POLITECNICO DI MILANO

Project reporting document

MeteoCal

February 8, 2015

Manuel Grillo
Francesco Fermi
Jerry M. Jude

0. Index

Table of Contents of Project Planning document v1.0

1. Introduction

This is the Project Reporting Document of the MeteoCal project. It contains a posthumous analysis of the project. This analysis consists in a cost estimation about:

- the size of the software (measured in LOC: Lines Of Code) via the FP (Function Points) analysis;
- the effort required (adopting a waterfall model) using the constructive cost model (i.e COCOMO).

Usually, the project-planning phase is done before the actual development of the project. In this case, the purpose is to perform the estimations to compare them with the real data, obtained at the end odf the development phase.

2. Function Points analysis

2.1. Unadjusted Function Point Count

The unadjusted function point count (UFPC) has two function types—data and transactional. These function types are further defined as follows:

- Data functions:
  - Internal Logical Files (ILFs);
  - External Interface Files (EIFs).
- Transactional functions:
  - External Inputs (EIs);
  - External Outputs (EOs);
  - External Inquiries (EIQs).

The unadjusted function point count (UFPC) reflects the specific countable functionality provided to the user by the project or application.

The application's specific user functionality is evaluated in terms of “what” is delivered by the application, not “how” it is delivered. Only user-requested and defined components are counted.

2.1.1. Count data functions

Data functions represent the functionality provided to the user to meet internal and external data requirements. Data functions are either internal logical files or external interface files. These are defined as follows:

- An internal logical file (ILF) is a user identifiable group of logically related data or control information maintained within the boundary of the application. The primary intent of an ILF is to hold data maintained through one or more elementary processes of the application being counted.
- An external interface file (EIF) is a user identifiable group of logically related data or control information referenced by the application, but maintained within the boundary of another application. The primary intent of an EIF is to hold data referenced through one or more elementary processes within the boundary of the application counted. This means an EIF counted for an application must be in an ILF in another application.

The primary difference between an internal logical file and an external interface file is that an EIF is not maintained by the application being counted, while an ILF is.

2.1.1.1. Identify the ILFs

ILFs are thus user identifiable groups of logically related data that resides entirely within the applications boundary and is maintained through External Inputs. We can attempt to detect them observing the JPA entities that have been written:

- User related ILFs: User (with Invitation and Group).
- Event related ILFs: Event, Answer (with AnswerPK).
- Weather related ILFs: Weather, WeatherCondition.
- Notification related ILFs: Information.

2.1.1.2. Identify the EIFs

ILFs are thus user identifiable groups of logically related data that is used for reference purposes only. The data resides entirely outside the application and is maintained by another application. The External Interface File is an Internal Logical File for another application. Since MeteoCal uses the OpenWeatherMap forecast service, we can consider the requested JSONs as EIFs:

- Short JSON forecast (today);
- Complete JSON forecast (1 to 16 days from today).

2.1.1.3. Determine the ILF or EIF complexity and their contribution to the UFPC

This is what we derive:

Type	Name	Rating	Description
ILF	User (w/ Invitation, Group)	Medium	Low number of fields, but it has a complex structure wrt its interactions w/ the other entities.
ILF	Event	Medium	It has a medium number of fields but It'is the core-structure of the program.
ILF	Answer (w/ AnswerPK)	Simple	Low number of fields w/ an "acceptable" structure.
ILF	Weather	Simple	Low number of fields w/ a simple structure
ILF	WeatherCondition	Simple	"Accettable" structure.
ILF	Information	Simple	Simple structure w/ a low number of fields.
EIF	Short JSON forecast	Simple	Constant complexity.
EIF	Complete JSON forecast	Simple	Constant complexity.

Once a function has been rated, its weights in terms of UFPs can be obtained from the table below:

	Weight [UFPs]		
Function type	**Simple**	**Medium**	**Complex**
Internal Logic File	7	10	15
External Logic File	5	7	10

We can thus summarize everything as follows:

Name	Rating	UFPs
User	Medium	10
Event	Medium	10
Answer	Simple	7
Weather	Simple	7
WeatherCondition	Simple	7
Information	Simple	7
Short JSON forecast	Simple	5
Complete JSON forecast	Simple	5
	Total	58

Thus, the total data UFPs are:

$$Dt = 58\ UFP$$

2.1.2. Count transactional functions

Transactional functions represent the functionality provided to the user to process data. Transactional functions are either external inputs, external outputs, or external inquiries. These are defined as follows:

- An external input (EI) is an elementary process that processes data or control information that comes from outside the application's boundary. The primary intent of an EI is to maintain one or more ILFs and/or to alter the behavior of the system.
- An external output (EO) is an elementary process that sends data or control information outside the application's boundary. The primary intent of an external output is to present information to a user through processing logic other than or in addition to the retrieval of data or control information. The processing logic must contain at least one mathematical formula or calculation, or create derived data. An external output may also maintain one or more ILFs and/or alter the behavior of the system.
- An external inquiry (EQ) is 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. The processing logic contains no mathematical formula or calculation, and creates no derived data. No ILF is maintained during the processing, nor is the behavior of the system altered.

2.1.2.1. Identify the EIs
EIs are thus elementary processes in which data crosses the boundary from outside to inside. This data may come from a data input screen, electronically or another application. The data can be either control information or business information. If the data is business information, it is used to maintain one or more internal logical files. If the data is control information it does not have to update an internal logical file. The functionalities of this type are:

- Sign up;
- Event creation and deletion;
- Event modification;
- Accept and decline an invitation;
- Decline a previously accepted invitation;
- Read an information;
- Toggle calendar privacy;
- Import a calendar.
- Log in and log out;

2.1.2.2. Identify the EIQs
EIQs are thus elementary processes with both input and output components that result in data retrieval from one or more internal logical files and external interface files. This information is sent outside the application boundary. The input process does not update any Internal Logical Files and the output side does not contain derived data. The functionalities of this type are:

- Assistance request (Reset a forgotten password and recover the user name);
- Search for a user;
- Load the user's profile;
- Load another user's page;
- Load calendar status;
- Load the notifications (both information and invitation).

2.1.2.3. Identify the EOs
EOs are thus elementary processes in which derived data passes across the boundary from inside to outside. The data creates reports or output files sent to other applications. These reports and files are created from one or more internal logical files and external interface file. Derived Data is data that is processed beyond direct retrieval and editing of information from internal logical files or external interface files. Derived data is the result of algorithms, and/or calculations. Derived data occurs when one or more data elements are combined with a formula to generate or derive an additional data element(s).

The functionalities of this type are:

- Send an email notification;
- Download a calendar.

2.1.2.4. Determine the EI, EIQ or EO complexity and their contribution to the UFPC

This is what we derive:

Type	Name	#	Rating	Description
EI	Sign up	1	Simple	n/a
EI	Event creation and modification	2	Complex	Among the other things, it has to check the time-slot availability. The algorithm is complex.
EI	Event deletion	1	Medium	It has to modify multiple entities.
EI	Accept and decline an invitation	2	Simple	n/a
EI	Decline a previously accepted invitation	1	Simple	n/a
EI	Read an information	1	Simple	n/a
EI	Toggle calendar privacy	1	Simple	n/a
EI	Import a calendar	1	Medium	It has to check the time-slot availability and the type of the events, in addition they partially "adapted".
EI	Log in and log out	2	Simple	n/a
EIQ	Assistance request	2	Simple	n/a
EIQ	Search for a user	1	Simple	n/a
EIQ	Load the user's profile	1	Medium	The calendar is lazily loaded.
EIQ	Load another user's page	1	Medium	It has to check the time-slot availability and the type of the events; in addition, the calendar is lazily loaded.
EIQ	Load calendar status	1	Simple	n/a
EIQ	Load the notifications	2	Medium	It has to load information from multiple entities and it has to load the required buttons.
EO	Send an email notification	1	Simple	n/a
EO	Download a calendar	1	Medium	It has to generate an ad-hoc file.

Once a function has been rated, its weights in terms of UFPs can be obtained from the Albrecht’s table:

	Weight [UFPs]		
Function type	**Simple**	**Medium**	**Complex**
External Input	3	4	6
External Inquiry	3	4	6
External Output	4	5	7

We can thus summarize everything using the number of different FPs and their weight given in the previous table, obtaining the UFPs, as follows:

Name	Rating	#	UFPs	Subtot
Sign up	Simple	1	3	3
Event creation and modification	Complex	2	6	12
Event deletion	Medium	1	4	4
Accept and decline an invitation	Simple	2	3	6
Decline a previously accepted invitation	Simple	1	3	3
Read an information	Simple	1	3	3
Toggle calendar privacy	Simple	1	3	3
Import a calendar	Medium	1	4	4
Log in and log out	Simple	2	3	6
Assistance request	Simple	2	3	6
Search for a user	Simple	1	3	3
Load the user's profile	Medium	1	4	4
Load another user's page	Medium	1	4	4
Load calendar status	Simple	1	3	3
Load the notifications	Medium	2	4	8
Send an email notification	Simple	1	4	4
Download a calendar	Medium	1	7	7
			Total	83

Thus, the total transactional UFPs are:

$$Tr = 83\ UFP$$

2.2. Size calculation

At this point, we are going to convert the values from UFPs to LOCs. To do it we use the QSM Function Points Languages Table, which contains updated function point language gearing factors.

Environmental factors can result in significant variation in the source statements per function point. For this reason, QSM recommends that organizations collect both code counts and final function point counts for completed software projects and incorporate this data into project estimates. Where there is no completed project data available for estimation, they provide the following industry gearing factor information (where sufficient project data exists):

- Average;
- Median ;
- Range (low/high).

These three measures should allow us to assess the amount of variation, the central tendency, and any skew to the distribution of gearing factors for each language.

The QSM Function Points value for Java EE is:

Language	QSM SLOC/FP Data			
	Avg	**Median**	**Low**	**High**
J2EE	46	49	15	67

The gearing factor is simply the average number of new plus modified (Effective) Source Lines of Code per function point in the completed project. Gearing factors are calculated by dividing the effective code count for a completed project by the final function point count. SLOC counts represent logical, not physical line counts.

Using an average value of:

$$G = 46\ {}^{SLOC}/_{FP}$$

we can derive an approximate number of SLOC in this way:

$$S' = G(Dt + Tr) = 46\frac{SLOC}{FP}(58\,UFP + 83\,UFP) = 46\frac{SLOC}{\cancel{FP}}\cdot 141\,(\cancel{U})\cancel{FP} = 6486\,SLOC$$

2.3. Comparison

At this point we only have to compare the approximate number of SLOC:

$$S{\sim}6\ KLOC$$

with the actual one. The analysis of the LOCs in the code of the final project is:

```
          83 text files.
          66 unique files.
          61 files ignored.

http://cloc.sourceforge.net v 1.62  T=1.16 s (55.4 files/s, 6684.5 lines/s)
-------------------------------------------------------------------------------
Language                     files          blank        comment           code
-------------------------------------------------------------------------------
Java                            45           1313           1183           4223
JavaServer Faces                 9            129              0            568
XML                              7              1             19            117
Maven                            1              2              8            103
CSS                              2             15             18             30
-------------------------------------------------------------------------------
SUM:                            64           1460           1228           5041
-------------------------------------------------------------------------------
```

Thus, the result is:

$$L = 5041\,LOC \sim 5\,KLOC$$

The predicted value is thus pessimistic, but we also have to consider that:

- A more precise analysis could be done using the adjusted FPs;
- No other history data about KLOC or FPs has been previously collected;
- Some optimizations (e.g. functional programming optimizations) may have reduced the actual number of logical line of code;

Thus, a “calibration” may be required.

We can make a final correction from -35% to $+35\%$ based on 14 parameters F_i. The result of $(Dt + Tr)$ can be used as an independent variable to obtain the effort estimate in FPs:

$$S'' = G(Dt + Tr)\left(0.65 + 0.01 \cdot \sum_{i=1}^{14} F_i\right)$$

Nevertheless, we decide to do not pursuit this way, since it might not be a good idea: in this scenario, an error by $\sim 20\%$ may be considered acceptable, especially due to the fact that is pessimistic.

3. Constructive cost model

3.1. Basic COCOMO approach

The effort estimation required by the MeteoCal project is done using the COCOMO, which involves the use of formulas:

$$M \approx a \cdot S^b \text{ – } \textit{effort needed to develop the application}$$

$$T \approx c \cdot S^d \text{ – } \textit{time needed to develop the application}$$

$$N \approx \frac{M}{T} \text{ – } \textit{number of people to be allocated to the project}$$

that depends on some parameters, themselves biased on the complexity of the application:

app-type	a	b	c	d
organic	2,4	1,05	2,5	0,38
semi-detached	3	1,12	2,5	0,35
embedded	3,6	1,2	2,5	0,32

In this scenario:

$$N = 3\,p$$

$$S \equiv L \sim 5\ KLOC$$

Since the application uses already existent components, auto-generated code, and has requirements that are not too rigid, it can be considered of the simplest type (i.e. organic):

$$M \approx a \cdot S^b = 2.4 \cdot 5^{1.05} \sim 13.01\ mnth \cdot p$$

$$N \approx \frac{M}{T} \rightarrow T \approx \frac{M}{N} = \frac{13.01\ mnth \cdot \cancel{p}}{3\ \cancel{p}} \sim 4.34\ mnth$$

Since:

$$1\ mnth \cdot p = 152\ h \cdot p \rightarrow mnth = \frac{152\ h \cdot \cancel{p}}{\cancel{p}} = 152\ h$$

We obtain:

$$T \approx 4.34\ mnth = 4.34 \cdot 152\ h = 659.6\ h$$

3.1.1. Count transactional functions

The time reporting amount – which considers the system design, implementation, integration and testing – is about:

$$T' = T_{design} + T_{development+integration+test} = 147\ h + 440\ h = 587\ h$$

which is not too far from the estimated one, but is not considerable correct, since the hypothesis of COCOMO does not hold.

3.2. COCOMO II approach

The COCOMO II model makes its estimates of required effort based primarily on the size of the software project measured in thousand of SLOC (i.e. KSLOC):

$$E = a \cdot EAF \cdot S^b$$

Where EAF is the Effort Adjustment Factor derived from the cost drivers D_i^c:

$$EAF = \prod_{i=1}^{16} D_i^c$$

The cost drivers are divided in:

- product factors (see Subsection 3.2.2. – Product factors);
- platform factors (see Subsection 3.2.3. – Platform factors);
- personnel factors (see Subsection 3.2.4 – Personnel factors);
- project factors (see Subsection 3.2.5 – Project factors).

The result will be:

$$EAF = 0.88 \cdot 1.0 \cdot 0.88 \cdot 1.00 \cdot 1.00 \cdot 1.31 \cdot 1.00 \cdot 0.87 \cdot 1.22 \cdot 1.00 \cdot 0.84 \cdot 1.00 \cdot 1.25 \cdot 1.10 \cdot 1.12 \cdot 1.25 \cdot 1.00 = 1.7411$$

We can therefore assume that:

$$a = 2.94$$

Then, b is an exponent derived from the scale drivers (please refer to Subsection 3.2.1. – Scaling drivers):

$$b = 1.15$$

Thus:

$$E = a \cdot EAF \cdot S^b = 2.94 \cdot 1.7411 \cdot 5.041^{1.15} = 32.89\, mnth \cdot p$$

Given:

$$N = 3\, p \ \rightarrow T' \approx \frac{E}{N} = \frac{32.89\, mnth \cdot \cancel{p}}{3\, \cancel{p}} \sim 10.96\, mnth$$

Since:

$$1\, mnth \cdot p = 152\, h \cdot p \rightarrow mnth = \frac{152\, h \cdot \cancel{p}}{\cancel{p}} = 152\, h$$

We finally obtain:

$$T' \approx 10.96\, mnth = 10.96 \cdot 152\, h = 1666\, h$$

The COCOMO II model is off by several times the actual effort spent.

3.2.1. Scaling drivers

The selection of scale drivers is based on the rationale that they are a significant source of exponential variation on a project's effort or productivity variation. Each scale driver has a range of rating levels, from Very Low to Extra High. Each rating level has a weight, W, and the specific value of the weight is called a scale factor. A project's scale factors, W_i , are summed across all of the factors, and used to determine a scale exponent, b, via the following formula:

$$b = 1.01 + 0.01 \cdot \sum_{i=0}^{4} W_i$$

The scale factors are going to be:

Scale factors (Wi)	Rating (val.)	Description
PREC (precedenteness)	3 (nominal)	Somewhat unprecedented: - Considerable organizational understanding of product objective, and experience in working with related software systems. - Moderate concurrent development of associated new hardware and operational procedures. - Some need for innovative data processing architectures and algorithms.
FLEX (development flexibility)	3 (nominal)	Some relaxation: - Considerable need for software conformance with pre-established requirements, and for software conformance with external interface specifications. - Medium premium on early completion.
RESL (architecture / risk resolution)	3 (nominal)	Often (60%).
TEAM (team cohesion)	1 (very high)	Higly cooperative: - Strong consistency of stakeholder objectives, and ability, willingness of stakeholders to accommodate other stakeholders' objectives. - Basic experience of stakeholders in operating as a team. - Considerable stakeholder teambuilding to achieve shared vision and commitments.
PMAT (process maturity)	4 (low)	Low process maturity.

Thus:

$$b = 1.01 + 0.01(3 + 3 + 3 + 1 + 4) = 1.01 + 0.01 \cdot 14 = 1.15$$

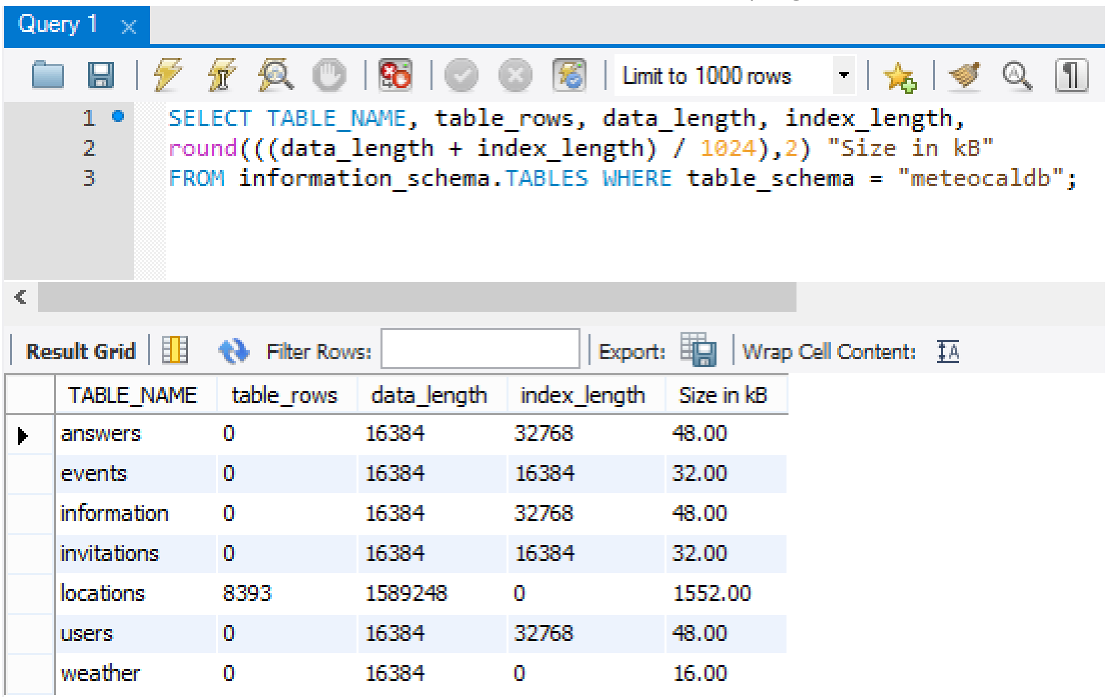
3.2.2. Product factors

Cost drivers (Ci)	Rating (val)	Description
RELY (required software reliability)	0,88 (low)	Low, easily recoverable losses.
DATA (database size)	1,00 (nominal)	*
CPLX (product complexity)	0,88 (low)	- Control operations: straightforward besting of structured programming operators. Mostly simple predicates. - Computational operations: evaluation of moderate-level expressions. - Device-dependent operations: NO cognizance needed of particular processor or I/O device characteristics. I/O done at GET/PUT level. - Data management operations: single file sub-setting with no data structure changes, no edits, no intermediate files. Moderately complex COTS-DB queries, updates. - User Interface management operations: use of simple graphic user interface (GUI) builders.
RUSE (required reusability)	1,00 (nominal)	Across project.
DOCU (documentation match to life-cycle needs)	1,00 (nominal)	Right-sized to life-cycle needs.

*the DATA cost driver attempts to capture the affect large data requirements have on product development. The rating is determined by calculating D/S. The reason the size of the database is important to consider it because of the effort required to generate the test data that will be used to exercise the program.

With the empty DB:

test data that will be used to exercise the program.



```
SELECT TABLE_NAME, table_rows, data_length, index_length,
round(((data_length + index_length) / 1024),2) "Size in kB"
FROM information_schema.TABLES WHERE table_schema = "meteocaldb";
```

TABLE_NAME	table_rows	data_length	index_length	Size in kB
answers	0	16384	32768	48.00
events	0	16384	16384	32.00
information	0	16384	32768	48.00
invitations	0	16384	16384	32.00
locations	8393	1589248	0	1552.00
users	0	16384	32768	48.00
weather	0	16384	0	16.00

We obtain (excluding the `locations` table):

$$D = 224KB = 224 \cdot 1024\ B = 229376\ B$$

$$\frac{D}{S} = \frac{229376\ B}{5041\ KSLOC} \sim 45.5$$

Since:

$$10 \leq \frac{D}{S} \leq 100$$

The result is “nominal”.

3.2.3. Platform factors

Cost drivers (Ci)	Rating (val)	Description
TIME (execution time constraint)	1,31 (very high)	85% of the execution time resource.
STOR (main storage constraint)	1,00 (nominal)	≤ 50% use of available storage.
PVOL (platform volatility)	0,87 (low)	- Major change every 12 months; - minor change every 1 month.

3.2.4. Personnel factors

Cost drivers (Ci)	Rating (val)	Description
ACAP (analyst capability)	1,22 (low)	Analysts fall in the 35th percentile.
PCAP (programmer capability)	1,00 (nominal)	Programmer team in the 55th percentile.
AEXP (applications experience)	1,00 (nominal)	1 year.
PEXP (platform experience)	1,25 (very low)	≤ 2 months.
LTEX (language and tool experience)	1,10 (low)	6 months.
PCON (personnel continuity)	0,84 (very high)	Project's annual personnel turnover: 3%

3.2.5. Project factors

Cost drivers (Ci)	Rating (val)	Description
TOOL (use of software tools)	1,12 (low)	Simple, frontend, backend CASE, little integration.
SITE (multisite development)	1,10 (low)	- Communication: wideband electronic communication, but after technical issues only some phone and email. - Collocation: international.
SCED (requirement development schedule)	1,00 (nominal)	Stretch-out of 130%.

4. References:

"*Function Point counting practices manual*" (IFPUG)

QSM Function Point language conversion table (http://www.qsm.com/resources/function-point-languages-table)

"*COCOMO II Model Definition Manual*" (Dr. Barry Boehm, University of Southern California)