

# PROGETTO INGEGNERIA DEL SOFTWARE



## Stima degli Sforzi

*"La più vasta raccolta di musica, sempre a tua disposizione"*

*Gianluca Amato 1691171*  
*Patrizio Conte 1618639*

# Indice

<b>1. Prefazione</b>	<b>2</b>
<b>2. Introduzione al modello <i>COCOMO</i> per la stima degli sforzi e dei costi</b>	<b>3</b>
<b>3. Analisi preliminare degli Sforzi</b>	<b>5</b>
3.1 Calcolo del fattore Size . . . . .	5
3.1.1 Calcolo del fattore di aggiustamento . . . . .	9
3.1.2 Calcolo degli Adjusted Function Points . . . . .	11
3.2 Calcolo del fattore B . . . . .	11
3.3 Calcolo del fattore EAF . . . . .	12
3.4 Totale . . . . .	13
<b>4. Analisi preliminare del tempo di calendario</b>	<b>14</b>
<b>5. Analisi Preliminari dei Costi</b>	<b>15</b>

# 1. Prefazione

Lo scopo di questo documento è di mostrare un'analisi preliminare degli sforzi e dei costi necessari al completamento del progetto e di rendere la stima più precisa durante le fasi successive del progetto.

## 2. Introduzione al modello *COCOMO* per la stima dei costi

In questo capitolo si vuole dare una spiegazione approssimativa del modello **COCOMO**. *CoCoMo* (Constructive Cost Model) è un modello di stima dei costi e stima degli sforzi di sviluppo del software, è dunque un modello predittivo basato su regressione statistica che considera vari parametri pesati mediante una griglia di valutazione. Abbiamo due versioni del modello COCOMO:

- La prima **COCOMO 81**: viene pubblicata nel 1981, anche chiamato semplicemente come COCOMO. COCOMO 81 è da considerarsi un modello statico, per quanto concerne le variabili di ingresso e uscita, e analitico, in quanto può essere anche applicato a parti di un progetto. Esistono tre diversi livelli di *COCOMO* i quali si differenziano per la precisione con cui vengono effettuate le stime e, quindi, per la finezza dei modelli applicati nella valutazione dei parametri del modello. Di seguito vediamo più in dettaglio i vari livelli:
  - **Basic COCOMO** [1]: è dei tre, il livello più facile da utilizzare ma anche il meno preciso. La stima viene fatta partendo dalla dimensione del software da sviluppare.
  - **Intermediate COCOMO** [2]: questo modello calcola lo sforzo di sviluppo del software, oltre che come funzione della grandezza del programma, anche di un insieme di 'indici dei costi'.
  - **Advanced COCOMO** [3]: questo modello incorpora tutte le caratteristiche di [2], con l'aggiunta della valutazione dell'impatto dei vari costi per ogni fase del processo software.

Per ciascun livello di **COCOMO** esistono tre diverse possibili *development modes* o *complessità*:

- **Organic**
- **Semi-detached**
- **Embedded**
- La seconda **COCOMO 2**: viene rilasciata a partire dalla fine degli anni '90. In **COCOMO 2** alcuni dei più importanti fattori che contribuiscono alla durata e al costo di un progetto sono gli *Scale Drivers*, che rimpiazzano i *development modes* di COCOMO. Inoltre il modello COCOMO presentava alcune limitazioni, come il modello di processo che in COCOMO è strettamente orientato a modelli di ciclo di sviluppo tradizionali. **COCOMO**

**2** comprende un insieme di modelli applicabili al variare delle modalità di realizzazione del software:

- **Application Composition Model:** Utile per la stima di sistemi ottenuti a partire dal riuso di componenti.
- **Early Design Model:** Basata su Function Point; utile per ottenere stime in fase di analisi.
- **Post-Architecture Model:** Modelli utilizzabili dopo la definizione dell'architettura per ottenere stime più affidabili.

Per la stima dei tempi e dei costi necessari allo sviluppo del progetto *Spotify* si utilizzerà la tecnica dell'**Early Design Model** di **COCOMO 2**.

## 3. Analisi preliminare degli Sforzi

In questo capitolo si intende fornire al committente una prima valutazione approssimativa dei tempi e dei costi necessari allo sviluppo del progetto, pertanto le stime non dovranno essere considerate vincolanti, ma serviranno per dare un'idea generale della grandezza del progetto.

La formula per la stima dell'*Early Design Model* è simile a quella classica di COCOMO 81, ed è la seguente:

$$PM = A * Size^B * EAF \quad (1)$$

Dove:

- **PM**: sono i Person-Months, cioè la stima dei mesi necessari a completare il progetto;
- **A**: è una costante con valore 2,94 nella calibrazione iniziale [COCOMO II]
- **Size**: è la dimensione del sistema espressa in *KSLOC*, ovvero in migliaia di righe di codice. I Function Points vengono convertiti negli equivalenti *SLOC* e poi in *KSLOC*.
- **B**: varia da 1,1 a 1,24 a seconda di alcuni aspetti del progetto come: l'esperienza dell'organizzazione che gestisce il progetto, flessibilità nello sviluppo, etc.
- **EAF** (Effort Adjustment Factor): è un fattore moltiplicativo che si basa su alcune caratteristiche del progetto come: capacità sviluppatori, requisiti non funzionali, familiarità con piattaforma di sviluppo, etc.

Per calcolare il valore di PM dobbiamo calcolare i tre valori mancanti che sono:

- *Size*
- *B*
- *EAF*

### 3.1 Calcolo del fattore Size

Per il calcolo del fattore *Size* dobbiamo calcolare i **Function Points**. Esistono due tipi di Function Points: Adjusted e Unadjusted; per calcolare i primi abbiamo bisogno dei secondi. Per calcolare gli UFP (Unadjusted Function Points) utilizzeremo il metodo descritto dall'IFPUG (*International Function Point Users Group*) versione 4.2.1.

Possiamo quindi identificare i seguenti componenti:

- **Data Function**

- **ILF** (Internal Logical Files): raggruppamento logico di dati che ha lo scopo di memorizzare informazioni a supporto di processi elementari.
- **EIF** (External Interface Files): strutture logiche di dati provenienti da sistemi esterni.

- **Transazionali**

- **EI** (External Input): Transazioni elementari necessarie al mantenimento delle strutture logiche legate agli ILF.
- **EO** (External Output): Transazioni elementari che fanno fluire informazioni dall'interno del sistema verso l'esterno.
- **EQ** (External Inquiries): Transazioni elementari necessarie a selezionare dati e visualizzarli all'utente finale. I dati possono provenire ed essere aggregati da più ILF e EI.

Ci sono poi dei calcoli accessori da effettuare per ognuna delle caratteristiche precedenti:

- **RET** (Record Element Type): Sottogruppi legati ad un ILF o EIF.
- **DET** (Data Element Type): Sono i tipi di dati elementari utilizzati all'interno di un ILF e EIF.
- **FTR** (File Type Reference): Numero di ILF o EIF legate ad una particola transazione.

Per assegnare i pesi ad ogni requisito abbiamo bisogno di calcolare la complessità funzionale di ogni componente. Il calcolo della complessità funzionale di EI, EO, EQ è basato sul numero di **FTR**, (File Types Referenced), e sul numero di **DET**, (Data Element Types). Invece la complessità funzionale degli ILF ed EIF viene assegnata a seconda del numero di elementi di tipo dati (DET) e dal numero di **RET**, (Record Element Type).

Le seguenti tabelle collegano il numero di FTR e di DET alla complessità da assegnare alle funzioni:

**Calcolo Complessità EI**

	<b>1-4 DET</b>	<b>5-15 DET</b>	<b>16 o più DET</b>
<b>0-1 FTR</b>	Bassa	Bassa	Media
<b>2 FTR</b>	Bassa	Media	Alta
<b>3 o più FTR</b>	Media	Alta	Alta

**Calcolo Complessità EO e EQ**

	<b>1-5 DET</b>	<b>6-19 DET</b>	<b>20 o più DET</b>
<b>0-1 FTR</b>	Bassa	Bassa	Media
<b>2-3 FTR</b>	Bassa	Media	Alta
<b>4 o più FTR</b>	Media	Alta	Alta

Invece per collegare DET e RET dobbiamo calcolare la complessità in questo modo:

**Calcolo Complessità ILF e EIF**

	<b>1-19 DET</b>	<b>20-50 DET</b>	<b>51 o più DET</b>
<b>1 RET</b>	Bassa	Bassa	Media
<b>2-5 RET</b>	Bassa	Media	Alta
<b>6 o più RET</b>	Media	Alta	Alta

Per ottenere il numero di Function Point **non pesati**, si utilizza la seguente tabella di conversione che trasforma la complessità in un determinato numero di Function Point:

<b>Complessità Funzionale</b>	<b>ILF</b>	<b>EIF</b>	<b>EI</b>	<b>EQ</b>	<b>EO</b>
<b>Bassa</b>	7	5	3	3	4
<b>Media</b>	10	7	4	4	5
<b>Alta</b>	15	10	6	6	7

Quindi procediamo a calcolare la complessità di ogni requisito come descritto sopra per poi calcolare il numero di Function Points assegnati a tale complessità così da avere il numero di Function Points, non pesati, totali.



Nelle tabelle seguenti vediamo i requisiti (descritti nel *Documento dei Requisiti*) suddivisi nei vari componenti con la complessità assegnata.

External Input (EI)	
Requisiti	Complessità
Req_F_01_Effettua_Registrazione	Alta
Req_F_01.01_Recupera_DatiDiAccesso	Media
Req_F_02.02_Effettua_Log-out	Bassa
Req_F_02.01_Effettua_Download	Media
Req_F_02.06_Accesso_Playlist-utente	Bassa
Req_F_02.07_Accesso_RaccoltaBrani	Media
Req_F_02.09_Modifica_Playlist	Media
Req_F_02.14_Modifica_Profilo	Media
Req_F_02.12_Visualizza_Profilo	Media
Req_F_02.17_Carica_FotoArtista	Bassa
Req_F_02.18_Carica_AlbumArtista	Alta
Req_F_02.19_Carica_Brano	Alta
Req_F_02.20_Accesso_Statistiche	Media
Req_F_03.01_Accesso_PlaylistML	Media

External Output (EO)	
Requisiti	Complessità
Req_F_02_Effettua_Autenticazione	Bassa
Req_F_02.01_Modifica_DatiDiAccesso	Media
Req_F_02.08_Crea_Playlist	Bassa
Req_F_02.13_Modifica_MetodoPagamento	Media
Req_F_02.21_Elimina_Utente	Media
Req_F_02.22_Elimina_Playlist	Media
Req_F_04.01_NuovoAbbonamento	Alta
Req_F_04.02_ModificaAbbonamento	Alta

External Inquiries (EQ)	
Requisiti	Complessità
Req_F_02.02_Riproduzione_Offline	Media
Req_F_02.03_Riproduzione_BranoSingolo	Media
Req_F_02.04_Riproduzione_Sequenziale	Alta
Req_F_02.05_Riproduzione_Shuffle	Alta
Req_F_02.11_Modifica_GestioneQualitàRiproduzione	Media
Req_F_03.02_ApprendimentoAutomatico	Alta

External Interface Files (EIF)	
Tipi di Interfaccia	Complessità
Interfaccia Database utenti	Media
Interfaccia Content Delivery Network	Alta
Interfaccia applicazioni bancarie	Alta
Interfaccia Client Mail	Alta
Interfaccia Gestione Playlist-ML	Media
Interfaccia per API esterne	Media

Internal Logical Files (ILF)	
Tipi di Documento	Complessità
Log di errori	Media

Una volta calcolata la complessità per ogni componente possiamo passare al calcolo totale degli UFP (Unadjusted Function Points) che forniscono un'indicazione della dimensione del sistema in termini funzionali.

Calcolo UFP				
	Complessità			Totale
Funzioni	Bassa	Media	Alta	
External Input (EI)	3*3	8*4	3*6	59
External Output (EO)	2*4	4*5	2*7	42
External Inquiries (EQ)	0*3	3*4	3*6	30
External Interface Files (EIF)	0*5	3*7	3*10	51
Internal Logical Files (ILF)	0*7	1*10	0*15	10
<b>TOTALE UFP</b>				192

### 3.1.1 Calcolo del fattore di aggiustamento

Il **Fattore di Aggiustamento** (Value adjustment factor, **VAF**) è un valore che moltiplicato per il numero di UFP ci permette di ottenere il numero definitivo di questi ultimi, portando così a termine la procedura di conteggio. Il VAF è basato su 14 caratteristiche generali del sistema che spiegano e danno un valore alle funzionalità generali che devono essere contate nell'applicazione.

Ogni caratteristica ha una propria dettagliata descrizione che consente di determinare il grado di influenza che quella caratteristica ha sul progetto; questo grado di influenza varia in una scala da 0 a 5:

- 0: Influenza nulla
- 5: Influenza molto forte

La seguente tabella mostra l'elenco delle 14 caratteristiche considerate nel VAF, con descrizione e valore assegnato per ognuna di esse.

Caratteristiche del sistema	Descrizione	Valore Assegnato
Data Communication	Si valuta in quale misura l'applicazione riceve e trasmette dati attraverso sistemi di comunicazione	5
Distributed Functions	Indica come sono distribuiti i dati e le funzioni di elaborazione all'interno dei confini dell'applicazione	4
Performance Objectives	Indica se all'applicazione sono posti vincoli stringenti su tempi di risposta o <i>throughput</i>	5
Heavily used configuration	Indica se l'applicazione è stata progettata in funzione di una particolare configurazione hardware	1
Transaction rate	Indica se un eventuale alta frequenza di transazioni ha influenzato le fasi di progettazione	4
Online data entry	Indica se l'applicazione fornisce funzioni per l'inserimento da parte dell'utente	5
End-user efficiency	Indica se l'applicazione è progettata e orientata per l'utente finale	5
Online update	Indica se l'applicazione fornisce l'aggiornamento interattivo	4
Complex processing	Indica quanto è complessa l'elaborazione interna	4
Reusability	Indica quanto codice può essere riutilizzato od è stato sfruttato da altre applicazioni	3
Installation ease	Indica quanto può essere complicato il processo d'installazione	1
Operational ease	Indica se l'applicazione minimizza la necessità di attività manuali	1
Multiple sites	Indica se il sistema è progettato per installazioni multiple	4
Facilitate change	Indica la facilità di modifica	2
<b>Total Degree of Influence (TDI)</b>		48

Calcolato il TDI possiamo calcolare il VAF così:

$$VAF = (TDI/100) + 0,65 = (48/100) + 0,65 = 1,13 \quad (2)$$

### 3.1.2 Calcolo degli Adjusted Function Points

In base al fattore di aggiustamento si calcolano gli Adjusted Function Point (AFP), che nel seguito chiameremo solo FP. Il valore totale di FP sarà quindi dato dalla seguente formula:

$$FP = UFP * VAF = 192 * 1,13 = 216,96 \approx 217 \quad (3)$$

Ora possiamo calcolare il fattore *Size* in KSLOC:

$$Size = FP * [(L_1 * BC_1) + ... + (L_n * BC_n)]/1000 \quad (4)$$

Dove:

- **FP**: sono i Function Point (Adjusted);
- **L<sub>1</sub> ... L<sub>n</sub>**: sono le percentuali di uso dei linguaggi di programmazione utilizzati nel progetto;
- **BC<sub>1</sub> ... BC<sub>n</sub>**: sono i corrispettivi Backfiring Coefficients dei linguaggi di programmazione utilizzati nel progetto, descritti in dettaglio [qui](#).

I linguaggi di programmazione che saranno usati sono:

- Java: per il 90% con BC di 53,33;
- Python: per il 5% con BC sempre di 53,33;
- SQL: per il restante 5% con BC di 35.

Quindi:

$$\begin{aligned} Size &= 217 * [(0,95 * 53,33) + (0,05 * 53,33) + (0,05 * 35)]/1000 = \\ &= 217 * (50,66 + 2,67 + 1,75)/1000 = \\ &= 217 * 55,08/1000 = \\ &= 11,952 \approx \mathbf{11,95} \end{aligned} \quad (5)$$

## 3.2 Calcolo del fattore B

Il valore del fattore B è dato dalla seguente formula:

$$B = 0.91 + 0.01 * \sum SD_i \quad (6)$$

Dove:

- SD rappresenta la somma dei valori assegnati agli Scale Drivers. Ogni Scale Driver ha dei livelli di valutazione che possono andare da:
  - 5: Very Low a

– 0: Extra High

La seguente tabella mostra l'elenco dei parametri, una breve descrizione di essi, il valore descrittivo assegnato per questo progetto e il valore numerico associato da 5 a 0.

Parametro	Descrizione	Valore Descrittivo	Valore Associato
Lavori precedenti	Il prodotto è simile a prodotti precedentemente sviluppati	Thoroughly unprecedented	5
Flessibilità di sviluppo	Possibilità di flessibilità durante lo sviluppo	Some relaxation	3
Architettura / Risoluzione rischi	Risoluzione dei rischi e precisione dell'architettura	Often	3
Coesione del Team	Consistenza degli obiettivi degli stakeholders e capacità degli stakeholders di operare con il team	Highly cooperative	1
Maturità del processo	Livello del SW-CMM ( <b>Software Capability Maturity Model</b> ) secondo il Software Engineering Institute	SW-CMM Level 1 (Lower Half)	5
<b>Totale</b>			<b>17</b>

Quindi il valore del fattore B è:

$$B = 0,91 + 0.01 * 17 = \mathbf{1.08} \quad (7)$$

### 3.3 Calcolo del fattore EAF

Il valore del fattore EAF è dato dalla seguente formula:

$$EAF = PERS * RCPX * RUSE * PDIF * PREX * FCIL * SCED \quad (8)$$

La seguente tabella mostra il nome dei vari parametri, una breve descrizione per ciascuno di essi, il livello di valutazione e il valore associato [COCOMO II]:

Parametro	Descrizione	Livello di valutazione	Valore As-sociato
PERS	Capacità del personale	Nominal	1.00
RCPX	Affidabilità e complessità del prodotto	High	1.33
RUSE	Livello di riuso richiesto	Low	0.95
PDIF	Difficoltà della piattaforma	Low	0.87
PREX	Esperienza del personale	Nominal	1.00
FCIL	Strumenti di sviluppo a disposizione della squadra e frequenza di comunicazione nel team	High	0.87
SCED	Richieste di schedulazione	Nominal	1.00

Quindi:

$$\begin{aligned}
 EAF &= PERS * RCPX * RUSE * PDIF * PREX * FCIL * SCED = \\
 &= 1,00 * 1,33 * 0,95 * 0,87 * 1,00 * 0,87 * 1,00 = \\
 &= 0,9563 \approx \mathbf{0,96}
 \end{aligned}
 \tag{9}$$

### 3.4 Totale

Ora che abbiamo calcolato i tre fattori mancanti, possiamo calcolare il valore della formula 1 per avere il totale dello sforzo del progetto misurato in Person-Months.

$$PM = 2,94 * 11,95^{1,08} * 0,96 = 41,1316 \approx \mathbf{41,13}
 \tag{10}$$

## 4. Analisi preliminare del tempo di calendario

Conoscendo lo sforzo in PM, si può stimare il tempo di calendario indicativo (in mesi) con la seguente formula:

$$TDEV = C * PM^{[D+0,2*(B-1,01)]} \quad (11)$$

Dove:

- **TDEV**: è il Time to Develop (in mesi);
- **C, D**: costanti secondo la calibrazione di COCOMO II;
- **PM**: sono i Person-Months.

Quindi:

$$TDEV = 3 * 41,13^{[0,33+0,2*(1,08-1,01)]} = 10.7745 \approx \mathbf{11} \quad (12)$$

## 5. Analisi Preliminari dei Costi

Conoscendo lo sforzo in PM, possiamo stimare il costo dello sviluppo del progetto.

Assumendo che:

- lo stipendio medio di un membro del team di sviluppo sia 2200€/mese;
- i costi degli overheads per lo staff coinvolto (costi di viaggio, ufficio, mensa, etc) siano di circa 20000€;
- i costi dell'architettura fisica sono di 800000€(vedere *Documento di Visione*)

Il costo indicativo totale del progetto sarà:

$$Costo = 41,13 * 2.200 + 20.000 + 800.000 = \mathbf{910.486\text{€}} \quad (13)$$

A cui va aggiunto il costo per l'utilizzo del *Content Delivery Network (CDN)* per la distribuzione dei contenuti multimediali.