GuessBid

Project Reporting

Milica Shulevska
POLITECNICO DI MILANO | SOFTWARE ENGINEERING 2
30.06.2015

CONTENTS

1 INTRODUCTION

1.1 PURPOSE

This document has the aim to provide the application of the Function Point approach to the GuessBid project. The project's size is used to apply formulas from COCOMO and compare it with the actual effort needed for this application. The results of this analysis is included below.

1.2 SCOPE

The support for this document is the RASD (Requirements Analysis Specification Document), Design Document and the Project itself.

1.3 DEFINITIONS, ACRONYMS AND ABBREVIATIONS

The following acronyms will be used through the whole document:

- RASD: Requirements Analysis and Specification Document

1.4 REFERENCES

1. COCOMO II Model Definition Manual
   http://csse.usc.edu/csse/research/COCOMOII/cocomo2000.0/CII_modelman2000.0.pdf

2 FUNCTION POINTS

To find the number of lines of code (LOC) of the project, we first need to determine the Function points.

In this section are presented the Function Points values:

- **Internal Logic File (ILF)** – represents homogeneous set of data used and managed by the application.
- **External Interface File (EIF)** – represents homogeneous set of data used by the application but generated and maintained by other applications.
- **External Input** – elementary operation to elaborate data coming from the external environment.
- **External Inquires** – represents elementary operation that generates data for the external environment.
- **External Output** – represents elementary operation that generates data for the external environment.

The values from the following table are used:

Function	Simple	Medium	Complex
N. Inputs	3	4	6
N. Outputs	4	5	7
N. Inquiry	3	4	6
N. ILF	7	10	15
N. ELF	5	7	10

Table 1: Function Points weights

2.1 INTERNAL LOGIC FILES (ILF)

The application keeps information about:

- Users
- Auctions
- Bids
- Notifications

Users, Auctions and Bids are simple and Notification is of medium structure so we have:

3*7 + 1*10 = 31 FPs

2.2 External Logic Files (ELF)

The application does not use any information which comes from an external source. No function points here.

2.3 External Inputs

The application receives inputs from the users for:

- Log in; simple

- Log out; simple

- Sign up; simple

- Create auction; medium

- Bid on an auction; medium

3*3 + 2*4 = 17 FPs

2.4 External Inquiries

The application creates views for the users on their request for:

- View auctions is simple;
- View credit is simple;
- View notifications is medium;

2*3 + 1*4 = 10 FPs

2.5 External Outputs

Outputs given from the application are:

- Notification of the winner of the auction

1*4 = 4 FPs

2.6 Unadjusted Function Points

Total sum of function points is: **31 + 17 + 10 + 4 = 62 FPs**

UFP = 62 FPs

2.7 Estimation of SLOC

In order to obtain the estimation of the lines of code we have to use the following simple formula:

LOC = UFP * AVG

Where AVG is a language dependent factor. We use the AVG specific for J2EE, which is the programming language mainly used to develop the project. The coefficient was taken from: http://www.qsm.com/resources/function-point-languages-table (we used the value in the Avg column). The final estimation turns out to be:

- SLOC = UFP * AVG = 62*46 = 2852 or in KSLOC = 2.852

The actual size of the project is 2702. The estimation seems to be more or less precise. The difference is just 150 lines which is 5.5% of the total real size.

3 COCOMO

In COCOMO II, effort is expressed as person-months (PM). The effort is computed as follows:

$$PM = A * Size^{E} * \prod_{i=1}^{n} EM_i$$

Where A=2.94(for COCOMO II.2000). The size is expressed in KSLOC. This is derived from estimating the size of software modules that will constitute the application program. The way in which the exponent E and the effort multipliers (EM) are calculated is explained below.

3.1 EXPONENT

The exponent E is an aggregation of five scale factors (SF) that account for the relative economies or diseconomies of scale encountered for software projects of different sizes:

- If E < 1.0, the project exhibits economies of scale.
- If E = 1.0, the economies and diseconomies of scale are in balance.
- If E > 1.0, the project exhibits diseconomies of scale.

The exponent E is calculated as follows:

$$E = B + 0.01 * \sum_{j=1}^{5} SF\ j$$

Where *B*=0.91 (for COCOMO II.2000)

Each scale factors has a range of rating levels, from Very Low to Extra High. Each rating level has a weight. The specific value of the weight is called a scale factor (SF). The project's scale factors, the selected scale factors ratings, are summed and used to determine a scale exponent. The following table was used to retrieve weights:

Scale Factors	Very Low	Low	Nominal	High	Very High	Extra High
PREC SFi	Thoroughly unprecedented **6.20**	Largely unprecedented **4.96**	Somewhat unprecedented **3.72**	Generally familiar **2.48**	Largely familiar **1.24**	Thoroughly familiar **0.00**
FLEX SFi	Rigorous **5.07**	Occasional relaxation **4.05**	Some relaxation **3.04**	General conformity **2.03**	Some conformity **1.01**	General goals **0.00**
RESL SFi	Little (20%) **7.07**	Some (40%) **5.65**	Often (60%) **4.24**	Generally (75%) **2.83**	Mostly (90%) **1.41**	Full (100%) **0.00**
TEAM SFi	Very difficult interactions	Some difficult interactions	Basically cooperative interactions	Largely cooperative	Highly cooperative	Seamless interactions

	5.48	**4.38**	**3.29**	**2.19**	**1.10**	**0.00**
PMAT SFi	The estimated Process Maturity Level (EPML) or:					
	SW-CMM Level 1 Lower **7.80**	SW-CMM Level 1 Upper **6.24**	SW-CMM Level 2 **4.68**	SW-CMM Level 3 **3.12**	SW-CMM Level 4 **1.56**	SW-CMM Level 5 **0.00**

Table 2: Scale factors and weights

This is the list of the appropriate scale factors rating levels for this project:

1. Precedentedness (PREC): *Nominal 3.72*
   I have never developed a web-based project and I did not have any experience with J2EE environment, but I had some experience in Java development.
2. Development Flexibility (FLEX): *High 2.03*
   There weren't many constraints in the development of this project.
3. Architecture/Risk Resolution (RESL): *Nominal 4.24*
   Since I did not have previous experience it was difficult to predict risks in the new environment. I needed more time to establish the system architecture.
4. Team Cohesion (TEAM): *Extra High 0.00*
   I am the only person in this team so there were not any communication issues.
5. Process Maturity (PMAT): *Nominal 4.68*
   Processes have been planned and documented at the project level.

   After choosing the rating levels for each scale factor, the exponent is calculated.

$$E = B + 0.01 * \sum_{j=1}^{5} SF\,j$$

E = 0.91 + 0.01* (3.72 + 2.03 + 4.24 + 0.00 + 4.68) = 1.0567

3.2 Effort Adjustment Factor

The seventeen effort multipliers (EM) are used in the COCOMO II model to adjust the nominal effort, Person-Months, to reflect the software product under development.

Each multiplicative cost driver is defined below by a rating level and a corresponding effort multiplier.

Cost Driver	Rating					
	Very Low	Low	Nominal	High	Very High	Extra High
RELY	0.75	0.88	1.00	1.15	1.39	
DATA		0.93	1.00	1.09	1.19	
CPLX	0.75	0.88	1.00	1.15	1.30	1.66
RUSE		0.91	1.00	1.14	1.29	1.49
DOCU	0.89	0.95	1.00	1.06	1.13	
TIME			1.00	1.11	1.31	1.67
STOR			1.00	1.06	1.21	1.57
PVOL		0.87	1.00	1.15	1.30	
ACAP	1.50	1.22	1.00	0.83	0.67	
PCAP	1.37	1.16	1.00	0.87	0.74	
PCON	1.24	1.10	1.00	0.92	0.84	
AEXP	1.22	1.10	1.00	0.89	0.81	
PEXP	1.25	1.12	1.00	0.88	0.81	
LTEX	1.22	1.10	1.00	0.91	0.84	
TOOL	1.24	1.12	1.00	0.86	0.72	
SITE	1.25	1.10	1.00	0.92	0.84	0.78
SCED	1.29	1.10	1.00	1.00	1.00	

Table 3: Cost drivers

1. Required Software Reliability (RELY)

Rating level: low

Effort multiplier: **0.88**

2. Data Base Size (DATA) Rating level: low

Effort multiplier: **0.93**

3. Product Complexity (CPLX) Rating level: low

Effort multiplier: **0.88**

4. Developed for Reusability (RUSE)

Rating level: nominal

Effort multiplier: **1.00**

5. Documentation match to lifecycle needs (DOCU)

Rating level: nominal

Effort multiplier: **1.00**

6. Execution Time Constraint (TIME)

Rating level: nominal

Effort multiplier: **1.00**

7. Main Storage Constraint (STOR)

Rating level: nominal

Effort multiplier: **1.00**

8. Platform Volatility (PVOL)

Rating level: low

Effort multiplier: **0.87**

9. Analyst Capability (ACAP)

Rating level: high

Effort multiplier: **0.83**

10. Programmer Capability (PCAP)

Rating level: high

Effort multiplier: **0.87**

11. Personnel Continuity (PCON)

Rating level: extra high

Effort multiplier: **/**

12. Applications Experience (APEX)

Rating level: very low

Effort multiplier: **1.22**

13. Platform Experience (PLEX)

Rating level: very low

Effort multiplier: **1.19**

14. Language and Tool Experience (LTEX)

Rating level: low

Effort multiplier: **1.10**

15. Use of Software Tools (TOOL)

Rating level: nominal

Effort multiplier: **1.00**

16. Multisite Development (SITE)

Rating level: very low

Effort multiplier: **1.25**

17. Required Development Schedule (SCED)

Rating level: nominal

Effort multiplier: **1.00**

The Effort Adjustment Factor (EAF) is defined as the product of the effort multipliers corresponding to each of the cost drivers for the project. Given the previous cost drivers analysis:

EAF = 0.903

3.3 Effort Computation

In order to compute the effort we need to know the KSLOC. The KSLOC is just SLOC/1000.

KSLOC = 2702/1000 = 2.702

From the previous sections we know:

A = 2.94; E = 1.0567; EAF = 0.903

The effort is computed as:

$PM = A*EAF*(KSLOC)^{E} = 2.94*0.903*(2.702)^{1.0567} = 7.589$ personMonths

3.4 Schedule Estimation

COCOMOII computes the time to develop the project as:

$$TDEV = \left[C * (PM_{NS})^{D+0.2*(E-B)}\right] * \frac{SCED\%}{100}$$

Where C=3.67; D=0.28; B=0.91; PM_{NS} is the estimated PM excluding the SCED effort multipliers. In this case PM = 7.5

With the values for this project:

$$TDEV = \left[3.67 * (7.5)^{0.28+0.2*(1.0567-0.91)}\right] * 1 \approx 6.84 \text{ months}$$

Therefore, the number of people that should have worked on the project is:

$$\frac{PM}{TDEV} = \frac{7.5\ personMonths}{6.84\ months} = 1.09\ persons$$

3.5 SCHEDULE COMPARISON

COCOMO II treats the number of person-hours per person-month, PH/PM, as an adjustable factor with a nominal value of 152 hours per Person-Month.

The number of person-hours can be computed corresponding to the previously computed effort:

PH = 152*7.5 = 1140 person-hours.

The time that was devoted to the project is 250 hours.

Phase	RASD	Design Document	Implementation/Testing
Hours	60	40	150

Table 4: Hours spent per phase.

The estimation here it is not even close but this may be due to the fact that COCOMO II has been developed to deal with big projects and teams.