

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	40 ore	30 ore	70 ore
Revisione	10 ore	5 ore	15 ore
DOCUMENT DESIGN			
Prima scrittura	43 ore	34 ore	77 ore
Revisione	14 ore	5 ore	29 ore
IMPLEMENTAZIONE			
Implementazione Progetto	80 ore	130 ore	210 ore
TEST			
Test progetto	20 ore	15 ore	35 ore
Test progetto assegnatoci	8 ore	12 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	Linee di codice sorgente		2346
a_b	Parametro		2.40
b_b	Parametro		1.05
	Stima nominale $a_b \text{LOC}^{b_b}$	M_{Nom}	65,92
Product Attributes			
RELY	Required reliability	NOMINAL	1.00
DATA	Database size	LOW	0.94
CPLX	Product complexity	LOW	0.85
Hardware Attributes			
TIME	Execution time constraints	NOMINAL	1.00
STORE	Main storage constraints	NOMINAL	1.00
VIRT	Virtual machine volatility	LOW	0.87
TURN	Computer turnaround time	LOW	0.87
Personnel attributes			
ACAP	Analyst capability	NOMINAL	1.19
AEXP	Application experience	HIGH	1.00

RCAP	Programmer capability	HIGH	1.00
VEXP	Virtual machine experience	LOW	1.21
LEXP	Programming language experience	NOMINAL	1.07
Project attribute			
MODP	Use of modern	HIGH	1.00
TOOL	Use of software tools	VERY HIGH	0.91
SCED	Required development schedule	EXTRA HIGH	1.10
Totale	Coefficiente TOTALE	$\prod_{i=1,..,15} C_j$	0,93
	Stima M	Mesi-uomo	61,49

Considerazioni:

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

$$T = cb \cdot KLOC^{db} = 2.5 * (23.46)^{0.38} = 8,29 \text{ mesi} \rightarrow (\text{circa}) 8 \text{ mesi}$$

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:

$$N = M / T = 61,49[\text{mesi} \cdot \text{uomo}] / 8.29[\text{mesi}] = 7,41[\text{persone}]$$

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 successive di revisione del progetto dopo la fase di test.

In secondo luogo potrebbe esserci stato un riutilizzo del codice durante lo sviluppo dell'applicazione (sulle pagine html), le quali appaiono un po' elevate (come indicato al paragrafo 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.

Function points [Abrecht, 1979] are basic data from which productivity metrics could be computed. FP data is used in two ways:

- as an estimation variable that is used to "size" each element of the software,
- as baseline metrics collected from past projects and used in conjunction with estimation variables to develop cost and effort projections.

Measurement Parameter	Count	Weighting Factor			
		Simple	Average	Complex	
Number of User Inputs	8 x	3	4	6	= 24
Number of User Outputs	2 x	4	5	7	= 8
Number of User Inquiries	2 x	3	4	6	= 6
Number of Files	3 x	7	10	15	= 21
Number of External Interfaces	0 x	5	7	10	= 0
Count = Total					59

Note: By clicking on the buttons above more information about the measurement parameters will be available.

Influence	0	1	2	3	4	5
	No	Incidental	Moderate	Average	Significant	Essential

Rate each factor (F_i, i=1 to 14) on a scale of 0 to 5.

F1. Does the system require reliable backup and recovery?	0
F2. Are data communications required?	5
F3. Are there distributed processing functions?	0
F4. Is performance critical?	0
F5. Will the system run in a existing, heavily utilized operational environment?	2
F6. Does the system require on-line data entry?	0
F7. Does the on-line data entry require the input transaction to be built over multiple screens or operations?	0
F8. Are the master files updated on-line?	0
F9. Are the inputs, outputs, files or inquiries complex?	4
F10. Is the internal processing complex?	3
F11. Is the code designed to be reusable?	4
F12. Are conversion and installation included in the design?	2
F13. Is the system designed for multiple installations in different organizations?	1
F14. Is the application designed to facilitate change and ease of use by the user?	2

Result: According to the input your project has: **51 FP**

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.

```

http://cloc.sourceforge.net v 1.60  T=0.37 s (150.2 files/s, 6681.4 lines/s)
-----
Language             files      blank      comment      code
-----
JavaServer Faces      40         143           0        1631
Java                   8          135           0         368
XML                    5           18           0          89
CSS                    2           13          20          65
HTML                   1            1           0           8
-----
SUM:                  56         310          20       2161
-----
MacBook-Air-di-Fabio:Downloads fabio$ █

```

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.