

PROJECT REPORTING

BidWin Project 2014/2015

Emanuele La Malfa emanuele.lamalfa@mail.polimi.it
Davide Malvestiti davide.malvestiti@mail.polimi.it

1.FUNCTION POINTS ANALYSIS

Internal Logic Files

We have a database composed by 7 tables: Users, Auction, Objects, Bid, Notifications, Group and User_groups and Participation. All of them - but Group which is indeed very simple - can be considered of medium complexity.

Total 6 average, 1 low $\Rightarrow 6 \cdot 10 + 7 = 67$

External Interfaces

No external interfaces

External Inquiries

No internal inquiries

External Inputs

User inputs include Registration, Login/Logout, Search Engine, Create Objects, Create Auctions, Recharge pocket:

Total: 5 Simple, 1 High $\Rightarrow 6 + 3 \cdot 5 = 21$

External Outputs

Bids Information, Auctions Information - history + current state -, Users Information, notifications
4 Low $\Rightarrow 4 \cdot 5 = 20$

2.Count and comparison with actual data

Total count: 108

Unadjusted SLOC count, using J2EE as language: $108 \cdot 46 = 4968$

Comparison with actual SLOC count (Obtained with CLOC, on the sources folder, excluding all non programming languages)

6213 Java SLOC

We have to subtract from that count about 800 lines of code which belong to the crud used in the admin session (partially created automatically)

So the actual count is

5413 Java SLOC

Just for completeness we add the result of the command cloc - which also includes XML and SQL in the total count-.

```
cloc-1.64.exe /bidwin/
```

100 files
196 text files.

classified 196 files
Duplicate file check 196 files (138 known unique)
Unique: 100 files
138 unique files.

Counting: 100
145 files ignored.

<http://cloc.sourceforge.net> v 1.64 T=3.06 s (38.2 files/s, 3730.9 lines/s)

Language	files	blank	comment	code
Java	54	804	1916	3720
JavaServer Faces	48	202	0	2493
XML	11	9	111	1649
SQL	1	48	75	222
CSS	1	10	0	72
Visualforce Component	1	0	0	28
Ant	1	12	54	5
SUM:	117	1085	2156	8189

3.COCOMO2 Analysis

We will derive the required effort using the COCOMO2 post architecture model, using the actual source line count.

Master formula:

$PM = 2.94 * [Size]^E * \text{product}(\text{effort multipliers})$

Where

$E = 0.91 + 0.01 * \text{sum}(\text{scale factors})$

Size = 3KSLOC (considering 0 adapted SLOC, and 0 breakage factor due to requirements changes)
All terms regarding adapted sources have been set to zero because we're developing a product from scratch.

Scale and Cost drivers

According to the Cocomo specifications, for each of the cost and scale drivers, a value between very low and very high is chosen. Each value is mapped to a weight (that can be found on the tables in the linked manual) and computed in the formulas above.

To perform these calculation conveniently we used an online calculator. In the screenshot below, the values we chosen for every driver can be seen, as well as the result of the calculation.



COCOMO II - Constructive Cost Model

Software Size Sizing Method **Source Lines of Code** ▼

[SLOC](#) % Design % Code % Assessment Software Unfamiliarity
Modified Modified Integration and Understanding (0-1)
 Required Assimilation (0% - 50%)
 (0% - 8%)

New

Reused

Modified

Software Scale Drivers

Precedentedness **Nominal** ▼ Architecture / Risk Resolution **Nominal** ▼ Process Maturity **Low** ▼

Development Flexibility **Very High** ▼ Team Cohesion **Very High** ▼

Software Cost Drivers

Product		Personnel		Platform	
Required Software Reliability	Very Low ▼	Analyst Capability	Nominal ▼	Time Constraint	Nominal ▼
Data Base Size	Low ▼	Programmer Capability	Nominal ▼	Storage Constraint	Nominal ▼
Product Complexity	Low ▼	Personnel Continuity	Very High ▼	Platform Volatility	Low ▼
Developed for Reusability	Low ▼	Application Experience	Low ▼	Project	
Documentation Match to Lifecycle Needs	Nominal ▼	Platform Experience	Low ▼	Use of Software Tools	Very High ▼
		Language and Toolset Experience	Nominal ▼	Multisite Development	Very Low ▼
				Required Development Schedule	Low ▼

Maintenance **Off** ▼

Software Labor Rates

Cost per Person-Month (Dollars)

Calculate

Results

Software Development (Elaboration and Construction)

Staffing Profile

Effort = 10.4 Person-months

Schedule = 6.5 Months

Cost = \$0

Your project is too small to display a staffing profile due to truncation.

Total Equivalent Size = 5413 SLOC

Acquisition Phase Distribution

Phase	Effort (Person-months)	Schedule (Months)	Average Staff	Cost (Dollars)
Inception	0.6	0.8	0.8	\$0
Elaboration	2.5	2.4	1.0	\$0
Construction	7.9	4.0	2.0	\$0
Transition	1.3	0.8	1.5	\$0

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

Phase/Activity	Inception	Elaboration	Construction	Transition
Management	0.1	0.3	0.8	0.2
Environment/CM	0.1	0.2	0.4	0.1
Requirements	0.2	0.5	0.6	0.1
Design	0.1	0.9	1.3	0.1
Implementation	0.1	0.3	2.7	0.2
Assessment	0.1	0.3	1.9	0.3
Deployment	0.0	0.1	0.2	0.4

Your output file is http://csse.usc.edu/tools/data/COCOMO_June_28_2015_06_19_49_714674.bt

4.CONCLUSIONS

We kept count of the amount of time each members spent on the project.
Here's the result:

RASD: about 25 hours each

DD: about 25 hours each

Implementation: about 80 hours each

Acceptance: about 7 hours each

The result is **274 hours of work**.

Please note that the project is a prototype, and since it was developed for scholastic purposes, the difference between expected and actual hours is justified.

5.REFERENCE

COCOMO2.2000 manual:

http://csse.usc.edu/csse/research/COCOMOII/cocomo2000.0/CII_modelman2000.0.pdf

FP->KSLOC:

<http://www.qsm.com/resources/function-point-languages-table>