



Project management

Alberto Gianoli

Dove?

- ❖ Pressmann, cap. 5 e qualcosa da cap. 21
- ❖ Sommerville, cap. 5

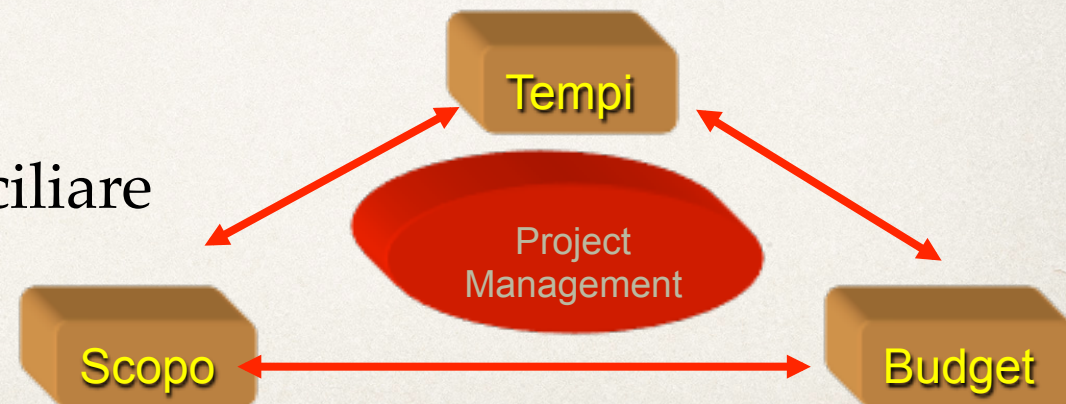
Un po' di management...

- ❖ **Software Project Management**

- ❖ intendiamo tutte le attività necessarie ad assicurare che un progetto software sia sviluppato rispettando le scadenze prefissate e risponda a determinati standard
 - ❖ aspetti sia tecnici che economici
- ❖ Un progetto diretto bene può anche fallire (qualche volta), ma un progetto diretto male fallisce quasi sicuramente
- ❖ E' anche questione di esperienza.....

Cosa intendiamo per progetto?

- * In generale per progetto (software e non solo) intendiamo un insieme ben definito di attività
 - * ha un inizio
 - * ha una fine
 - * ha uno scopo
 - * viene portato avanti da un insieme di persone
 - * utilizza un insieme di risorse
 - * non è un lavoro di routine
- * Il project management deve conciliare tempi, scopo e budget



Perché ci interessa la gestione dei progetti?

- ❖ L'ingegneria del software è anche una attività economica, quindi soggetta a vincoli di tipo economico
- ❖ Ricordate? Meno di 1 / 5 dei progetti è un completo successo (funziona, in tempo e con costi corretti)
- ❖ N.B.: lo scopo è illustrare l'attività di gestione di progetti, non farvi diventare dei manager.... (per quello serve l'esperienza)

Problemi

- ❖ Il software è intangibile. Per valutare i progressi di un progetto bisogna basarsi su qualcosa: la documentazione
- ❖ L'ingegneria del software non è (ancora) riconosciuta come disciplina , non come ingegneria elettrica, civile, ...
- ❖ Non esiste un vero standard di processo per la produzione del software
- ❖ Ogni progetto è un caso a sé

Motivi per cui un progetto può essere in ritardo

- ❖ Deadline non realistica, fissata e/o imposta da qualcuno esterno allo staff tecnico
- ❖ Cambiamenti dei requisiti imposti dal cliente
- ❖ Stima ottimistica (troppo) del lavoro e delle risorse necessarie a compierlo
- ❖ Rischi che non sono stati presi in considerazione all'inizio del progetto
 - ❖ difficoltà tecniche imprevedibili
 - ❖ difficoltà umane imprevedibili
- ❖ Problemi di comunicazione nel gruppo di sviluppo
- ❖ Incapacità dalla direzione del progetto di riconoscere che c'è un ritardo, e mancata attuazione di contromisure

Worst case scenario

- ❖ supponiamo di dover portare a termine un progetto in 12 mesi
- ❖ dopo aver analizzato attentamente il lavoro necessario e i rischi, arrivate alla conclusione che un tempo più realistico per portare a termine il progetto e' di almeno 18 mesi
- ❖ che si fa???

Worst case scenario

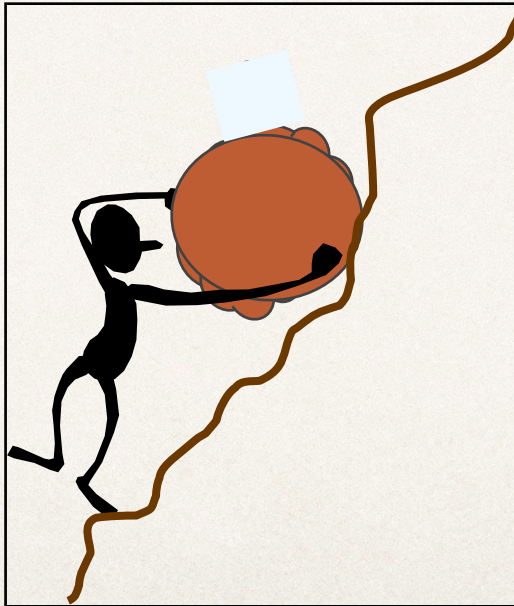
- ❖ **Possibilità # 1:** aumento il budget e aggiungo risorse
 - ❖ non sempre è fattibile; se si attua in ritardo aumentano i rischi di un lavoro di scarsa qualità dati i tempi stretti
- ❖ **Possibilità # 2:** elimino parte delle funzionalità non essenziali
 - ❖ intanto sviluppo un prototipo che implementi l'indispensabile; il resto lo aggiungo nei restanti 6 mesi
- ❖ **Possibilità # 3:** faccio finta di nulla e provo ugualmente a farcela in 12 mesi
 - ❖ a meno di miracoli il fallimento è assicurato....

Principi fondamentali per la pianificazione

- ❖ Un progetto deve essere ripartito in attività e compiti di dimensioni ragionevoli
- ❖ Bisogna determinare le dipendenze tra attività e compiti
- ❖ Determinare quali compiti si possono svolgere in parallelo e quali in sequenza
- ❖ Alcune attività dipendono da altre per poter iniziare
- ❖ A ogni compito bisogna assegnare delle “unità di lavoro” (p.e. mesi/uomo)
- ❖ Ogni compito deve avere una data di inizio e una di fine
- ❖ A ogni progetto è assegnato un numero definito di persone
- ❖ Non bisogna assegnare più persone del necessario
- ❖ Ogni compito deve essere assegnato a qualcuno
- ❖ Ogni compito deve avere un risultato predefinito
- ❖ A ogni compito si deve associare almeno un *punto di controllo*
- ❖ Un punto di controllo è passato quando la qualità di uno o più compiti è approvata

Perché c'è bisogno di un team

- ❖ La maggior parte dei progetti software sono troppo impegnativi per poter essere realizzati da una sola persona



Gli attori sulla scena

- ❖ Senior managers
 - ❖ di solito definiscono gli aspetti economici del progetto
- ❖ Project managers
 - ❖ pianificano, organizzano, controllano lo sviluppo del progetto
- ❖ Practitioners
 - ❖ chi ha le competenze tecniche per realizzare parti del progetto
- ❖ Customers
 - ❖ il cliente che stabilisce i requisiti del software (definiscono il problema che bisogna risolvere)
- ❖ End users
 - ❖ chi userà il sistema una volta sviluppato

Il project manager

Di cosa si occupa?

- ❖ Stesura della proposta del progetto
- ❖ Stima del costo del progetto (insieme ai senior managers)
- ❖ Planning e scheduling
 - ❖ cioè partiziona il progetto, individua le milestones e i deliverables
- ❖ Monitoraggio e revisioni
- ❖ Selezione dello staff e assegnazione ai singoli compiti
- ❖ Stesura dei rapporti e delle presentazioni

Abbiamo un piano?

1. Introduzione

- ❖ definizione degli obiettivi del progetto, e dei vincoli prefissati (costi, tempo, risorse, ...)

2. Organizzazione

- ❖ definisce l'organizzazione del team di sviluppo: quali sono le persone e quali sono i loro ruoli

3. Analisi dei rischi

- ❖ elenco dei rischi previsti, della probabilità che accadano, delle strategie per ridurli o affrontarli

4. Risorse HW e SW richieste

- ❖ stime di costo per acquisire le risorse; stime temporali (tempi di consegna)

Abbiamo un piano?

5. Suddivisione del lavoro

- ❖ Suddivisione in attività, vengono identificati i *deliverables* e le *milestones*

6. Scheduling del progetto

- ❖ identificare le dipendenze tra le attività, stima tempo richiesto per le milestones, assegnazione personale alle attività

7. Controllo e rapporto sulle attività

- ❖ elenca i rapporti che devono essere prodotti per i manager, quando devono essere prodotti e che meccanismi di controllo sullo stato di avanzamento delle attività sono previste

Abbiamo un piano?

Fattori di successo

❖ coinvolgimento del cliente	16%
❖ supporto direzione esecutiva	14%
❖ definizione chiara requisiti	13%
❖ pianificazione corretta	10%
❖ aspettative realistiche	8%
❖ personale competente	7%

Fattori di fallimento

❖ requisiti incompleti	13%
❖ cliente non coinvolto	12%
❖ mancanza risorse	11%
❖ aspettative non realistiche	10%
❖ mancanza supp. esecutivo	9%
❖ cambiamenti requisiti	9%

Organizzare le attività

- ❖ Le attività del progetto devono produrre dei documenti affinché manager possano osservarne l'avanzamento
- ❖ Milestones: sono i punti finali di ogni attività
- ❖ Deliverables: sono i risultati del progetto consegnati ai clienti
- ❖ Nel modello a cascata è banale definire le milestones (attività già ben definite); con altri modelli la musica cambia

Scheduling

- ❖ Occorre suddividere il progetto in tasks e stimare il tempo e le risorse necessari per completare ogni compito
- ❖ organizzare i compiti concorrentemente permette un uso ottimale della forza lavoro
- ❖ minimizzare le dipendenze tra task permette di evitare la propagazione a catena dei ritardi
- ❖ Lo scheduling “ottimale” spesso è un’arte: dipende dall’istinto e dall’esperienza del project manager

Problemi dello scheduling

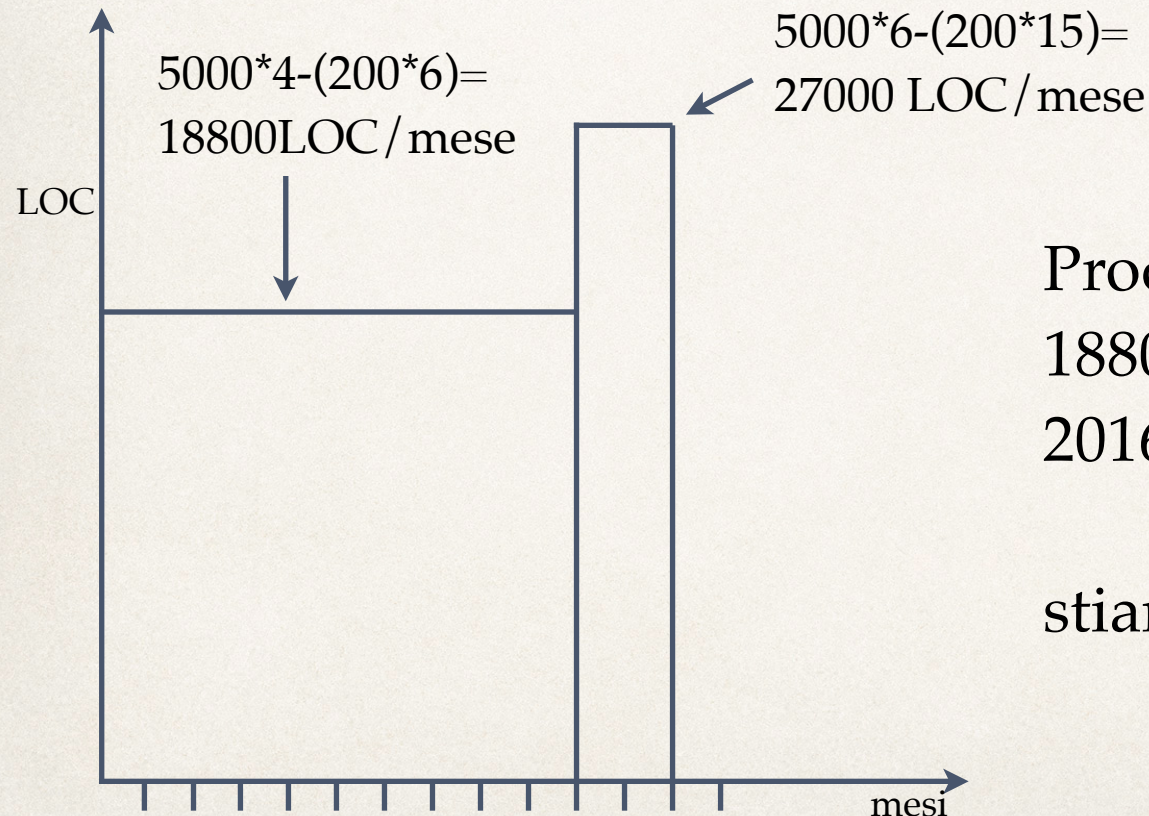
- ❖ Stimare la difficoltà di un problema e i costi di sviluppo di una soluzione è difficile
- ❖ la produttività non è proporzionale al numero di persone che lavorano ad un task
- ❖ Variante della legge di Murphy: ciò che è inatteso si verifica puntualmente. Occorre sempre essere pronti ad affrontare l'imprevisto
- ❖ Regola del 40-20-40
 - ❖ 40% del tempo per l'analisi e la progettazione
 - ❖ 20% del tempo per la scrittura del codice
 - ❖ 40% per il collaudo

Esempio

- ❖ Consideriamo 4 programmatori, con una capacità di scrivere 5000 linee di codice (LOC) in un mese
- ❖ Li facciamo lavorare in gruppo
 - ❖ per tenersi aggiornati tra loro devono creare canali di comunicazione a scapito della produttività
 - ❖ stimiamo che producano 200 LOC / mese in meno per ogni canale
 - ❖ al netto: $5000 \cdot 4 - (200 \cdot 6) = 18800$ LOC / mese
 - ❖ perso il 6% rispetto alla somma delle singole produttività

Esempio

- * supponiamo che il progetto duri un anno, ma dopo dieci mesi aggiungiamo due nuovi membri



Produttività totale:

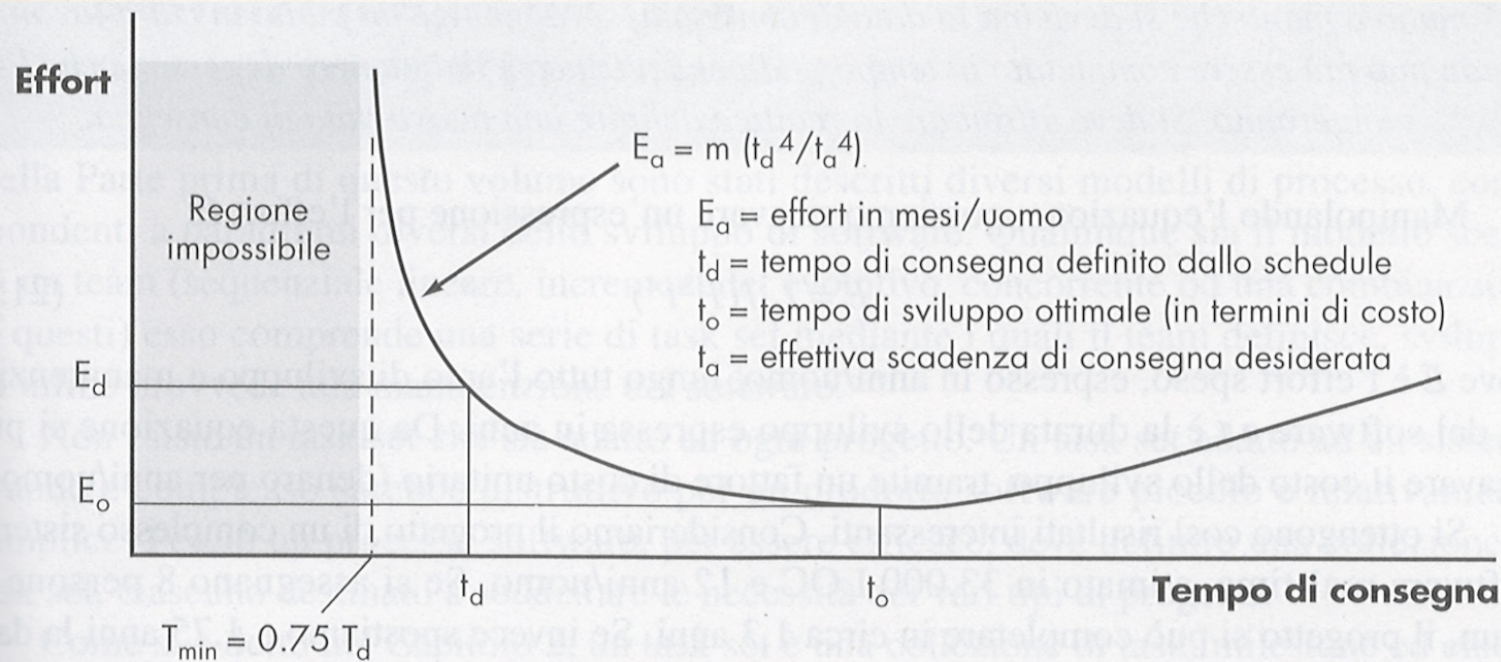
$$18800*10/12 + 27000*2/12 = 20167\text{ LOC/mese}$$

stiamo perdendo ~6.9%

Possiamo generalizzare?

- ❖ No! La relazione tra numero di persone e produttività non è lineare
- ❖ Il lavoro di gruppo è controproducente (perdita di tempo)?
 - ❖ No, perché la comunicazione tra i membri del team serve a migliorare la qualità del software
- ❖ Le revisioni tecniche formali possono portare miglioramenti all'analisi, riducendo il numero di errori e quindi il tempo di collaudo e di debug

Relazione tra persone e lavoro: curva PNR



L numero di righe di codice

P parametro di produttività (valori tipici da 2000 a 28000)

$$L = P \times E^{1/3} \times t^{4/3} \leftarrow t \text{ tempo}$$

A. Gianoli - Ingegneria del Software E impegno espresso in mesi/uomo o anni/uomo

Tipologie di team

- ❖ **Democratico Decentralizzato**

- ❖ assenza di un leader permanente
- ❖ consenso di gruppo sulle soluzioni e sulla organizzazione del lavoro
- ❖ comunicazione orizzontale

- ❖ **Vantaggi**

- ❖ attitudine positiva a ricercare presto gli errori
- ❖ funziona bene per problemi “difficili” (p.e. per la ricerca)

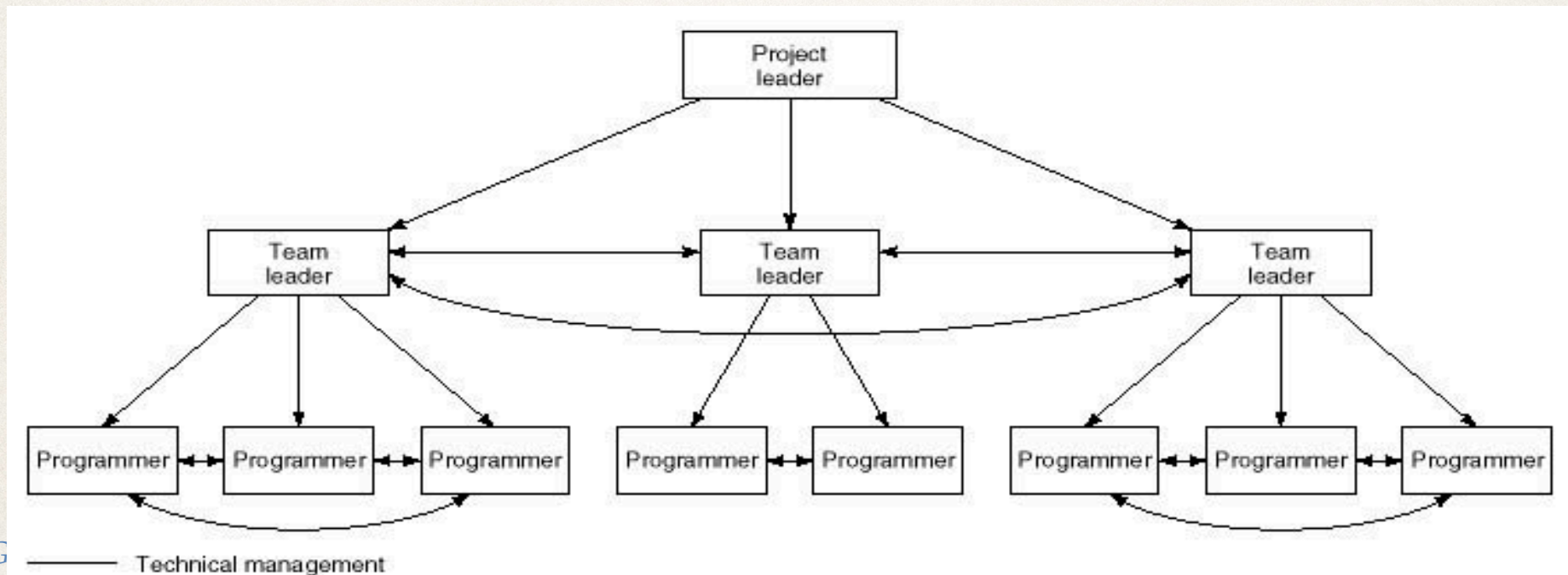
- ❖ **Svantaggi**

- ❖ difficile da imporre
- ❖ non è scalabile

Tipologie di team

❖ Controllato Decentralizzato

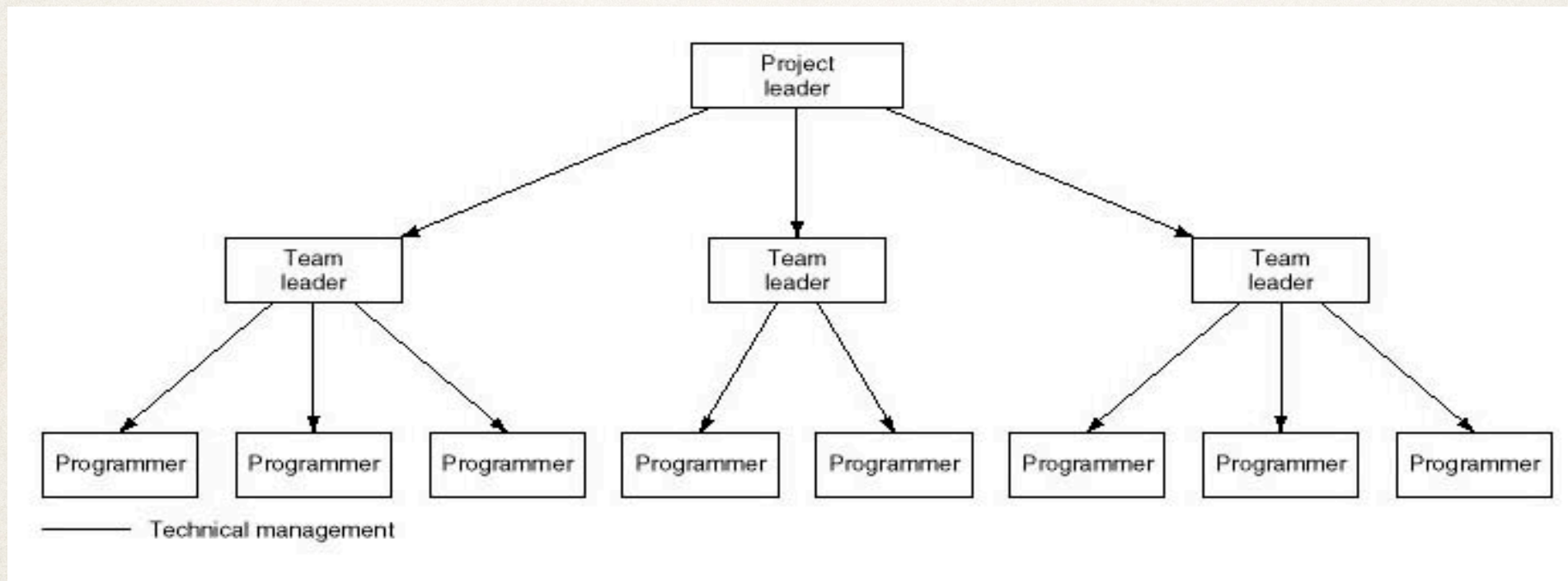
- ❖ un leader riconosciuto che coordina il lavoro
- ❖ la risoluzione dei problemi è di gruppo, ma l'implementazione delle soluzioni è assegnata dal leader ai vari sottogruppi
- ❖ comunicazione orizzontale tra i sottogruppi e verticale con il leader



Tipologie di team

❖ Controllato Centralizzato

- ❖ il team leader decide sulle soluzioni e sull'organizzazione
- ❖ comunicazione verbale tra team leader e gli altri membri



Ruoli in un team Controllato Decentralizzato

- ❖ Project manager
 - ❖ pianifica, coordina e supervisiona le attività del team
- ❖ Technical staff
 - ❖ conduce l'analisi e lo sviluppo (da 2 a 5 persone)
- ❖ Backup engineer
 - ❖ supporta il project manager ed è responsabile della validazione
- ❖ Software librarian
 - ❖ mantiene e controlla la documentazione, i listati del codice, i dati..

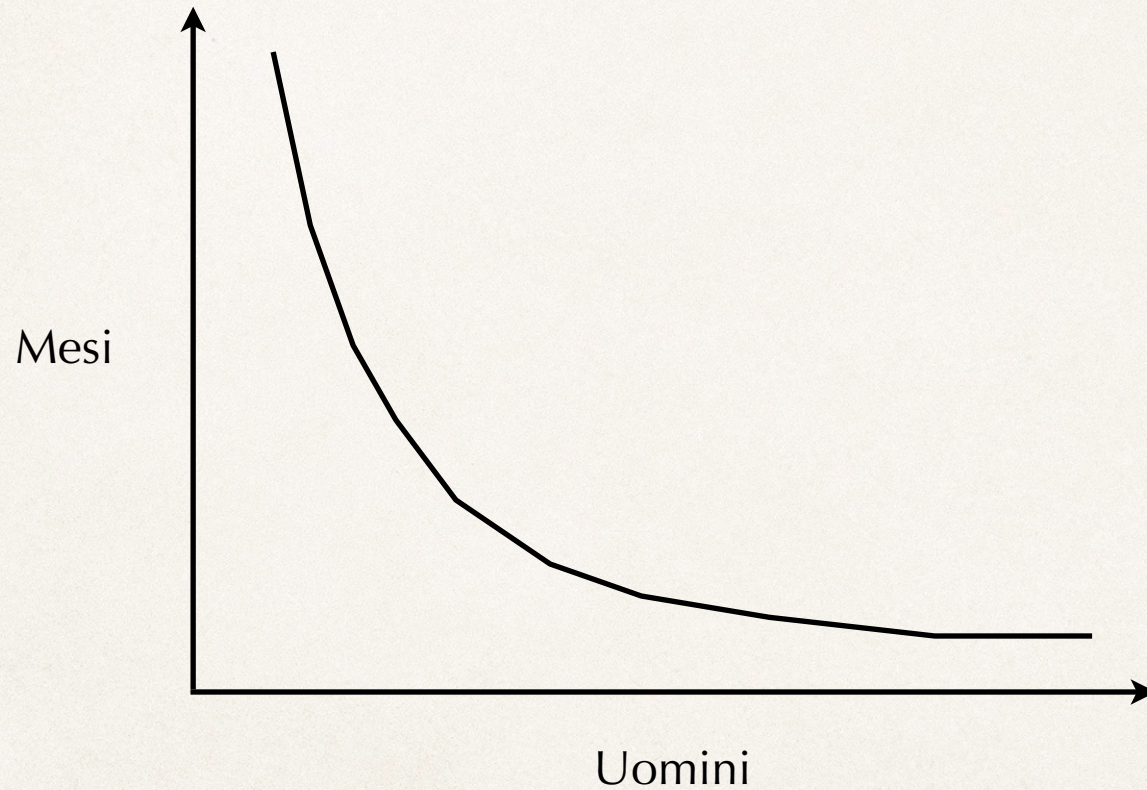
Mese-uomo: cosa bisogna ricordare

- ❖ Tra le cause di fallimento di progetti software, la mancanza di tempo sufficiente è significativa
- ❖ Perché?
 - ❖ Le tecniche di stima dei tempi non sono affidabili; inoltre non si considerano gli imprevisti
 - ❖ Durante la stima si confonde l'impegno con i progressi fatti: si assume che uomini e mesi siano interscambiabili
 - ❖ Non si tiene traccia dei progressi compiuti in maniera corretta
 - ❖ Se si è in ritardo in un punto, si tende ad aggiungere personale, e abbiamo visto che può peggiorare le cose

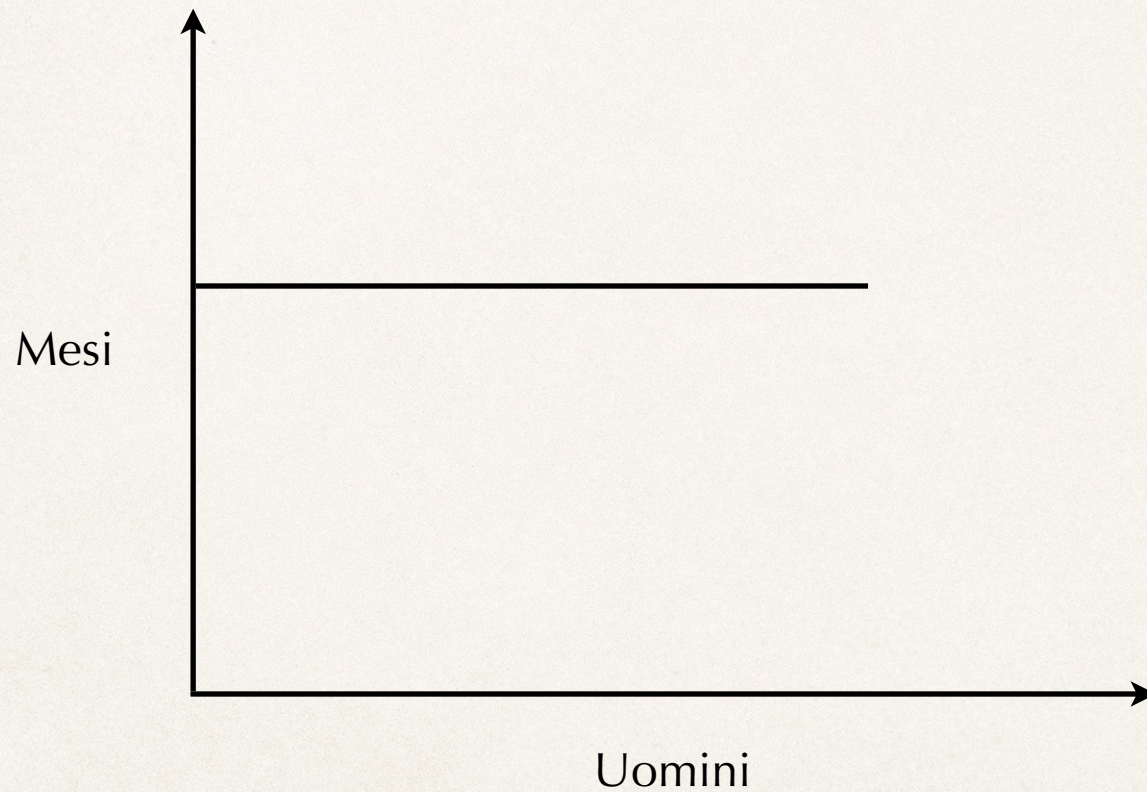
Mese-uomo

- ❖ Come misurare l'effort (impegno) richiesto per un lavoro? La misura tipica è il mese-uomo
- ❖ In effetti facilita il calcolo dei costi del personale: basta fare $\text{numero_mesi} * \text{numero_uomini}$
- ❖ Peccato che i progressi compiuti **non** siano proporzionali a questa unità
- ❖ Quand'è che mesi e uomini sono effettivamente interscambiabili? Solo per compiti che possono essere perfettamente partizionati tra i lavoratori, *e che non richiedono comunicazione tra essi*

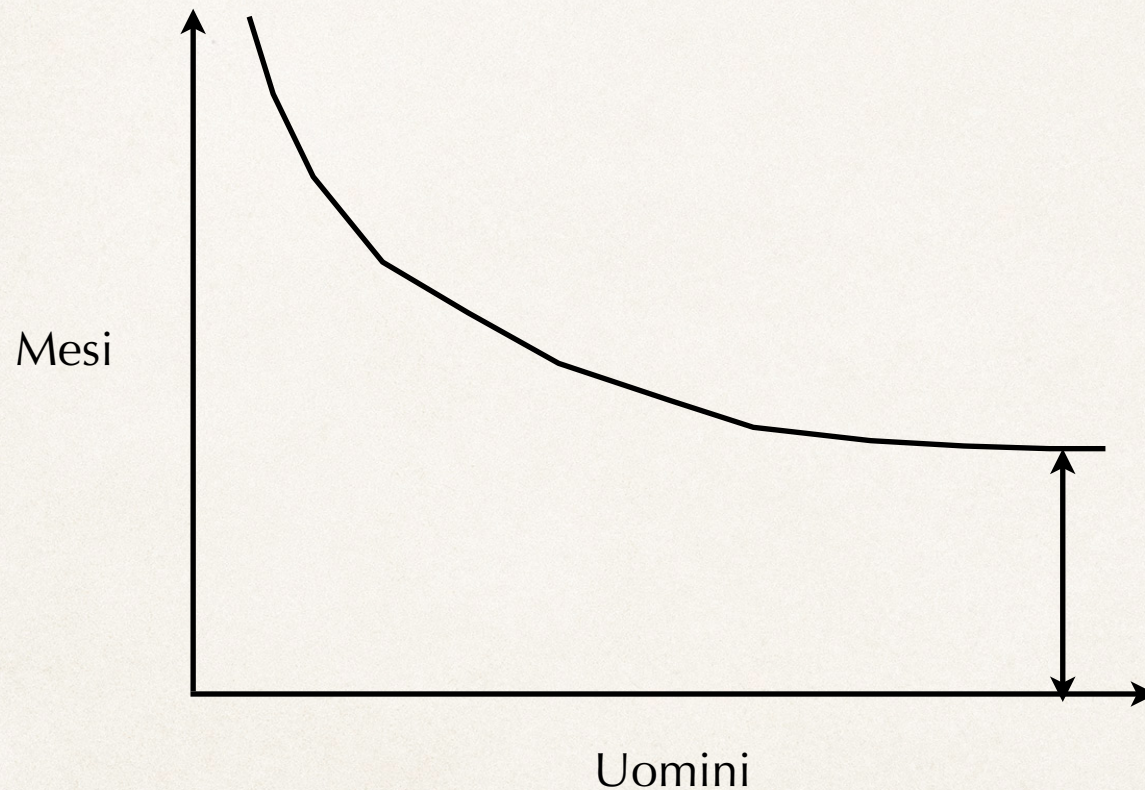
Mesi vs Uomini: task perfettamente partizionabile



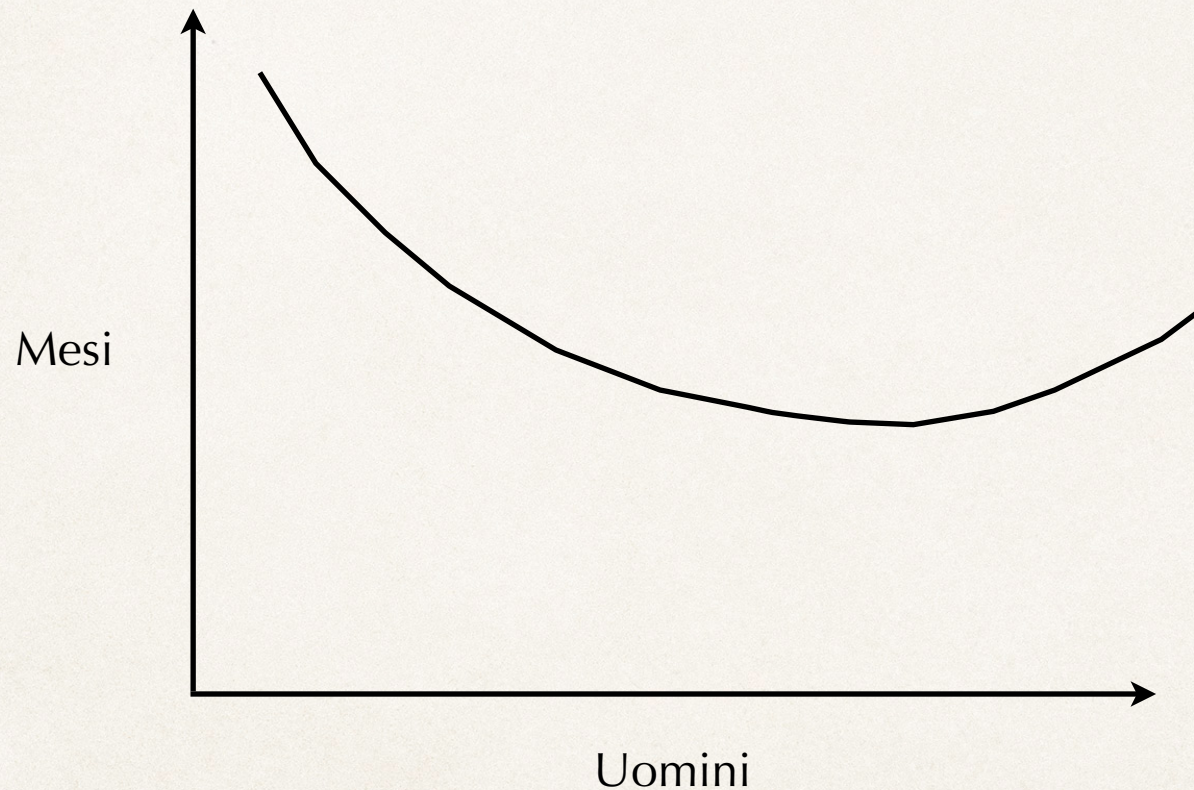
Mesi vs Uomini: task non partizionabile



Mesi vs Uomini: task partizionabile che richiede comunicazione

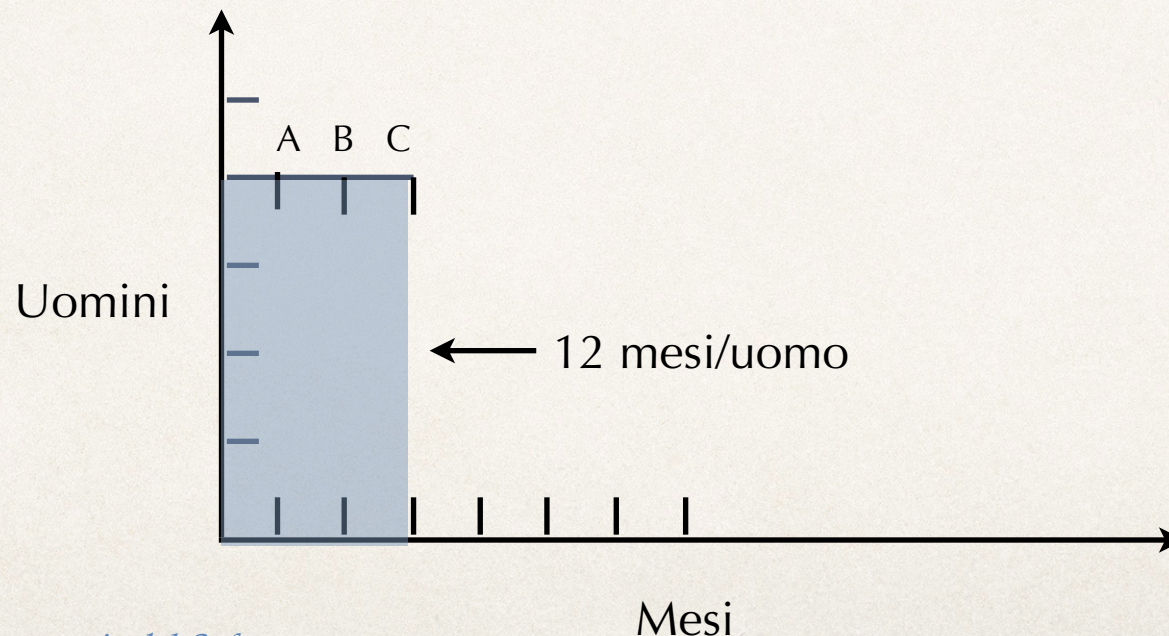


Mesi vs Uomini: task partizionabile in cui tutti parlano con tutti



Esempio

- ❖ Proviamo a chiarire con un esempio: consideriamo un progetto perfettamente partizionabile con una stima di 12 mesi/uomo. Per il progetto sono assegnati 4 programmatori per 3 mesi. Alla fine di ogni mese sono stabilite delle milestones: A, B, C



Esempio

- ❖ Imprevisto: la prima milestone A viene raggiunta dopo due mesi

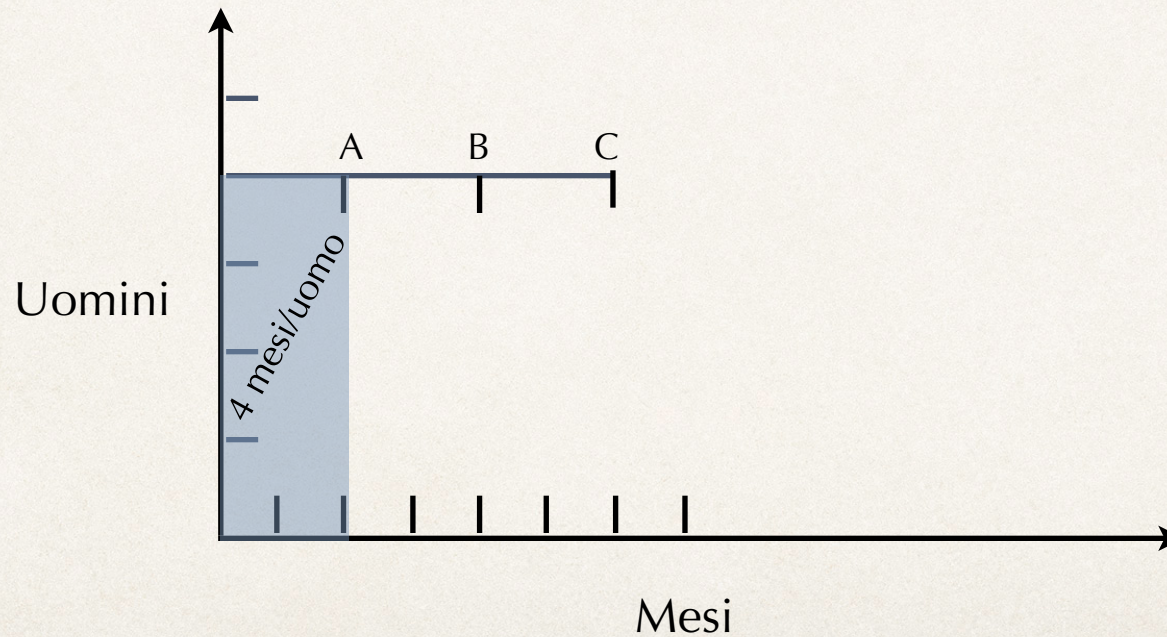


Esempio

- ❖ Facciamo l'ipotesi che l'errore sia stato solo nella valutazione dei tempi della prima milestone (caso A)
- ❖ Cosa fare per finire il progetto in tempo?
 - ❖ restano 8 mesi/uomo di lavoro
 - ❖ bisogna finire in 1 mese
 - ❖ sono necessari 8 programmatori
- ❖ Occorre aggiungere 4 programmatori ai 4 assegnati al progetto

Esempio

- ❖ Consideriamo invece il caso di errore uniformemente distribuito su tutta la durata del progetto (caso B)



Esempio

- ❖ Questo caso è equivalente a dire che tutti i tempi sono raddoppiati
- ❖ I mesi/uomo rimanenti non sono 8 ma sono 16
- ❖ Per finire il progetto entro il terzo mese come originariamente pianificato servono 16 persone che lavorino nell'ultimo mese
- ❖ Bisogna assegnare al progetto altri 12 programmatori...

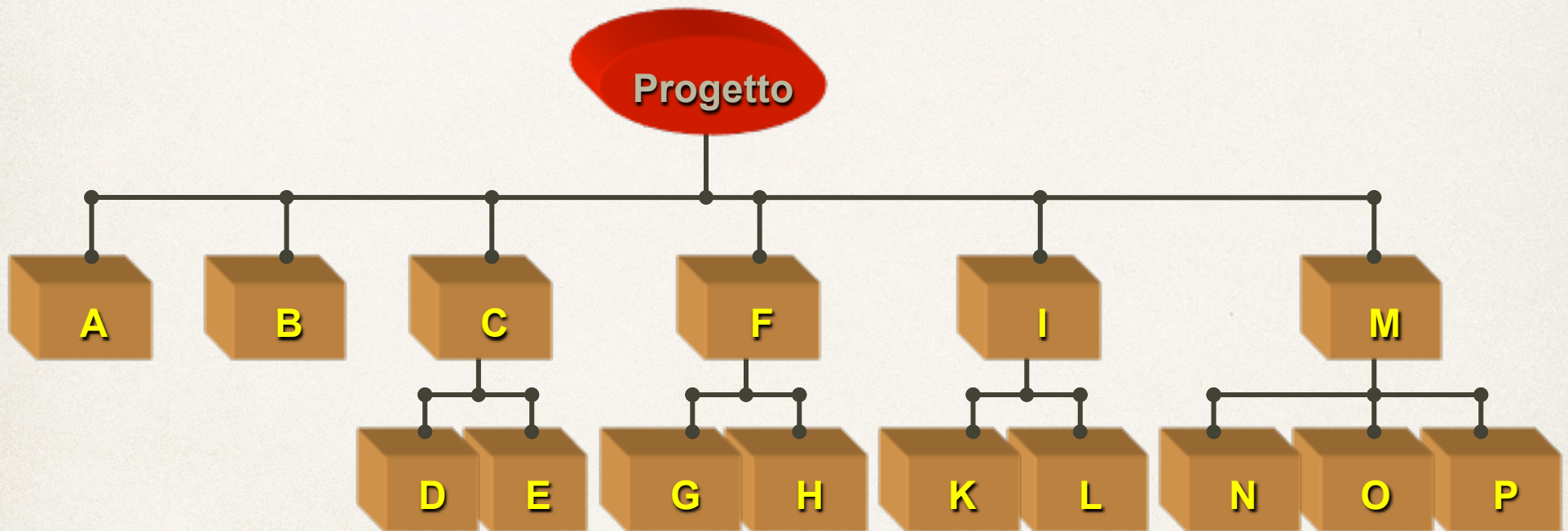
Esempio

- ❖ Fin qui abbiamo considerato un caso ideale. Torniamo al caso A e facciamo i conti con la realtà...
 - ❖ le 4 nuove persone vanno istruite
 - ❖ per istruirle uno dei programmatori deve usare parte del suo tempo per questo scopo. Gli servono 2 settimane.
- ❖ Quant'è il lavoro svolto nel terzo mese?
 - ❖ $(3 \times 1) + (1 \times 0,5) + (4 \times 0,5) = 5,5$
- ❖ Come potete vedere il conto semplicistico dei mesi/uomo può portarci fuori strada

Work Breakdown Structure (WBS)

- ❖ Applicare alla pianificazione il principio del “dividi e conquista”: il problema va risolto mediante la soluzione di sotto-problemi e la seguente integrazione delle soluzioni
- ❖ Definisce:
 - ❖ quali sono le attività da eseguire
 - ❖ quali sono i rapporti gerarchici tra le attività
- ❖ Facilita l'attività di pianificazione in quante, grazie alla decomposizione in attività più semplici, migliora la capacità di stimare:
 - ❖ tempi di realizzazione
 - ❖ costi di realizzazione
 - ❖ scopi realizzabili

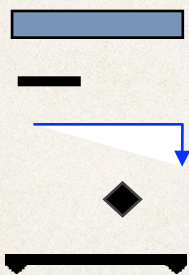
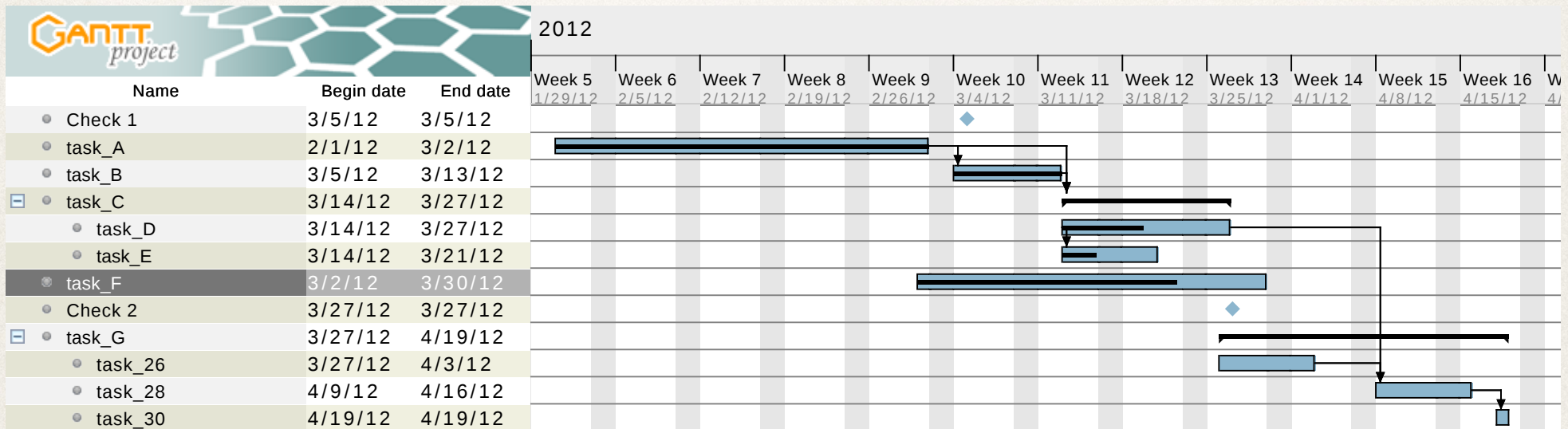
WBS



Diagrammi Gantt

- * Vogliamo aggiungere al WBS se seguenti informazioni
 - * informazioni sui tempi
 - * vincoli di precedenza nell'esecuzione delle attività
 - * informazioni sull'avanzamento delle attività
 - * punti di controllo (checkpoint, milestones)
- * Il diagramma di Gantt sono uno strumento sia in fase di pianificazione che in fase di monitoraggio
- * Sono diagrammi bi-dimensionali
 - * sull'asse X vi è il tempo (giorni, settimane, mesi)
 - * sull'asse Y vi sono le attività di progetto, prese dal WBS
 - * l'origine degli assi è l'inizio del progetto

Diagrammi Gantt



Rappresenta la durata stimata di una attività

Rappresenta lo stato di avanzamento dell'attività rispetto alla durata stimata

Esprime i vincoli di precedenza nella esecuzione di due attività

Rappresenta un punto di controllo del progetto

Rappresenta una attività composta

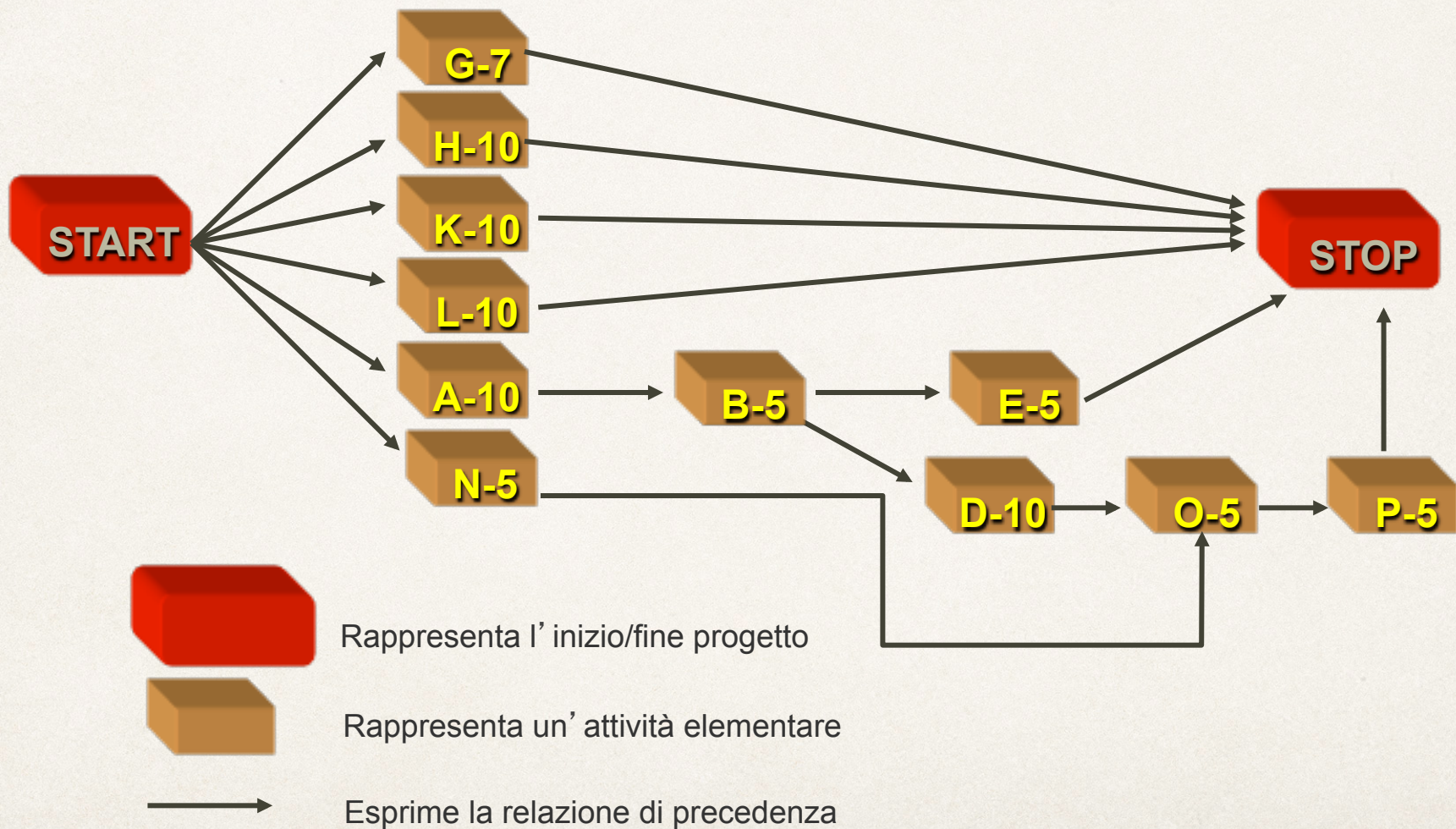
Gantt

- ❖ Esistono molte varianti del diagramma Gantt: tendono a sovrapporre in vario modo diversi modelli rappresentativi
- ❖ p.e.: sul diagramma si possono aggiungere i nomi degli esecutori delle attività, gli attrezzi necessari o vincoli di altro tipo
- ❖ Il vantaggio di questi diagramma consiste nell'avere in un colpo d'occhio le informazioni di sintesi circa il progetto e il suo andamento
- ❖ Esistono molti tool di supporto alla definizione, verifica e modifica di diagrammi Gantt

Diagrammi Pert

- ❖ Contengono le stesse informazioni dei diagrammi Gantt
- ❖ Di solito:
 - ♦ non forniscono informazioni sulla posizione temporale esatta delle attività del progetto, ma focalizzano l'attenzione sulla durata e sui vincoli di precedenza
 - ♦ mostrano solo le attività elementari, non mostrano le attività composte
 - ♦ sono uno strumento più snello da utilizzare, utili soprattutto nelle fasi di pianificazione preliminare

Diagrammi Pert



Algoritmi di trasformazione e pratiche di ottimizzazione

- ❖ **Processo:** è la descrizione concettuale delle attività da eseguire e dei manufatti in input e output necessari a ognuna di queste. Consideriamolo come una n-tupla $P=(A,I,O)$ dove
 - ❖ $A=\{A_i\}$ è l'insieme delle attività elementari A_i da eseguire
 - ❖ $I=\cup I_i$ è l'unione degli I_i , con I_i che è l'insieme dei manufatti in input necessari all'esecuzione dell'attività A_i
 - ❖ $O=\cup O_i$ è l'unione degli O_i , con O_i che è l'insieme dei manufatti in output risultanti dall'esecuzione dell'attività A_i
- ❖ **Piano di Progetto:** indica la sequenza temporale potenziale con cui sono eseguite le attività A dichiarate nel modello di processo
- ❖ **Piano Esecutivo:** indica la sequenza temporale reale delle attività A , considerati i vincoli di progetto e le decisioni manageriali circa tempi, costi, uomini (risorse)

Piano di progetto ed esecutivo

- * Lo scopo è essere flessibili durante la gestione di un progetto
 - * Il **piano di progetto** rappresenta l'ottimizzazione ottimale del progetto: pianifica la sequenza di esecuzione tenendo conto dei soli vincoli di precedenza imposti dal fabbisogno dei manufatti delle attività e non considerando vincoli di tempi, costi e personale
 - * Il **piano esecutivo** parte dalla pianificazione ottimale (che deriva dal piano di progetto) e tenendo conto delle precedenze la rimodula sulla base dei vincoli di tempi, costi e uomini
- * Questi piani si rappresentano utilizzando diagrammi Gantt o Pert.

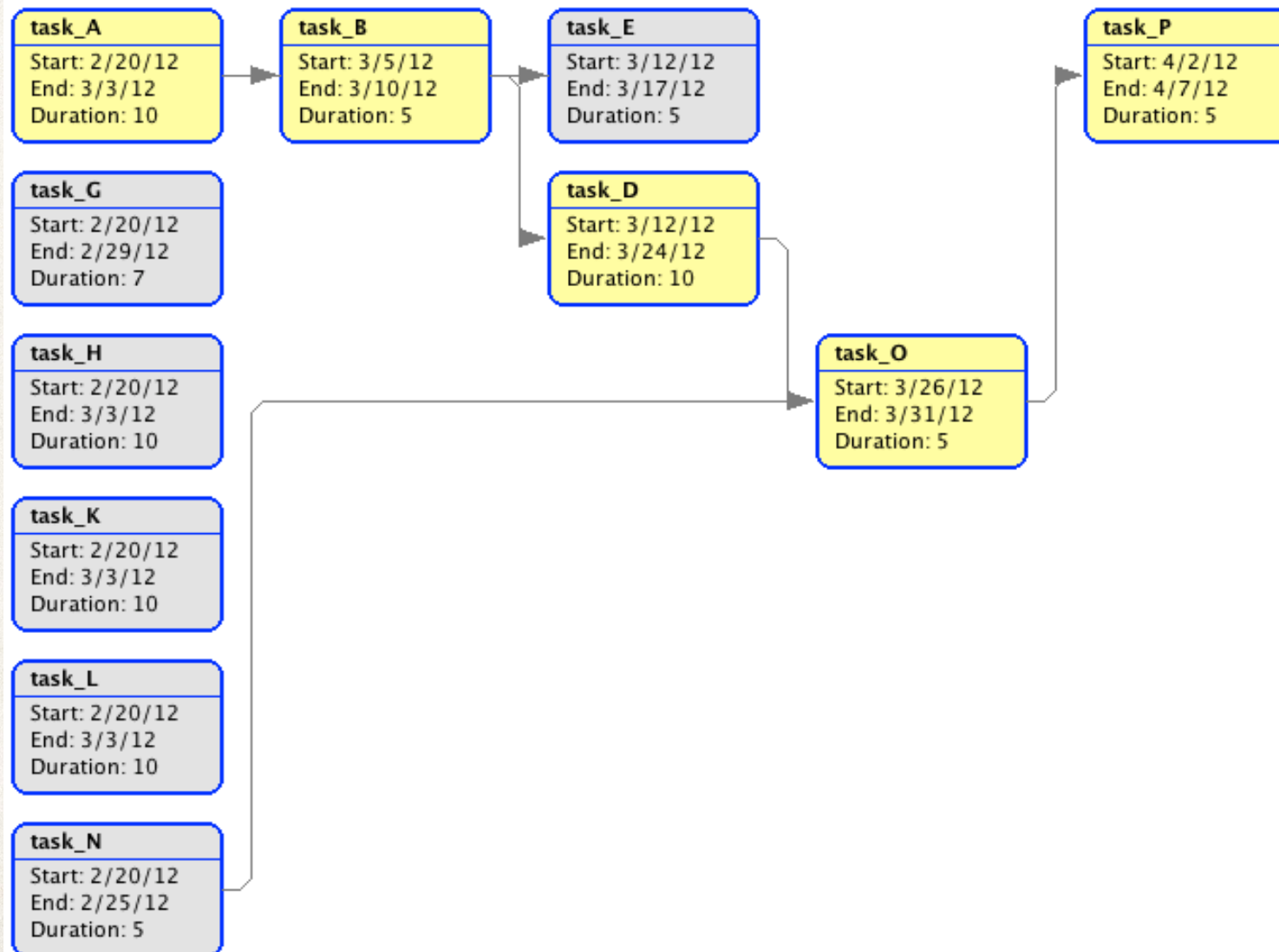
Trasformare il Processo in Piano di Progetto

1. Diagramma PERT: nodo START
2. inizializzare l'insieme $MD = \{\text{insieme tutti manufatti disponibili}\}$
3. per ogni attività $A_i \in A$ non ancora inclusa nel PERT
 - a. se $I_i \subset MD$ allora
 - (i) inserire nel diagramma PERT il nodo N_i corrispondente all'attività A_i
 - (ii) per ogni nodo N_j già nel diagramma se $O_j \cap I_i \neq \emptyset$ collegare N_i a N_j
 - (iii) $MD = MD \cup O_i$
4. Porre il nodo STOP
5. Collegare a STOP tutti gli N_i senza frecce uscenti

Esempio

Task	Sub-task	durata	dipende da
A		10	
B		5	A
C	D	10	B
C	E	5	B
F	G	7	
F	H	10	
I	K	10	
I	L	10	
M	N	5	
M	O	5	D,N
M	P	5	P

Esempio



Trasformazione del Piano di Progetto in Piano Esecutivo

- ❖ Il Piano di Progetto è la prima versione del piano esecutivo
- ❖ Le eventuali modifiche derivano dall'analisi dei tempi, costi, uomini disponibili
 - le attività in sequenza nel Piano di Progetto devono rimanere tali anche nel Piano Esecutivo
 - le attività in parallelo nel Piano di Progetto
 - ♦ restano in parallelo nel Piano Esecutivo se ci sono risorse sufficienti
 - ♦ sono rese sequenziali se non sono disponibili risorse a sufficienza

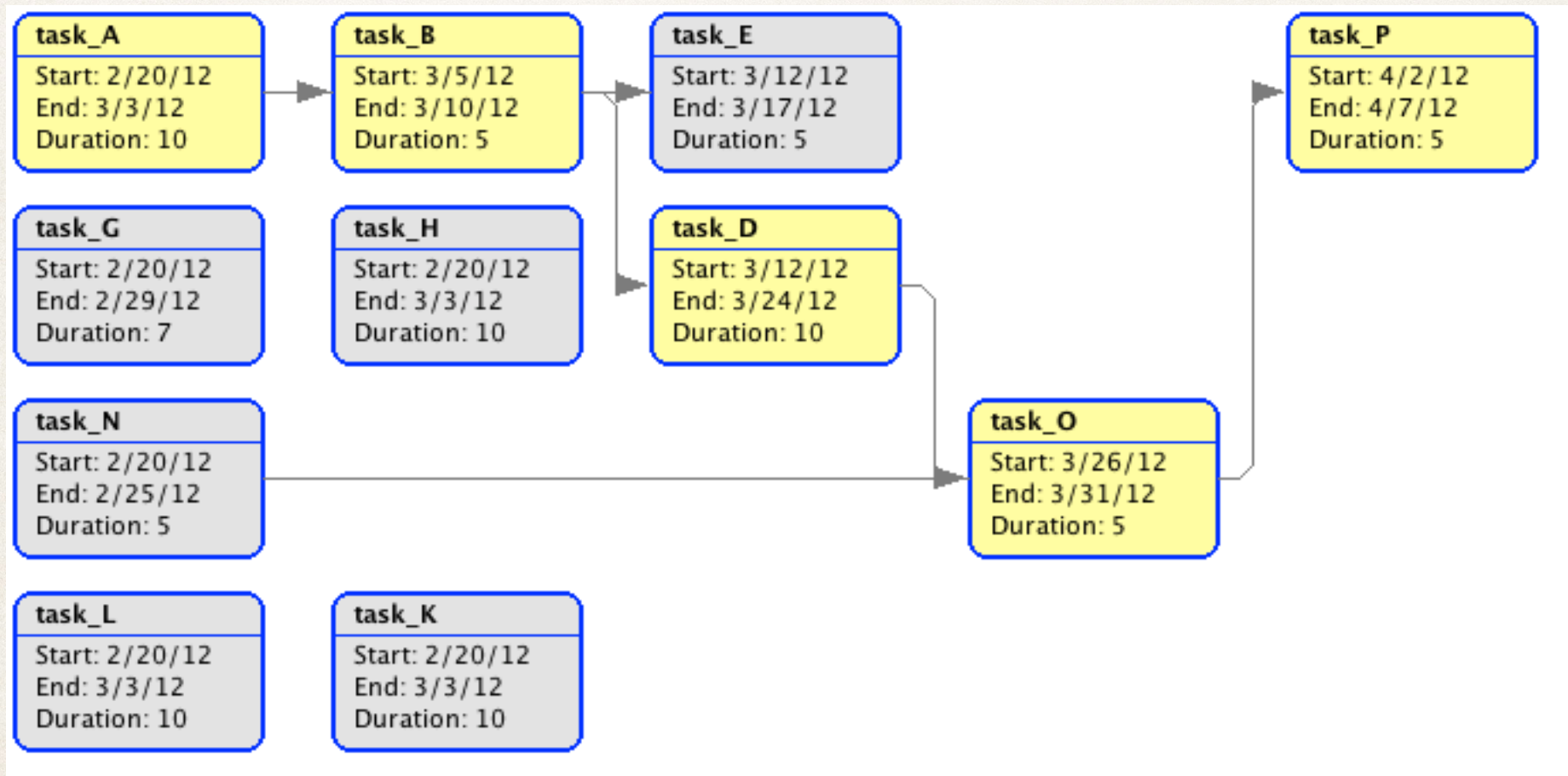
Pratiche di ottimizzazione

- ❖ Il piano esecutivo può venire rivisto più volte, in conseguenza del contesto in cui si opera. A ogni modifica si verifica il rispetto dei vincoli di progetto
- ❖ Generalmente i raffinamenti del piano esecutivo vertono sui possibili parallelismi
- ❖ Nel caso si debbano eseguire in sequenza due attività potenzialmente eseguibili in parallelo bisogna valutare la priorità d'esecuzione. Se la priorità è pari si può valutare
 - ❖ **efficacia del progetto:** si esegue prima l'attività che produce manufatti intermedi che sono più immediatamente tracciabili con i manufatti di output; in questo modo è più vicina la produzione dei deliverable per il committente
 - ❖ **flessibilità della schedulazione:** si esegue per prima l'attività che produce manufatti con più alto scope (numero attività che utilizzano il manufatto) in modo che possano partire più attività

Pratiche di ottimizzazione

- ❖ Ogni raffinamento vuole ottimizzare il piano rispetto ai nuovi vincoli circa tempi, costi, uomini
 - Tempi: come ottimizzarli?
 - ♦ aumentando il personale
 - ♦ possibile ottimizzare un piano esecutivo che si discosta dal piano di progetto facendo tendere il primo a quest'ultimo
 - ♦ se non basta, rivisitare le attività sui cammini critici
 - Costi: come ottimizzarli?
 - ♦ bilanciare qualità prodotto finale tra riduzione risorse coinvolte e relativo aumento tempi di esecuzione
 - ♦ out-sourcing: demando l'esecuzione di alcune attività a terzi a costi inferiori
 - ♦ riduco le persone accettando aumenti di tempo
 - Uomini
 - ♦ posso ottimizzare l'allocazione nel tempo agendo sui possibili parallelismi
 - ♦ posso ridurre l'allocazione sacrificando tempi di progetto

Esempio: ottimizzazione persone

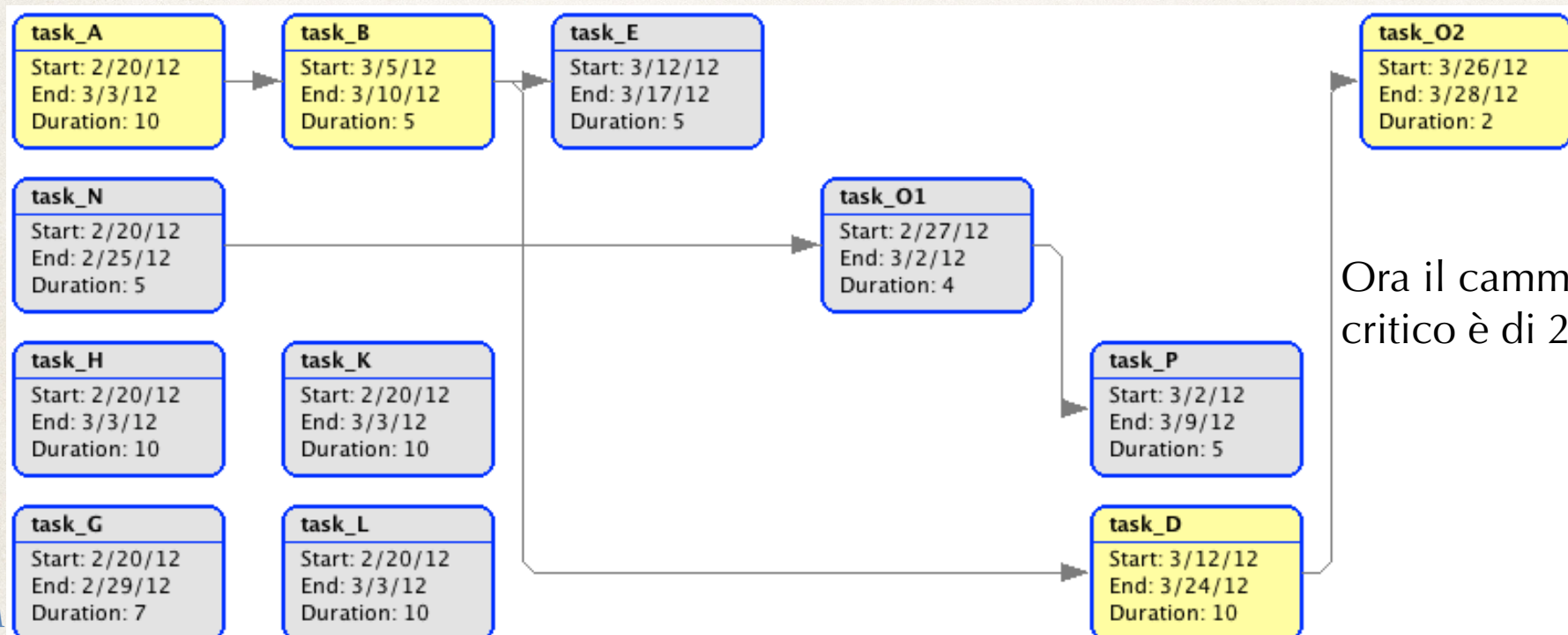


Analisi cammini critici

- ❖ Il cammino critico è quello che determina la lunghezza di un progetto. Un qualsiasi ritardo su di esso determina automaticamente uno slittamento dell'intero progetto
- ❖ In generale, l'aumentare del cammino critico decresce la flessibilità del progetto
- ❖ Per ottimizzare oltre quanto consentito dai parallelismi, bisogna investigare la possibilità di ristrutturare le attività per ridurre i cammini critici.
- ❖ Prima si fa e meno costa: è molto oneroso (e non sempre possibile) farlo quando il progetto è già in esecuzione. Andrebbe fatto prima di avviare l'esecuzione del progetto

Esempio: analisi cammini critici

- ❖ L'attività O può essere rivista, e spezzata in due attività (e le due parti possono essere maggiori del tutto...)
- ❖ O1, dipendente da N e di durata 4 giorni
- ❖ O2, dipendente da D e di durata 2 giorni



Conclusioni

- ❖ Una buona gestione è fondamentale per il successo di un progetto
- ❖ Purtroppo il software è intangibile, e questo non ci aiuta...
- ❖ I Manager hanno diversi compiti: tra i più significativi vi è la pianificazione, la stima dei costi e lo scheduling
- ❖ La pianificazione e la stima dei costi sono processi iterativi che continuano per tutta la durata del progetto