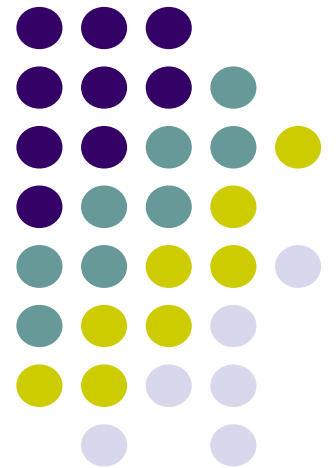


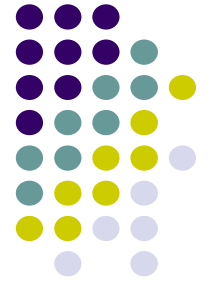
# Software project management

---

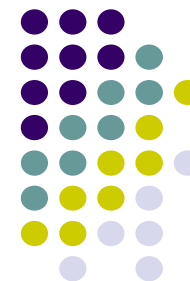
[www.vincenzocalabro.it](http://www.vincenzocalabro.it)



# Software project management

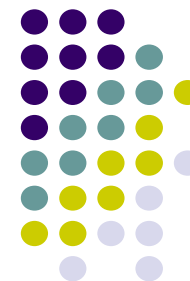


- Sono le **attività** necessarie per assicurare che un prodotto software sia sviluppato
  - rispettando le **scadenze** fissate
  - rispondendo a determinati **standard**
- Interazione di aspetti economici e tecnici
- Un progetto diretto bene qualche volta fallisce, uno diretto male fallisce sicuramente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



# Che 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”



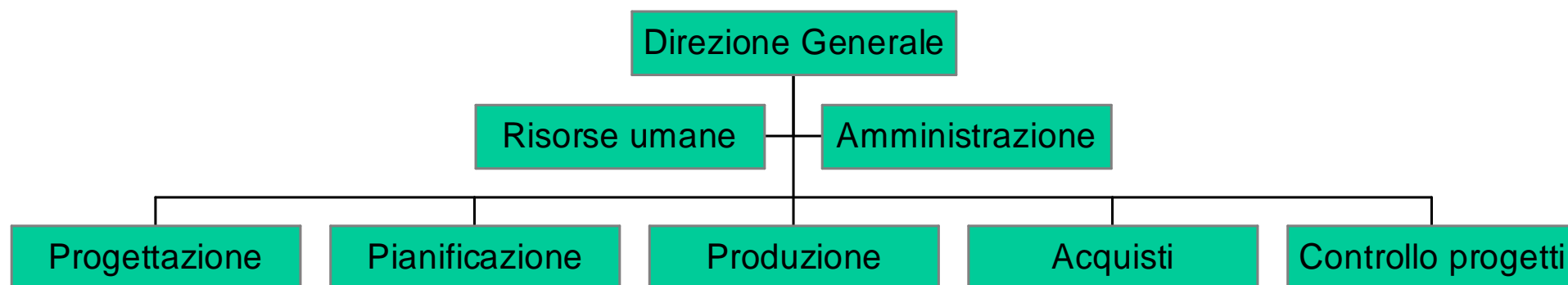
# Struttura organizzativa

- **Per funzione:** non è assegnato personale allo specifico progetto e non c'è un coordinatore delle attività;
- **Per progetto:** esiste un coordinatore (project manager) a tempo pieno ed il personale che opera sul progetto dipende direttamente da lui;
- **A matrice:** esiste un coordinatore a tempo pieno o parziale ed il personale che opera sul progetto è in parte assegnato al team di progetto.

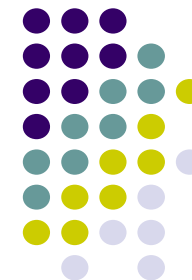
# Struttura Organizzativa



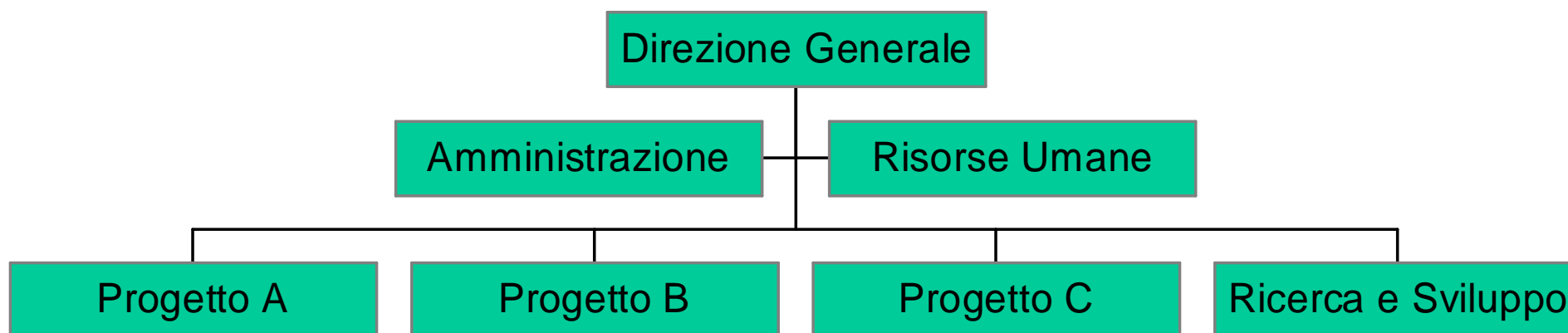
Organizzazione per funzione

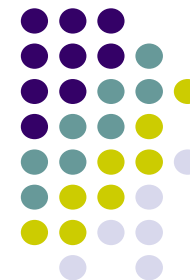


# Struttura Organizzativa



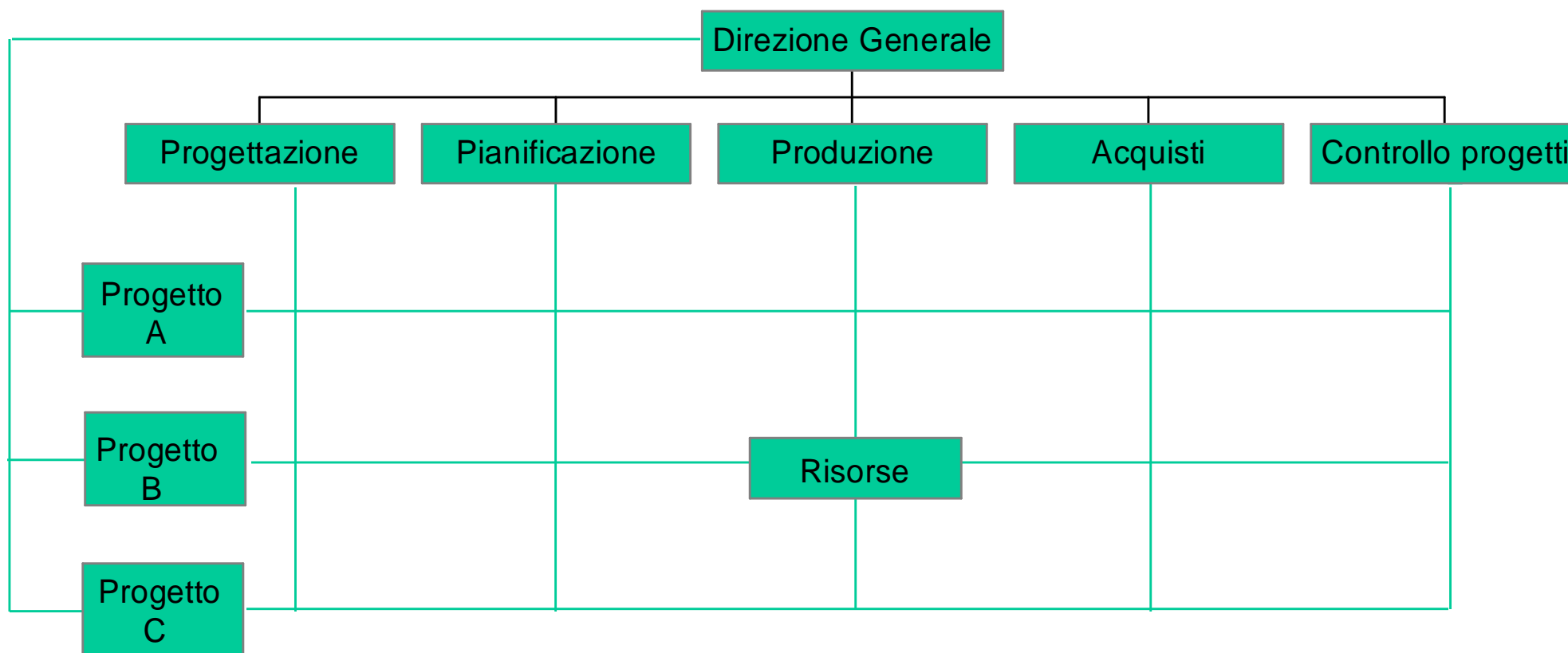
Organizzazione per progetto





# Struttura Organizzativa

Organizzazione a matrice

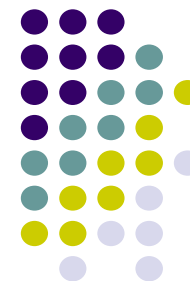




# Struttura Organizzativa

Tipo di organizzazione	Caratteristiche	Pregi	Difetti
Per funzione	<ul style="list-style-type: none"> <li>- Compiti e ruoli rigidi</li> <li>- Un solo diretto superiore</li> <li>- Specializzazione professionale</li> </ul>	<ul style="list-style-type: none"> <li>- centralizzazione di risorse simili</li> </ul>	<ul style="list-style-type: none"> <li>- Scarso coordinamento nell'ambito del progetto,</li> <li>- scarso coinvolgimento delle risorse (il progetto transita in più aree)</li> </ul>
Per progetto	<ul style="list-style-type: none"> <li>- Risorse allocate al team fino a conclusione del progetto</li> </ul>	<ul style="list-style-type: none"> <li>- Buon coordinamento e comunicazione all'interno del team</li> <li>- elevata autorità del project manager</li> </ul>	<ul style="list-style-type: none"> <li>- Poco efficiente utilizzo delle risorse</li> </ul>
A matrice	<ul style="list-style-type: none"> <li>- Centralità del progetto</li> <li>- partecipazione al progetto delle diverse funzioni aziendali</li> <li>- due superiori diretti</li> </ul>	<ul style="list-style-type: none"> <li>- Ottimizzazione delle risorse</li> </ul>	<ul style="list-style-type: none"> <li>- Conflitti tra project manager e Functional manager</li> </ul>





# Struttura Organizzativa

- Quale struttura scegliere:
  - Struttura per funzione - progetti semplici che richiedono elevata specializzazione del personale
  - Struttura per progetti - progetti complessi o molto grandi
  - Struttura a matrice - progetti mediamente complessi



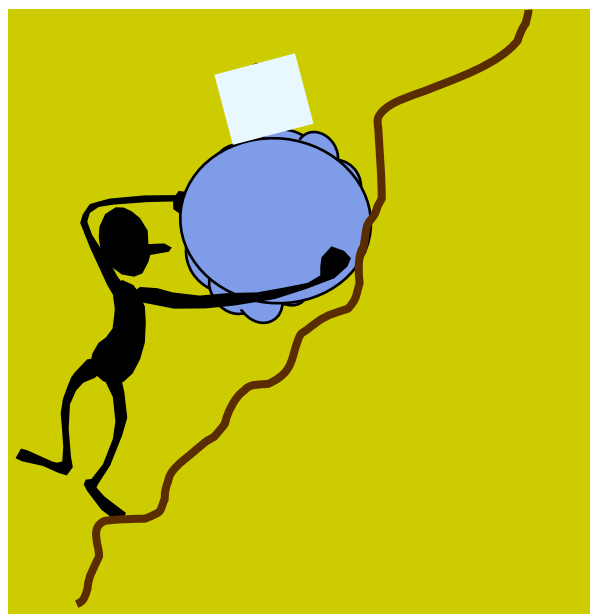
# I giocatori in campo...

- Business 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



# Perché c'è bisogno di un team?

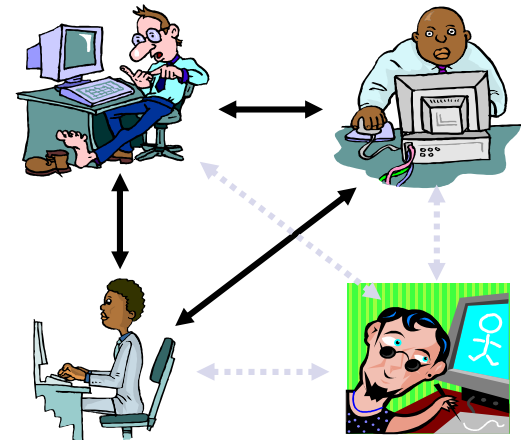
- La maggior parte dei progetti software sono troppo impegnativi per essere realizzati da una sola persona



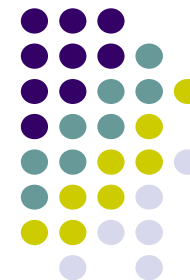
# The mythical man/month



- Perché non calcolare la “forza lavoro” in termini di mesi uomo necessari?
  - $\text{Persone/mese} * \text{Tempo allocato} = \text{Numero\_Persone\_Necessarie}$
- ✓ Alcuni compiti possono essere condivisi, altri no  
Esempio: raccogliere fragole vs. produrre un bambino
- ✓ overhead necessario per il coordinamento e la comunicazione



# Tipologie di team (1)



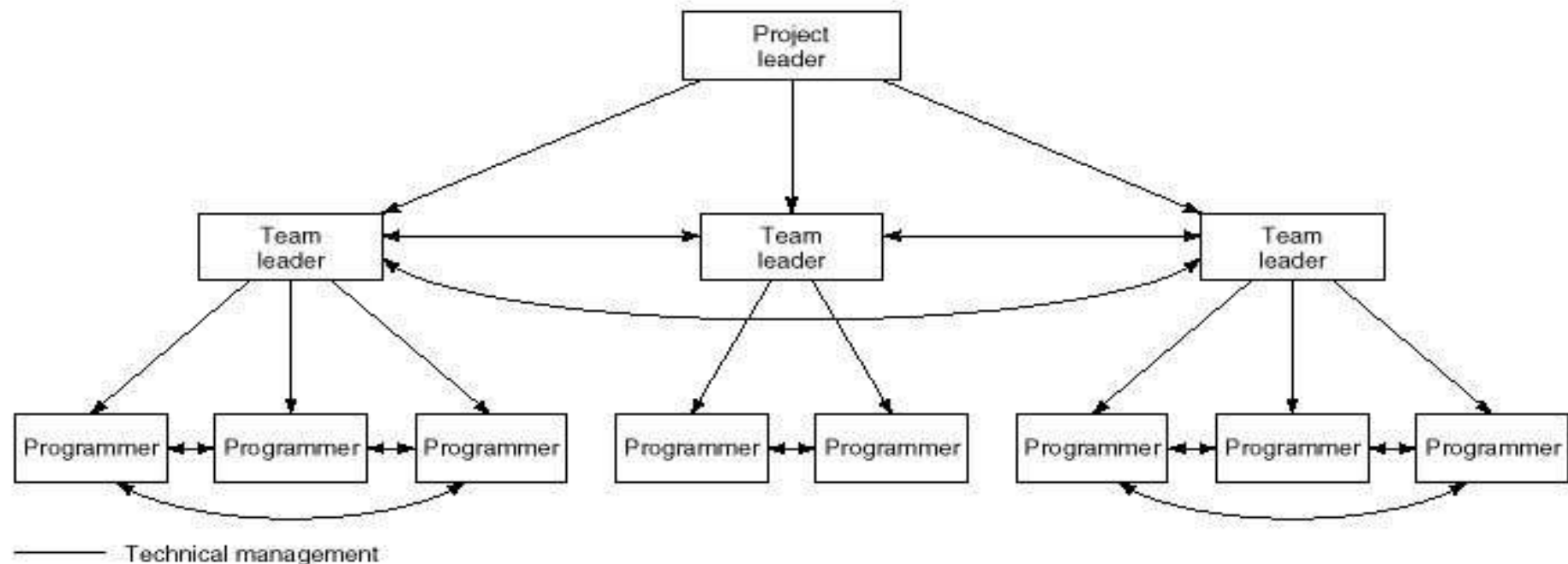
- Democratico Decentralizzato
  - Assenza di un leader permanente
  - Consenso di gruppo sulle soluzioni e sulla organizzazione del lavoro
  - Comunicazione orizzontale
- Vantaggi
  - Attitudine positiva a ricercare presto gli errori
  - Funziona bene per problemi “difficili” (ad esempio per la ricerca)
- Svantaggi
  - È difficile da imporre...
  - Non è scalabile...





- **Controllato Decentralizzato**

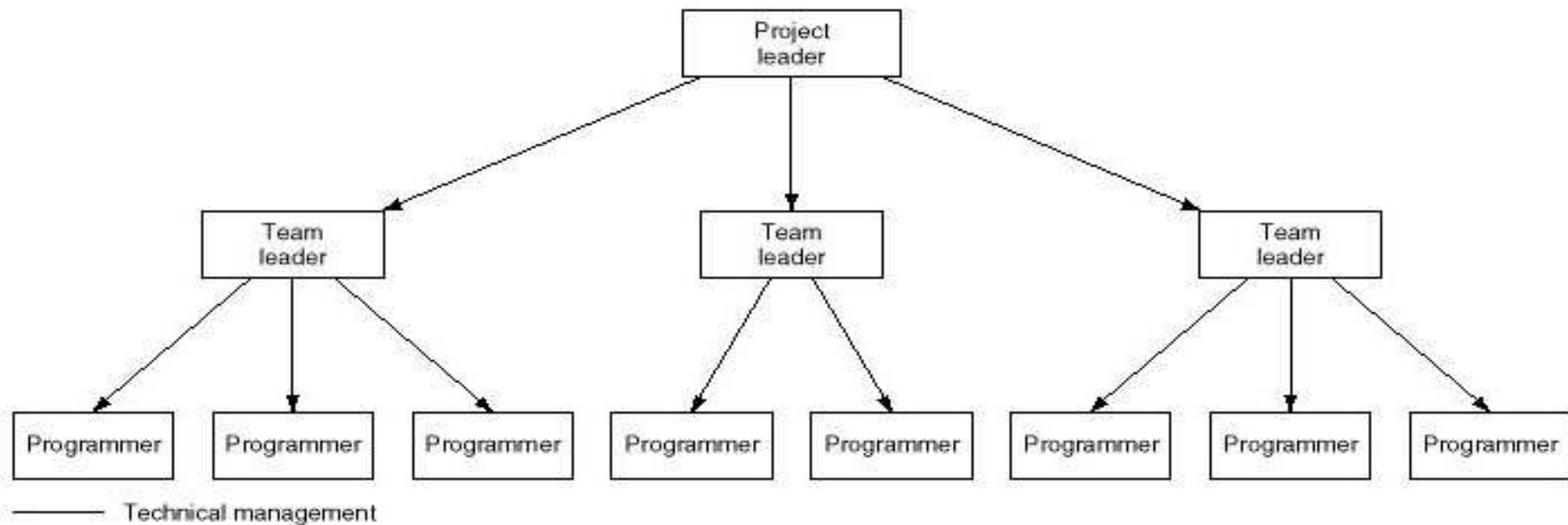
- Un leader riconosciuto, che coordina il lavoro
- La risoluzione dei problemi è di gruppo, ma l'implementazione delle soluzioni è assegnata a sottogruppi da parte del leader
- Comunicazione orizzontale nei sottogruppi e verticale con il leader



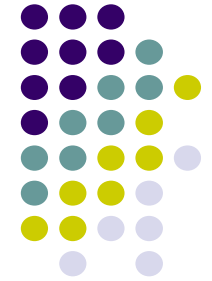


- **Controllato Centralizzato**

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



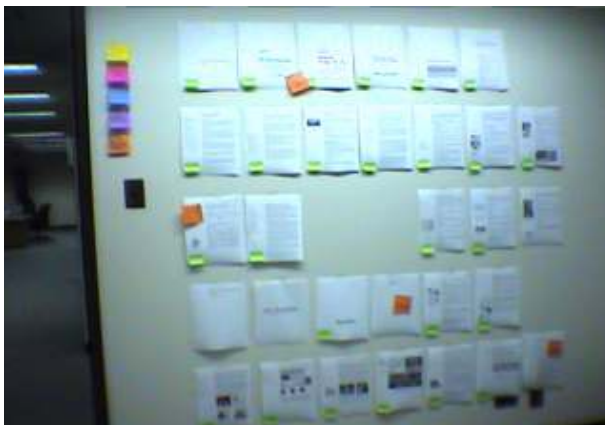
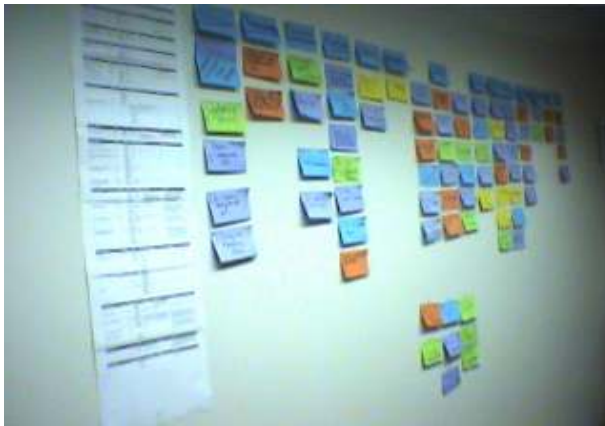
# Ruoli in un team Controllato Decentralizzato



- 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...



# spazio condiviso & risultati condivisi

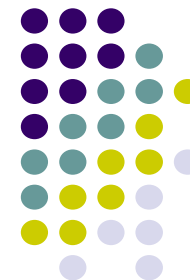


- Un team deve prima di tutto decidere gli strumenti che permettono la cooperazione
- La pianificazione
- Chi fa cosa
- Le scelte fatte
- Cosa è stato fatto

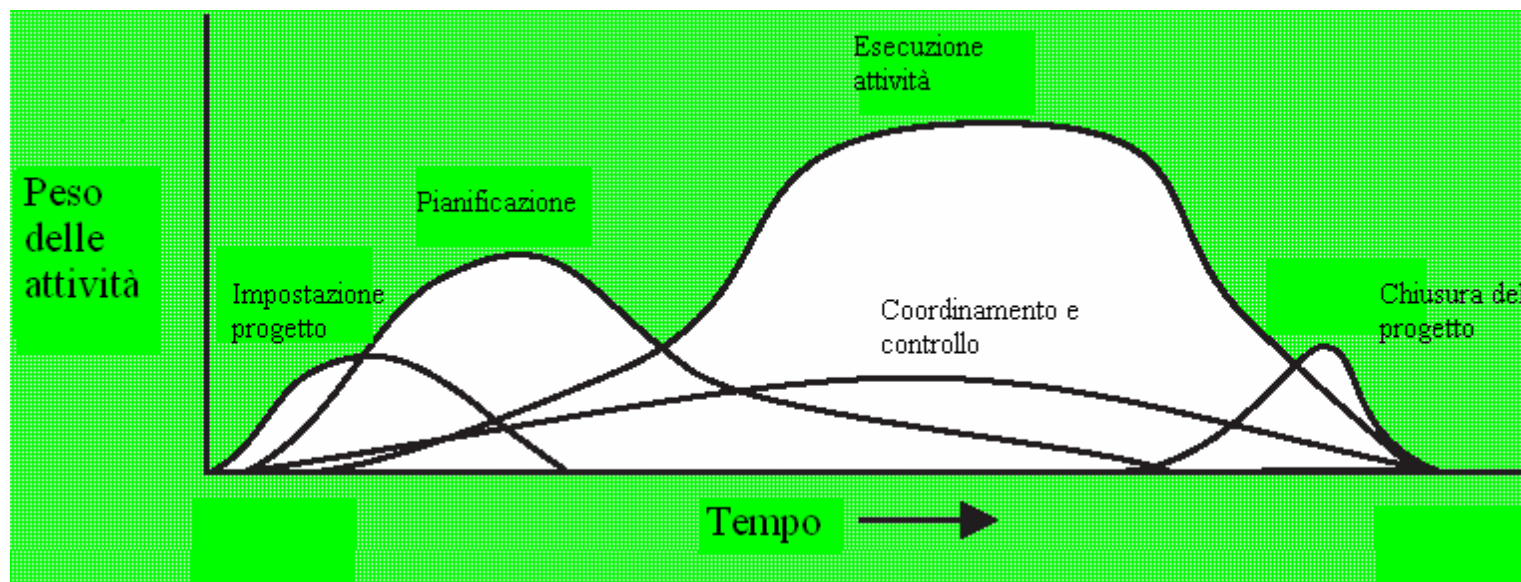
# Le attività del project manager

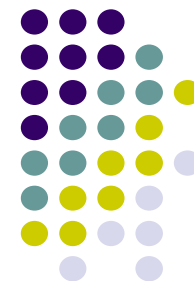


- Stesura della **proposta** di progetto
- Stima del **costo** del progetto
- Pianificazione (**planning**) e temporizzazione (**scheduling**)
- **Monitoraggio** e revisioni del progetto
- Selezione e valutazione del **personale**
- Stesura di **rapporti** e presentazioni



# Distribuzione temporale delle attività





# Stimare i costi di un progetto

- Dilaziona la stima fino a quando il progetto non è in stato avanzato di sviluppo
  - modello non praticabile: la stima dev'essere fatta all'inizio
- Basa la stima su progetti simili già sviluppati
  - similarità di problemi, clienti, ecc.
- Usa tecniche di decomposizione per generare stime di costo e di risorse necessarie
  - approccio “divide et impera”, calcolando il costo delle componenti
- Usa uno o più modelli empirici
  - basati su dati storici, es. COnstructive COst MOdel (Boehm, 1981)

# Stime quantitative: LOC



- KLOC = Migliaia di **linee di codice**
- Metriche:
  - \$ per KLOC
  - errori o difetti per KLOC
  - LOC per mese/persona
  - pagine di documentazione per KLOC
  - errori/mese-persona
  - \$/pagina di documentazione
- Il codice è il prodotto tangibile del processo di sviluppo, ed esiste già letteratura in proposito
- Dipende dal linguaggio di programmazione e penalizza programmi scritti bene e concisi

# Stime quantitative (dimensionali)



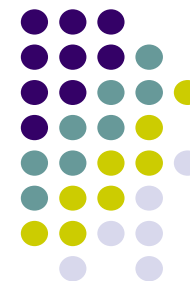
aspetti critici delle metriche dimensionali

1. Difficile stimare la dimensione in LOC
2. Non tiene conto diversa complessità e potenza delle istruzioni (di uno stesso linguaggio o di linguaggi diversi)
3. Difficile definire in modo preciso il criterio di conteggio (istruzioni spezzate su più righe, più istruzioni su una riga...)
4. Più produttività potrebbe portare a più codice senza qualità?

# Stime funzionali: FP



- **Function Points** (punti funzione)
- misurare le funzionalità offerte dall'applicazione, a partire da:
  - il dominio informativo
  - da un giudizio sulla complessità del software



# Parametri

- Numero di **input**
  - informazioni distinte fornite dall'utente e utilizzate dal programma come dati di ingresso
- Numero di **output**
  - informazioni distinte ritornate all'utente come risultato delle proprie elaborazioni
- Numero di **richieste**
  - numero di interrogazioni in linea che producono una risposta immediata del sistema
- Numero di **files**
  - file creati ed utilizzati internamente dal programma
- Numero di **interfacce esterne**
  - files o di altri insiemi di dati scambiati con altri programmi





# Function Points

Indici	Valore	Pesi			Vi
		Sempl.	Medio	Compl.	
N. input		3	4	6	
N. Output		4	5	7	
N. Richieste		3	4	6	
N. File		7	10	15	
N. Int. Est.		5	7	10	

$$FP = \sum Vi \times [0.65 + 0.01 \times \sum Fi]$$



# Fattori di influenza (FI)

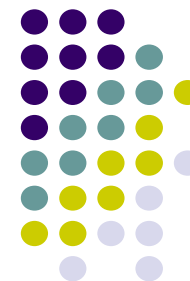
1. Il sistema richiede procedure di recovery e backup affidabili?
2. È richiesta la trasmissione di dati?
3. Vi sono funzionalità che richiedono elaborazioni distribuite?
4. Le prestazioni sono critiche?
5. Funzionerà in un sistema già carico?
6. Richieste funzionalità avanzate per input e lettura in linea dei dati?
7. Input dei dati mediante tramite interfacce a finestre?
8. Archivi principali aggiornati in tempo reale?
9. Informazioni complesse scambiate tra utente e programma?
10. Codice complesso?
11. Scritto per essere riusabile?
12. Il progetto include anche le attività di installazione e conversione?
13. Programma progettato per essere installato presso diversi utenti?
14. Programma progettato per facilitare uso e modifiche da parte dell'utente?



# LOC/FP

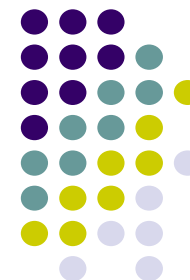
- Capers Jones ha determinato una tabella di corrispondenza tra Function Point e LOC per i alcuni linguaggi di programmazione:

LINGUAGGIO	LOC/FP
ASSEMBLER	320
JCL	220
C	128
ANSI COBOL 74	105
CICS	46
Visual Basic	40
C++	29
SQL	12



# LOC/FP

- A questi valori si applica un coefficiente correttivo basato su tre parametri che possono assumere valori da 1 a 5:
- **Complessità delle elaborazioni** (numero e complessità degli algoritmi): si attribuisce il valore 5 per elevata complessità
- **Complessità della base dei dati** (grado di interrelazione dei dati): si attribuisce il valore 5 per elevata complessità
- **Complessità del codice** (grado di strutturazione del codice): si attribuisce il valore 5 per codice non strutturato



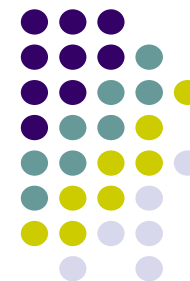
# LOC/FP

L'espressione per il calcolo del coefficiente correttivo è la seguente:

$$CC = 0,7 + 0,05 (CT - 3)$$

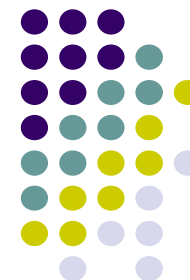
dove:

CT è la somma dei valori assegnati ai parametri



# Struttura del piano di progetto

1. Introduzione
2. Organizzazione del Progetto
3. Descrizione dei Processi Gestionali
4. Descrizione dei Processi Tecnici
5. Pianificazione del lavoro, delle risorse umane e del budget.



# 1. Introduzione

## 1.1 Overview del Progetto

- Descrizione di massima del progetto e del prodotto.

## 1.2 Deliverables del Progetto

- Tutti gli items che saranno consegnati, con data e luogo di consegna

## 1.3 Evoluzione del Progetto

- Piani per cambiamenti ipotizzabili e non

## 1.4 Materiale di riferimento

- Lista dei documenti cui ci si riferisce nel Piano di Progetto

## 1.5 Definizioni e Abbreviazioni



## 2. Organizzazione del progetto

### 2.1 Modello del Processo

- Relazioni tra le varie fasi del processo

### 2.2 Struttura Organizzativa

- Gestione interna, chart dell'organizzazione

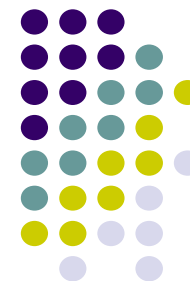
### 2.3 Interfacce Organizzative

- Relazioni con altre entità

### 2.4 Responsabilità di Progetto

- Principali funzioni e attività;
- Di che natura sono?
- Chi ne è il responsabile ?





# 3. Processi gestionali

## 3.1 Obiettivi e Priorità

## 3.2 Assunzioni, Dipendenze, Vincoli

- Fattori esterni

## 3.3 Gestione dei rischi

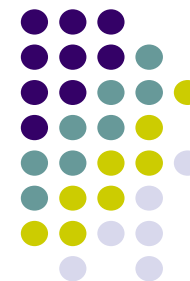
- Identificazione, Valutazione, Monitoraggio dei rischi

## 3.4 Meccanismi di monitoraggio e di controllo

- Meccanismi di reporting, format, flussi di informazione, revisioni

## 3.5 Pianificazione dello staff

- Skill necessari (cosa?, quanto?, quando?)



## 4. Processi tecnici

### 4.1 Metodi, Strumenti e Tecniche

- Sistemi di calcolo, metodi di sviluppo, struttura del team, ecc.
- Standards, linee guida, politiche.

### 4.2 Documentazione del Software

- Piano di documentazione, che deve includere milestones, e revisioni

### 4.3 Funzionalità di supporto al progetto

- Pianificazione della qualità
- Pianificazione della gestione delle configurazioni

# 5. Pianificazione del lavoro, delle risorse umane e del budget.



## 5.1 WBS (Work breakdown structure)

- Il progetto è scomposto in tasks; definizione di ciascun task

## 5.2 Dipendenze

- Relazioni di precedenza tra funzioni, attività e task

## 5.3 Risorse Necessarie

- Stima delle risorse necessarie, in termini di personale, di tempo di computazione, di hardware particolare, di supporto software ecc.

## 5.4 Allocazione del Budget e delle Risorse

- Associa ad ogni funzione, attività o task il costo relativo

## 5.5 Pianificazione

- Deadlines e Milestones

# Grazie per l'attenzione

---

[www.vincenzocalabro.it](http://www.vincenzocalabro.it)

