Gestione – parte IIB

Rif. Ghezzi et al.
6.2.3

università di ferrara
DA SEICENTO ANNI GUARDIAMO AVANTI.

Stima dei costi

- Difficile da effettuare nei primi stadi di un progetto:
  - Requisiti incerti
  - Possono essere necessarie tecnologie innovative:
    - Riducono i costi a lungo termine
    - Possono aumentare quelli a breve termine
  - Le persone e le loro capacità possono non essere note
- Le stime tendono a essere self-fulfilling ("auto-adempienti"):
  - Le risorse vengono allocate secondo le stime
  - Il progetto viene adattato alle risorse disponibili

Tecniche di stima dei costi

- Modellazione algoritmica
  - Modelli che legano il costo a metriche del software (LOC, function point, punti oggetto)
- Giudizio di esperti
  - Discussione e mediazione fra esperti delle tecniche prescelte
- Stima tramite analogia
  - Stima ottenuta dai costi di progetti analoghi
- Legge di Parkinson
  - Il lavoro si adatta al tempo disponibile: se il tempo disponibile è m mesi e sono disponibili n persone, il costo sarà m*n mesi-uomo
- Pricing to win
  - Si adatta lo sforzo alla disponibilità del cliente

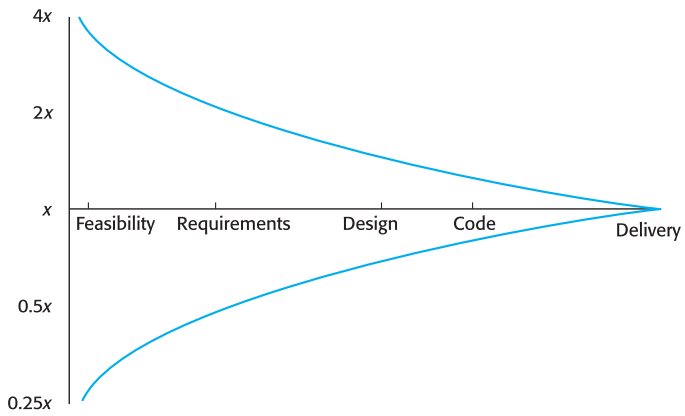
Stima top-down e bottom-up

- Top-down
  - Basata sulle caratteristiche di alto livello
  - Si può applicare appena è nota un'architettura generale del sistema
  - Tende a trascurare problemi tecnici di basso livello
- Bottom-up
  - Somma il costo dei singoli componenti
  - Richiede un'architettura dettagliata del sistema
  - Tende a trascurare i costi di integrazione, gestione della configurazione, documentazione

Modellazione algoritmica

- Formula generica:
  Sforzo = A * Dimensione^B * M
  - A costante dipendente dall'organizzazione e dal tipo di software
  - Dimensione è una stima della dimensione del codice
  - B è solitamente compreso fra 1 e 1.5
  - M dipende da
    - Processo
    - Prodotto
  - A * Dimensione^B spesso è detto *sforzo nominale*

Incertezza delle stime



Fattori che influenzano i costi

- Requisiti
  - Correttezza
  - Prestazioni
- Uso di componenti
- Linguaggio di programmazione
  - Produttività
  - Difficoltà
- Personale
  - Disponibilità
  - Capacità/esperienza

COCOMO

- Boehm, 1981
- COnstructive COst MOdel
- Stima della dimensione del codice basata su KDSI
- Modalità di sviluppo
  - organico
  - semi-distaccato
  - embedded

Caratteristiche delle modalità di sviluppo

Feature	Mode		
	Organic	Semidetached	Embedded
Organizational understanding of product objectives	Thorough	Considerable	General
Experience in working with related software systems	Extensive	Considerable	Moderate
Need for software conformance with pre-es tablished requirements	Basic	Considerable	Full
Need for software conformance with external interface specifications	Basic	Considerable	Full
Concurrent development of associated new hardware and operational procedures	Some	Moderate	Extensive
Need for inn ovative data processing architectures, algorithms	Minimal	Some	Considerable
Premium on early completion	Low	Medium	High
Product size range	<50 KDSI	<300 KDSI	All sizes

Sforzo nominale e totale

- Si determina la categoria di sviluppo considerando quale approssima meglio le caratteristiche del progetto
- A ogni categoria corrispondono sforzi nominali diversi
- Lo sforzo totale si determina moltiplicando quello nominale per moltiplicatori dipendenti da caratteristiche del prodotto e del processo

Development Mode	Nominal effort	Schedule
Organic	$(PM)_{NOM}=3.2(KDSI)^{1.05}$	$TDEV=2.5(PM_{DEV}))^{0.38}$
Semidetached	$(PM)_{NOM}=3.0(KDSI)^{1.12}$	$TDEV=2.5(PM_{DEV}))^{0.35}$
Embedded	$(PM)_{NOM}=2.8(KDSI)^{1.20}$	$TDEV=2.5(PM_{DEV}))^{0.32}$

Moltiplicatori

Cost Drivers	Ratings					
	Very low	Low	Nominal	High	Very High	Extra High
Product attributes						
Required software reliability	.75	.88	1.00	1.15	1.40	
Data base size		.94	1.00	1.08	1.16	
Product complexity	.70	.85	1.00	1.15	1.30	1.65
Comput er attributes						
Execution time constraints			1.00	1.11	1.30	1.66
Main storage constraints			1.00	1.06	1.21	1.56
Virtual machine volatility*		.87	1.00	1.15	1.30	
Computer turnaround time		.87	1.00	1.07	1.15	
Personnel attributes						
Anal yst capability	1.46	1.19	1.00	.86	.71	
Applications experience	1.29	1.13	1.00	.91	.82	
Programmer capability	1.42	1.17	1.00	.86	.70	
Virtual machine experience*	1.21	1.10	1.00	.90		
Programming language experience	1.14	1.07	1.00	.95		
Project attributes						
Use of modern programming practices	1.24	1.10	1.00	.91	.82	
Use of software tools	1.24	1.10	1.00	.91	.83	
Required development schedule	1.23	1.08	1.00	1.04	1.10	

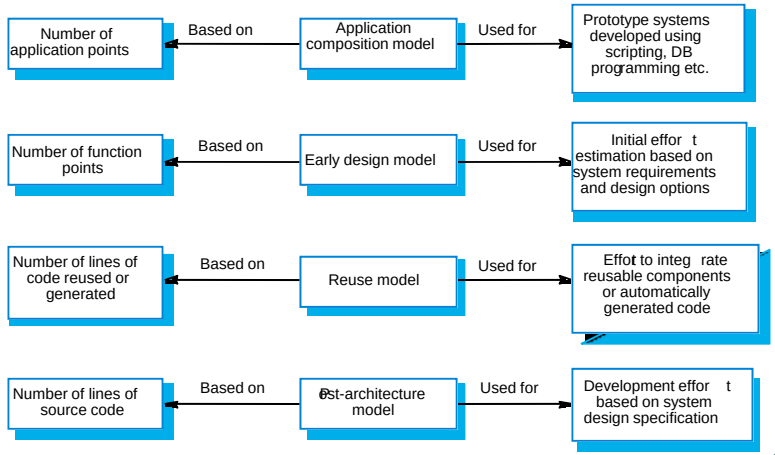
COCOMO: difetti

- Ipotesi forti sul processo di sviluppo
  - Processo a cascata
  - Software sviluppato su commissione
  - Requisiti noti e non soggetti a modifiche
- Non adatto a modelli moderni
  - Metodi evolutivi
  - Progetto OO
  - Riuso di componenti
- Stima basata sul numero di linee di codice

COCOMO II

- Composto da un insieme di sottomodelli
  - applicabili in fasi diverse del processo o per tipi diversi di processo
  - basati su misure o stime diverse
- Application Composition Model
- Early Design Model
- Reuse Model
- Post-Architecture Model

Modelli COCOMO II



Application Composition Model

- Per progetti di prototipi o con ampio riuso (GUI)
- Considera strumenti CASE
- Basato su metrica dei punti applicazione (o punti oggetto):
  - numero di schermate e report da produrre, ognuno pesato secondo fattore di complessità:
    - facile
    - medio
    - difficile

Application Composition Model

- PM = (NAP × (1 - %reuse/100)) / PROD

  dove
  - PM è lo sforzo in persone/mese
  - NAP è il numero di punti applicazione
  - PROD è la produttività
  - %reuse è la percentuale di codice riusato

Developer's experience and capability	Very low	Low	Nominal	High	Very high
ICASE maturity and capability	Very low	Low	Nominal	High	Very high
PROD (NOP/month)	4	7	13	25	50

Early Design Model

- Stima basata sui requisiti
- Basata su formula standard:
  - PM = A × Size^{B} × M dove
  - M = PERS × RCPX × RUSE × PDIF × PREX × FCIL × SCED;
  - A = 2.94, Size in KLOC, B varia da 1.1 a 1.24 a seconda di novità del progetto, flessibilità di sviluppo, processi di risoluzione dei rischi, coesione del team, livello di maturità del processo

Early design model

- Moltiplicatori:
  - RCPX – affidabilità e complessità del prodotto;
  - RUSE – riutilizzo necessario;
  - PDIF – difficoltà della piattaforma;
  - PREX – esperienza del personale;
  - PERS – capacità de personale;
  - SCED - tempistica;
  - FCIL – funzionalità di supporto.

Reuse model

- Considera codice che si può riusare senza modifiche e codice che richiede adattamenti per essere integrato.
- Due versioni:
  - Riuso black-box, dove il codice non viene modificato. Sforzo nullo.
  - Riuso white-box, dove il codice viene modificato. Si stima lo sforzo necessario per produrre una quantità equivalente di codice nuovo.

Stima 1

- Codice generato:
- PM = (ASLOC * AT/100)/ATPROD
  - ASLOC è il numero di righe di codice prodotto
  - AT è la percentuale di codice generato automaticamente
  - ATPROD è la produttività nella generazione di codice (stima: 2400 LOC/mese)

Stima 2

- Codice che deve essere compreso e integrato: dimensione equivalente
- ESLOC = ASLOC * (1-AT/100) * AAM.
  - ASLOC e AT come prima.
  - AAM adaptation adjustment multiplier (moltiplicatore di aggiustamento adattativo), somma di:
    - AAF, costo di modifica del codice esistente
    - SU, costo di comprensione del codice esistente
    - AA, costo di decisione sul riuso

Post-Architecture Model

- Modello più dettagliato
- PM = A $\times$ Size^{B} $\times$ M (come in Early Design Model), ma
  - Size è la somma di
    - righe di nuovo codice
    - ESLOC da riuso
    - righe da modificare per cambiamenti nei requisiti
  - M è prodotto di 17 fattori (anzichè 7 come in Early Design Model)

Esponente

- B=1.01 + S/100, dove

  S è la somma di valutazioni da 5 (molto basso) a 0 (molto alto) dei seguenti fattori:
  - precedenti
  - flessibilità di sviluppo
  - analisi dei rischi
  - coesione del team
  - maturità del processo

Moltiplicatori

- Attributi di prodotto
  - requisiti sul prodotto da sviluppare
- Attributi informatici
  - vincoli imposti al software dalla piattaforma hardware
- Attibuti del personale
  - tengono conto dell'esperieza e della capacità dei membri del team
- Attributi di progetto
  - caratteristiche particolari del progetto