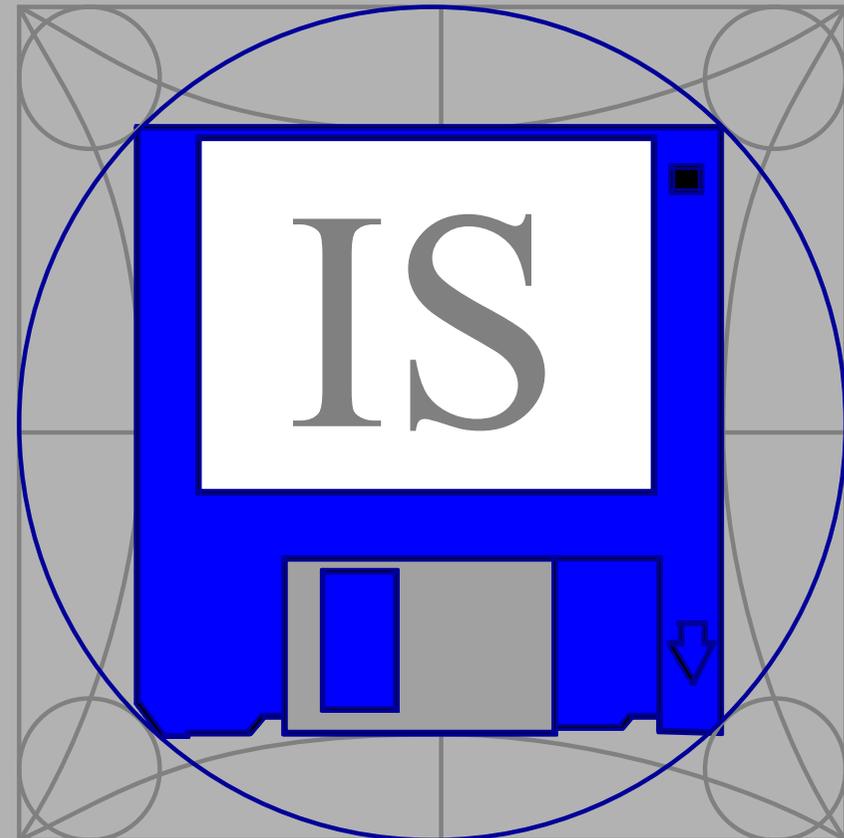


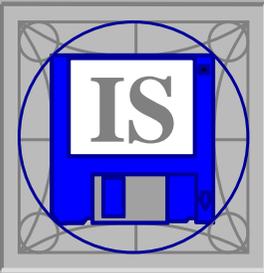
# Gestione di progetto

Ingegneria del Software

V. Ambriola, G.A. Cignoni,  
C. Montangero, L. Semini

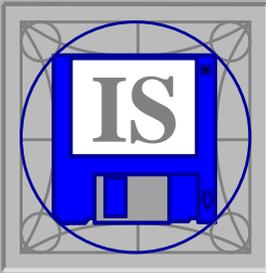
Aggiornamenti: T. Vardanega (UniPD)





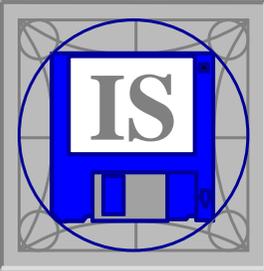
# Cosa è un progetto?

- ❑ **Ne abbiamo visto la definizione (Kerzner, SEMAT), e anche la funzione rispetto al ciclo di vita di prodotto**
- ❑ **Ora sappiamo che un progetto è un insieme ordinato di attività che istanziano processi di ciclo di vita**
  - **Le attività sono fatte di compiti assegnati a singoli individui**
  - **Il progetto nel suo complesso è sempre collaborativo**
- ❑ **Le attività sono pianificate prima di essere svolte**
  - **Ogni attività ha specifici obiettivi e vincoli che derivano dal processo di appartenenza**
  - **Ogni singola attività di progetto deve puntare all'economicità**



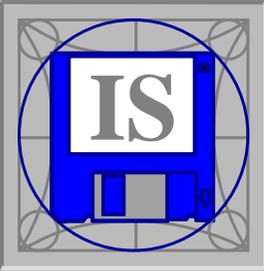
# Fondamenti di gestione

- ❑ Determinare i processi generali da istanziare nel *way of working*
- ❑ Stimare le risorse necessarie per svolgere le attività di progetto
  - Rapportandole sempre alle risorse disponibili
- ❑ Pianificare le attività e assegnarle alle persone
  - A partire dall'obiettivo: all'indietro e non in avanti
- ❑ Controllare le attività verificandone i risultati
  - Il più frequentemente possibile, senza rallentare il lavoro: *push, non pull*
  - Aggiornando la pianificazione alla luce dell'avanzamento



## Ruoli e funzioni

- Le organizzazioni specializzano il proprio personale per funzioni
  - Direzione, Amministrazione finanziaria, Sviluppo, Controllo di qualità, ...
  
- In un progetto, le persone assumono **ruoli**
  - Ogni ruolo assume responsabilità (*ownership*) su specifiche attività di specifici processi



# Ruoli: analisti e progettisti

## □ Analisti

- **Conoscono il dominio del problema e hanno esperienza professionale**
- **Hanno molta influenza sul successo del progetto**
- **Sono pochi: non seguono il progetto fino alla consegna**

## □ Progettisti

- **Hanno competenze tecniche e tecnologiche aggiornate**
- **Influiscono sulle scelte tecniche e tecnologiche**
- **Sono pochi: seguono lo sviluppo, non la manutenzione**



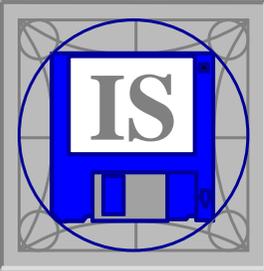
# Ruoli: programmatori e verificatori

## □ Programmatori

- Partecipano alla realizzazione e manutenzione del prodotto
- Hanno competenze tecniche, ma autonomia e responsabilità circoscritte
- Formano la categoria storicamente più popolosa

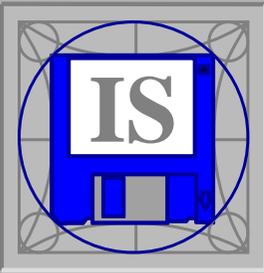
## □ Verificatori

- Sono presenti per l'intera durata del progetto
- Hanno competenze tecniche, esperienza professionale, conoscenza del *way of working*
- Hanno capacità di giudizio e di relazione



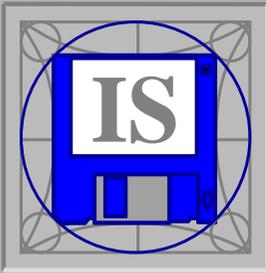
## Ruoli: responsabile

- **Governa il *team* e rappresenta il progetto verso l'esterno**
  - Accentra le responsabilità di scelta e approvazione
  - Partecipa al progetto per tutta la sua durata
- **Ha responsabilità su**
  - Pianificazione
  - Gestione delle risorse umane
  - Controllo, coordinamento e relazioni esterne
- **Deve avere conoscenze e capacità tecniche**
  - Per valutare rischi, scelte, alternative



## Ruoli: amministratore

- **Ha controllo sull'ambiente di lavoro**
  - Amministrazione delle infrastrutture di supporto
  - Risoluzione di problemi legati alla gestione dei processi
  - Salvaguardia della documentazione di progetto
  - Controllo di versioni e configurazioni
  
- **Funzione o ruolo?**
  - È funzione aziendale in organizzazioni strutturate, con più progetti simili (ragioni di standardizzazione)
  - Altrimenti solo ruolo di progetto



# Gestione qualità

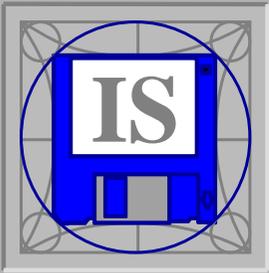
- ❑ **La funzione di più recente introduzione**
  - Funzione aziendale, non ruolo di progetto
- ❑ **La qualità ha più dimensioni**
  - Riguarda i prodotti e i processi
  - Interessa sia il committente che la direzione aziendale
- ❑ **La garanzia di qualità produce confidenza**
  - Richiede applicazione rigorosa dei processi adottati
  - E loro manutenzione migliorativa → ciclo PDCA

Di questo parleremo  
ampiamente più avanti

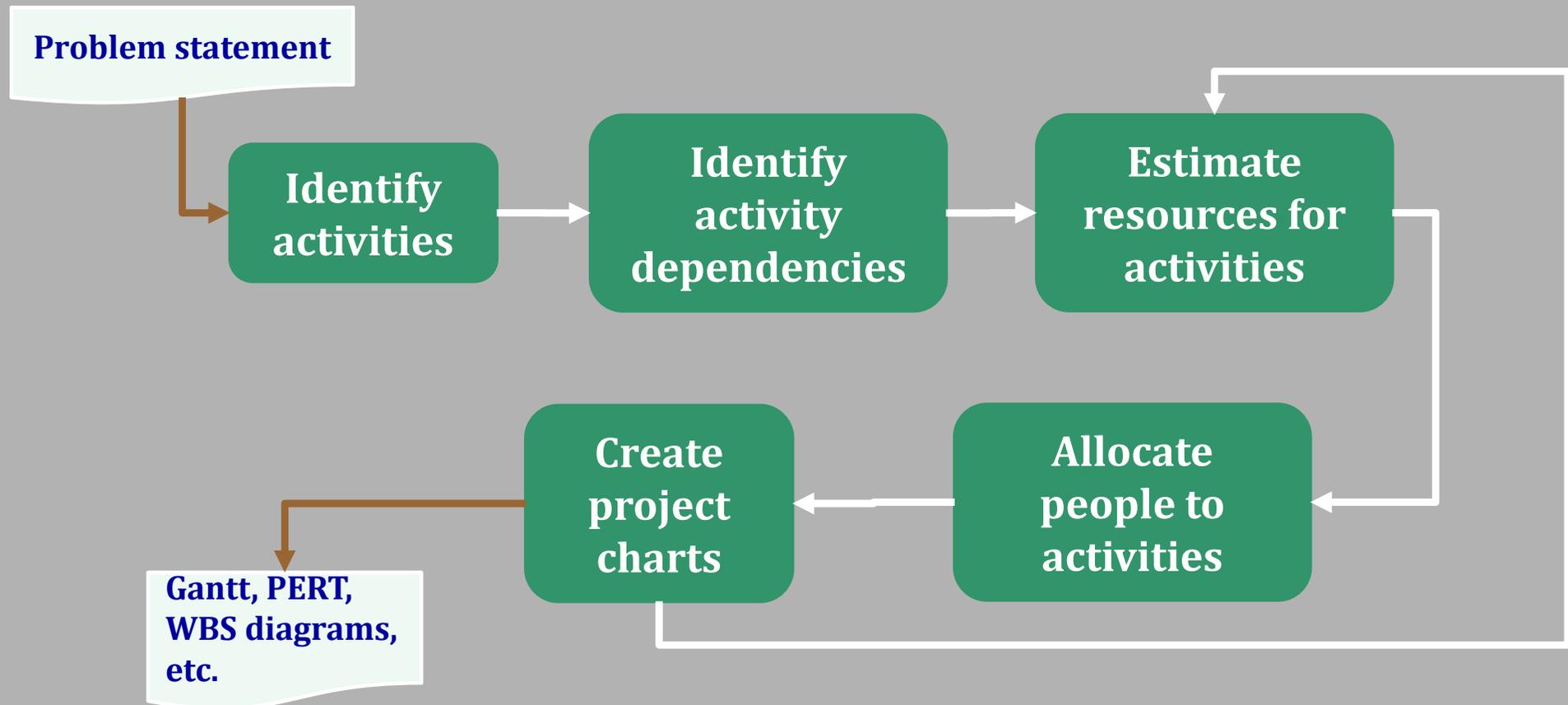


# Pianificazione di progetto – 1

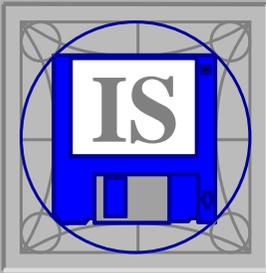
- **Definizione delle attività**
  - Per pianificarne lo svolgimento e valutare il progresso
  - Per avere una base su cui gestire l'allocazione delle risorse
  - Per stimare e controllare scadenze e costi
  
- **Strumenti per la pianificazione**
  - **Diagrammi di Gantt**
    - (*"Work, Wages and Profit"*, Henry L. Gantt, The Engineering Magazine, NY, 1910 )
  - ***Programme Evaluation and Review Technique (PERT)***
  - ***Work Breakdown Structure (WBS)***



# Pianificazione di progetto – 2

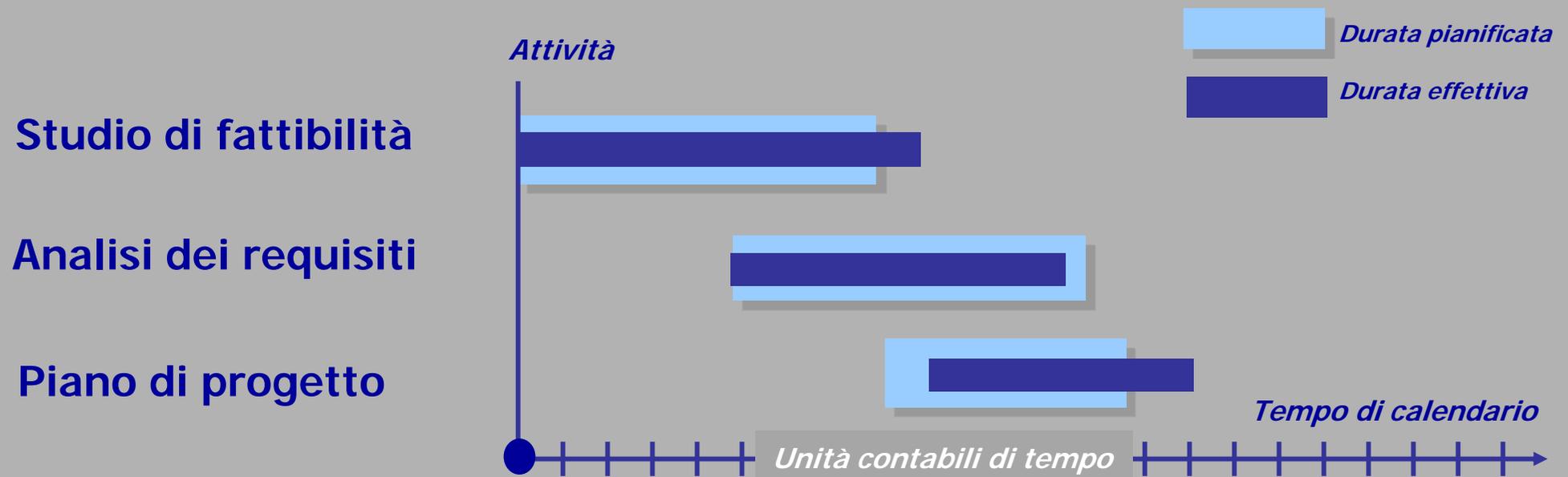


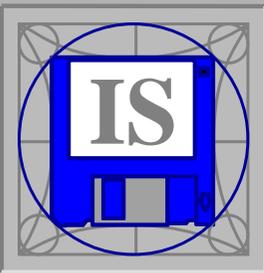
Tratto da: Ian Sommerville, *Software Engineering*, 8<sup>th</sup> ed.



# Diagrammi di Gantt

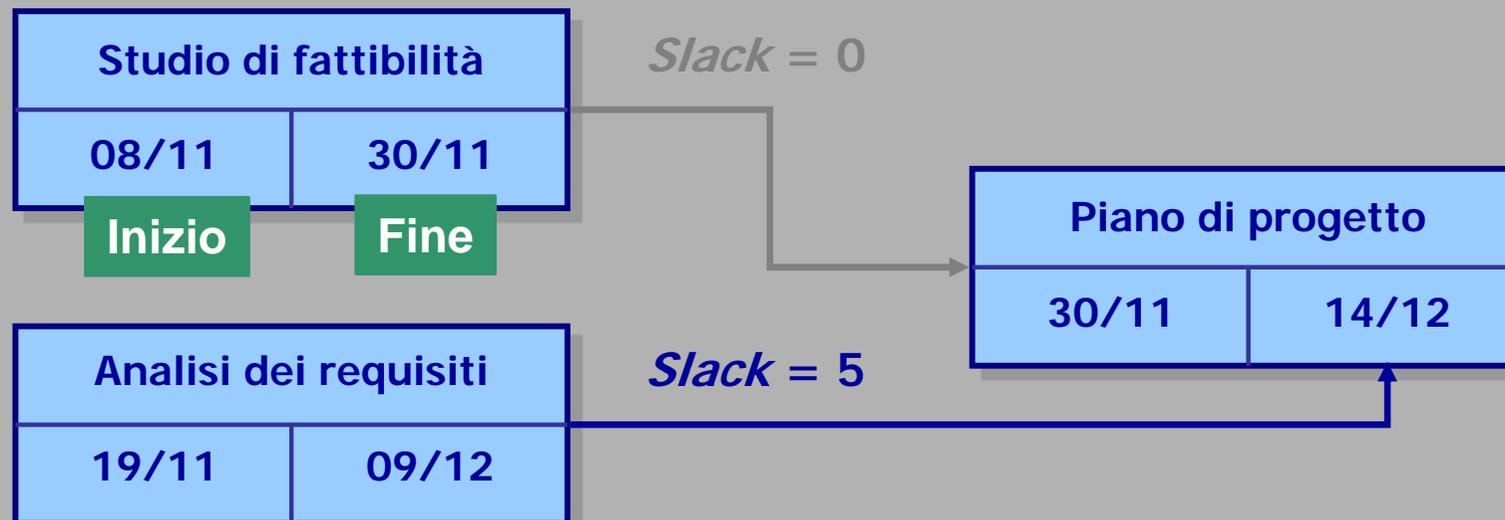
- **Dislocazione temporale delle attività**
  - Per rappresentarne la durata
  - Per rappresentarne sequenzialità e parallelismo
  - Per confrontare le stime con i progressi

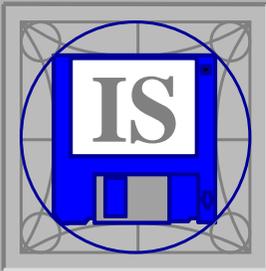




# Diagrammi PERT

- Dipendenze temporali tra attività
  - Per ragionare (all'indietro) sulle scadenze di progetto
  - *Slack time*
  - **Cammino critico** → sequenza di attività ordinata con prodotto importante e dipendenze temporale strette

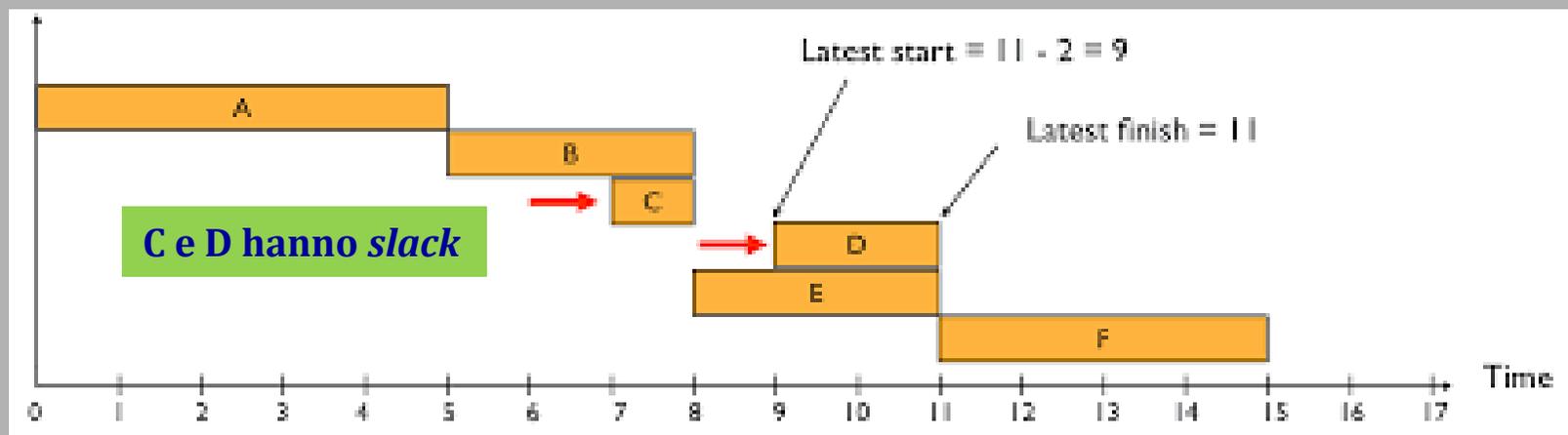
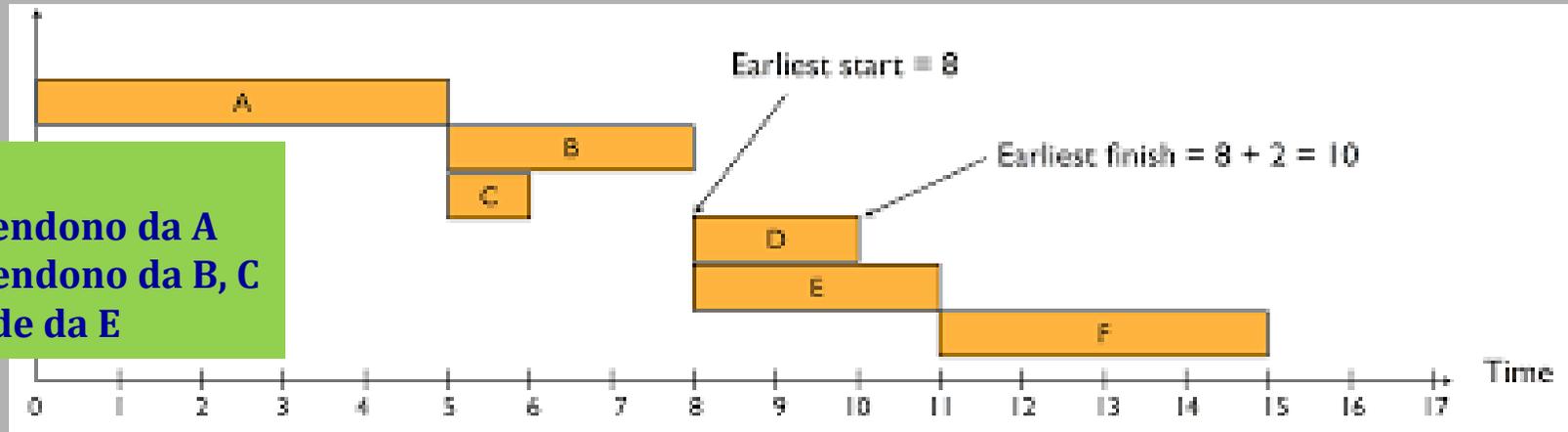


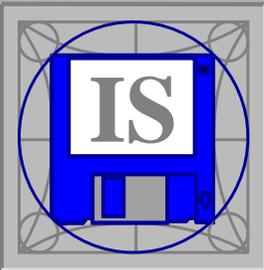


# Slack

## Vincoli

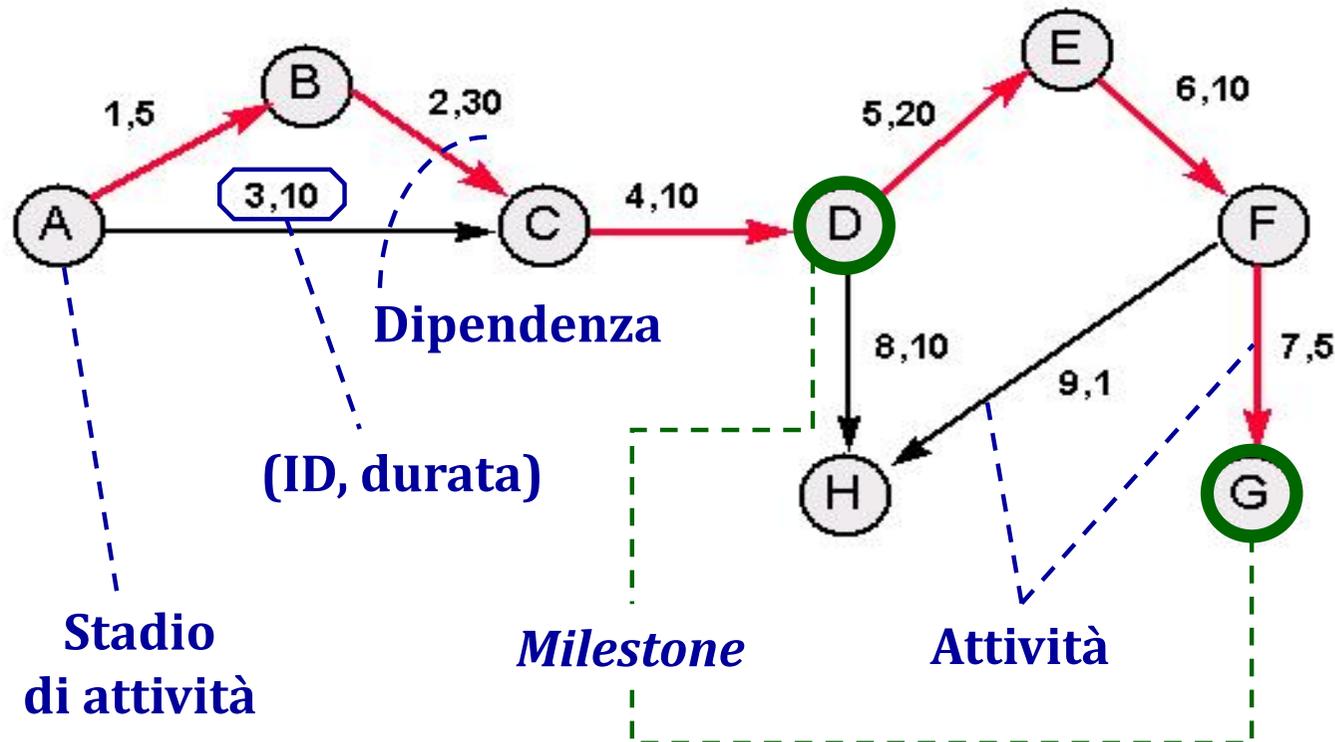
- B, C dipendono da A
- D, F dipendono da B, C
- F dipende da E

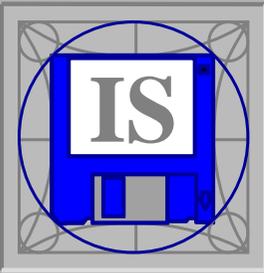




# Diagrammi PERT – esempio

Forma semplificata – in rosso il “cammino critico”

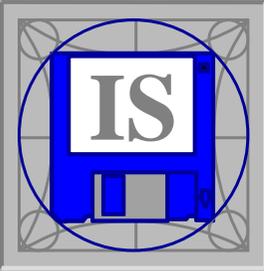




# *Work Breakdown Structure*

- **Struttura gerarchica delle attività**
  - Ogni attività si compone di sottoattività
  - Non necessariamente sequenziali
  - Univocamente identificate

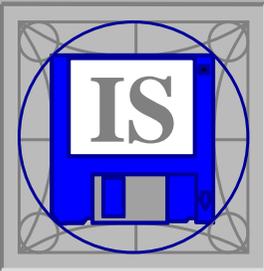




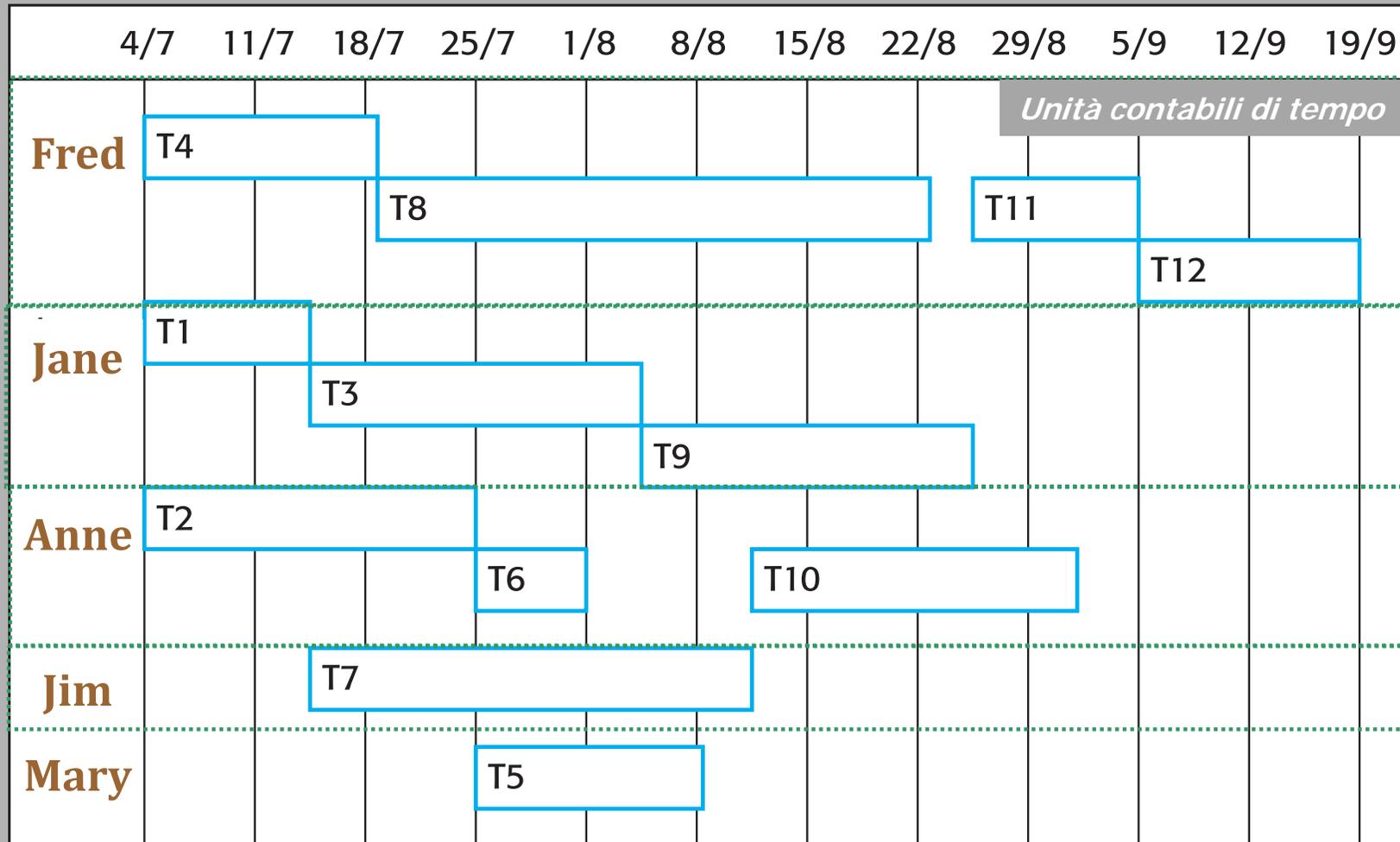
# Allocazione delle risorse

- ❑ **Assegnare attività a ruoli e ruoli a persone**
- ❑ **Difficoltà**
  - Non sottostimare
  - Non sovrastimare
- ❑ **Molte risorse sono impegnate su più progetti**
  - Aziendalmente, per non incorrere in sotto-utilizzo
  - Per voi, perché avete molti altri obblighi oltre a IS
- ❑ **Gestire più “cammini critici” su più progetti**

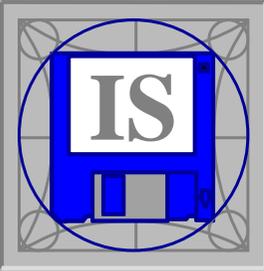




# Allocazione delle risorse – esempio

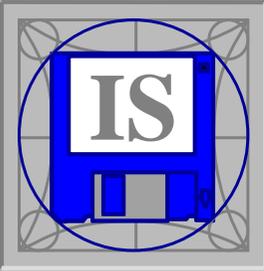


Tratto da: Ian Sommerville, *Software Engineering*, 8<sup>th</sup> ed.



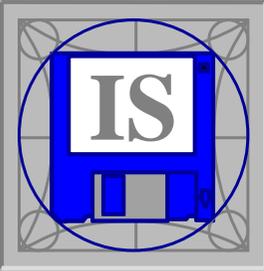
# Stima dei costi di progetto

- ❑ **Come pianificare?**
  - Con strumenti che permettano di organizzare le attività
  - Con strumenti che permettano di evidenziare le criticità
  - Con strumenti che permettano di studiare scenari
- ❑ **Come definire durata e costo delle attività?**
  - Calcolando il **tempo/persona** stimato necessario e rapportandolo al tempo di calendario
- ❑ **Come stimarlo?**
  - Esperienza, analogia, competizione, algoritmo predittivo, raffinamenti
- ❑ **Grana grossa sull'insieme, grana fine entro periodi brevi**



## Fattori di influenza sulle stime

- Dimensione del progetto
- Esperienza del dominio
- Familiarità con le tecnologie
- Produttività dell'ambiente di lavoro
- Qualità attesa



# Un modello algoritmico

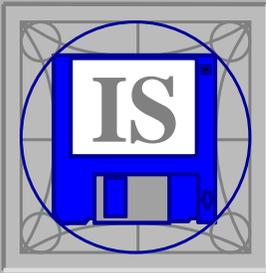
## □ *Constructive Cost Model (CoCoMo)*

- Stima le risorse necessarie esprimendone la misura in mesi/persona (*MP*)
- *Software Engineering Economics*, B. Boehm, Prentice-Hall, 1981

□  $MP = \alpha \times S^\beta \times \gamma$  (**mesi persona**)     $T = \delta \times MP^\varepsilon$  (**tempo di calendario**)

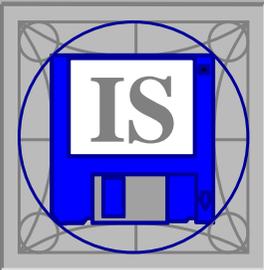
- $\alpha$ : tipo di (complessità) progetto
- $S$ : dimensione del prodotto (in *Kilo Delivered Source Instructions*)
- $\beta$ : impatto della complessità sullo sviluppo
- $\gamma$ : coefficiente di correzione (inizialmente fissato a 1)
- $\delta$ : fattore di espansione del tempo (inizialmente fissato a 2,5)
- $\varepsilon$ : coefficiente di complessità

1 mese/persona = 152 ore



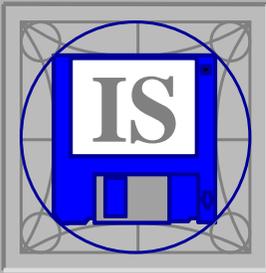
# CoCoMo base

- ❑ Assume modello di sviluppo sequenziale, da zero
- ❑ Bassa complessità di progetto: “*Simple*” [*Organic*]
  - Ciascuno può avere comprensione globale del prodotto
  - $\alpha = 2,4$     $\beta = 1,05$     $\varepsilon = 0,38$
- ❑ Complessità media: “*Moderate*” [*Semi-detached*]
  - Il prodotto può essere compreso solo per parti
  - $\alpha = 3,0$     $\beta = 1,12$     $\varepsilon = 0,35$
- ❑ Complessità elevata: “*Complex*” [*Embedded*]
  - Il prodotto interagisce con terze parti e ambiente esterno
  - $\alpha = 3,6$     $\beta = 1,20$     $\varepsilon = 0,32$



# Esempi di stime CoCoMo

	$\alpha$	$\beta$	$\gamma$	$S$	$MP$
<i>simple</i>	2.40	1.05	1.00	3.00	7.61
				3.50	8.94
				4.00	10.29
				4.50	11.64
<i>moderate</i>	3.00	1.12	1.00	3.00	10.27
				3.50	12.20
				4.00	14.17
				4.50	16.17
<i>embedded</i>	3.60	1.20	1.00	3.00	13.45
				3.50	16.19
				4.00	19.00
				4.50	21.89



# Piano di progetto – 1

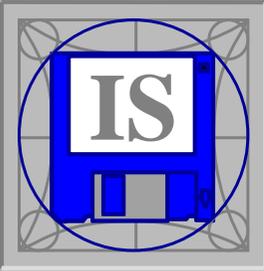
## □ Contenuti

- Risorse disponibili e loro assegnazione alle attività
- Scansione delle attività nel tempo

## □ Obiettivi

- Organizzare le attività con efficienza per produrre risultati efficaci
- Facilitare la misurazione dell'avanzamento fissando “*milestone*” nel tempo

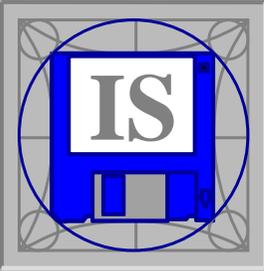




## Piano di progetto – 2

### □ **Struttura tipica del PdP**

- Introduzione (scopo e struttura)
- Organizzazione del progetto
- **Analisi dei rischi**
- Risorse disponibili (tempo e persone)
- Suddivisione del lavoro (*work breakdown*)
- Calendario delle attività (*project schedule*)
- Meccanismi di controllo e di rendicontazione



# Rischi di progetto

## □ Risultati negativi da evitare

- Sforamento dei costi
- Sforamento dei tempi
- Risultati insoddisfacenti

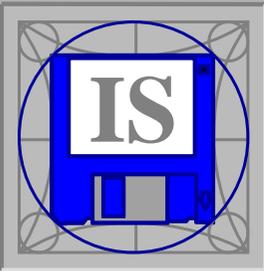
## □ Perché?

- **Fonte: studio Standish Group (1994-2004)**
  - Da leggere con cautela per i numeri assoluti; solido nella sostanza
- **Analisi delle maggiori cause di fallimento**



## Fonti di rischio

- ❑ **Tecnologie di lavoro e di produzione SW**
- ❑ **Rapporti interpersonali**
- ❑ **Organizzazione del lavoro**
- ❑ **Requisiti e rapporti con gli *stakeholder***
- ❑ **Tempi e costi**



# Gestione dei rischi – 1

## □ Identificazione

- Nel progetto, nel prodotto, nel mercato

## □ Analisi

- Probabilità di occorrenza, conseguenze possibili

## □ Pianificazione

- Come evitare i rischi o mitigarne gli effetti

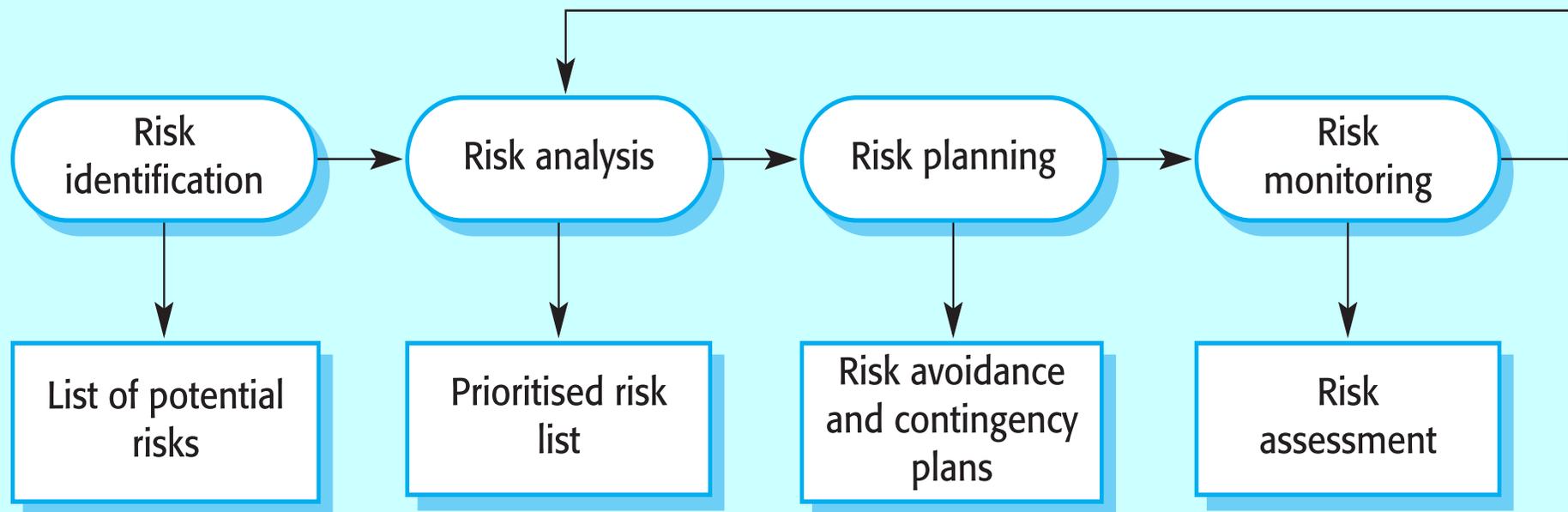
## □ Controllo

- Attenzione continua tramite rilevazione di indicatori
- Attuazione delle procedure di mitigazione
- Raffinamento delle strategie

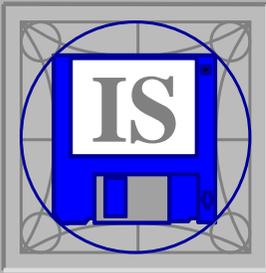




# Gestione dei rischi – 2



Tratto da: Ian Sommerville, *Software Engineering*, 8<sup>th</sup> ed.



## Secondo Standish Group (1994)

### □ Progetti di successo (dati USA)

- In tempo, senza costi aggiuntivi, prodotto soddisfacente
- 16.2%

### □ Progetti a rischio

- Fuori tempo, o con costi aggiuntivi, o con prodotto difettoso
- 52.7%

### □ Fallimenti

- Progetti cancellati prima della fine
- 31.1%

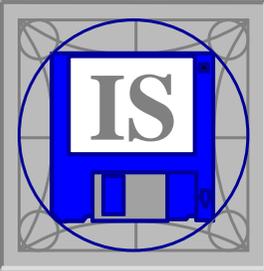
**ATTENZIONE:**

Vi è *bias* nei dati assoluti,  
ma alla base vi sono  
elementi di realtà



## Fattori di successo

- Coinvolgimento del cliente** 15.9%
- Supporto della direzione esecutiva** 13.9%
- Definizione chiara dei requisiti** 13.0%
- Pianificazione corretta** 9.6%
- Aspettative realistiche** 8.2%
- Personale competente** 7.2%



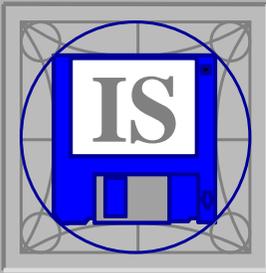
## Fattori di fallimento

- |   |       |
|---|-------|
| <input type="checkbox"/> Requisiti incompleti               | 13.1% |
| <input type="checkbox"/> Mancato coinvolgimento del cliente | 12.4% |
| <input type="checkbox"/> Mancanza di risorse                | 10.6% |
| <input type="checkbox"/> Aspettative non realistiche        | 9.9%  |
| <input type="checkbox"/> Mancanza di supporto esecutivo     | 9.3%  |
| <input type="checkbox"/> Fluttuazione dei requisiti         | 8.7%  |



## Secondo Standish Group (2004)

- ❑ **X edizione dell'indagine**
  - Oltre 40.000 progetti USA studiati in 10 anni
  - Valore complessivo : 255 miliardi \$ (250 nel 1994)
- ❑ **Progetti finiti con successo : 34% (era 16,2%)**
  - Importante miglioramento nelle tecniche di gestione
- ❑ **Progetti falliti : 15% (era 31,1%)**
  - Danno economico : 55 miliardi \$ (140 nel 1994)
  - Peggior eccesso di costo : 43% (189% nel 1994)



- ❑ ***Software Project Management Technology Report***,  
[www.slideshare.net/Samuel90/project-management-technology-report](http://www.slideshare.net/Samuel90/project-management-technology-report)
- ❑ **La stima dei costi dei sistemi informativi automatizzati**,  
[www.researchgate.net/publication/265986910\\_LA\\_STIMA\\_DEI\\_COSTI\\_DEI\\_SISTEMI\\_INFORMATIVI\\_AUTOMATIZZATI](http://www.researchgate.net/publication/265986910_LA_STIMA_DEI_COSTI_DEI_SISTEMI_INFORMATIVI_AUTOMATIZZATI)
- ❑ **B. Boehm et al., “Cost Models for Future Software Life Cycle Processes: CoCoMo II”, USC CSSE**,  
[sunset.usc.edu/csse/research/COCOMOII/cocomo\\_main.html](http://sunset.usc.edu/csse/research/COCOMOII/cocomo_main.html)
- ❑ **Standish Group, “The CHAOS Report”**  
[vedi pagina pubblica del corso]