Corso di Laurea in Ingegneria Informatica

Corso di Ingegneria del Software

Stima dei costi.

Analisi dei Punti Funzione

Sommario

- Stima dei costi
- Analisi a Punti Funzione
  - Definizione, contesto e obiettivi
  - Indicatori
  - Conteggio
  - Esempio di conteggio
- Modelli per la stima dei costi (cenni)

Riferimenti

C.Ghezzi, M. Jazayeri, D. Mandrioli; *Ingegneria del Software. Fondamenti e principi*, II edizione. Pearson.

Cap. 8 § § 8.1 8.2, 8.3

Stima dei costi (1/2)

- **La stima dei costi è essenziale in fase di pianificazione**
- Per un nuovo prodotto software da realizzare, serve a:
  - Stimare il prezzo del prodotto
  - Dimensionare il team di sviluppo del prodotto
  - ...
- **Per un prodotto esistente, serve a**:
  - Stimare il costo della manutenzione evolutiva
  - Stimare il suo valore (per es. per decidere se effettuarne una manutenzione adattativa o ri-svilupparlo ex novo, a fronte di cambiamenti tecnologici)

Stima dei costi (2/2)

- Malgrado l'importanza della stima del costo, è difficile stimare un valore che si discosta di poco dal costo reale, a causa della difficoltà di misurare i due fattori che maggiormente la influenzano:

  - **produttività del personale coinvolto nella produzione del software**
  - **complessità del software**

Metriche di produttività

- Una metrica di *produttività* ideale dovrebbe misurare la quantità di valore o di funzionalità prodotta (output) per unità di *Effort* (input).
  - *Effort:* quantità di lavoro necessaria a completare un "*Work package*". Normalmente espressa in giorni-uomo o mesi-uomo

- I **punti funzione** sono una metrica per la stima della "quantità di funzionalità" del software,
  - **si svincolano dalla tecnologia usata;**
  - **sono applicabili soprattutto nelle prima fasi del ciclo di sviluppo del software (studio di fattibilità, analisi dei requisiti).**

Function Point Analysis (1/3)

- La Function Point Analysis (analisi a punti funzione) usa i *punti funzione* per quantificare le **funzionalità** del sistema software.
- Si usano il numero di informazioni in entrata, in uscita, e memorizzate nel sistema.
  - Si analizzano le funzionalità già delineate durante la fase di analisi dei requisiti
  - Il conteggio dei FP è un processo iterativo che può essere di supporto per l'analisi dei requisiti
  - I Function Point (FP) sono una metrica "orientata" ai requisiti dettati dal cliente poichè misurano le funzionalità che questi richiede

Function Point Analysis (2/3)

- Non è richiesta la conoscenza del numero di linee di codice per effettuare le stime, sono sufficienti i requisiti funzionali
- Indipendenza dalla tecnologia e dall'*effort* richiesto in fase di sviluppo
- La *function point analysis* può essere adottata anche in fase di manutenzione (software esistente)
- L'analisi è adeguata per sistemi software di tipo gestionale

Function Point Analysis (3/3)

- I Function Point sono stati standardizzati:
  - L'ISO (International Standard Organization) e l'IFPUG (International Function Point User Group) ne hanno promosso l'uso e succesivamente la standardizzazione.
  - **L'IFPUG 4.1 Unadjusted Function Point Method** è stato approvato dall'ISO/IEC Joint Technical Committee 1 (JTC1) ed è divenuto PAS (Publicly Available Specification) nel 2001.
  - Il personale dedito al conteggio dei FP può essere anche certificato.

Conteggio dei Function Point (1/3)

- I Function Point sono conteggiati in due passi consecutivi:

1. **Valutazione degli Unadjusted Function Point (UFP)**

1. **Valutazione dei Fattori di Aggiustamento**

Conteggio dei Function Point (2/3)

1. **Valutazione degli Unadjusted Function Point (UFP)**
   - Gli UFP prendono in considerazione gli aspetti funzionali del sistema e li classificano secondo degli indicatori.
   - Questa attività prevede:
     a. Determinazione del tipo di conteggio
     b. Identificazione dell'ambito di conteggio e dei confini applicativi
     c. Identificazione delle funzioni dati
     d. Identificazione delle funzioni transazionali
     e. Conteggio UFP

Conteggio dei Function Point (3/3)

2. **Valutazione dei Fattori di Aggiustamento**
   - Il risultato che si ottiene con la valutazione degli UFP è raffinato con l'applicazione dei Fattori di Aggiustamento.
   - Attraverso dei fattori correttivi si adegua il valore ottenuto dagli UFP al sistema che si sta progettando.
   - Al termine di questa attività si ottiene il conteggio finale.

- Sia la la tecnica per il conteggio degli UFP, sia i fattori di aggiustamento, sono frutto di considerazioni empiriche

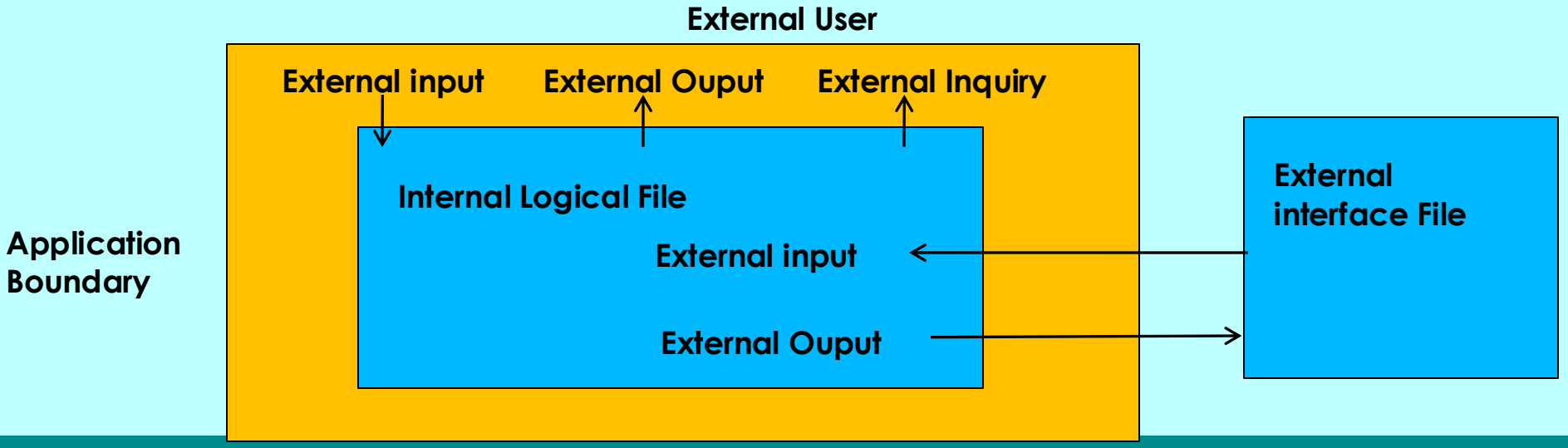
Determinazione del tipo di conteggio

- Il tipo di conteggio è legato allo stato in cui si trova il sistema
  - **Conteggio per sviluppo di progetto**
    - Misura le funzionalità da fornire agli utenti finali di un software da realizzare *ex novo* (più un eventuale conversione dei dati dalla vecchia applicazione).
  - **Conteggio per manutenzione evolutiva**
    - Misura le modifiche ad un'applicazione esistente e comprende le funzionalità fornite dagli utenti con l'aggiunta di nuove funzioni, cancellazione di vecchie funzioni e variazioni di funzioni esistenti.
  - **Conteggio applicativo**
    - Misura un'applicazione già installata. Questo conteggio, detto anche *baseline*, valuta le funzionalità fornite.

Confine applicativo

- Il confine applicativo identifica le "linee di divisione" fra:
  - **l'utente e l'applicazione che si vuole misurare**
  - **l'applicazione che si vuole misurare e eventuali altre applicazioni.**

- Il confine è individuato tenendo conto della percezione che ha l'utente delle funzionalià offerta dal sistema. Quindi il confine è legato agli aspetti di business piuttosto che a considerazioni tecnologiche (difficilmente comprese dall'utente).



Identificazione delle funzioni dati

- Per l'analisi dei FP si usano indici di due tipi: Dati e Transazioni.
- Dati
  - **Internal Logical File (ILF)** - aggregati logici di dati generati, usati e gestiti internamente dal sistema identificabili dal cliente. Esempi tipici sono: tabelle di un database, file di configurazione, file di errore, password mantenuti dal sistema. I file temporanei non rientrano in questo tipo perchè non visibili o accessibili dall'utente.
  - **External Interface File (EIF)** - aggregati logici di dati scambiati o condivisi con altre applicazioni identificabili dall'utente. Un EIF contato per un'applicazione deve essere un ILF per un'altra applicazione.

Identificazione delle funzioni transazionali (1/3)

- **Transazioni**
  - **External Input (EI)** – Informazioni distinte fornite dall'utente o da altre parti del sistema o da altre applicazioni, usate come dati di ingresso.
  - Le informazioni di ingresso possono alterare un ILF oppure agire sul funzionamento del sistema.
  - Esempi: dati transazionali (una vendita, un appuntamento inserito), messaggi di altre applicazioni, ingressi che portano informazioni di controllo (un sensore).
  - Non sono da considerare EI schemi di menu che sono usati per navigare ma non agiscono su un ILF, richieste di input che sono parte di una query, meccanismi per il refresh della schermata video.

Identificazione delle funzioni transazionali (2/3)

- **External Output (EO)** – output distinti che il sistema restituisce all'utente come risultato delle proprie elaborazioni.

  L'intento dell'EO è di presentare dati all'utente attraverso una logica elaborativa diversa, o aggiuntiva del semplice recupero di dati (da un ILF) o di informazioni di controllo. Pertanto la logica elaborativa deve contenere una formula o un procedimento di calcolo e creare dei dati derivati.

  Esempi sono: report su terminale o stampante che richiedono l'uso di algoritmi o calcoli, trasferimenti di file o dati da altre applicazioni verso l'utente, documenti di varia natura con calcoli (fatture, bollette, assegni, ...).

  Non sono da considerare EO: messaggi di conferma o errore, report generati da un semplice query, più modi di invocare lo stesso processo di output.

Identificazione delle funzioni transazionali (3/3)

- **External Inquiry (EQ)** – interrogazioni in linea che producono una risposta immediata del sistema.

L'intento primario è di presentare l'informazione all'utente attraverso il recupero da un ILF o EIF senza effettuare elaborazioni nè modifiche ad un ILF o EIF.

Esempi: dati prelevati da uno o più ILF/EIF e visualizzati, funzioni utente per la ricerca, file inviati ad altre applicazioni che non prevedono conteggi o contengono formule.

Non sono External Inquiry: schermate di menu per navigazione o selezione, messaggi di errore o conferma

Conteggio UFP (1/2)

- Il **conteggio degli Unadjusted Function Point** richiede che si contino:
  - **Numero degli ILF (NIFL)**
  - **Numero degli EIF (NEIF)**
  - **Numero degli EI (NEI)**
  - **Numero degli EO (NEO)**
  - **Numero degli EQ (NEQ)**

Conteggio UFP (2/2)

- **I valori precedentemente conteggiati sono pesati secondo la tabella seguente**, dove VPi sono i valori pesati per ciascun indice, ottenuti moltiplicando il campo Valore (cioè il numero fornito dal progettista relativo all'indice considerato) con uno dei tre pesi indicati, che variano a seconda che l'applicazione sia considerata semplice, media o complessa.

$$UFP = \sum_{i=1}^{5} V_i \cdot P_i$$

INDICE		PESI			VPI
	VALORE	SEMPLICE	MEDIO	COMPLESSO	
NIFL	V1	3	4	6	V1*P1
NEIF	V2	4	5	7	V2*P2
NEI	V3	3	4	6	V3*P3
NEO	V4	7	10	15	V4*P4
NEQ	V5	5	7	10	V5 *P5

Fattori correttivi

- Il conteggio UFP produce un valore che può essere raffinato secondo 14 **fattori correttivi** che tengono in conto la "complessità" del sistema; il loro valore varia da 0 a 5:
  - **0 ininfluente**
  - **1 incidenza scarsa**
  - **2 incidenza moderata**
  - **3 incidenza media**
  - **4 incidenza significativa**
  - **5 incidenza essenziale**

- L'assegnazione del grado di incidenza avviene secondo l'esperienza di chi effettua il conteggio

CARATTERISTICHE GENERALI	VALORE
COMUNICAZIONE DATI	
DISTRIBUZIONE ELABORAZIONE	
PRESTAZIONI	
UTILIZZO INTENSIVO CONFIGURAZIONE	
FREQUENZA DELLE TRANSAZIONI	
INSERIMENTO DATI INTERATTIVO	
EFFICIENZA PER L'UTENTE FINALE	
AGGIORNAMENTO INTERATTIVO	
COMPLESSITA' ELABORATIVA	
RIUSABILITA'	
FACILITA' INSTALLAZIONE	
FACILITA' GESTIONE OPERATIVA	
MOLTEPLICITà DI SITI	
FACILITA' DI MODIFICA	

Fattori correttivi: Prestazioni

- Prestazioni: misura gli obiettivi prestazionali dell'applicazione, richiesti o approvati dall'utente, sia in termine di tempo di risposta che di elaborazioni
  - **0 l'utente non ha specificato particolari requisiti di prestazioni**
  - **1 l'utente ha specificato i livelli di prestazioni nei requisiti, ma tali livelli non hanno richiesto attenzione speciale durante lo sviluppo**
  - **2 i tempi di risposta sono ciritici durante le ore di picco. Nessuno studio speciale per la CPU è stato richiesto. La scadenza di elaborazione è giornaliera**
  - **3 i tempi di risposta sono critici durante l'intero arco della giornata. Nessuno studio speciale per il calcolo della CPU è stato richiesto**
  - **4 le prestazioni dell'utente sono stringenti da richiedere una attività di analisi delle prestazioni durante il disegno**
  - **5 l'ottimizzazione delle prestazioni riveste estrema criticità; strumenti per la valutazione delle prestazioni devono essere utiizzati in fase di progettazione, sviluppo ed installazione**

Fattori correttivi: Efficienza per l'utente finale

- Esprime il grado di considerazione per i fattori umani e la facilità di utilizzo per l'utente dell'applicazione. Le funzioni online sono state progettate evidenziando *user-friendliness*, ed utilizzando tecniche quali: help e documentazione on line, menu, scrolling, uso del mouse, ecc.
  - **0 nessuno degli elementi indicati è stato preso in considerazione**
  - **1 uno, due o tre degli elementi indicati sono stati presi in considerazione**
  - **2 quattro o cinque degli elementi indicati sono stati presi in considerazione**
  - **3 sei o più degli elementi indicati sono stati presi in considerazione, ma non esistono richieste specifiche da parte degli utenti relativamente all'efficienza**
  - **4 sei o più degli elementi indicati sono stati presi in considerazione, gli aspetti legati ad un uso efficiente da parte dell' utente richiedono attenzione in fase di progettazione**
  - **5 sei o più degli elementi indicati sono stati presi in considerazione, l'interfacci utente gioca un ruolo determinante**

Fattori correttivi: Riusabilità

- L'applicazione e il codice dell'applicazione sono appositamente progettati per essere utilizzati in altre applicazioni.
  - **0 il codice non è riusabile**
  - **1 viene usato del codice riusabile nell'applicazione**
  - **2 meno del 10% dei moduli prodotti considerano i bisogni di più di un utente**
  - **3 più del 10% dei moduli prodotti considerano i bisogni di più di un utente**
  - **4 l'applicazione è stata realizzata e documentata in modo tale da facilitarne il riuso, la personalizzazione verso l'utente è solo a livello codice sorgente**
  - **5 l'applicazione è progettata e sviluppata per diventare generalizzata e usabile in contesti diversi**

Stima finale del conteggio

- **Il valore finale del conteggio FP è il prodotto fra il conteggio UFP e l'aliquota introdotta dai fattori correttivi**.

$$FP = UFP \times \left(0.65 + 0.01 \times \sum_{i=1}^{14} F_i \right)$$

- Cosa è possibile misurare con i FP?
  - Costo del progetto conoscendo il costo di un FP.
  - Produttività dei programmatori, ad esempio numero di FP per unità di tempo sviluppato da un programmatore.
  - Dimensione in linee di codice per lo sviluppo di un FP, FP/LoC

Dai Function Point alle LOC

- Alcuni studi hanno stimato il numero di linee di codice *logiche* necessarie per codificare un punto funzione nei vari linguaggi.
  - Il valore LLOC/FP è un valore medio.

LINGUAGGIO	LLOC/FP
Assembler	119
C	97
Cobol	61
C++	50
Java	53
VB.NET	52
HTML	34
...	...

Modelli per la stima dei costi

- I modelli per la stima dei costi del software mirano a valutare:
  - La durata del processo di sviluppo
  - Lo sforzo (*effort*) richiesto
  - Il numero di persone necessarie (progettisti, programmatori,...)
- **Il costo è calcolato a partire dalla stima delle linee di codice** del software che si desidera sviluppare. Il numero di linee di codice è stimato facendo riferimento a progetti passati.
  - Sebbene la stima del numero di linee di codice non sia idonea per misuare la produttività del software è, allo stesso tempo, una metrica semplice.

Modelli per la stima dei costi

- La stima dei costi si articola in generale in tre passi:

  1. **Stima della dimensione del software**.
  2. **Calibrazione della stima**
  3. **Applicazione degli strumenti del modello alla stima per determinare le grandezze di interesse** (sforzo totale, durata, distribuzione attività, etc.).

COnstructive COst MOdel - COCOMO

- La formula generale per il calcolo dello sforzo nominale è:

$$PM = c \times KLOC^{L}$$

  - PM, people-month
  - KLOC, migliaia di linee codice
  - *c* e *L* sono costanti dipendenti dal tipo di modello
- **La formula è calibrata al processo software** mediante dei fattori correttivi, classificabili per:
  - **Prodotto**, requisiti di affidabilità o di complessità dell'applicazione
  - **Computer,** esistenza di vincoli sulle memoria occupata e tempi di esecuzione
  - **Personale,** l'esperienza delle persone nell'area applicativa e sulle tecnlogie usate
  - **Progetto**, disponibilità di strumenti di sviluppo sofisticati

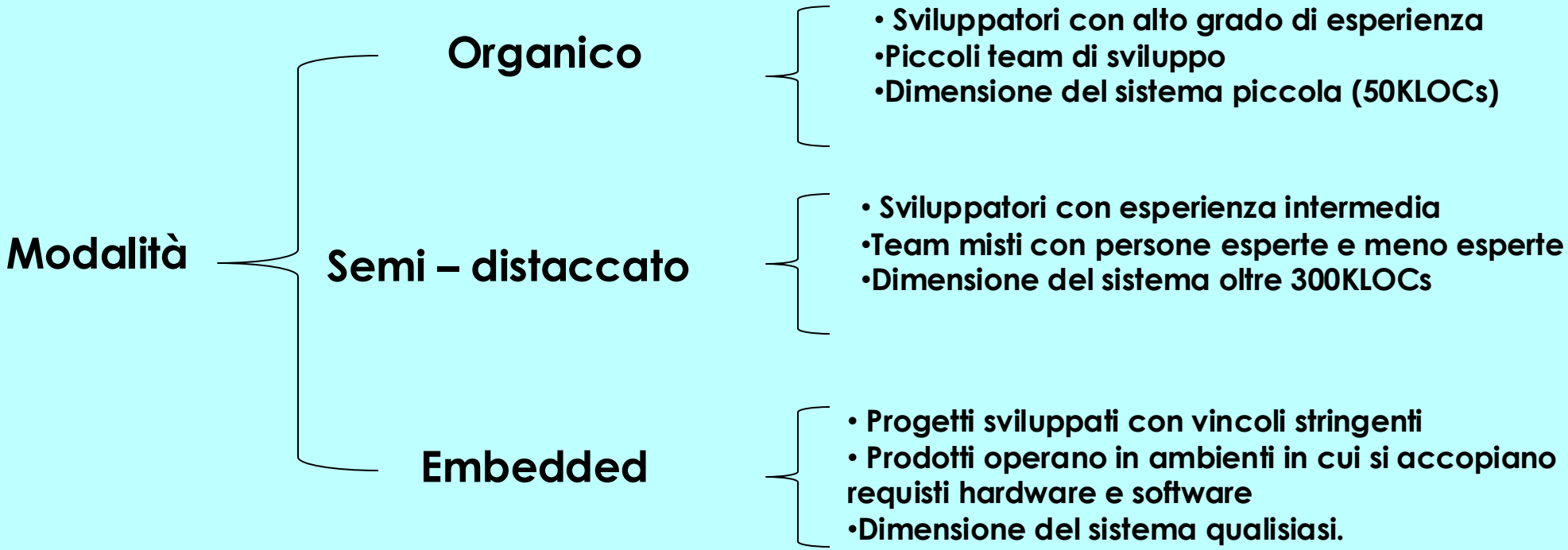
COCOMO

- Il COnstructive COst MOdel (COCOMO) è un modello per la stima dei costi, evoluto, successivamente, in COCOMO II.
- Il COCOMO assume che il ciclo di sviluppo sia a cascata
- Originariamente il COCOMO prevedeva tre diversi livelli di dettaglio:
  - **Basic**, utile per una veloce stima dei costi, ma approsimato
  - **Intermediate**, aggiunge alla stima effettuata dal basic dei fattori correttivi (Cost Drivers)
  - **Detail,** stima i costi alla stessa maniera dell'intermediate ma si sofferma su ogni fase del ciclo di sviluppo del software (analisi, progettazione,...)

COCOMO Intermediate (1/4)

- Nel modello COCOMO Intermediate i tre passi della stima dei costi si specializzano come segue:

  1a. Scelta della modalità di sviluppo. Il COCOMO definisce tre modalità di sviluppo (**organico, semi-distaccato, embedded**).



COCOMO Intermediate (2/4)

1b. Determinazione dello sforzo nominale, sulla base della modalità. Ad ogni modalità di sviluppo corrisponde nel COCOMO Intermediate una formula per il calcolo dei costi. Lo sforzo nominale è determinato in base ai valori della tabella seguente.

Modalità di sviluppo	Sforzo nominale	Pianificazione
Organico	$(PM)_{NOM} = 3.2(KDSI)^{1.05}$	$TDEV = 2.5(PM_{DEV})^{0.38}$
Semi-distaccato	$(PM)_{NOM} = 3.0(KDSI)^{1.12}$	$TDEV = 2.5(PM_{DEV})^{0.35}$
Embedded	$(PM)_{NOM} = 2.8(KDSI)^{1.20}$	$TDEV = 2.5(PM_{DEV})^{0.32}$

- TDEV = time for development

COCOMO Intermediate (3/4)

	Stime					
Fattori di costo	Molto basso	Basso	Normale	Alto	Molto alto	Estrem. alto
Attributi del prodotto						
Affidabilità richiesta del software	.75	.88	1.00	1.15	1.40	
Dimensione del database		.94	1.00	1.08	1.16	
Complessità del prodotto	.70	.85	1.00	1.15	1.30	1.65
Attributi del computer						
Vincoli di tempo d'esecuzione			1.00	1.11	1.30	1.66
Vincoli di memoria centrale						
Volatilità della piattaforma		.87	1.00	1.15	1.30	
Tempo di risposta del computer		.87	1.00	1.07	1.15	
Attributi del personale						
Capacità dell'analista	1.46	1.19	1.00	.86	. 71	
Esperienza con le applicazioni	1.29	1.13	1.00	.91	. 82	
Capacità del programmatore	1.42	1.17	1.00	.86	. 70	
Esperienza con la macchina virtuale*	1.21	1.10	1.00	.90		
Esperienza con il linguaggio di programmazione	1.14	1.07	1.00	.95		
Attributi del progetto						
Uso di pratiche moderne di programmazione	1.24	1.10	1.00	.91	. 82	
Uso di strumenti software	1.24	1.10	1.00	.91	. 83	
Pianificazione dello sviluppo	1.23	1.08	1.00	1.04	1.10	

2. **Correzione della stima.** Il valore PM ottenuto al passo 1 viene corretto per tener conto di fattori quali l'esperienza dei programmatori, e l'affidabilità richiesta.

I fattori correttivi sono riportati in tabella; per ognuno si sceglie un valore e si moltiplicano fra loro. Il prodotto ottenuto è moltiplicato per il valore PM calcolato in precedenza.

COCOMO Intermediate (4/4)

3. Applicazione degli strumenti del modello alla stima per determinare le grandezze di interesse.

Ad es., il calcolo del tempo di sviluppo (TDEV) si ottiene in funzione della modalità secondo le formule riportate nella terza colonna della tabella sottostante, in cui PM_{DEV} è calcolato a partire da PM_{NOM} applicando i fattori correttivi ricavati al passo 2.

Modalità di sviluppo	Sforzo nominale	Pianificazione
Organico	$(PM)_{NOM} = 3.2(KDSI)^{1.05}$	$TDEV = 2.5(PM_{DEV})^{0.38}$
Semi-distaccato	$(PM)_{NOM} = 3.0(KDSI)^{1.12}$	$TDEV = 2.5(PM_{DEV})^{0.35}$
Embedded	$(PM)_{NOM} = 2.8(KDSI)^{1.20}$	$TDEV = 2.5(PM_{DEV})^{0.32}$

COCOMO: considerazioni

- Il COCOMO è un modello empirico nato dagli studi effetuati negli anni 80 da Boehm
- È possibile ricalcolare rapidamente il valore della stima variando il valore dei fattori correttivi, ad esempio si può valutare l'incremento di costo che si avrebbe se si impiegassero dei programmatori molto esperti.
- Non fornisce una stima accurata dei costi, ma rimane comunque uno strumento valido per la stima dei costi che non può essere legata alla sola esperienza del project manager
- È necessaria una stima delle linee di codice per la stima dei costi.
- Un modello legato al ciclo di vita a cascata