



February 10th, 2015

Politecnico Di Milano
Computer Science

Authors:

- *Alessandro Cianferotti*
- *Alessandro Cimbelli*
- *Alessandro De Angelis*



Effort estimation

To estimate the effort we are going to use the Function Points method, which directly depends on the functionalities that the system offers. To be clearer we are going to calculate the FPs according to the weights into the table below.

Function types	Weights		
	Simple	Medium	Complex
N. Inputs	3	4	6
n. outputs	4	5	7
n. inquiry	3	4	6
n. Internal files	7	10	15
N external files	5	7	10

Calculus of the FPs:

1) Internal Logic Files

The application is going to store information about: Users, Events, Invites, Places and preferences that can be expressed about an outdoor event. We can consider, as simple entities, all of them except Event. In fact this is the core Entity of the application. The decision is to give a simple weight to all the other entities according to the fact that they have a very simple structure with few fields. While for the Event entity we are going to give a Complex weight. So the Fps for this kind of function will be: $5 * 7 = 35$ plus the complex one $1 * 15 = 15$.

Total number = 50 Fps.

2) External Logic Files

The application features also an external logic file to manage the interaction with the external forecast system for what concerns the acquisition about the weather conditions. This information is stored in only one entity: Forecast. We decide to give a complex weight for it.

Total number $1 * 10 = 10$ FPs.

3) External Inputs

The application interacts with customers and allow them according to the specification expressed in the RASD to do the following operations:

- Registration / Authentication : very simple operations that we will classify as simple so $2 * 3 = 6$ Fps .
- Management of an event : this function is composed by three specific operations that the customer can perform. The first two, that we will consider as Complex weight, are create and modify an event (Can have an high impact on the system, they need a lot of ILFs). While the last one, delete event, will be considered as simple. This according to the checks that the system is going to perform. So we will have $2 * 6 = 12$ Fps plus $1 * 3 = 3$ Fps. Tot = 15 Fps.
- Send Invitation, Accept/Decline Invitation: These three functions are not so complicated for the impact that they can have on the system so we give the lowest weight to cover them: $3 * 3 = 9$ Fps.
- Set Calendar as Public or Private: Very simple function so $1 * 3 = 3$ Fps.
- Import Calendar: this operation needs to modify a lot of entities into the system so, for this reason, we choose a Complex weight of 6 Fps.

Total number 39 Fps.

4) External Inquiries

The applications allow the customer to

- View all the public calendar
- View warning
- View invitation

The complexity of these operations can be considered as medium.

Total number $3 * 4 = 12$ Fps.

5) External Outputs

The application allows the export of the calendar. This operation only needs to collect information from ILFs.

Thus we can apply the case of medium complexity $1 * 5 = 5$ Fps.

The Application has also to guarantee a notification system through mails.

In particular or bad weather and invitation notification. This notification system can be assigned a medium complexity $1 * 5 = 5$ Fps.

Total number 10 Fps.

Final consideration

We have computed a total number of unadjusted number of FPs of 121 Fps.
Now looking at the following formula:

$$\text{LOC} = \text{AVC} * \text{n}^{\circ} \text{ Function Points}$$

Where:

- LOC = Lines of code.
- AVC = Constant depending on the programming language used to develop the application.
- Unadjusted Fps

Knowing that for java 2ee AVC is 43 we can calculate an approximate number of lines of code that will characterize our program.

$$\text{LOC} = 43 * 121 = 5203$$

After having estimated the lines of code of our project we will use COCOMO II to calculate the effort (measured in Person Month) and also the duration of the project.

The effort equation in cocomo ii is the following:

$$\text{Effort} = 2.94 * \text{EAF} * (\text{KSLOC})^E$$

Where:

EAF → Effort Adjustment Factor derived from Cost Drivers
(product of the effort multipliers corresponding to each
of the cost drivers for your project)

E → Exponent derived from Scale Drivers

To calculate the EAF we will make assumption based on our intuitions and on the values of the table below.

Cost Drivers	Ratings					
	Very Low	Low	Nominal	High	Very High	Extra High
Product attributes						
Required software reliability	0.75	0.88	1.00	1.15	1.40	
Size of application database		0.94	1.00	1.08	1.16	
Complexity of the product	0.70	0.85	1.00	1.15	1.30	1.65
Hardware attributes						
Run-time performance constraints			1.00	1.11	1.30	1.66
Memory constraints			1.00	1.06	1.21	1.56
Volatility of the virtual machine environment		0.87	1.00	1.15	1.30	
Required turnabout time		0.87	1.00	1.07	1.15	
Personnel attributes						
Analyst capability	1.46	1.19	1.00	0.86	0.71	
Applications experience	1.29	1.13	1.00	0.91	0.82	
Software engineer capability	1.42	1.17	1.00	0.86	0.70	
Virtual machine experience	1.21	1.10	1.00	0.90		
Programming language experience	1.14	1.07	1.00	0.95		
Project attributes						
Application of software engineering methods	1.24	1.10	1.00	0.91	0.82	
Use of software tools	1.24	1.10	1.00	0.91	0.83	
Required development schedule	1.23	1.08	1.00	1.04	1.10	

The assumption that we made are represented by the red squares on the table above.

We estimate our EAF factor as the product of all the multiplier

$$\text{EAF} = 1.00 * 0.94 * 1.00 * 1.00 * 1.00 * 0.86 * 0.91 * 0.86 * 1.07 * 0.82 * 1.00 * 1.04 = 0.5773$$

The value of the exponent E can be chosen according to the nature of our application. In fact we can consider our application as Organic due to the fact that we used components to develop the application. For example: PrimeFaces for the graphic interface, and java classes already made to interface with the external forecast system. The use of these components drive us to choose a value of E = 1.05

Software project	
Organic	1.05
Semi-detached	1.12
Embedded	1.20

Now that we have all the parameters we can estimate the effort expressed in Person Month

$$\text{Effort} = 2.94 * (0.5773) * (5.203)^{(1.05)} = 9.589 \text{ Person-Months}$$

Our working group is composed by 3 Person so we can calculate an estimation of the project s duration as following:

$$\text{Duration} = (9.589[\text{Person-Months}]) / (3[\text{Person}]) = 3.19 \text{ Months.}$$

The duration is more or less the time given to us to develop our application.