

Politecnico Di Milano

Scuola Di Ingegneria Dell'Informazione
Corso Di Laurea Magistrale In Ingegneria Informatica



Progetto di Ingegneria del Software 2 TravelDream

Planning Report

Autori:

Francesca Garziera

Matricola 818925

Stefano Gianelli

Matricola 817398

Responsabile:

Prof. Luca Mottola

Anno Accademico 2013–2014

Indice

1	Function Point	2
1.1	Utente	2
1.1.1	Internal Logic File (ILF)	2
1.1.2	External input	2
1.1.3	External inquiry	2
1.1.4	External Interface File (EIF)	3
1.1.5	External outputs	3
1.2	Dipendente	3
1.2.1	External input	3
1.2.2	External inquiry	3
1.2.3	External output	3
2	Linee di codice	4
3	COCOMO	5

1 Function Point

Per effettuare una stima della dimensione del progetto utilizziamo il metodo dei Function Point. Le funzionalità vengono suddivise in base all'attore che compie l'operazione: utente o dipendente. Per ognuno vengono identificati gli Internal Logic File (ILF), External Interface File (EIF), External Input, External Inquiry ed External Output.

Ad ogni funzionalità viene associato un peso in base alla complessità dell'operazione e del numero delle entità coinvolte: Simple (S), Medium (M), Complex (C).

Vengono considerate Medium tutte le operazioni che effettuano numerosi controlli sugli input forniti dall'utente, diversi controlli di coerenza con le entità già presenti nel database o che coinvolgono diverse relazioni. Vengono considerate Complex tutte le operazioni che utilizzano delle altre funzioni di media complessità. Le funzionalità restanti vengono considerate Simple.

Le tabelle del database vengono considerate Simple perchè composto da un numero limitato di attributi.

1.1 Utente

1.1.1 Internal Logic File (ILF)

- le 21 tabelle del database (S)

1.1.2 External input

- login / logout (S)
- registrazione (S)
- modifica password (S)
- inserimento dati utente (S)
- modifica data arrivo / durata pacchetto (M): viene considerata di media complessità per la presenza di diversi controlli sulle date inserite dall'utente
- elimina destinazione (S)
- modifica hotel (S)
- modifica nome pacchetto (S)
- modifica città di partenza (M)
- modifica numero partecipanti (M)
- aggiunta collegamento (M)
- aggiunta escursione (M)
- rimozione escursione (S)
- eliminazione pacchetto (S)

1.1.3 External inquiry

- elenco (S)
- visualizzazione pacchetto (C)

1.1.4 External Interface File (EIF)

- coordinate città da Google Maps (S)

1.1.5 External outputs

- salvataggio pacchetto predefinito (M)
- creazione pacchetto personalizzato (M)
- aggiunta destinazione (C)
- acquisto pacchetto (M)
- condivisione pacchetto (M)
- salvataggio pacchetto condiviso (M)

1.2 Dipendente

1.2.1 External input

- aggiunta hotel (S)
- aggiunta collegamento (S)
- aggiunta escursione (S)
- aggiunta città (S)
- rimozione hotel, collegamento, escursione, città e pacchetto predefinito (M)
- modifica prezzo pacchetto predefinito (S)
- aggiunta / rimozione città di partenza (S)
- aggiunta / rimozione data partenza (S)
- aggiunta / rimozione durata (S)
- conferma pacchetto utente (S)

1.2.2 External inquiry

- elenco (S)
- visualizzazione dettagli pacchetto (S)
- visualizzazione prodotto (S)

1.2.3 External output

- creazione pacchetto predefinito (S)

È stata utilizzata la tabella con i vari pesi dei Function Point presente sulle slide del corso, alla quale abbiamo aggiunto il conteggio delle funzionalità divise per tipo e complessità.

In tabella viene mostrato il risultato complessivo di FP ottenuti e la conversione in LOC, effettuata impiegando il fattore di conversione per il linguaggio Java preso dal sito <http://www.cs.helsinki.fi/u/taina/oftu/fp.html>

		Simple	Medium	Complex	Totale per tipo
External Input	Pesi	3	4	6	
	Conteggio	22	7	-	
	Totale	66	28	-	94
External Output	Pesi	4	5	7	
	Conteggio	1	5	1	
	Totale	4	25	7	36
External Inquiry	Pesi	3	4	6	
	Conteggio	4	-	1	
	Totale	12	-	6	18
External Interface File (EIF)	Pesi	7	10	15	
	Conteggio	1	-	-	
	Totale	7	-	-	7
Internal Logical File (ILF)	Pesi	5	7	10	
	Conteggio	21	-	-	
	Totale	105	-	-	105
Totale					260
Converter					31
LOC					8060

Tabella 1: Pesi e conteggi degli FP

2 Linee di codice

Per il calcolo del numero di righe di codice abbiamo incluso solamente le effettive linee di codice di ogni metodo (quindi esclusi commenti, spazi bianchi, interfacce, ecc.), le linee di codice in HTML ed escluso i file SQL e CSS. Per il calcolo delle righe di codice Java è stato utilizzato Eclipse Metrics¹, mentre per le righe di codice HTML è stato usato CodeAnalyzer².

- EJB: 1450
- EJBClient: 259
- Web
 - Beans: 1034
 - HTML: 4929

Totale: 7672

È possibile notare che il numero di righe di codice è simile a quello previsto con gli FP. Questo risultato è stato ottenuto dopo numerose prove di conteggio delle linee di codice con diversi software (LocMetrics, cloc, Universal Code Line Counter). Alla fine si è deciso di utilizzare Eclipse Metrics in quanto effettua un conteggio più accurato delle linee di codice, escludendo gli import, le interfacce, ecc.

Sul web sono state trovate diverse tabelle per il calcolo degli FP e per la conversione in LOC. Si è deciso di utilizzare i precedenti parametri perchè sono quelli che fornivano una stima più veritiera delle linee di codice effettive.

¹<http://eclipse-metrics.sourceforge.net/>

²<http://sourceforge.net/projects/codeanalyze-gpl/>

3 COCOMO

Anche in questo caso viene impiegata la tabella presente sulle slide e utilizzata la classe “organic”, in quanto il team di sviluppo è molto ridotto e i requisiti non sono particolarmente vincolanti. Per i calcoli sono state utilizzate come Size le righe di codice effettive.

	a_b	b_b	c_b	d_b
Organic mode	2,4	1,05	2,5	0,38
Semi-detached mode	3,0	1,12	2,5	0,35
Embedded mode	3,6	1,2	2,5	0,32

Tabella 2: Coefficienti COCOMO

- $M = a_b S^{b_b} = 20,38756$ [mesi/uomo]
- $T = c_b S^{d_b} = 5,422571$ [mesi]
- $N = \frac{M}{T} = 3,759759$ [persone]

Dall’analisi risulta che per lo sviluppo del progetto sarebbero stati necessari 20 mesi / uomo, per una durata complessiva del progetto di 5 mesi e mezzo. Da questi risultati si ricava che per il progetto il team consigliato sarebbe stato di 4 persone.

	Stefano	Francesca
RASD	84	91
DD	47	55
Implementazione	170	120
Testing	5	15
Totale	306	281

Tabella 3: Riepilogo ore

Se consideriamo che un mese uomo é formato da 20 giorni ed ogni giorno é composto da 8 ore si ottiene il seguente risultato:

$$Ore\ totali = 5,42 \cdot 20 \cdot 8 = 870\ ore$$

Considerando le ore effettive si può notare una decisa discordanza rispetto al risultato ricavato tramite COCOMO. Secondo noi le ragioni vanno ricercate nell’imprecisione del metodo (in quanto effettua una stima veloce ma grezza dell’effort, in quanto si basa solamente sul numero di righe di codice senza tenere conto delle abilità del team, gli strumenti utilizzati, ecc.), nell’imprecisione dell’unità di misura (in quanto non esiste una corrispondenza certa tra i mesi / uomo e le ore effettive di lavoro svolto) ed al fatto che non viene specificato se il team si dedica solamente a questo progetto o ne segue altri in parallelo (condizione verosimile in un’impresa).