

**Metriche e modelli di stima
dei costi del software**

Indice

Metriche e modelli di stima dei costi del software	3
Il processo di stima dei costi del software	3
Stima dei costi e gestione del software	3
Incertezza della stima dei costi del software	4
Approcci alla stima dei costi del software	5
Sistema di misure per la stima dei costi del software	5
Metriche della stima dei costi del software	5
Tassonomia delle metriche di stima dei costi del software	6
Dimensione del prodotto	6
Produttività	6
Sforzo e tempo di sviluppo	7
Tassonomia di modelli algoritmici di stima dei costi del software	7
Criteri di valutazione di modelli di stima	7
Modelli statistici	8
Modelli storico-empirici	8
Modelli teorici	9
Modelli composti di stima dei costi del software	9
Modelli COCOMO ('81)	9
La metodologia dei Punti Funzione	10
Altri modelli e metodologie	11
Approfondimenti	11
Altre risorse per approfondimenti	11
Note	12
Bibliografia	12

Metriche e modelli di stima dei costi del software

You can't control what you can't measure. 1 (DeMarco 1986)

Scopo della lezione



- inquadramento del processo di stima dei costi nel processo produttivo del software
- analisi delle principali metriche di interesse alla stima dei costi del software (SCS)
- tassonomia e inquadramento storico dei principali modelli di SCS
- introduzione ai modelli composti di SCS più attualmente in uso (COCOMO + Punti Funzione)

Il processo di stima dei costi del software

Stima dei costi del software (SCS)



comprensione e controllo

- comprensione dei costi  miglior controllo
- *controllo*
 -  *maturità* del processo
 -  *qualità* del processo e del prodotto
- **comprensione dei costi del software:**
 - **approccio a scatola nera**
individuazione dei pesi relativi dei *fattori di costo*
 - **approccio a scatola trasparente**
distribuzione dei costi fra le risorse, attività, fasi, etc.
- **entrambi gli approcci sono utili: complementari**

Stima dei costi e gestione del software



importanza di una buona SCS: imprescindibile per decidere, pianificare, difendere, controllare il processo di sviluppo

- Sinergia di SCS e gestione del processo di sviluppo del software: processi dinamici



Dinamica dell'interazione fra SCS e gestione SW

- Produttività di sviluppo o del ciclo di vita?

Incertezza della stima dei costi del software



Quanto può essere incerta la stima dei costi del software?

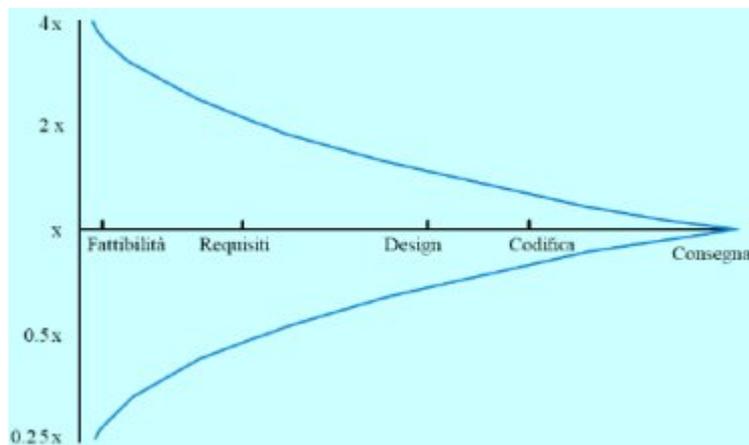


Figura 11.1: Riduzione d'incertezza della stima (Boehm et al. 1995)

Difficoltà inerenti alla stima dei costi del software

Conseguenze di errori di stima

- per eccesso (di cautela: sovrastima dei costi)
occasioni mancate, perdita di *quote di mercato*
- per difetto (di cautela = eccesso di "ottimismo")
(sottostima di costi e tempi: fenomeno molto più diffuso)
 - conseguenze economiche
 - conseguenze tecniche
 - conseguenze manageriali: legge di Brooks
"... poiché il processo di sviluppo del software è complesso, aggiungere personale allo staff oltre un certo numero fa aumentare, anziché diminuire, la durata del progetto".

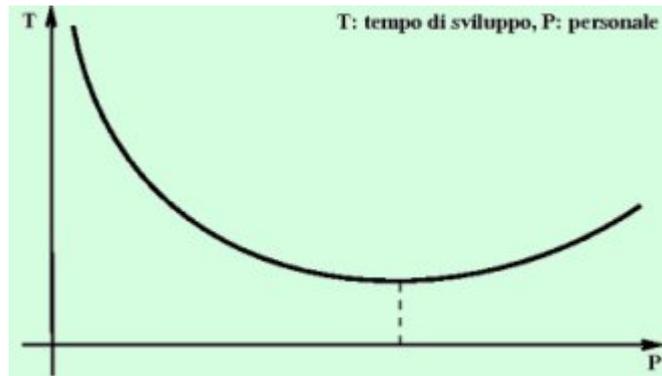


Figura 11.2: Legge di Brooks (Brooks 1975)

Approcci alla stima dei costi del software

- **ricorso al giudizio di esperti**
per una migliore oggettività: *metodo Delphi*
- **stima per analogia**
punti deboli:
 - scarsa sensibilità agli aspetti *nuovi* del progetto
 - presuppone l'esistenza di database di progetti *analoghi*
- **uso di modelli algoritmici**
stima in funzione di *cost driver*: dipende dall'accuratezza della loro stima, è poco sensibile ad aspetti nuovi, *tuttavia* è più oggettiva, ripetibile, e meglio suscettibile di *calibrazione*

Sistema di misure per la stima dei costi del software

Parte integrante di un sistema di misure per il controllo di qualità, del processo e del prodotto, v. il modello [ISO 9126](#)

Aspetti salienti della costruzione di un sistema di gestione della misura per la SCS

- Valore del sistema di gestione della misura
- Costo del sistema di gestione della misura
- Profilo del personale per la gestione della misura, competenze richieste
- Posizionamento della gestione della misura nell'organizzazione produttiva
- Un piano operativo per la gestione di misure

Metriche della stima dei costi del software



mattoni nella costruzione dei modelli di stima

importanza delle metriche per la stima dei costi:

affidabilità predittiva dei modelli di stima

- ipotesi:
 - **esistenza di relazioni fra metriche di fasi diverse del ciclo di vita**
 - **dette relazioni sono formalizzabili in modelli teorico-empirici**
- **scostamento stima - misura** ➔ **correzione e calibrazione dei modelli di stima**

Tassonomia delle metriche di stima dei costi del software

- **metriche di input:**
 - **metriche di base: I (SLOC, KDSI, ...)**
 - **metriche calcolate: FP, SSM**
- **metriche di output: T, S**
- **metriche di elaborazione:**
complessità ciclomatica (McCabe), ...

Dimensione del prodotto



I (SLOC, KDSI): modalità di conteggio

tre gruppi ortogonali di opzioni:

- **a livello di linea**
linee fisiche , linee logiche
- **a livello di programma**
linee eseguibili , + def. dati , + commenti , + JCL
- **a livello di progetto**
*linee nuove , + modificate , + riusate ,
+ codice temporaneo , + codice di supporto*

Produttività



P, "equazione dello sforzo": $S = I/P$

- **produttività di che cosa?**
 - **quantità di SLOC:**
metrica concreta, concetto *erroneo* di produttività
 - **quantità di "funzionalità", Function Points (FP):**
metrica astratta, concetto *corretto* di produttività (*Allen J. Albrecht, 1979*)
- **concetto economico di produttività:**
quantità di **bene o servizio** prodotto per unità di risorsa (tempo · persona)

Sforzo e tempo di sviluppo

- sforzo S (mesi - persona) e tempo di sviluppo T (mesi) correlati alla quantità media di personale FSP (Full-time Software Personnel)
- "curva di lavoro" $FSP(t) \cong$ curva di Rayleigh :

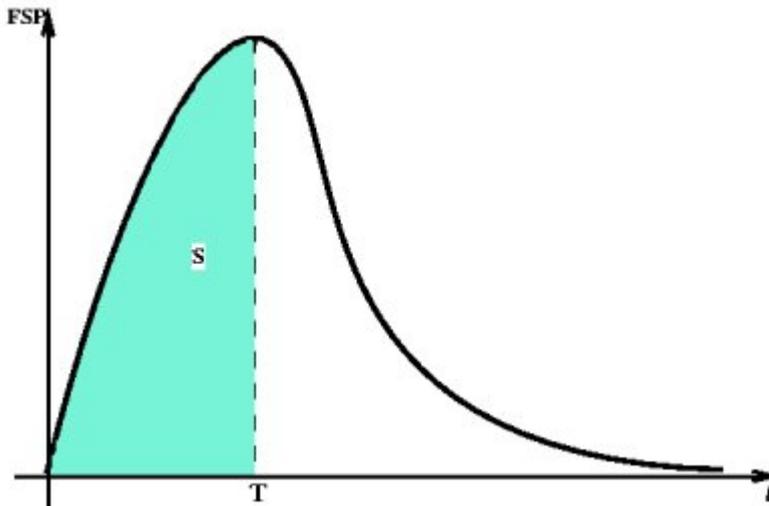


Figura 11.3: Curva di Rayleigh

Tassonomia di modelli algoritmici di stima dei costi del software

- Criteri di valutazione dei modelli algoritmici di stima: soggettivi, oggettivi
- Modelli statistici
- Modelli storico-empirici
- Modelli teorici
- Modelli composti

Criteri di valutazione di modelli di stima

Criteri soggettivi:

- validità
- oggettività
- facilità d'uso
- robustezza
- trasportabilità

Criteri oggettivi:

estimatori statistici di validità del modello a fronte di un campione di misura: i criteri più in uso sono

i seguenti

◆ errore relativo assoluto medio	$\overline{MRE} \leq 0.25$
◆ coefficiente di determinazione multipla	$R^2 \sim 1$
◆ predizione a livello r	$PRED(0.25) \geq 0.75$
◆ valor medio relativo della deviazione standard	$\overline{RMS} \leq 0.25$

Criteri statistici di validità di un modello

Modelli statistici



Identificati mediante analisi di regressione

Modelli statistici lineari

$$S = a_0 + \sum_{i=1}^n a_i x_i$$

Forma di un modello statistico lineare

- i primi risalgono agli anni '60
- hanno prestazioni scadenti

Modelli statistici non lineari

- con:
- equazione nominale di stima** (da correggere poi con $m(\mathbf{X})$):
- $$S_{\text{nom}} = a + b \cdot I^c$$
- ◆ I : KSLOC
 - ◆ a, b, c : costanti del modello (regressione, ma talvolta dipendono dai *cost driver*)
 - ◆ $m(\mathbf{X})$: fattore di correzione, con \mathbf{X} il vettore dei *cost driver*

Forma frequente di un modello statistico non lineare dello sforzo

Modelli storico-empirici



Basati sulla raccolta di dati storici e sull'analogia

classico il modello di Wolverton (1975):

- ◆ **matrice di costo** del software $C_{i,j}$
 - i : **tipo** di funzionalità: controllo, I/O, pre/post-processing, algoritmo, gestione dati, criticità delle prestazioni
 - j : **difficoltà**: OE, OM, OH, NE, NM, NH
- ◆ $C_{i,j}$: costo unitario (*i.e.* per linea di codice)
- ◆ $i(k), j(k)$: stima di **tipo** e **difficoltà** del modulo k ($1, \dots, n$)
- ◆ $\hat{I}(k)$: stima della **dimensione** del modulo k
- ◆ $C(k) = \hat{I}(k) \cdot C_{i(k),j(k)}$: costo del modulo k

$$\text{Costo totale} = \sum_{k=1}^n C(k)$$

Modello di Wolverton

Modelli teorici

Derivano da ipotesi sull'operare della mente umana nel processo di sviluppo del software, o su leggi matematiche che si presume lo governino

- modello di Halstead (1972): *Software Science Effort Model*
- modello di Putnam (1978): *Resource Allocation Model*
- Modello di Jensen (1984): variante del modello di Putnam

Modelli composti di stima dei costi del software



Combinazione di idee, metodi e risultati dalle tre precedenti categorie di modelli: statistici, storico-empirici e teorici.

Modelli COCOMO ('81)

COnstructive **CO**st **MO**del: risultante dall'analisi statistica di 63 progetti (Boehm 1981)

- famiglia di nove modelli basata su due classificazioni ortogonali:
 - **livello di dettaglio**
dell'informazione espressa nelle metriche di input
tre livelli: *di base, intermedio, dettagliato*
 - **modalità di sviluppo**:
tre modalità: *organic, semi-detached, embedded*
- forma generale dei modelli COCOMO:

$$S = a \cdot I^b \cdot \prod_i C_i \quad T = c \cdot S^d$$

Modelli COCOMO '81

- *problema*: la stima della dimensione del prodotto I (KDSI)
- *soluzione*: stima dei Punti Funzione (v. appresso) + livello del linguaggio

La metodologia dei Punti Funzione

Presupposti

motivazione: misura economica della produttività $P = F/S$
 dove F è una misura astratta della quantità di funzionalità prodotta

Requisiti per F (Allen Albrecht, 1979):

- **astrazione**: indipendenza dalla tecnologia
- **completezza**: misura di tutte le funzioni consegnate
- **proprietà**: misura solo delle funzioni consegnate

prerequisito per la stima dei PF: specifica dei requisiti

PF, una metrica funzionale:

• cinque tipi t di entità funzionali:

1. file logici interni
2. file logici di interfaccia
3. input esterni
4. inquiry esterne
5. output esterni

• tre livelli di complessità c : bassa, media, alta

Calcolo dei PF nominali, o *Unadjusted Function Points* (UFP):

$$UFP = \sum_{t=1}^5 \sum_{c=1}^3 \pi_{t,c} \cdot n_{t,c}$$

Calcolo dei Punti Funzione nominali

dove $n_{t,c}$ è il numero di entità funzionali di tipo t e complessità c , mentre $\pi_{t,c}$ è il valore dei PF nominali per una entità funzionale di tale tipo e complessità, dato dalla tabella

Type of Component	Complexity of Components		
	Low	Average	High
External Inputs	3	4	6
External Outputs	4	5	7
External Inquiries	3	4	6
Internal Logical Files	7	10	15
External Interface Files	5	7	10

PF nominali per tipo e complessità delle entità funzionali

Correzione dei PF: le CGS

Caratteristiche Generali del Sistema: 14 proprietà *non funzionali* del prodotto, della cui influenza si tiene conto per correggere la stima dei PF entro il $\mp 35\%$

Calcolo dei PF corretti:

detta v_i l'influenza della CGS i , con $0 \leq v_i \leq 5$ ($i = 1, \dots, 14$), si calcola:

$$N = \sum_{i=1}^{14} v_i, \quad \text{VFC} = 0.01 \cdot N + 0.65, \quad \text{PF} = \text{UFP} \cdot \text{VFC}$$

Correzione del calcolo dei PF

Uso dei PF per la stima della dimensione in linee di codice

Livello λ di un linguaggio di programmazione L:

reciproco (a meno di un fattore di scala) del numero medio n_L di istruzioni che realizzano 1 PF in L: $\lambda \cdot n_L = 320$.

Calcolo di I_L (KDSI nel linguaggio L di livello λ) dai PF:

$$I_L = n_L \cdot \text{PF} / 1000 = 0.32 \cdot \text{PF} / \lambda$$

Limitazioni dei PF

Altri modelli e metodologie

- [COCOMO II, COCOMO Suite](#) : (Boehm et al. 2000)
- [Feature Points](#) (Capers Jones, 1986)
- [COSMIC Full Function Points](#) : Standard (ISO/IEC 2003)
- [System Dynamics](#) : originata da (Forrester 1961)
- [Software Project Dynamics](#) (Abdel-Hamid et al. 1991)

Approfondimenti



Altre risorse per approfondimenti su metriche, modelli e metodologie di stima dei costi del software

Altre risorse per approfondimenti

- **Ulteriori letture consigliate:**
(Sedehi 1997), (Buglione 2003), (Garmus et al. 2000)
- **Capitoli sulla stima dei costi in testi di Ingegneria del software:**
Ch. 26 in (Sommerville 2005), *liberamente disponibile*
Cap. 18 in (Pressman 2004)
Ch. 7 in (Van Vliet 2000)
- **Altri siti per la ricerca di documentazione:**
(GUFPI-ISMA), (IFPUG)
(R.S. Pressman & Associates)
- **Analisi comparativa di metodologie proprietarie:** (ProjectExperts 2006)

Note

1. Non si può controllare ciò che non si può misurare.

Prentice Hall: <http://vig.prenhall.com>

Bibliografia

- **Abdel-Hamid, T. & Madnik, S.E.**, 1991. *Software Project Dynamics: An Integrated Approach*. Prentice Hall.
- **Albrecht, A.J.**, 1979. Measuring application development productivity. In: *Proc. IBM Applications Development Symp.*. CA, USA: Guide Int. and Share, Inc., IBM Corp., 83.
- **Boehm, B.W.**, 1981. *Software Engineering Economics*. Prentice Hall.
- **Boehm, B.W., Abts, C., Brown, A.W., Chulani, A., Clark, B.K., Horowitz, E., Madachy, R., Reifer, D., Steefe, B.**, 2000. *Software Cost Estimation with COCOMO II*. Prentice Hall.
- **Boehm, B.W., Clark, B.K., Horowitz, E., Westland, C., Madachy, R., Selby, R.**, 1995. Cost models for future software life cycle processes: COCOMO 2.0. *Annals of Software Engineering*, 1:1, 57-94.
Web: <http://www.springerlink.com/content/y2386315010g7113>
- **Brooks, F.P. Jr.**, 1975. *The Mythical Man-Month*. Addison-Wesley.
- **Buglione, L.**, 2003. *Misurare il software, Seconda Ed.*. Milano: FrancoAngeli.
Web: http://www.geocities.com/lbu_measure/libri/mis.htm
- **DeMarco, T.**, 1986. *Controlling Software Projects: Management, Measurement, and Estimates*. Prentice Hall.
- **Forrester, J.W.**, 1961. *Industrial Dynamics*. New York, USA: MIT Press and John Wiley & Sons, Inc..
- **Garmus, D., Herron, D.**, 2000. *Function Point Analysis: Measurement Practices for Successful Software Projects*. Addison Wesley Professional, IT Series.
Web: <http://www.aw.com/cseng/series/it>
- **GUFPI-ISMA**. *Gruppo Utenti Function Point Italia*. Italian Software Metrics Association. Web: <http://www.gufpi-isma.org>.
- **IFPUG**. *International Function Point Users Group*. Web: <http://www.ifpug.org>.
- **ISO/IEC**, 2003. *Software engineering -- COSMIC-FFP -- A functional size measurement method*. Geneva, CH: ISO/IEC, ISO/IEC 19761.
Web: <http://www.iso.org>
- **Kitchenham, B.A. & Känsälä, K.**, 1993. Inter-item Correlations among Function Points. In: *Proc. 15th Int'l Conference on Software Engineering*. Baltimore, ML, USA: IEEE Computer Society / ACM Press, 477-480.
Web: <http://portal.acm.org/citation.cfm?id=257572.257677>
- **Pressman, R.S.**, 2004. Cap. 18: Stime per il progetto software. In: *Principi di Ingegneria del software, 4/Ed.*. McGraw-Hill Italia.
Web: <http://www.mcgraw-hill.it>
- **ProjectExperts** (2006). *Twenty Years of Better IT Estimating Software*. Web: http://www.projectexperts.com/articles/estimating_sw.htm.
- **R.S. Pressman & Associates, Inc.**. *Software Engineering Resources*. Web: <http://www.rspa.com/spi>.

- **Sedehi, H.**, 1997. *Ingegneria economica del software*. Milano; Roma: Apogeo; EUCOS, Libreria Italiana (2003).
Web: <http://www.libreriaitaliana.it>
- **Sommerville, I.**, 2005. Chapter 26: Software Cost Estimation. *In: Software Engineering, 7th Ed.*. Addison-Wesley, Pearson Education.
Web: <http://www.comp.lancs.ac.uk/computing/resources/lanS/SE7/SampleChapters/ch26.pdf>
- **Van Vliet, H.**, 2000. Chapter 7: Cost Estimation. *In: Software Engineering - Principles and Practice, 2nd Ed.*. John Wiley & Sons.
Web: <http://www.wiley.co.uk/vanvliet>