**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 => 6\*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 => 6 + 3\*5 = 21

**External Outputs**

Bids Information, Auctions Infromation - history + current state -, Users Infromation, notifications

4 Low => 4\*5 = 20

**2.Count and comparison with actual data**

Total count: 108

**Unadjusted SLOC count, using J2EE as language:** 108 \* 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 filesDuplicate 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 \* product(effort multipliers)

Where

E = 0.91 + 0.01 \* sum(scale factors)

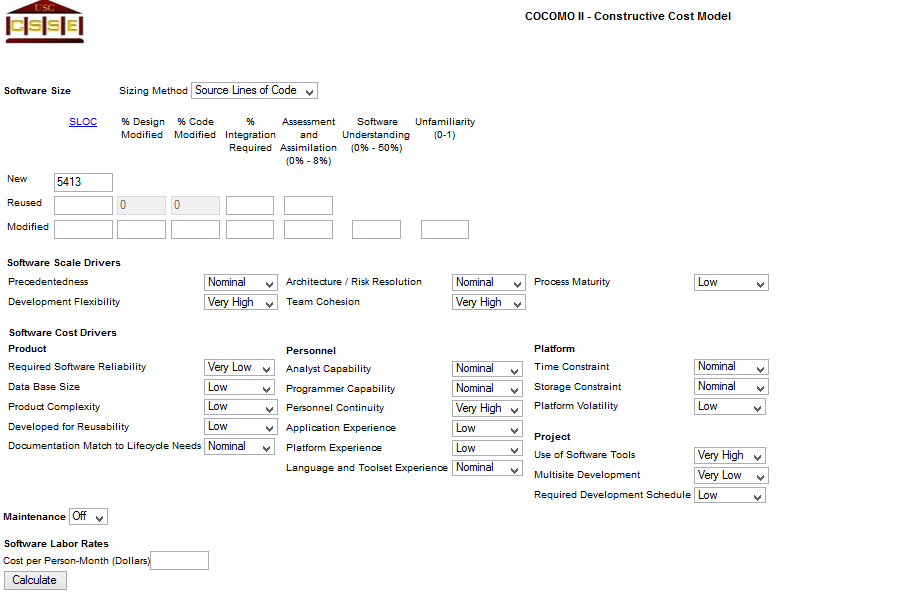
Size = 3KSLOC (considering 0 adapted SLOC, and 0 breakage factor due to requirements changes)

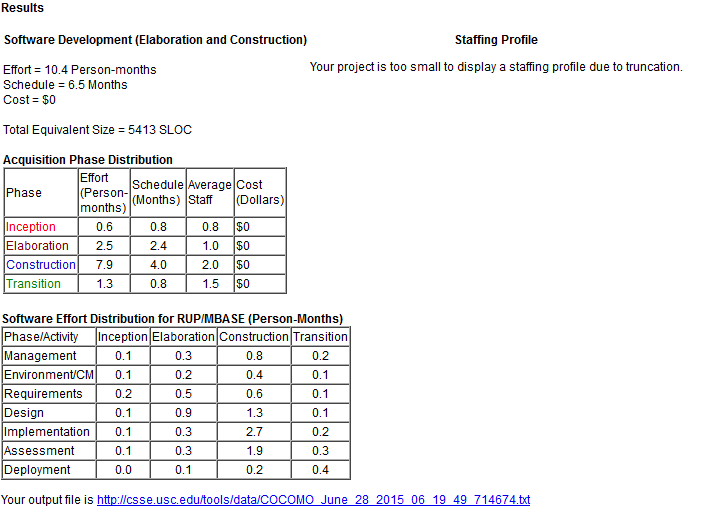
All terms regarding adapted sources have been set to zero because we’re developing a product from scratch.

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.





**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