# Software Project Management

## Lab (5)

**Topics to be covered:**

- Effort estimations
  - COCOMO 81
  - COCOMO II
- Exercises

# Lab (5)

# Effort Estimations

In the previous lab, the techniques for estimating size were discussed. The second aspect of estimation involves effort estimation. There are 2 techniques for effort estimation: COCOMO 81 and COCOMO II, both of which are further explained in this lab.

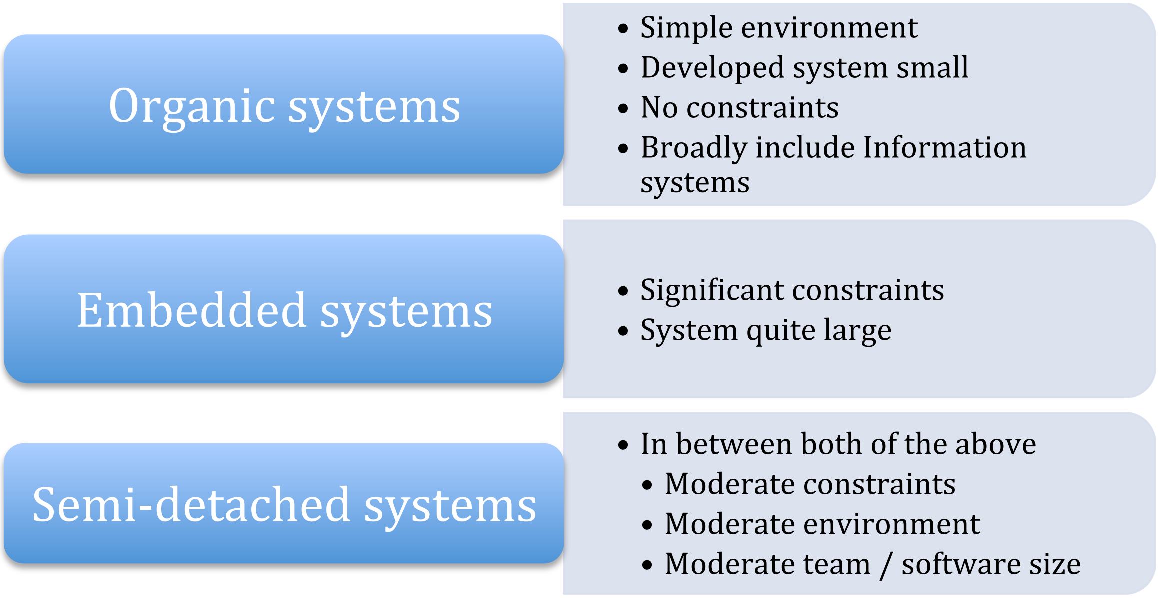
## A. COCOMO 81:

- Called the Constructive Cost Model
- This algorithm is used to estimate the amount of effort needed to complete a project
- In order to calculate effort, the previously determined size (using either Fp or Albrecht II) is used
- The following equation is used to calculate the effort:

$$effort = C\ (Size)^{K}$$

Such that C and K are constants that are determined based on the nature of the system.

A system may be one of 3 things: Organic, Semi-detached, or Embedded.



Organic systems

- Simple environment
- Developed system small
- No constraints
- Broadly include Information systems

Embedded systems

- Significant constraints
- System quite large

Semi-detached systems

- In between both of the above
  - Moderate constraints
  - Moderate environment
  - Moderate team / software size

The following table is used to determine the values of C and K:

**Table 1: Values of C and K based on the nature of the system**

| System type | C | K |
|---|---|---|
| **Organic** | 2.4 | 1.05 |
| **Semi-detached** | 3.0 | 1.12 |
| **Embedded** | 3.6 | 1.20 |

## Example:

A new project has 'average' novelty for the software supplier that is going to execute it and is this given a nominal rating on this account for precedentedness. Development flexibility is high, but requirements may change radically and so the risk resolution exponent is rated very low.

The development team are allocated in the same office and this leads to team cohesion being rated as very high, but the software house as a whole tends to be very informal in its standards and procedures and the process maturity driver has therefore been given a rating of 'low'.

What would the estimate of effort be, if the size of the application were estimated as in the region of 2000 lines of code?

## Solution:

Based on the description of the system that is provided, we can determine that the system is not of small size and also not extremely large. We know that the system isn't simple and also it doesn't have a set or rigid constraints. As a result, we can assume that this is a semi-detached system.

On that note, using table 5 (provided above), we can conclude that C = 3.0 and K = 1.12

We plug in these values into the equation to get:

$$effort = C\ (Size)^{K}$$

$$= 3.0\ (\ \underline{\boldsymbol{2}}\ )^{1.12}$$

$$= 14{,}937.4$$

## B. COCOMO II:

In an attempt to further refine effort estimations, the COCOMO II method was developed to take into account a wide range of process models that are included throughout a project's implementation. More specifically, the COCOMO II model takes into account 3 stages:

- Application composition: This has to do with the interface or external feature design
- Early design: This has to do with designing the core and fundamental structure of the system
- Post architecture: This has to do with any final modifications or fine-tuning that would be done to the system.

Given the aforementioned, the equation for calculating effort:

$$\text{Effort (PM)} = \text{A (Size)}^{\text{sf}} \text{ x (Product of exponent multipliers)}$$

- PM: Stands for person-months, which is the unit for effort
- Size: estimated using FPI or FPII (Unit is KLOC)
- A is a constant where A = 2.94
- The exponent multipliers are used to represent the previously mentioned stages, more specifically the early design and post architecture.
- Sf: better known as scale factors. An equation is used to calculate these scale factors as follows:

$$\text{sf} = \text{B} + 0.01 * \sum(exponent\ driver\ ratings)$$

- B is a constant where B = 0.91

### 1. Exponent driver ratings:

There are 5 main factors that determine the exponent driver rating. Each of those components is given one of the following ratings: very low, low, nominal, high, very high, extra high. For each of those ratings there is a corresponding numerical value. After determining the numerical value of each factor, they are all added up and the sum is substituted in the sf equation.

The following are the 5 exponent driver factors:

- Precedentedness (PREC):
  - Represents how similar the current project is to previous ones. The more unique or uncertain the project is, the higher the rating

- Development Flexibility (FLEX):
  - Represents the flexibility of the system in terms of the number of available ways to meet the requirements. The less flexible it is, the higher the rating should be.
- Architecture / Risk reduction (RESL):
  - Represents the ambiguity or uncertainty of the requirements. The more prone they are to change, the higher the rating.
- Team cohesion (TEAM):
  - Represents how close the team members are to each other (i.e. a team distributed across countries vs. a close small team). The more dispersed they are, the higher the rating.
- Process maturity (PMAT):
  - Represents how organized and structure the project is. The less uncertainty, the lower the rating.

Table 2 shows the multipliers for each of the 6 ratings that can be assigned to each of the 5 factors.

**Table 2: COCOMO II Scale factor values**

| Driver | Very low | Low | Nominal | High | Very High | Extra High |
|---|---|---|---|---|---|---|
| **PREC** | 6.20 | 4.96 | 3.72 | 2.48 | 1.24 | 0.00 |
| **FLEX** | 5.07 | 4.05 | 3.04 | 2.03 | 1.01 | 0.00 |
| **RESL** | 7.07 | 5.65 | 4.24 | 2.83 | 1.41 | 0.00 |
| **TEAM** | 5.48 | 4.38 | 3.29 | 2.19 | 1.10 | 0.00 |
| **PMAT** | 7.80 | 6.24 | 4.68 | 3.12 | 1.56 | 0.00 |

## 2. Exponent multipliers:

After determining the project size and scale factor rating, the exponent multipliers are to be determined which consider the software's early design and post architecture stages. Each of these stages has a corresponding table, which identifies the factors regarding the particular stage in question along with the multipliers for each of the 7 ratings: extra low, very low, low, nominal, high, very high, and extra high.

Table 3 shows the 7 factors involved in the early design effort multipliers as follows:

Table 3: COCOMOII Early design effort multipliers

| Code | Effort Modifier | Extra low | Very low | Low | Nominal | High | Very High | Extra High |
|---|---|---|---|---|---|---|---|---|
| **RCPX** | Product reliability & complexity | 0.49 | 6.0.60 | 0.83 | 1.00 | 1.33 | 1.91 | 2.72 |
| **RUSE** | Required reusability | | | 0.95 | 1.00 | 1.07 | 1.15 | 1.24 |
| **PDIF** | Platform difficulty | | | 0.87 | 1.00 | 1.29 | 1.81 | 2.61 |
| **PERS** | Personnel capability | 2.12 | 1.62 | 1.26 | 1.00 | 0.83 | 0.83 | 0.50 |
| **PREX** | Personnel experience | 1.59 | 1.33 | 1.12 | 1.00 | 0.87 | 0.74 | 0.62 |
| **FCIL** | Facilities available | 1.43 | 1.30 | 1.10 | 1.00 | 0.87 | 0.73 | 0.62 |
| **SCED** | Schedule pressure | | 1.43 | 1.14 | 1.00 | 1.00 | 1.00 | |

Table 4 shows the 17 factors involved in the post architecture effort multipliers as follows:

Table 4: COCOMO II Post architecture effort multipliers

| Type | Code | Modifier | Rating | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | V. Low | Low | Nominal | High | V. High | Extra high |
| Product | RELY | Required software reliability | 0.82 | 0.92 | 1 | 1.10 | 1.26 | |
| | DATA | Database size | | 0.90 | 1.00 | 1.14 | 1.28 | |
| | DOCU | Doc. Match to life cycle needs | 0.81 | 0.91 | 1.00 | 1.11 | 1.23 | |
| | CPLX | Product complexity | 0.73 | 0.87 | 1.00 | 1.17 | 1.34 | 1.74 |
| | REUSE | Required reusability | | 0.95 | 1.00 | 1.07 | 1.15 | 1.24 |
| Computer | TIME | Execution time constraint | | | 1.00 | 1.11 | 1.29 | 1.63 |
| | STOR | Main storage constraint | | | 1.00 | 1.05 | 1.17 | 1.46 |
| | PVOL | Platform volatility | | 0.87 | 1.00 | 1.15 | 1.30 | |
| Personnel | ACAP | Analyst capabilities | 1.42 | 1.19 | 1.00 | 0.85 | 0.71 | |
| | AEXP | Application experience | 1.22 | 1.10 | 1.00 | 0.88 | 0.81 | |
| | PCAP | Programmer capabilities | 1.34 | 1.15 | 1.00 | 0.88 | 0.76 | |
| | PEXP | Platform experience | 1.19 | 1.09 | 1.00 | 0.91 | 0.85 | |
| | LEXP | Programming lang. experience | 1.20 | 1.09 | 1.00 | 0.91 | 0.84 | |
| | PCON | Personnel continuity | 1.29 | 1.12 | 1.00 | 0.90 | 0.81 | |
| Project | TOOL | Use of software tools | 1.17 | 1.09 | 1.00 | 0.90 | 0.78 | |
| | SITE | Multisite development | 1.22 | 1.09 | 1.00 | 0.93 | 0.86 | 0.80 |
| | SCED | Schedule pressure | 1.43 | 1.14 | 1.00 | 1.00 | 1.00 | |

After determining the exponent multipliers as needed, the acquired ratings are all added together and fed into the original equation to get the effort in person-months. Both multipliers for the early design as well as the post architecture are all added together.

## Example:

A software supplier has to produce an application that controls a piece of equipment in a factory. A high degree of reliability is needed as a malfunction could injure the operators. The algorithms to control the equipment are also complex. The product reliability and complexity are therefore rated as very high. The company would like to take the opportunity to exploit fully the investment that they made in the project.

By reusing the control system, with suitable modifications, on future contracts. The reusability requirement is therefore rated as very high. Developers are familiar with the platform and the possibility of potential problems; in that respect, it is regarded as low. The current staff is generally very capable and is rated in this respect as very high, but the project is in a somewhat novel application domain for them, so experience is rated as nominal. The toolsets available to the developers are judged to be typical for the size of company and are rated as nominal, as is the degree of schedule pressure to meet the deadline.

i. What would be the value for each of the effort multipliers?
ii. What would be the impact on effort multipliers on a project estimated as taking 200 'staff-month'?

## Solution:

i.

| Effort Modifier | Rating | Multiplier |
|---|---|---|
| **Product reliability & complexity** | V. High | 1.91 |
| **Required reusability** | V. High | 1.15 |
| **Platform difficulty** | Low | - |
| **Personnel capability** | V. High | 1.62 |
| **Personnel experience** | Nominal | 1 |
| **Facilities available** | Nominal | 1 |
| **Schedule pressure** | Nominal | 1 |

Product = 1.91 x 1.15 x 1.62 x 1 x 1 x 1 = **3.56**

ii.

**Effort** = 200 * 3.56 = **712 PM**

## Exercises:

1. A software supplier has to produce an application that controls a piece of equipment in a factory. A high degree of reliability is needed as a malfunction could injure the operators. The algorithms to control the equipment are also complex. The product reliability and complexity are therefore rated as very high. Assume that the application consists of the following modules. Calculate the size estimated (using Albrecht II) and the needed effort (using COCOMO) to complete the application.

| **Module** | **Inputs** | **Entity types accessed** | **Outputs** |
|---|---|---|---|
| A | 1 | 2 | 10 |
| B | 5 | 1 | 1 |
| C | 7 | 1 | 7 |

2. A new project has ‘average’ novelty for the software supplier that is going to execute it and is thus given a nominal rating on this account for precedentedness. Development flexibility is high, but requirements may change radically and so the risk resolution exponent is rated very low. The development team are all located in the same office and this leads to team cohesion being rated as very high, but the software house as a whole tends to be very informal in its standards and procedures and the process maturity driver has therefore been given a rating of low

    i. What would be the scale factor in this case?
    ii. What would the estimate of effort be if the size if the size of the application was estimated as in region of 2000 lines of code?