact on each step of the development is considered

COCOMO Steps

1. Obtain an initial estimate based on K LOC (source code delivered)
2. Determine a set of 15 multiplying factors
3. Adjust the effort estimated (multiply it with all factors)

Computing the Initial Estimate

- Also known as the nominal estimate or nominal effort
- Based on person-month: $E_i = a * (KLOC)^b$ where a and b are constants and depend on the project type
- Project types
  - Organic projects: organization has a lot of experience doing such projects; requirements are less stringent (small and straightforward projects)
  - Semi-detached projects: medium-size, more complex (e.g., an operating system, or a compiler project)
  - Embedded system projects: stringent requirements, fairly complex, organization has little or no experience in that area

COCOMO Project Constants

- Intermediate COCOMO constants for different types of projects:

Project Type	*a*	*b*
Organic	3.2	1.05
Semi-detached	3.0	1.12
Embedded	2.8	1.20

- Example 1: if a software product is organic and it is estimated to be 8,000 LOC, the initial effort is calculated as: $E_i = 3.2 \times 8^{1.05} = 28$ person-months
- Example 2: if a software product is considered embedded and is estimated to be 10,000 LOC, its nominal effort is as follows: $E_i = 2.8 \times 10^{1.20} = 44$ person-months

COCOMO Multiplying Factors

- Also known as cost driver attributes
- Depend on
  - product attributes (3 of them)
  - personnel attributes (5 of them)
  - computer attributes (4 of them)
  - project attributes (3 of them)
- Example: Reliability (a product attribute) rating scales: 0.75 (very low), 0.88 (low), 1.00 (nominal), 1.15 (high), 1.40 (very high)

  Thus if the reliability requirement for a project is low, then the multiplying factor will be 0.80 and will result in a lower Person-month value.

Duration Estimation

- Objective: to determine the total duration of a project (and the duration of different phases of that project)
- Duration cannot be determined by person-month cost because person/month are not interchangeable (relationship is not linear)
- COCOMO formula for duration calculation: $M = a \times E^b$ where E is the estimated effort and

Project Type	*a*	*b*
Organic	2.5	0.38
Semi-detached	2.5	0.35
Embedded	2.5	0.32

Duration Estimation Example

Suppose the overall size of an organic software is estimated to be 20,000:

- Efforts: $E = 3.2 \times 20^{1.05} = 74$ person-months
- Duration: $D = 2.5 \times E^{0.38} = 13$ months

Cost Estimation

- To calculate costs, multiply efforts (person-months) by the average salary of the software developers
- The calculated efforts for the last example was $E = 74$
- Suppose the average monthly salary of each software developer is \$6,000.00:

  Costs = $C = 74 \times 6,000 = 440,000$