FUNCTION POINTS AND COCOMO 2

FUNCTION POINTS

After defining the user stories of the entire **Eco-system**, we need to define the modules and associated functions in order to estimate **function points**.

Function points measure the **offered functionalities**, which will be converted into the appropriate number of **LOCs** with respect to the programming language.

DATA FUNCTIONALITIES

We first consider the **data functionalities,** so we consider ILFs and EIFs:

- **ILF**: user-identifiable group of logically related data or control information maintained within the boundary of the application
- **EIF**: user-identifiable group of logically related data or control information referenced by the application, but maintained within the boundary of another application

To evaluate the ILFs and the EIFs complexity we use two parameters in conjunction with the functionalities we use or FP:

- **Data Element Type (DET):** user-identifiable single field within a ILF/EIF
- **Record Element Type (RET):** user-identifiable group of fields within a ILF/EIF

Below we show our **data models**, and then we will show our **data functionalities** in association with each of them.

DATA MODELS

CHEMICAL AGENTS

- Registration date
- Value
- Type
- Sensor name
- Sensor UID
- Latitude
- Longitude

USER

- CF

- Type
- Name
- Surname
- Sex
- Birthdate
- Birthplace
- Email
- Phone
- Password

ANNOUNCEMENT

- Start date
- End date
- Zones
- Description

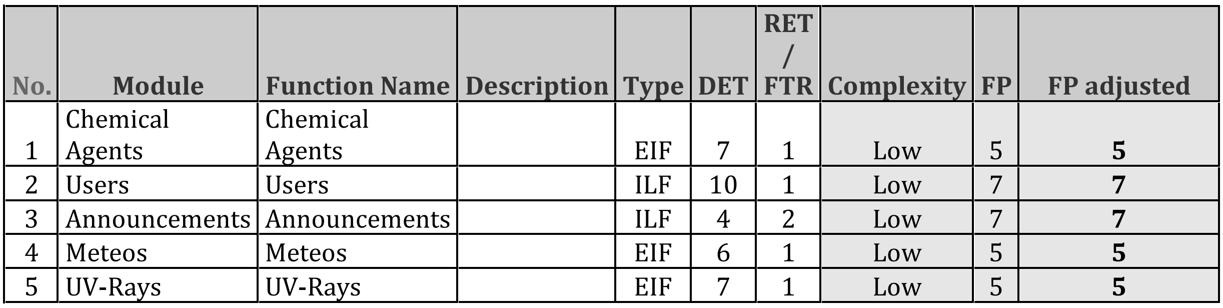
METEO

- Date
- Hours
- Description
- Temperature
- Humidity
- Wind speed

UV RAYS

- Value
- Value time
- Max value
- Max time
- Ozone value
- Ozone time
- Date

DATA FUNCTIONALITIES IN ECO



No.	Module	Function Name	Description	Type	DET	RET / FTR	Complexity	FP	FP adjusted
1	Chemical Agents	Chemical Agents		EIF	7	1	Low	5	**5**
2	Users	Users		ILF	10	1	Low	7	**7**
3	Announcements	Announcements		ILF	4	2	Low	7	**7**
4	Meteos	Meteos		EIF	6	1	Low	5	**5**
5	UV-Rays	UV-Rays		EIF	7	1	Low	5	**5**

1) **Chemical Agents**: this is an **EIF**, because we take the data from an external Web Service using REST APIs, in particular the Web Service is the AQI Service. The number of DETs is 7 because of the Chemical Agent Model, and we only have 1 RET.
2) **Users:** this is an **ILF**, because we define users within the boundary of our Eco application. We have 10 DETs because of the User model, and we only have 1 RET.
3) **Annoucements:** this is an **ILF**, because we define announcements within the boundary of our Eco application. We have 4 DETs because of the Announcement model, and we have 2 RET, because of the interaction with the User model, given that an announcement is published by a specific user with a certain CF.
4) **Meteos:** this is an **EIF**, because we take the data from an external Web Service using REST APIs, in particular the Web Service is the OpenWeather Service. The number of DETs is 6 because of the Meteos Model, and we only have 1 RET.
5) **UV rays:** this is an **EIF**, because we take the data from an external Web Service using REST APIs, in particular the Web Service is the OpenUV Service. The number of DETs is 7 because of the UV Rays Model, and we only have 1 RET.

TRANSACTIONS

Now we consider **transactions,** so we consider EI, EO and EQ:

- **External Input (EI):** Elementary process that processes data or control information that comes from outside the application boundary. The primary intent of an EI is to maintain one or more ILFs and/or to alter the behavior of the system.
- **External Output (EO):** An elementary process that sends data or control information outside the application boundary. The primary intent of an external output is to present information to a user through processing logic. The processing logic must contain at least one mathematical formula or calculation, create derived data, maintain one or more ILFs or alter the behavior of the system.
- **External Inquiry (EQ):** An elementary process that sends data or control information outside the application boundary. The primary intent of an external inquiry is to

present information to a user through the retrieval of data or control information from an ILF of EIF. The processing logic contains no mathematical formulas or calculations, and does not create derived data. No ILF is maintained during the processing, nor is the behavior of the system altered

For all these transactions we have two components:

- File **Type Referenced (FTR)** which is the read/written ILF or ELF
- the **DETs** involved in the transactions

TRANSACTIONS IN ECO

No.	Module	Function Name	Description	Type	DET	RET / FTR	Complexity	FP	FP adjusted
1	Announcements	Visualize all announcements		EQ	3	2	Low	3	**3**
2	Announcements	Insert an announcement		EI	5	2	Average	4	**4**
3	Announcements	Modify an announcement		EI	5	2	Average	4	**4**
4	Announcements	Delete an announcement		EI	3	2	Low	3	**3**
5	Announcements	Visualize an announcement by CF		EQ	5	2	Low	3	**3**
6	Announcements	Filter announcements		EQ	5	2	Low	3	**3**
7	Chemical Agents	Visualize all chemical agents with average		EO	5	1	Low	4	**4**
8	Meteos	Visualize meteo		EQ	4	1	Low	3	**3**
9	Meteos	Visualize meteo forecast		EQ	3	1	Low	3	**3**
10	UV-Rays	Visualize UV-Rays		EQ	2	1	Low	3	**3**
11	UV-Rays	Visualize UV-Rays forecast		EQ	2	1	Low	3	**3**
12	Users	Registration		EI	8	1	Low	3	**3**
13	Users	Password change		EI	2	1	Low	3	**3**
14	Users	Password forgotten		EI	2	1	Low	3	**3**
15	Users	Registration of a new operator		EI	8	1	Low	3	**3**

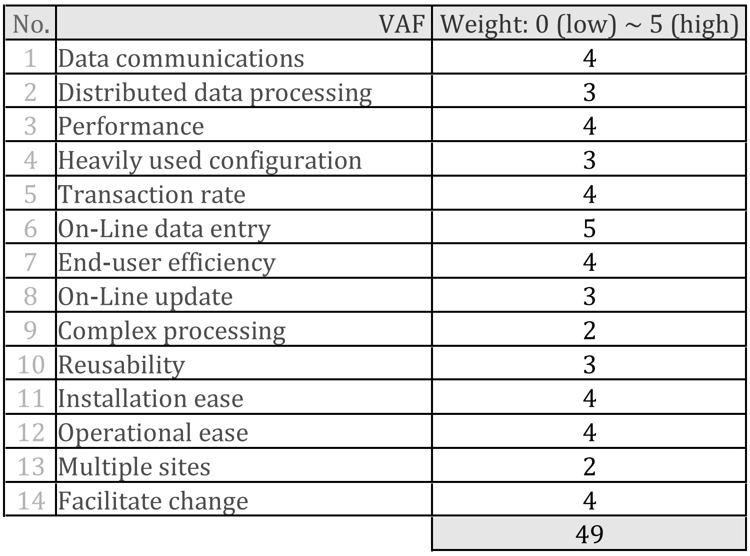
1) **Visualize all announcements:** this is an EQ, because it doesn't evaluate derived data. It uses 3 DETs because we show, for each announcement, the CF of the operator that has published the announcement, the starting date and the ending date, while it uses 2 FTR, because of the interaction with the user, because, as we have already said, each announcement is associated with the CF of the operator who has published it.
2) **Insert an announcement:** this is an EI, because the operator is inserting data that comes from outside the application boundary. It uses 5 DETs because the operator inserts, for each new announcement, his CF, the starting date, the ending date, the zones associated to the announcement and the description, while it uses 2 FTR, because of the interaction with the user, because, as we have already said, each announcement is associated with the CF of the operator who has published it.
3) **Modify an announcement:** this is an EI, because the operator is inserting data that comes from outside the application boundary. It uses 5 DETs because the operator inserts, for each announcement that he wants to update, his CF, the starting date, the ending date, the zones associated to the announcement and the description, while it uses 2 FTR, because of the interaction with the user, because, as we have already said, each announcement is associated with the CF of the operator who has published it.
4) **Delete an announcement:** this is an EI, because the operator is inserting data that comes from outside the application boundary. It uses 3 DETs because each announcement is uniquely identifies by the CF of the operator, the starting date and the ending date, while it uses 2 FTR, because of the interaction with the user, because, as we have already said, each announcement is associated with the CF of the operator who has published it.
5) **Visualize an announcement by CF:** this is an EQ, because it doesn't evaluate derived data. It uses 5 DETs because we show the CF of the operator that has published the announcement, the starting date, the ending date, the zones and the description, while it uses 2 FTR, because of the interaction with the user, because, as we have already said, each announcement is associated with the CF of the operator who has published it.
6) **Filter announcements:** this is an EQ, because it doesn't evaluate derived data. It uses 5 DETs because, in the general case, we allow the user to filter announcements by CF of the operator that has published the announcement, by starting date, by ending date, by zones and by description, while it uses 2 FTR, because of the interaction with the user, because, as we have already said, each announcement is associated with the CF of the operator who has published it.
7) **Visualize all chemical agents with average:** this is an EO, because it evaluates derived data, in our case the average. It uses 5 DETs because, for each sensor and for each chemical agent type associated to that sensor, we show the name of the

sensor, the type of chemical agent retrieved by that sensor, the value, the latitude, the longitude. It only uses 1 FTR.

8) **Visualize meteo:** this is an EQ, because it doesn't evaluate derived data. It uses 4 DETs because we show the temperature, the humidity, the wind, the description. It only uses 1 FTR.
9) **Visualize meteo forecast:** this is an EQ, because it doesn't evaluate derived data. It uses 3 DETs because we show, for each day of the week, the minimum temperature, the maximum temperature and the wind speed. It only uses 1 FTR.
10) **Visualize UV rays:** this is an EQ, because it doesn't evaluate derived data. It uses 2 DETs because we show the current value of the UV rays and the time at which we have retrieved that value. It only uses 1 FTR.
11) **Visualize UV rays forecast:** this is an EQ, because it doesn't evaluate derived data. It uses 2 DETs because we show the current value of the UV rays and the time at which we have retrieved that value. It only uses 1 FTR.
12) **Registration:** this is an EI, because the user needs to insert data from the outside of the application boundary. It uses 8 DETs because each user needs to insert the name, the surname, the email, the phone, the password, the sex, the birthdate, the birthplace. It only uses 1 FTR.
13) **Password change:** this is an EI, because the user needs to insert data from the outside of the application boundary. It uses 2 DETs because each user needs to insert the old password and the new password. It only uses 1 FTR.
14) **Password forgotten:** this is an EI, because the user needs to insert data from the outside of the application boundary. It uses 2 DETs because each user needs to insert the email and his CF. It only uses 1 FTR.
15) **Insert new operator:** this is an EI, because the admin needs to insert data from the outside of the application boundary. It uses 8 DETs because, for each new operator that the admin needs to insert, he needs the name, the surname, the email, the phone, the password, the sex, the birthdate, the birthplace. It only uses 1 FTR.

FP COMPUTATION

Each functionality, data and transactions, have **three possible weights** different for each of the above categories, and these weights are **low, medium and high**.
In order to evaluate our function points we will use the formula of **Adjusted Function Points**, in which we have to consider **14 factors** that can influence the project. Below we report our choices related to each one of the indicators:



No.	VAF	Weight: 0 (low) ~ 5 (high)
1	Data communications	4
2	Distributed data processing	3
3	Performance	4
4	Heavily used configuration	3
5	Transaction rate	4
6	On-Line data entry	5
7	End-user efficiency	4
8	On-Line update	3
9	Complex processing	2
10	Reusability	3
11	Installation ease	4
12	Operational ease	4
13	Multiple sites	2
14	Facilitate change	4
		49

1) **Data communications:** How many communication facilities are there to aid in the transfer or exchange of information with the application or system? **4**, because of the number of external services used
2) **Distributed data processing:** How are distributed data and processing functions handled? **3**, because of the architecture and of the communication protocols used for the communication between containers
3) **Performance:** Did the user require response time or throughput? **4**, because the system is very well performing under stress conditions
4) **Heavily used configuration:** How heavily used is the current hardware platform where the application will be executed? **3**, Not so high
5) **Transaction rate:** How frequently are transactions executed daily, weekly, monthly, etc.? **4,** because of the high number of interactions between the operator, the citizens and the system
6) **On line data entry:** What percentage of the information is entered On-Line? **5**, because announcements are inserted very frequently
7) **End user efficiency:** Was the application designed for end-user efficiency? **4**, because delays in the REST requests are very little
8) **On line update:** How many ILF's are updated by On-Line transaction? **3**, because only users and announcements are ILFs, and they are updated not so frequently
9) **Complex processing:** Does the application have extensive logical or mathematical processing? **2**, because we only evaluate the average values for chemical agents.
10) **Reusability:** Was the application developed to meet one or many user's needs? **3**, because we have developed the application for both citizens and users.
11) **Installation ease:** How difficult is conversion and installation? **4**, very easy, deployed with Docker
12) **Operational ease:** How effective and/or automated are start-up, back up, and recovery procedures? **4,** because the system is very comprehensible in all his parts.
13) **Multiple sites:** Was the application specifically designed, developed, and supported to be installed at multiple sites for multiple organizations? **2**, because the

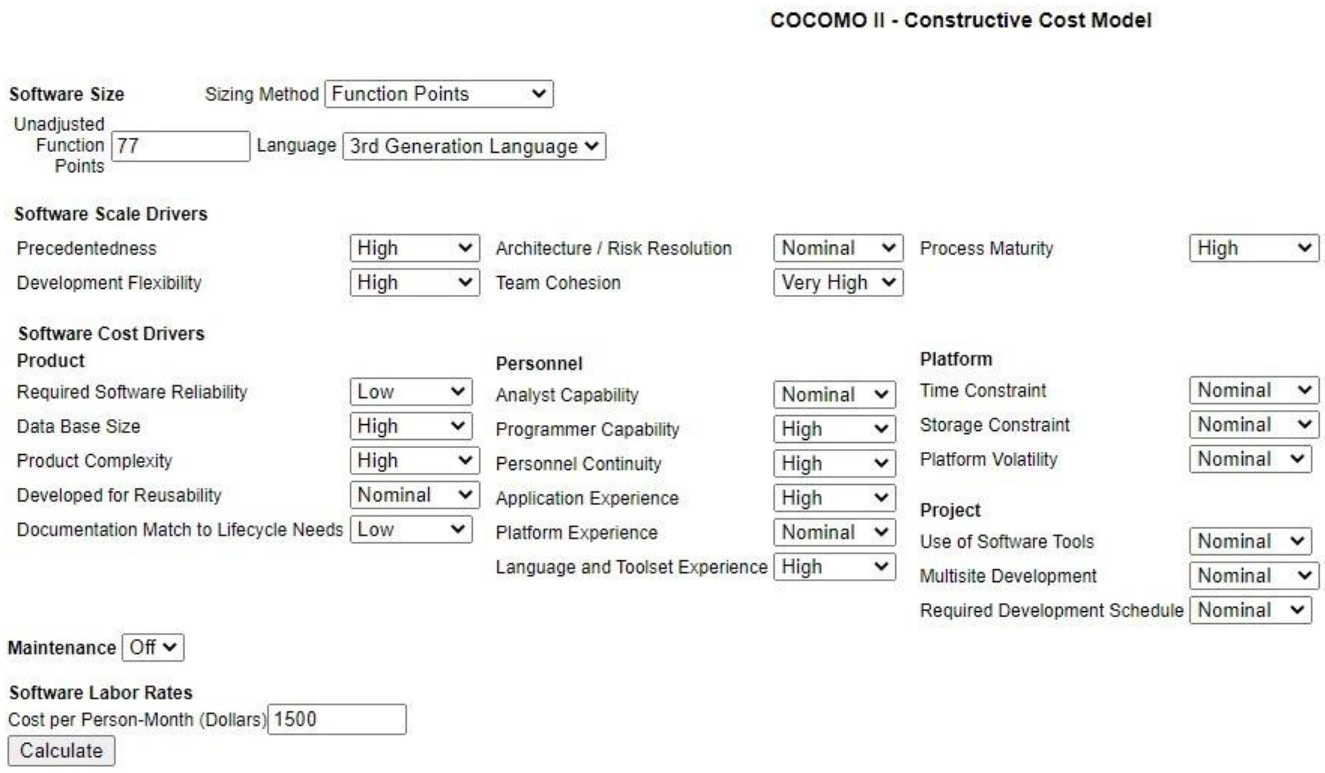
application was developed only for needings of the city of Rome, and their public administration.

14) **Facilitate change:** Was the application specifically designed, developed, and supported to facilitate change? **4**, because the application was developed by using a modular approach.

The value for the **unadjusted function points** is **77**.
The value for the **adjusted function points** is **87.78**.

COCOMO 2

Cocomo is a cost model based on the waterfall development process that relies on statistics to estimate **effort** and **cost** for a software project. Below we report our results with Cocomo, starting by the value of the unadjusted function points and the complete setup, in particular we have considered a **third generation language**, because our application was mainly developed in **Javascript/Node.js**:



COCOMO II - Constructive Cost Model

Software Size
Sizing Method: Function Points
Unadjusted Function Points: 77
Language: 3rd Generation Language

Software Scale Drivers
Precedentedness: High
Development Flexibility: High
Architecture / Risk Resolution: Nominal
Team Cohesion: Very High
Process Maturity: High

Software Cost Drivers

Product
Required Software Reliability: Low
Data Base Size: High
Product Complexity: High
Developed for Reusability: Nominal
Documentation Match to Lifecycle Needs: Low

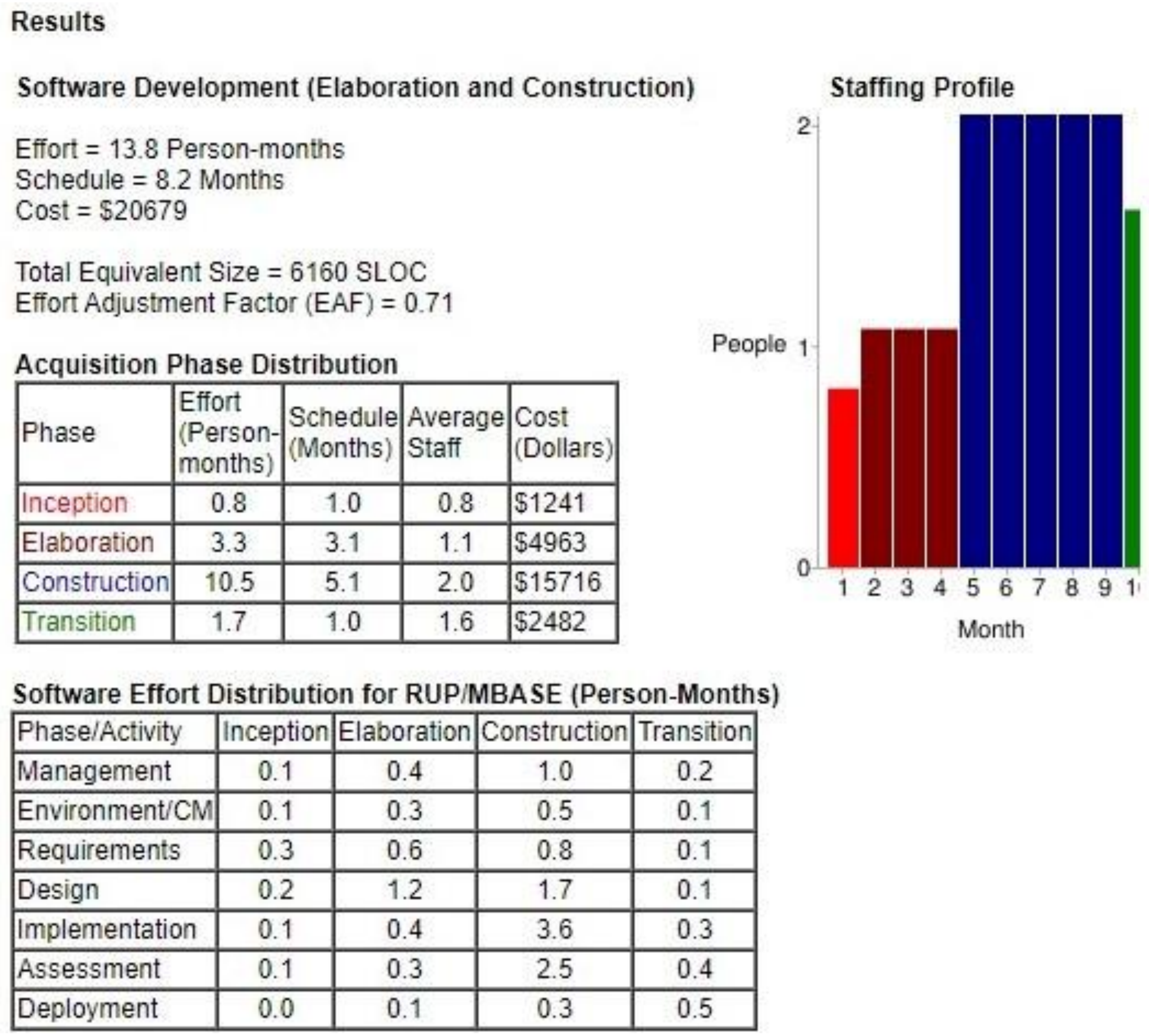
Personnel
Analyst Capability: Nominal
Programmer Capability: High
Personnel Continuity: High
Application Experience: High
Platform Experience: Nominal
Language and Toolset Experience: High

Platform
Time Constraint: Nominal
Storage Constraint: Nominal
Platform Volatility: Nominal

Project
Use of Software Tools: Nominal
Multisite Development: Nominal
Required Development Schedule: Nominal

Maintenance: Off

Software Labor Rates
Cost per Person-Month (Dollars): 1500



Results

Software Development (Elaboration and Construction)

Effort = 13.8 Person-months
Schedule = 8.2 Months
Cost = $20679

Total Equivalent Size = 6160 SLOC
Effort Adjustment Factor (EAF) = 0.71

Acquisition Phase Distribution

Phase	Effort (Person-months)	Schedule (Months)	Average Staff	Cost (Dollars)
Inception	0.8	1.0	0.8	$1241
Elaboration	3.3	3.1	1.1	$4963
Construction	10.5	5.1	2.0	$15716
Transition	1.7	1.0	1.6	$2482

Software Effort Distribution for RUP/MBASE (Person-Months)

Phase/Activity	Inception	Elaboration	Construction	Transition
Management	0.1	0.4	1.0	0.2
Environment/CM	0.1	0.3	0.5	0.1
Requirements	0.3	0.6	0.8	0.1
Design	0.2	1.2	1.7	0.1
Implementation	0.1	0.4	3.6	0.3
Assessment	0.1	0.3	2.5	0.4
Deployment	0.0	0.1	0.3	0.5

Staffing Profile

Above we have our results by the Cocomo analysis, and in particular if we look at the **total equivalent size** (in **LOC**) we have a value **6160 LOC**. By looking at the **real number of LOC** of our project, we have the following:

- **Eco server: about 2101 LOC**
- **Eco threshold: about 391 LOC**
- **Eco app: about 5345 LOC**

We have a total amount of LOC of about 7837 LOC.