Travel Dream

.

Matteo Camagni , Fabio Dellea

Matricole: 820203 817045

Anno: 2013 / 2014

Consegna: 29 / 11 / 2013

INDICE

1. Report di Lavoro 2
2. Modello CoCoMo 3
3. Function Point 5

Final Reporting

# 1 – Report di lavoro

Di seguito viene riportato il riassunto delle ore di lavoro spese per la realizzazione del progetto.

|  |  |  |  |
| --- | --- | --- | --- |
| Attività | Matteo Camagni | Fabio Dellea | Ore Gruppo |
| RASD Prima scrittura  Revisione | 40 ore  10 ore | 30 ore  5 ore | 70 ore  15 ore |
| DOCUMENT DESIGN Prima scrittura  Revisione | 43 ore  14 ore | 34 ore  5 ore | 77 ore  29 ore |
| IMPLEMENTAZIONE Implementazione Progetto | 80 ore | 130 ore | 210 ore |
| TEST Test progetto  Test progetto assegnatoci | 20 ore  8 ore | 15 ore  12 ore | 35 ore  20 ore |
| REPORTING Modelli di Analisi | 4 ore | 4 ore | 8 ore |
| Totale | ***175 ore*** | ***206 ore*** | ***381 ore*** |

# 2 - Modello COCOMO

Di seguito viene riportata l’analisi del progetto sviluppata secondo il modello COCOMO, versione intermediate. Tale analisi viene fatta come ultimo step, mira alla valutazione delle ore di lavoro effettivamente svolte rispetto ad una stima teorica basata sulle dimensioni del progetto.

Si assume che data la struttura e l’esperienza del team di sviluppo del prodotto TravelDream si può considerare l’applicazione di tipo Organic.

|  |  |  |  |
| --- | --- | --- | --- |
| Coefficienti | Significato | Descrizione | Valori |
| LOC  ab  bb | Linee di codice sorgente  Parametro  Parametro  Stima nominale abLOCbb | **MNom** | 2346  2.40  1.05  65,92 |
| Product Attributes RELY  DATA  CPLEX | Required reliability  Database size  Product complexity | NOMINAL  LOW  LOW | 1.00  0.94  0.85 |
| Hardware Attributes TIME  STORE  VIRT  TURN | Execution time constraints  Main storage constraints  Virtual machine volatility  Computer turnaround time | NOMINAL  NOMINAL  LOW  LOW | 1.00  1.00  0.87  0.87 |
| Personnel attributes ACAP  AEXP  RCAP  VEXP  LEXP | Analyst capability  Application experience  Programmer capability  Virtual machine experience  Programming language experience | NOMINAL  HIGH  HIGH  LOW  NOMINAL | 1.19  1.00  1.00  1.21  1.07 |
| Project attribute MODP  TOOL  SCED | Use of modern  Use of software tools  Required development schedule | HIGH  VERY HIGH  EXTRA HIGH | 1.00  0.91  1.10 |
| Totale | ***Coefficiente TOTALE***  ***Stima M*** | ***∏i=1,..,15 Cj***  ***Mesi-uomo*** | ***0,93***  ***61,49*** |

Considerazioni:

Possiamo quindi ricavare una stima del tempo richiesto e si ottiene, utilizzando i paramentri relativi ad un’applicazione Organic:

T = cb · KLOCdb = 2.5 \* (23.46) 0.38 = 8,29 mesi -> ( circa ) 8 mesi

Data la stima delle ore uomo ( mesi ) richieste e del tempo nel quale il progetto dovrebbe essere svolto si ottiene una stima del personale richiesto per la sua realizzazione:

N = M / T = 61,49[mesi · uomo] / 8.29[mesi] = 7,41[persone]

Come si può notare da tali stime, i valori ottenuti sono maggiori rispetto a quelli effettivamente impiegati per la realizzazione del progetto.

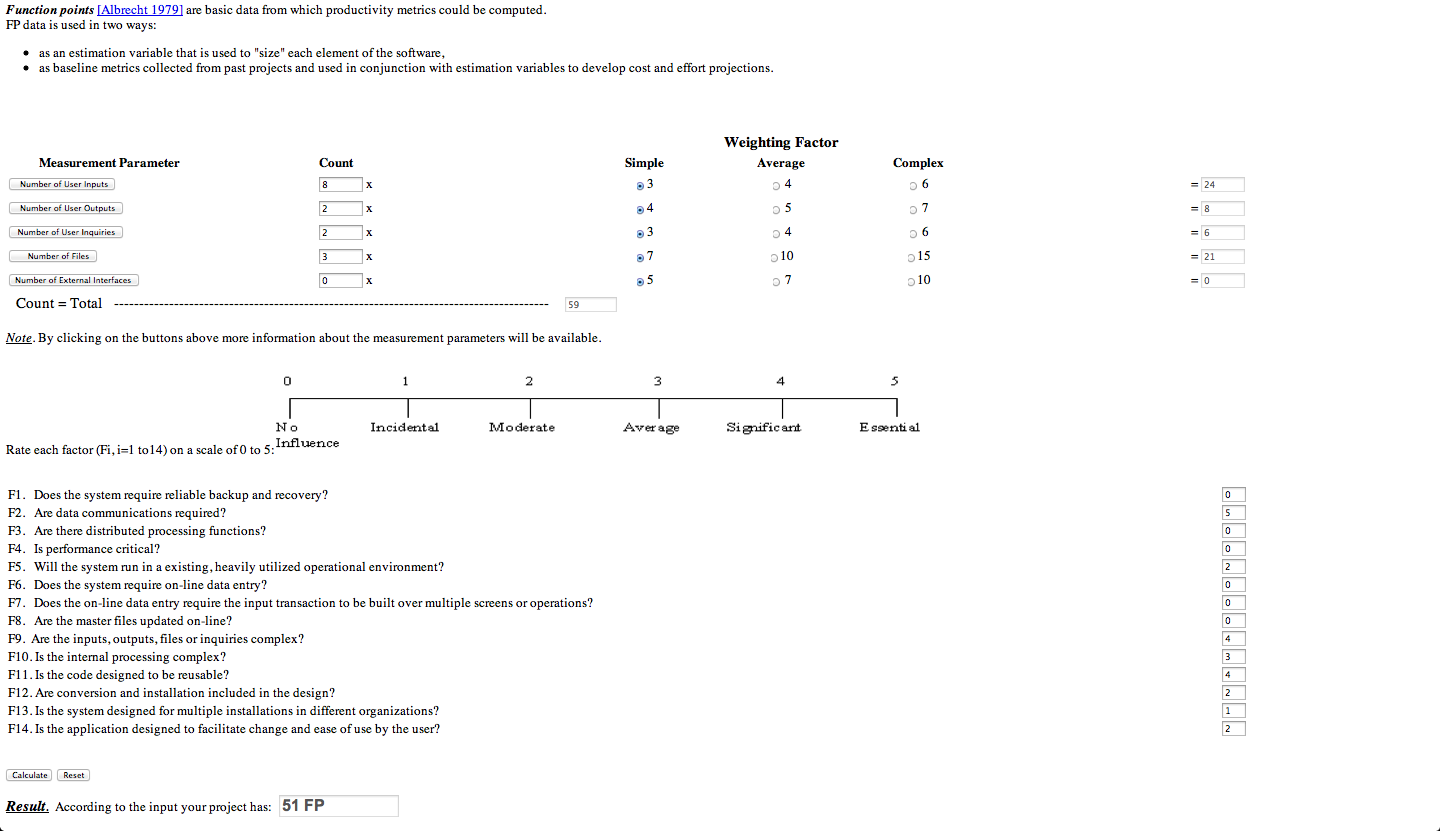
Tale discrepanza può essere spiegata dal fatto che nella realizzazione di questo progetto sono mancate le fasi di analisi dei requisiti con il cliente, l’interazione con esso e le fasi succesive di revisione del progetto dopo la fase di test.

In secondo luogo potrebbe esserci stato un riuso del codice durante lo sviluppo dell’applicazione ( sulle pagine html ), le quali appiano un po’ elevate (come indicato alparagrafo sotto).

# 3 - FUNCTION POINT

Abbiamo usato il metodo di Albrecht per la stima dei function point del nostro prodotto.

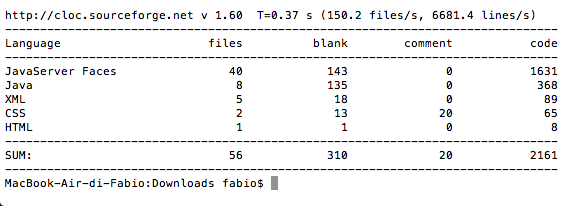
Leggendo accuratamente il RASD abbiamo individuato tre internal logical file (ILF), ovvero le 3 tabelle della base di dati, utente, wishlist e prodotto, 8 user input, corrispondenti a registrazione, login, inserimento / modifica / eliminazione prodotti, creazione pacchetti, personalizzazione pacchetti e acquisto prodotti, 2 user inquiries, corrispondenti alla ricerca e alla ricerca senza registrazione e infine 2 user outputs, la visualizzazione prodotto e la visualizzazione wishlist. Abbiamo deciso di pesarli come semplici, in quanto il progetto è stato sviluppato nelle sue funzionalità in maniera basilare, senza dovizia di particolari. Di conseguenza il peso di calcolo sarà 3 per gli user inputs, 4 per gli user outputs, 3 per le user inquiries e 7 per gli ILF. Così facendo il nostro risultato è 59 UFP ovvero un risultato Unadjusted Function Points, da pesare ulteriormente secondo vari parametri. Dopo aver pesato il risultato secondo 14 parametri (vedi immagine), il risultato è di 51 FP totali per il nostro sistema.



Il risultato è stato prevedibilmente corretto al ribasso rispetto al risultato UFP: questo è principalmente dovuto all'assenza di vincoli non funzionali particolarmente stringenti per il progetto sviluppato.

Il risultato in FP è poi stato utilizzato per fare una stima delle linee di codice previste per un progetto come il nostro. Usando come fonte la tabella presente sul sito www.qsm.com/resources/function-point-languages-table abbiamo usato come fattore di conversione FP/LOC il valore medio previsto per il linguaggio utilizzato (Java EE), ovvero 46 linee di codice per ogni FP.

Il risultato finale è di 2346 linee di codice previste. Usando lo script Piton cloc (vedi immagine , script reperibile all'indirizzo cloc.sourceforge.net) , abbiamo trovato che il numero di righe di codice del nostro progetto è 2161.



Il risultato della stima effettuata attraverso il metodo dei Function Point e poi convertito tramite tabella risulta quindi essere piuttosto accurato, in quanto si scosta solo circa dell'8,5% rispetto al valore effettivo ottenuto a fine progetto.