

Copyright © 2004 Moreno Marzolla

This work is licensed under the Creative Commons Attribution-Noncommercial-Share Alike 2.5 Italy License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-sa/2.5/it/> or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.

Cos'è un progetto

- Un progetto è un insieme ben definito di attività che
 - ha un inizio
 - ha una fine
 - realizza un obiettivo
 - è realizzato da un equipe di persone
 - utilizza un certo insieme di risorse
 - non è riconducibile a routine

Software Project Management

- Sono le attività necessarie per assicurare che un prodotto software sia sviluppato rispettando le **scadenze** fissate e risponda a determinati **standard**
- Interazione di aspetti *economici e tecnici*
- Un progetto diretto bene qualche volta fallisce, uno diretto male fallisce sicuramente
- L'importanza dell'esperienza

Perché è importante la gestione dei progetti?

- L'Ingegneria del Software è una attività economica, quindi è soggetta a vincoli di tipo economico
- Progetti ben gestiti *talvolta* falliscono. Progetti malgestiti falliscono *sicuramente*.
- L'obbiettivo del corso è di illustrare l'attività di gestione dei progetti, non insegnare a diventare manager...
 - ...lo diventerete con l'esperienza.

Problemi

- Un prodotto software è intangibile: per valutare i progressi ci si deve basare sulla documentazione
- L'ingegneria del software non è ancora riconosciuta come disciplina solida al pari dell'ingegneria meccanica, elettrica...
- Non ci sono standard per il processo di produzione software
- Ogni progetto ha una storia a sé

Motivi per cui un progetto può essere in ritardo

- Una *scadenza non realistica*, fissata e imposta da una persona esterna allo staff tecnico
- *Mutamenti* nei requisiti imposti dal cliente
- Una *stima ottimistica* del lavoro e delle risorse necessarie a compierlo
- *Rischi* che non sono stati tenuti nel dovuto conto all'inizio del progetto
 - Difficoltà tecniche imprevedibili
 - Difficoltà umane imprevedibili
- *Problemi di comunicazione* fra i membri dello staff
- L'incapacità da parte della direzione del progetto di *riconoscere che c'è un ritardo*, e la conseguente mancata applicazione di misure che pongano rimedio

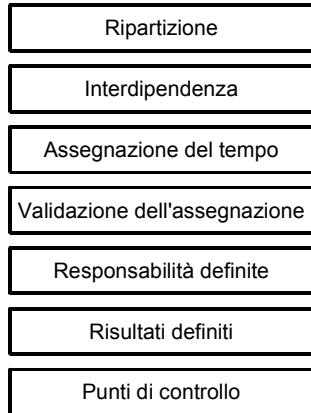
Lo scenario peggiore / 1

- Supponiamo di dover sviluppare un software entro 9 mesi
- Dopo un'accurata stima e analisi dei rischi, giungete alla conclusione che la realizzazione di quanto richiesto richiede non meno di 14 mesi
- Che fare?

Lo scenario peggiore / 2

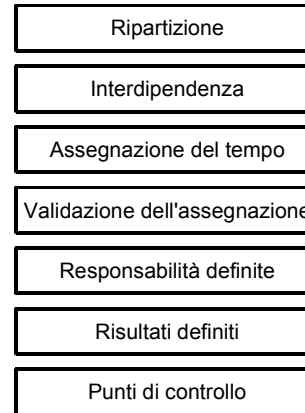
- **Alternativa 1:**
Aumentare il budget e aggiungere risorse
 - Però si aumentano i rischi di ottenere una qualità scarsa dati i tempi troppo stretti. Inoltre non sempre si può fare.
- **Alternativa 2:**
Eliminare parte delle funzionalità non essenziali
 - Ossia fornire all'inizio un prototipo che implementi solo l'indispensabile; il resto verrà completato entro 14 mesi
- **Alternativa 3:**
Far finta di niente e tentare in 9 mesi
 - L'insuccesso è (pressoché) assicurato

Principi fondamentali per la pianificazione temporale / 1



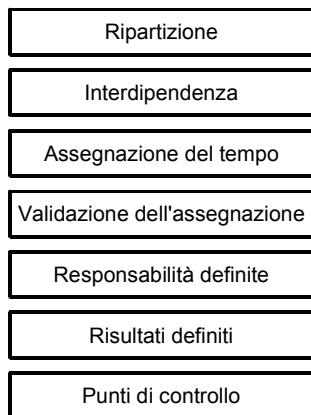
- Un progetto deve essere ripartito in attività e compiti di dimensioni ragionevoli

Principi fondamentali per la pianificazione temporale / 2



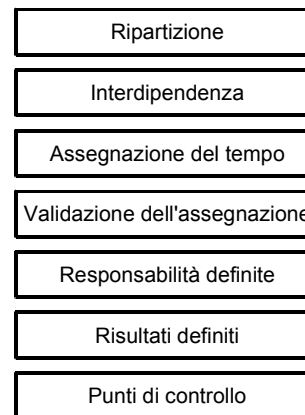
- Occorre determinare le dipendenze reciproche fra le attività e i compiti.
- Alcuni compiti devono essere svolti in sequenza, altri in parallelo
- Alcune attività non possono cominciare prima del termine di altre

Principi fondamentali per la pianificazione temporale / 3



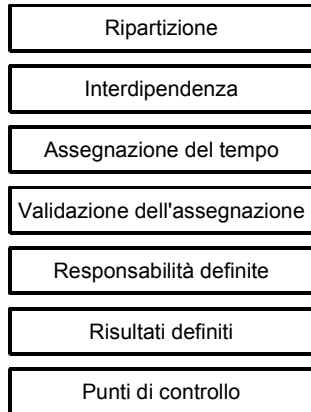
- A ogni compito si deve assegnare un certo numero di “unità di lavoro”
 - Es. giorni/uomo
- Ad ogni compito occorre associare la data di inizio e fine

Principi fondamentali per la pianificazione temporale / 4



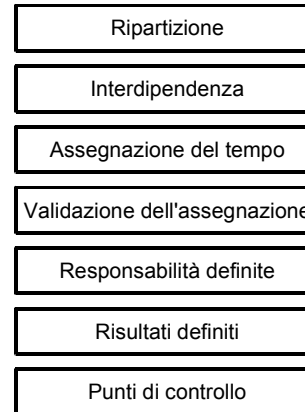
- A ogni progetto è assegnato un numero definito di membri nello staff
- E' necessario assicurarsi di non aver assegnato più persone di quelle necessarie

Principi fondamentali per la pianificazione temporale / 5



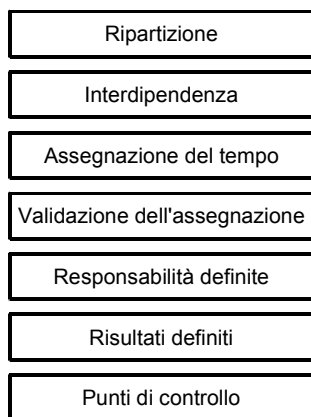
- Ogni compito deve essere assegnato ad un membro del team

Principi fondamentali per la pianificazione temporale / 6



- Ogni compito deve produrre un risultato predefinito

Principi fondamentali per la pianificazione temporale / 7



- Ad ogni compito si deve associare un *“punto di controllo”*
- Un punto di controllo è compiuto quando la qualità di uno o più semilavorati è stata esaminata e approvata

I giocatori in campo...

- Senior managers
 - definiscono i termini economici del progetto
- Project managers
 - pianificano, motivano, organizzano e controllano lo sviluppo
- Practitioners
 - hanno le competenze tecniche per realizzare il sistema
- Customers
 - specificano i requisiti del software da sviluppare
- End users
 - interagiscono con il sistema una volta realizzato

Attività del project manager

- Stesura della proposta di progetto
- Stima del costo del progetto
- Pianificazione (planning) e temporizzazione (scheduling)
 - Individuare le attività, le *milestones* e i prodotti intermedi (*deliverables*) del progetto
- Monitoraggio e revisioni del progetto
- Selezione e valutazione del personale
- Stesura di rapporti e presentazioni

Pianificazione del progetto

Stabilire i vincoli del progetto
Stimare i parametri del progetto
Definire milestones e deliverables
While (il progetto è completato o cancellato) **loop**
 Definire lo schedule
 Iniziare le attività in base allo schedule
 Attendere...
 Valutare i progressi del progetto
 Rivedere le stime dei parametri del progetto
 Aggiornare lo schedule
 Rinegoziare i vincoli del progetto e i deliverables
if (ci sono problemi) **then**
 Iniziare una revisione tecnica
endif
endloop

Il Piano del Progetto / 1

1. Introduzione
 - Definisce gli obiettivi del progetto e fissa i vincoli (costi, tempo...)
2. Organizzazione del progetto
 - Come è organizzato il team di sviluppo, quali sono le persone coinvolte e i loro ruoli
3. Analisi dei rischi
 - Quali sono i rischi previsti, la probabilità con cui si possono manifestare, quali sono le strategie per ridurre questi rischi

Il Piano del Progetto / 2

4. Risorse Hardware e Software richieste
 - Se l'hardware deve essere acquisito, il suo prezzo e il tempo stimato di consegna devono essere presi in considerazione
5. Suddivisione del lavoro
 - Il progetto viene suddiviso in attività; vengono identificati i *deliverables* e le *milestones*
6. Scheduling del progetto
 - Descrive le dipendenze tra le attività, il tempo stimato richiesto per il completamento delle *milestones* e l'allocazione di personale alle attività

Il Piano del Progetto / 3

- 7. Controllo e Rapporto sulle attività
 - Descrive 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.

Fattori di Fallimento

- Requisiti incompleti 13.1%
- Mancato coinvolgimento del cliente 12.4%
- Mancanza di risorse 10.6%
- Aspettative non realistiche 9.9%
- Mancanza di supporto esecutivo 9.3%
- Cambiamenti dei requisiti 8.7%

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%

Organizzazione delle attività

- Le attività del progetto devono produrre dei documenti affinché i manager possano osservarne lo stato di avanzamento
- *Milestones* sono i punti finali di ciascuna attività
- *Deliverables* sono i risultati del progetto consegnati ai clienti
- Il modello a cascata consente di definire banalmente le milestones
 - Le attività sono già ben definite

Scheduling di un progetto

- Suddividere il progetto in “*compiti*” (*tasks*) e stimare il tempo e le risorse necessarie per completare ogni compito
- Organizzare in compiti concorrentemente per fare un uso ottimale della forza lavoro
- Minimizzare le dipendenze tra task per evitare i ritardi causati da un task in attesa della terminazione di un altro
- Uno scheduling “ottimale” spesso dipende dall'intuizione e dall'esperienza del project manager

Problemi di scheduling

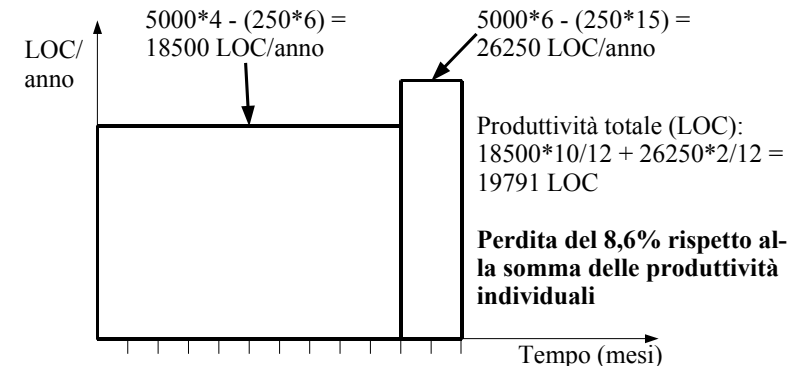
- Stimare la difficoltà di un problema (e quindi i costi di sviluppare una soluzione) è difficile
- La produttività **non** è proporzionale al numero di persone che lavorano ad un task
- *Ciò che è inatteso si verifica puntualmente.* Cercare sempre di sviluppare delle strategie per affrontare gli imprevisti
- Regola 40-20-40
 - 40% del tempo è impiegato nell'analisi e la progettazione
 - 20% del tempo è speso nella stesura del codice
 - 40% del tempo è speso nei collaudi finali

Un esempio / 1

- Prendiamo quattro tecnici, ciascuno con una produttività stimata di 5000 LOC/anno
- Li riuniamo in gruppo
 - Si creano canali di comunicazione che assorbono produttività
 - Stimiamo 250 LOC/anno in meno per ogni canale
 - Produttività totale: $5000*4 - (250*6) = 18500$ LOC/anno
 - Perdita del 7,5% rispetto la somma delle produttività individuali

Un esempio / 2

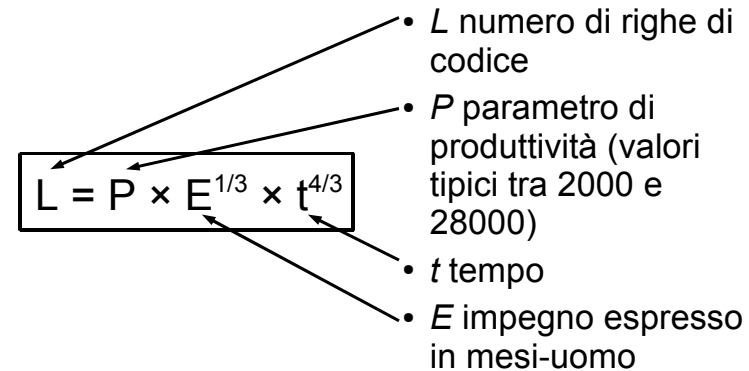
- Quando mancano due mesi alla fine, aggiungiamo due nuovi membri



Generalizzare non è possibile

- La relazione tra numero di persone e produttività non è lineare
- Dobbiamo concludere che il lavoro di gruppo è controproducente?
 - No, dato che la comunicazione tra membri del team serve a migliorare la qualità del software
 - Le revisioni tecniche formali possono portare ad un miglioramento dell'analisi e possono ridurre il numero di errori e quindi il tempo di collaudo e debug

Una relazione empirica

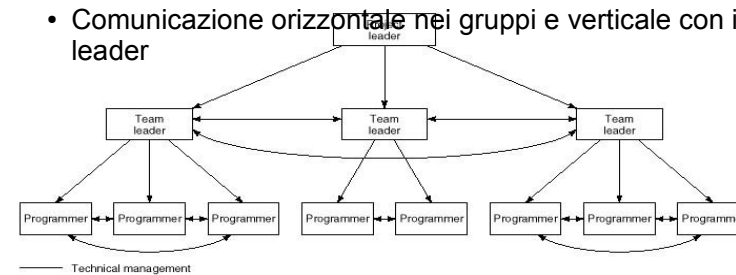


Tipologie di team / 1

- 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” (ad esempio per la ricerca)
- Svantaggi
 - È difficile da imporre...
 - Non è scalabile...

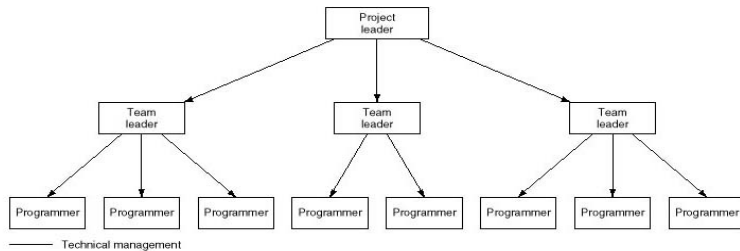
Tipologie di team / 2

- Controllato Decentralizzato
 - Un leader riconosciuto, che coordina il lavoro
 - La risoluzione dei problemi è di gruppo, ma l'implementazione delle soluzioni è assegnata a sottogruppi
 - Comunicazione orizzontale nei gruppi e verticale con il leader



Tipologie di team / 3

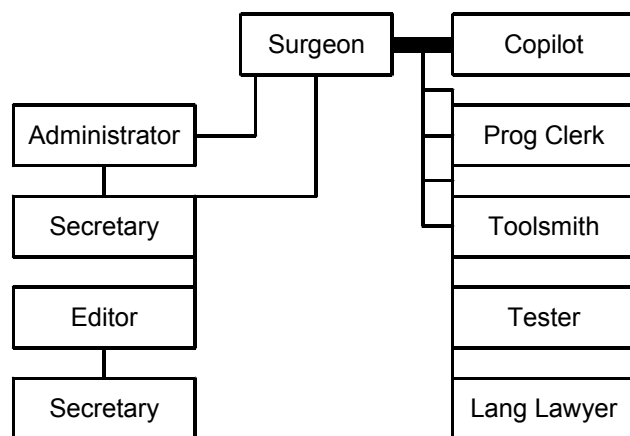
- Controllato Centralizzato
 - Il team leader decide sulle soluzioni e sull'organizzazione
 - Comunicazione verticale tra team leader e gli altri membri



Tipologie di team / 4

- Proposta di Mills
 - Surgeon (chief programmer)
 - Copilot
 - Administrator
 - Editor
 - Secretaries
 - Program clerk
 - Toolsmith
 - Tester
 - Language Lawyer

Pattern di comunicazione



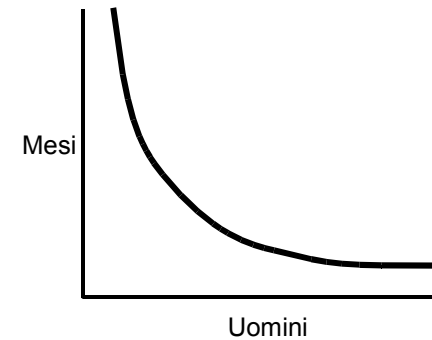
Il mitico *mese-uomo*

- Molti progetti software sono falliti a causa di mancanza di tempo sufficiente
- Perché?
 - Le tecniche di stima dei tempi di completamento non sono affidabili. In più, si fa l'assunzione implicita che tutto andrà bene
 - Le tecniche di stima confondono l'impegno con i progressi fatti. Cioè, assumono che uomini e mesi siano intercambiabili
 - I progressi sono mal verificati.
 - Se qualcosa è in ritardo, la prima cosa che si fa è aggiungere personale. Ciò non fa che peggiorare le cose.

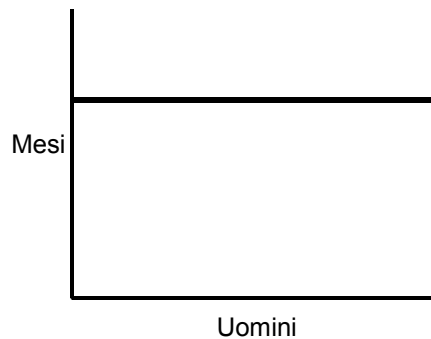
Il mese-uomo

- Il *mese-uomo* è l'unità di misura tipica dell'impegno (effort) richiesto per un compito
- I costi in effetti sono proporzionali a:
 $numero_mesi * numero_uomini$
- I progressi compiuti **NO**
- Mesi e uomini sono intercambiabili solo per compiti che possono essere partizionati perfettamente tra i lavoratori, *senza bisogno di comunicazione tra essi*.

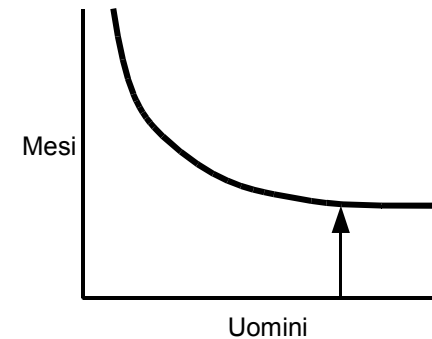
Mesi vs Uomini: task perfettamente partizionabile



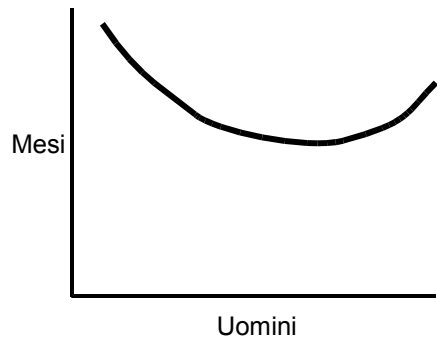
Mesi vs Uomini: task non partizionabile



Mesi vs Uomini: task partizionabile che richiede comunicazione

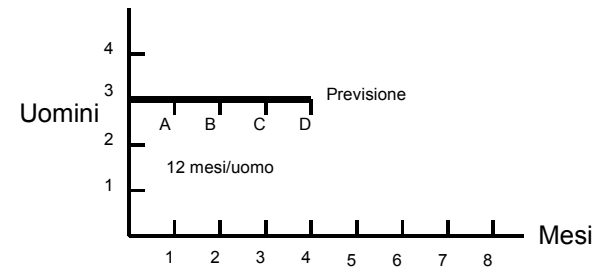


Mesi vs Uomini: task partizionabile in cui tutti comunicano con tutti



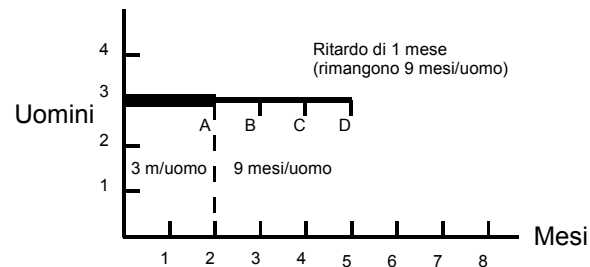
Esempio

- Un progetto perfettamente partizionabile richiede 12 mesi/uomo. Viene assegnato a 3 uomini per 4 mesi. Sono fissate delle scadenze A, B, C, D alla fine di ogni mese.



Esempio

- Supponiamo che la prima scadenza A venga raggiunta dopo due mesi

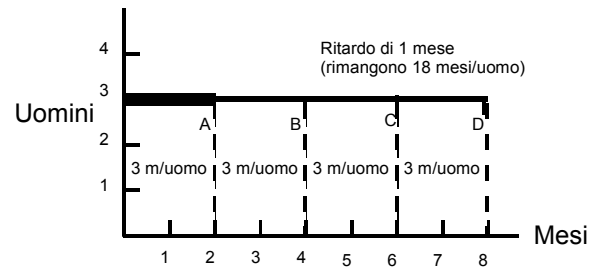


Esempio

- Supponiamo che l'errore stia solo nella valutazione della prima scadenza, la quale ha richiesto più del previsto (caso A)
- Se vogliamo finire il progetto in tempo:
 - Rimangono 9 mesi/uomo di lavoro
 - Devono essere completati in 2 mesi
 - Gli uomini richiesti sono allora: $9 / 2 = 4,5$
- Aggiungere 2 uomini ai 3 assegnati al progetto.

Esempio

- Supponiamo che l'errore sia uniformemente distribuito lungo tutta la durata del progetto (caso B)



Esempio

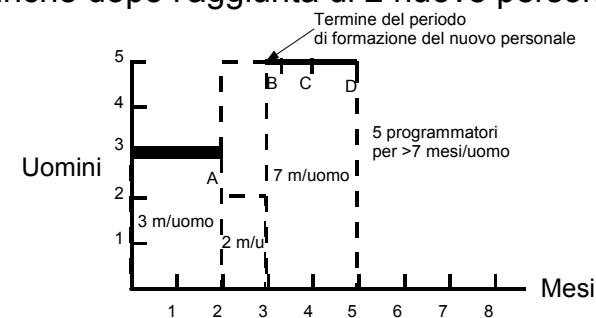
- In questo caso, tutti i tempi vanno dilatati per due
- I mesi uomo rimanenti sono dunque 18
- Per finire il progetto in tempo, entro il quarto mese:
 - Gli uomini richiesti sono $18 / 2 = 9$
- Aggiungere 6 uomini ai 3 assegnati al progetto

Esempio

- Riconsideriamo il caso A in condizioni più realistiche.
 - Le due nuove persone devono essere istruite
 - Una delle persone deve essere distolta dal proprio lavoro per istruire i nuovi arrivati. Supponiamo questo richieda 1 mese.
 - I rimanenti 2 programmatori lavoreranno durante il mese di addestramento
 - Al termine dell'addestramento, rimangono almeno 7 mesi/uomo di lavoro ancora da compiere, e 5 uomini
 - Ripartizionare il lavoro richiede altro tempo
 - In definitiva, rimangono molti più di 7 mesi/uomo di lavoro

Esempio

- Considerando il lavoro extra per ripartizionare i compiti, alla fine il lavoro è comunque in ritardo anche dopo l'aggiunta di 2 nuove persone



Esempio

- Volendo ignorare il tempo per ripartizionare il task, e volendo finire il lavoro entro il quarto mese, servirebbero 7 uomini
 - 7 mesi/uomo di lavoro ancora da compiere dopo il terzo mese. Il tutto da fare in un mese.
- Ma a questo punto ci ritroviamo con un team di 7 persone, contro le 3 originariamente previste.

La lezione di tutto ciò

- La massima del giorno

“Adding manpower to a late project makes it later”

Frederick P. Brooks, Jr
“The Mythical Man-Month”

Selezione del personale

- Potrebbe non essere possibile assumere le persone “ideali” per lavorare su un dato progetto
 - Il budget del progetto potrebbe non consentire l'uso di personale altamente qualificato (e pagato...)
 - Può non essere disponibile il personale con un adeguato livello di preparazione
 - Può essere desiderabile sviluppare la preparazione del personale facendolo lavorare su un progetto software

Diagrammi di Gantt e Activity Network

- Sono notazioni grafiche usate per illustrare lo schedule di un progetto
- Mostrano il progetto suddiviso in compiti (tasks).
 - I compiti non devono essere troppo piccoli; dovrebbero richiedere una settimana o due per essere completati
- Le Activity Network (*grafi delle attività, o grafi delle dipendenze*) mostrano le dipendenze tra task e il percorso critico (*critical path*)
- I diagrammi di Gantt mostrano lo scheduling come calendario lavori

Esempio

- Considerare un progetto composto dai task seguenti, con le durate e le dipendenze indicate:

Task	giorni	Dipende da:
T1	10	
T2	7	T1
T3	15	T2
T4	10	T2, T5
T5	5	T1
T6	4	T3, T4
T7	3	T4
T8	10	T6, T7

Grafo delle dipendenze tra task

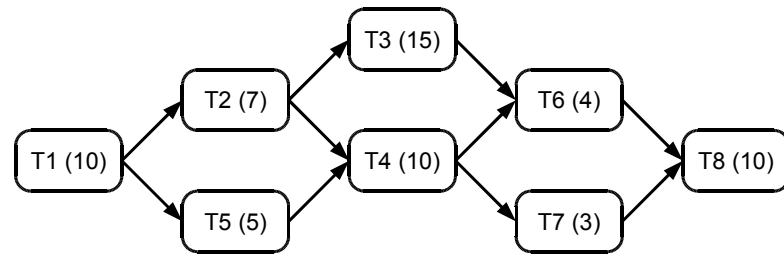
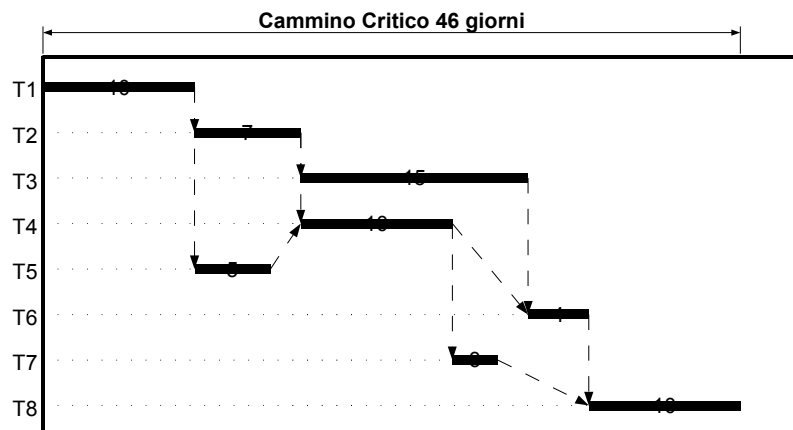
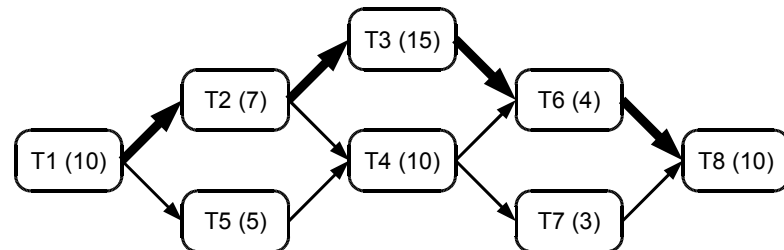


Diagramma dei tempi (Gantt chart)



Cammino Critico



Conclusioni

- Una buona gestione è fondamentale per il successo di un progetto
- La natura intangibile del software non aiuta...
- I Manager hanno diversi compiti, ma i più significativi riguardano la pianificazione, la stima dei costi e lo scheduling
- La pianificazione e la stima dei costi sono processi iterativi che continuano lungo tutta la durata del progetto.