

COCOMO II

Note basate su

[COCOMO II Model Definition Manual - Center for Software Engineering USC](#)

[COCOMO II \(slide\), Annalisa Fasolino, Univ, Napoli Federico II](#)
e altre fonti.

Vedere anche lo strumento:

<http://csse.usc.edu/tools/COCOMOII.php>

COCOMO II

1. COCOMO II
2. Scale drivers e Cost drivers
3. Modelli di pre-architettura
 1. Modello di progettazione iniziale
 2. Modello di composizione delle applicazioni
 3. Tabella per la valutazione della produttività
 4. Moltiplicatori
 5. Modello di riuso
4. Modello di post-architettura
5. Tabelle
6. Esempio
7. Valutazioni finali

Cost Drivers

Gli attributi correttivi, già 15 in COCOMO 81, qui diventano **17** e sono detti *Cost drivers*.

Anch'essi verranno prima misurati in una scala ordinale e poi trasformati in coefficienti reali, $C_{j=1..17}$.

Scale Drivers

In COCOMO II, alcuni ulteriori importanti fattori che contribuiscono alla durata e al costo di un progetto sono gli “**Scale Drivers**”.

Ciascuno di questi va posto a un valore adeguato al progetto da sviluppare; essi determinano i parametri da utilizzare nella equazione dello Sforzo:

Scale Drivers

1. Precedentedness
2. Development Flexibility
3. Architecture / Risk Resolution
4. Team Cohesion
5. Process Maturity

Scale Drivers vs. Development Modes

- Precedentedness
- Development Flexibility

nei fatti producono la stessa influenza
che in COCOMO 81 hanno i 3

*Development Mode: Organic, Semi-
detached ed Embedded.*

Modelli di COCOMO II

COCOMO II comprende modelli applicabili al variare dello stadio del progetto (analisi o pre-architettura e post-architettura) e delle modalità di realizzazione del software.

Ciò consente di ottenere stime adeguatamente dettagliate e affidabili.

Modelli di COCOMO II

I modelli compresi da COCOMO II sono:

- **Modelli di pre-architettura**
 - **Modello di progettazione iniziale.** Utile per ottenere stime in fase di analisi di sviluppi da zero.
 - **Modello di composizione delle applicazioni.** Utile per la stima di sistemi ottenuti a partire da componenti (riusabili).
 - **Modelli di riutilizzo.** Utili per stimare il costo relativo all'integrazione nei diversi tipi di riuso.
- **Modelli di post-architettura.** Modelli utilizzabili a valle della definizione dell'architettura per ottenere stime più affidabili.

Modello di progettazione iniziale

Lo si applica quando i requisiti utente sono stati definiti e la progettazione viene iniziata.

È spesso utilizzato per confrontare diverse soluzioni progettuali possibili.

$$M = a * EAF * S^b$$

- a si ottiene similmente all'omonimo coefficiente di COCOMO 81
- S è misurato in KSLOC (tipicamente S è calcolato partendo da FP o AP)
- b varia tra 1.1 e 1.24 in base ad una serie di fattori quali la **innovatività del progetto**, la **flessibilità**, le **tecniche di gestione del rischio**, la **maturità del processo** ...
- EAF è lo Effort Adjustment Factor:
 $EAF = PERS \times RCPX \times RUSE \times PDIF \times PREX \times FCIL \times SCED$

Moltiplicatori

I moltiplicatori che concorrono a EAF riflettono vari aspetti:

- PERS: capacità del personale
- RCPX: complessità del prodotto
- RUSE: livello di riuso
- PDIF: difficoltà legate alla piattaforma utilizzata
- PREX: esperienza del personale
- FCIL: funzionalità di supporto (Facility)
- SCED: rigidità dello schedulo

Ogni moltiplicatore è valutato su una scala da 1 (molto basso) a 6 (molto alto).

Esempio

Mediamente $a = 2,94$.

Un progetto con *Cost Drivers* e *Scale drivers* tutti nominali avrebbe fattore di aggiustamento dello sforzo $EAF=1.0$ ed esponente $b = 1.0997$.

Per 8000 linee di codice, COCOMO II dovrebbe stimare: 29 Persone Mese.

$$M = 2.94 * 1.0 * 8^{1.0997}$$

% Esempio

$$T = 3.67 * M^{SE}$$

SE è la equazione di schedulo ad esponente derivata da 5 Scale Drivers

Ottenuto **SE** = 0.3179, si arriva a quanto segue:

$$T = 3.67 * (42.3)^{0.3179} = 12.1 \text{ months}$$

Numero medio di persone **mP** = (42.3 PersoneMese) / (12.1 Mesi) = 3.5 persone

SCED

Trattasi di un *cost driver* che significa “*accelerated from the nominal schedule*”. È usato per portare in conto il fatto che un progetto sviluppato in modo *accelerato* richiede maggiore sforzo che nello schedulo ottimo.

Uno SCED = VERY LOW corrisponde, in COCOMO II (2000), a un moltiplicatore di Sforzo pari a 1.43; ciò significa che si intende terminare il progetto nel 75% del tempo ottimo stimato.

% Esempio

Continuando l'esempio precedente, ma assumendo SCED= VERY LOW, COCOMO produce le seguenti stime:

$$T = 75\% * 12.1 \text{ Mesi} = 9.1 \text{ Mesi}$$

$$EAF = 1.34 * 1.09 * 1.43 = 2.09$$

$$M = 2.94 * (2.09) * (8)^{1.0997} = 60.4 \text{ PersoneMese}$$

$$mP = (60.4 \text{ PersoneMese}) / (9.1 \text{ Mesi}) = 6.7 \text{ persone}$$

Nota che il calcolo di T non è basato direttamente sullo Sforzo ma sulla Schedulo nominale.

Modello di composizione delle applicazioni

È un modello utile per stimare i costi di realizzazione di un prototipo o di realizzazione di un software a partire da componenti esistenti riutilizzabili, **senza** contare il contributo di queste ultime.

$$M = (AP * (1 - reuse)) / Prod$$

- *M* è lo sforzo in mesi-uomo,
- *AP* il numero di Application Point,
- *reuse* è la percentuale di riuso.
- *Prod* la produttività

Tabella per la valutazione della % di produttività

Esperienza e Capacità degli sviluppatori	Molto bassa	Bassa	Normale	Alta	Molto Alta
Maturità e Capacità degli strumenti CASE impiegati	Molto bassa	Bassa	Normale	Alta	Molto Alta
Produttività (NOP/ mese)	4	7	13	25	50

Modelli di riuso

Considerano il riuso di codice senza cambiamenti (black-box reuse) e il riuso di codice che deve essere adattato per integrarlo con nuovo codice (white-box reuse).

Ci sono due versioni del modello di stima:

- **Black-box reuse model.** Si effettua una stima dello sforzo (M) di integrazione del codice riusato.
- **White-box reuse model.** Si effettua una stima del numero equivalente di linee di nuovo codice (a partire dal numero di linee riusate). Tale valore viene poi aggiunto a quello delle linee generate ex novo.

Modello di riuso. **Per codice generato automaticamente**

Detto M lo sforzo di integrazione:

$$M_{autom} = TS * automGen/integProd$$

- ***TS***: numero di linee di codice complessivo;
- ***automGenS***: percentuale di codice automaticamente generato;
- ***integProd***: produttività degli ingegneri che si occupano dell'integrazione del codice (tipico: 2400 istruzioni al mese a persona).

Per codice modificato e riutilizzato

$$ES = ARS * (1 - automGenS) * AAM$$

- **ES**: numero equivalente di righe di nuovo codice (da utilizzare nelle formule che richiedono il numero di linee di codice generato ex novo);
- **ARS**: numero di linee di codice riutilizzato (ma da adattare);
- **automGenS**: percentuale di codice automaticamente generato;
- **AAM**: coefficiente che indica la difficoltà di adattamento del codice. É la somma di **AAF** (costo per effettuare le modifiche), **SU** (costo per comprendere il codice esistente), **AA** (costo legato alle decisioni sul riutilizzo eventuale).
In pratica, si paragona lo sforzo di adattamento del codice riusato ad uno sforzo di scrittura di nuovo codice.

Modello di Post-Architettura (P.A.)

Viene utilizzato quando è già disponibile un progetto architettonico iniziale ed è conosciuta la struttura dei sottosistemi.

Viene riutilizzata la stessa formula del modello di composizione:

$$M = a * EAF * S^b$$

Stavolta però, conoscendosi l'architettura, è possibile effettuare una migliore stima di S.

Modello di Post-Architettura (P.A.)

- La **dimensione del codice (S)** è stimata (per ogni sottosistema) sulla base della somma di 3 componenti:
 - Numero di linee di codice da sviluppare da zero
 - Stima di numero di linee di codice equivalente calcolate tramite il modello di riuso
 - Stima del numero di linee di codice che devono essere adattate ai requisiti specifici dell'applicazione.

Modello di Post-Architettura (P.A.)

- L'esponente ***b*** invece varia in modo continuo (a partire da 1.01) e dipende da: 5 fattori di scala (*Scale drivers*, valutati in una scala tra 0 e 5 - da extra alto a molto basso); in COCOMO 81 ***b*** assumeva solo 3 possibili valori per modello.
- Il coefficiente ***EAF*** dipende da 17 attributi.

Fattori di scala (*Scale drivers*) in COCOMO II

Precedenti	Indica l'esperienza precedente dell'organizzazione con questo tipo di progetto (da bassa a alta)
Flessibilità di sviluppo	Indica se si usa un processo prefissato col cliente, oppure c'è libertà di scelta.
Architettura/ Risoluzione dei rischi	Indica se viene eseguita o meno una accurata analisi dei rischi
Coesione del Team	Indica il livello di conoscenza e familiarità del team di sviluppo
Maturità del Processo	Indica il grado di maturità del processo adottato (secondo CMM)

Moltiplicatori (*Cost drivers*) in COCOMO II

- **Attributi di prodotto**
 - RELY, Affidabilità richiesta
 - CPLX, Complessità dei moduli
 - DOCU, documentazione
 - DATA, dimensione del database
 - RUSE, percentuale di componenti riusabili
- **Attributi informatici**
 - TIME, tempo
 - PVOL, precarietà della piattaforma di sviluppo
 - STOR, vincoli di memoria
- **Attributi di personale**
 - ACAP, capacità degli analisti
 - PCON, continuità del personale
 - PCAP, capacità dei programmatori
 - PEXP, esperienza dei programmatori
 - AEXP, esperienza degli analisti
 - LTEX, esperienza del linguaggio e degli strumenti
- **Attributi di progetto**
 - TOOL, utilizzo di strumenti
 - SCED, concentrazione della pianificazione di sviluppo
 - SITE, qualità delle comunicazioni tra i siti di sviluppo

(Esempio di) Effetti dei Cost drivers sulla stima

Valore dell'esponente	1.17
Dimensione del sistema	128,000 DSI
Stima Iniziale COCOMO senza fattori di costo	730 mesi-persona

Affidabilità	Molto alta	1.39
Complessità	Molto alta	1.3
Vincoli di memoria	Alti	1.21
Uso di tool CASE	Bassa	1.12
Schedulazione	Accelerata	1.29
Stima di COCOMO corretta	2306 mesi-persona	

Affidabilità	Molto bassa	0.75
Complessità	Molto bassa	0.75
Vincoli di memoria	Nessuno	1
Uso di tool CASE	Molto alta	0.72
Schedulazione	Normale	1
Stima di COCOMO corretta	295 mesi-persona	

Qualche osservazione

COCOMO 81 e COCOMO II sono stati proposti sulla base di esperienze e dati storici raccolti dagli autori del modello presso specifica/che organizzazioni.

Il modello dovrebbe dunque essere tarato prima dell'uso in altre organizzazioni.

Prevedono molti attributi la cui valutazione è spesso difficile e imprecisa.

Anche per gli attributi sarebbe necessaria una taratura in base a dati storici di progetto.

Ci è utile COCOMO?

Nella pratica è difficile che le aziende posseggano tali dati. Chi può permetterselo, paga un esperto di COCOMO per adattare il modello ed usarlo nella propria realtà.

Committente vs. Produttore? Ruolo del pubblico.