



UNIVERSITÀ  
DEGLI STUDI  
DI PALERMO

# Project Management

Ingegneria del Software



# Software project management

- ~ Riga<sup>r</sup>da le attivit<sup>a</sup> volte a garantire che il software sia consegnato puntuamente e in tempo utile di calendario e in conformit<sup>a</sup> dei requisiti delle organizzazioni che sviluppa il software.
- ~ La gestione del progetto è necessaria perché lo sviluppo del software è sempre soggetto ai vincoli di budget e di pianificazione stabiliti dall'organizzazione che sviluppa il software.
- ~ La buona gestione non può garantire il successo di un progetto ma la cattiva gestione di solito ne determina il fallimento



# Criteri di successo per un sw

- ~ consegnare il software al cliente nei tempi concordati.
- ~ Mantenere i costi complessivi entro il budget.
- ~ consegnare software che soddisfi le aspettative del cliente.
- ~ Mantenere un team di sviluppo coerente e ben funzionante.
- ~ Sono obiettivi di tutti i progetti software ma...sw è diversa

# Cosa distingue il progetto sw dagli altri

- ~ Il prodotto è intangibile.
  - Il software non può essere visto o toccato. I responsabili di progetti software non possono vedere i progressi osservando semplicemente l'artefatto che si sta costruendo.
- ~ Molti progetti software sono progetti unici.
  - Grandi progetti software sono di solito diversi per certi versi dai progetti precedenti. Anche i manager che hanno molte esperienze precedenti possono avere difficoltà ad anticipare i problemi.
- ~ I processi software sono variabili e specifici dell'organizzazione.
  - Non siamo ancora in grado di prevedere con certezza quando un particolare processo software potrebbe causare problemi di sviluppo.

# Fattori che influenzano il project management

- ~ Non si può definire un compito standard per un manager.
- ~ Fattori che influiscono sulla gestione dei progetti:
  - Dimensione dell' azienda
  - Clienti del software
  - Dimensione software
  - Tipo di software
  - Cultura organizzativa
  - Processi di sviluppo software
- ~ Questi fattori significano che i project manager di diverse organizzazioni possono lavorare in modi molto diversi (alcune attività però sono comuni a tutte le aziende).



# Attività comuni di management

## ~ Pianificazione del progetto

- I project manager sono responsabili della pianificazione, della stima e della tempistica dello sviluppo dei progetti e devono assegnare le persone alle attività.

## ~ Gestione dei rischi

- I responsabili di progetto valutano i rischi che possono interessare un progetto, controllano questi rischi e intervengono in caso di problemi.

## ~ Gestione delle persone

- I responsabili di progetto devono scegliere le persone per il proprio team e stabilire modalità di lavoro che portino a prestazioni di squadra efficaci.



# Attività di management

## ~ Reporting

- Di norma, i responsabili di progetto hanno la responsabilità di riferire in merito allo stato di avanzamento di un progetto ai clienti e ai responsabili dell'azienda che sviluppa il software. Devono produrre report e scrivere documenti riassuntivi sulle informazioni critiche.

## ~ Scrittura delle proposte

- La prima fase di un progetto software può consistere nella stesura di una proposta per aggiudicarsi un contratto per la realizzazione di un'opera. La proposta descrive gli obiettivi e le modalità di attuazione del progetto.
- Di solito include la stima dei costi, dei tempi e spiega perché il contratto dovrebbe essere assegnato ad una particolare società o team di sviluppo.



# Project Planning



# Project Planning

- ~ La pianificazione del progetto consiste nel suddividere il lavoro in parti e assegnarle ai membri del team di progetto, anticipare i problemi che potrebbero sorgere e preparare soluzioni provvisorie a tali problemi.
- ~ Il piano di progetto, che viene creato all'inizio di un progetto, è utilizzato per comunicare al team di progetto e ai clienti come il lavoro sarà fatto e per aiutare a valutare lo stato di avanzamento del progetto.



# Fasi della pianificazione

- ~ **Fase della proposta**, quando si presenta un' offerta per un contratto per sviluppare o fornire un sistema software.
- ~ **Fase di avvio del progetto**, quando si deve pianificare chi lavorerà al progetto, come il progetto sarà suddiviso in incrementi, come verranno distribuite le risorse all'interno dell' azienda, ecc.
- ~ **Fase di verifica del progetto**, periodicamente durante tutto il progetto, quando si modifica il piano alla luce dell' esperienza acquisita e delle informazioni ottenute dal monitoraggio dello stato di avanzamento del lavoro.



# Proposta di pianificazione

- ~ La pianificazione può essere fatta solo con requisiti software di massima, è puramente speculativa.
- ~ L'obiettivo della pianificazione in questa fase è quello di fornire ai clienti informazioni che saranno utilizzate per fissare il prezzo del sistema.
- ~ Fissare i prezzi del progetto comporta la stima di quanto il software costerà per essere sviluppato, tenendo conto di fattori quali i costi del personale, i costi hardware, software, ecc.



# Pianificazione di avvio progetto

- ~ In questa fase, siamo a conoscenza dei requisiti di sistema, ma non abbiamo informazioni sulla progettazione o sull'implementazione.
- ~ Bisogna creare un piano sufficientemente dettagliato per prendere decisioni sul budget del progetto e sul personale.
- ~ Questo piano costituisce la base per l'assegnazione delle risorse del progetto e quindi dei costi.
- ~ Il piano di avvio dovrebbe anche definire i meccanismi di monitoraggio del progetto



# Pianificazione dello sviluppo

- ~ il piano del progetto dovrebbe essere regolarmente modificato man mano che il progetto progredisce e si sa qualcosa di più sul software e il suo sviluppo.
- ~ il calendario del progetto, la stima dei costi e i rischi devono essere regolarmente riveduti



# Costo del software

- ~ Le stime sono fatte per scoprire il costo, per lo sviluppatore, di produrre un sistema software.
- Bisogna prendere in considerazione i costi hardware, software, viaggio, formazione ed effort effettivo.
- ~ Non esiste una relazione semplice tra il costo di sviluppo e il prezzo addebitato al cliente.
- ~ Considerazioni organizzative, economiche, politiche e commerciali più ampie influenzano il prezzo praticato.



# Fattori che incidono sui prezzi dei software

Factor	Description
Termini contrattuali	Un cliente può essere disposto a consentire allo sviluppatore di mantenere la proprietà del codice sorgente e di riutilizzarlo in altri progetti. Il prezzo praticato può quindi essere inferiore se il codice sorgente del software viene consegnato al cliente.
Incertezza sulla stima dei costi	Se un' organizzazione non è sicura della sua stima dei costi, può aumentare il suo prezzo di un certo valore sopra il normale profitto.
Salute finanziaria	Gli sviluppatori in difficoltà finanziarie possono abbassare il prezzo per ottenere un contratto. E' meglio realizzare un profitto o un risultato di pareggio inferiore al normale piuttosto che abbandonare l' attività. Il flusso di cassa è più importante del profitto in tempi economici difficili.



# Fattori che incidono sui prezzi dei software

Factor	Description
Opportunità di mercato	Una società di software può quotare un prezzo basso perché vuole entrare in un nuovo segmento del mercato del software. Accettare un basso profitto su un progetto può dare all'organizzazione la possibilità di realizzare un maggiore profitto in seguito. L'esperienza acquisita può anche aiutarla a sviluppare nuovi prodotti.
Volatilità dei requisiti	Se i requisiti sono suscettibili di cambiamento, un' organizzazione può abbassare il suo prezzo per vincere un contratto. Una volta aggiudicato l' appalto, si possono addebitare prezzi elevati per le modifiche dei requisiti.

# Fattori che influiscono sul prezzo - scenario

*Una piccola società di software, PharmaSoft, ha 10 ingegneri. Ha appena ultimato un grosso progetto, ma ha contratti che richiedono solo cinque membri nel team di sviluppo. Sta partecipando a una gara per aggiudicarsi un importante appalto con una grande società farmaceutica che richiede 30 anni-uomo in due anni. Il progetto non partirà prima di un anno ma, se la società se lo aggiudicherà, le sue finanze rifioriranno.*

*PharmaSoft ha l'opportunità di fare un'offerta per un progetto che richiede sei persone e che deve essere completato in dieci mesi. I costi (inclusi gli overhead di questo progetto) sono stimati in 1,2 milioni di euro. Per essere più concorrenziale, PharmaSoft offre al cliente un prezzo di 0,8 milioni di euro. Questo significa che, anche se la società ha una perdita con questo appalto, tuttavia potrà mantenere il personale specializzato per progetti futuri più redditizi.*



# Costo del sw - strategie

## ~ Sotto prezzo

- un' azienda può sottoprezzare un sistema per ottenere un contratto che permetta loro di mantenere il personale per opportunità future.
- un' impresa può sottoprezzare un sistema per accedere a una nuova area di mercato

## ~ Aumento dei prezzi

- il prezzo può essere aumentato quando l' acquirente desidera un contratto a prezzo fisso e quindi il venditore aumenta il prezzo per tenere conto di rischi imprevisti.

# Pricing to win (prezzo per vincere)

- ~ Il prezzo del software è calcolato in base a quello che lo sviluppatore di software crede che l'acquirente sia disposto a pagare
- ~ Se i costi di sviluppo sono inferiori, la funzionalità del software può essere ridotta di conseguenza in modo da poter aggiungere funzionalità aggiuntive in una versione successiva.
- ~ I costi aggiuntivi possono essere aggiunti a seguito dell'evoluzione dei requisiti e possono essere valutati ad un prezzo più elevato per compensare il deficit del prezzo iniziale.



# Lo sviluppo plan-driven

- ~ Lo sviluppo plan-driven o plan-based è un approccio all'ingegneria del software dove il processo di sviluppo è pianificato in dettaglio.
- ~ Lo sviluppo guidato dai piani si basa su tecniche di ingegneria di gestione del progetto ed è il modo tradizionale di gestire grandi progetti di sviluppo software.
- ~ Viene creato un piano di progetto che tiene in considerazione
  - il lavoro da svolgere, chi lo farà, il programma di sviluppo e i prodotti di lavoro.
- ~ I manager utilizzano il piano per supportare il processo decisionale del progetto e per misurare i progressi.



# Pro e contro

## ~ Argomenti a favore di un approccio plan-driven

- la pianificazione precoce permette di prendere in seria considerazione i problemi organizzativi (disponibilità di personale, altri progetti, ecc.)
- i problemi e le dipendenze potenziali vengono scoperti prima dell'inizio del progetto, piuttosto che quando il progetto è in corso.

## ~ L'argomento principale contro lo sviluppo pianificato

- molte decisioni iniziali devono essere riviste a causa dei cambiamenti nell'ambiente in cui il software deve essere sviluppato e utilizzato.



# Pro e contro

- ~ L'approccio migliore alla pianificazione della progettazione può consistere in un opportuno mix di sviluppo agile e sviluppo guidato dai piani
- ~ Il mix dipende dal tipo di progetto e dalle capacità delle persone
  - i grandi sistemi
    - ~ richiedono un'approfondita analisi e possono essere certificati prima di essere messi a disposizione degli utenti
    - ~ lo sviluppo di questi sistemi dovrebbe essere principalmente guidato dai piani
  - sistemi di piccole e medie dimensioni
    - ~ ambiente competitivo in rapida evoluzione
    - ~ dovrebbero essere sviluppati principalmente con metodi agili
  - se più società sono impegnate nello sviluppo di un progetto di solito si usa un approccio guidato da piani per coordinare il lavoro tra i vari siti di sviluppo



# Piano di progettazione

- ~ In un progetto di sviluppo pianificato, un piano di progetto indica le risorse disponibili per il progetto, la ripartizione dei lavori e un calendario per l'esecuzione dei lavori.
- ~ Scaletta del (documento) piano di progettazione
  - § Introduzione
  - § Organizzazione del progetto
  - § Analisi dei rischi
  - § Requisiti di risorse hardware e software
  - § Ripartizione del lavoro
  - § Programma del progetto
  - § Meccanismi di monitoraggio e comunicazione

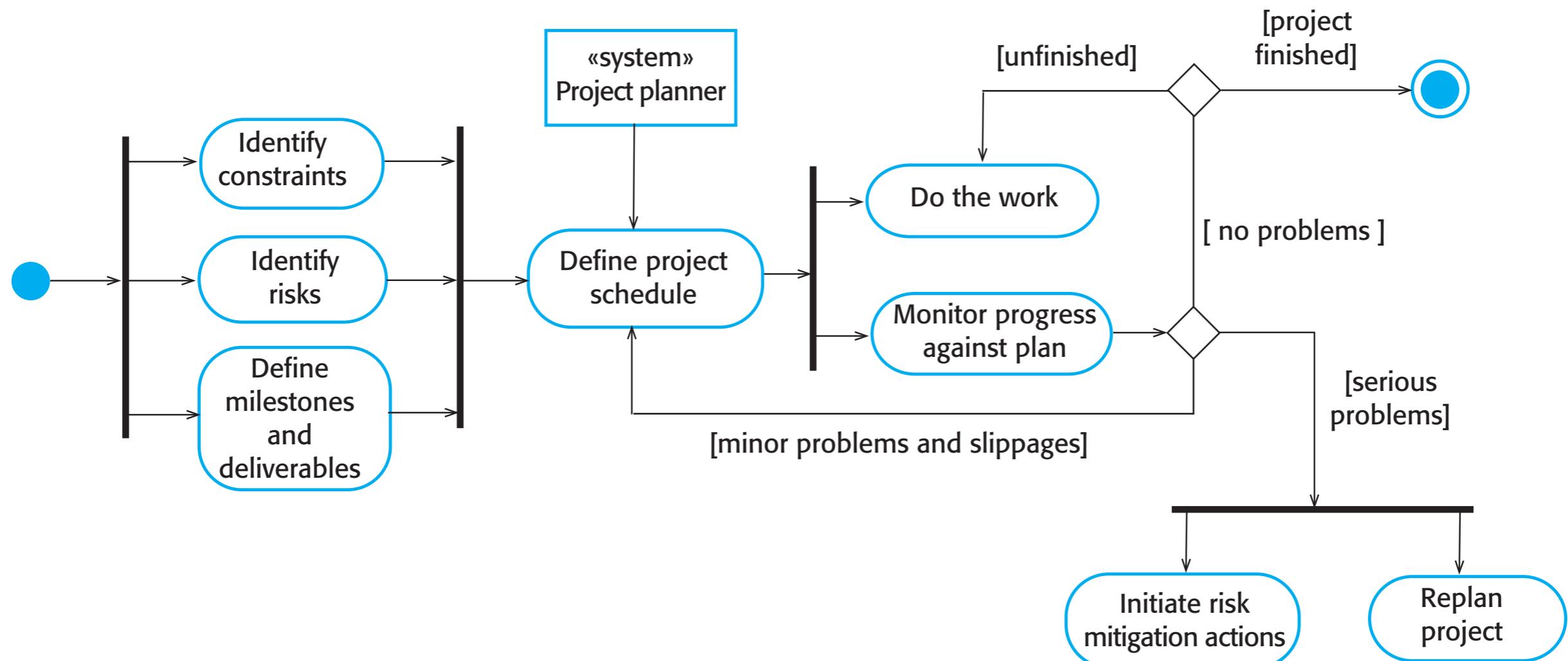
# Piani di progettazione supplementari

Plan	Description
Configuration management plan	Describes the configuration management procedures and structures to be used.
Deployment plan	Describes how the software and associated hardware (if required) will be deployed in the customer's environment. This should include a plan for migrating data from existing systems.
Maintenance plan	Predicts the maintenance requirements, costs, and effort.
Quality plan	Describes the quality procedures and standards that will be used in a project.
Validation plan	Describes the approach, resources, and schedule used for system validation.

# Processo di pianificazione

- ~ La pianificazione del progetto è un **processo iterativo** (vedi figura) che inizia quando si crea una pianificazione iniziale del progetto durante la fase di avvio del progetto.
- ~ I cambiamenti di piano sono inevitabili.
- ~ Poiché durante il progetto sono disponibili maggiori informazioni sul sistema e sul team di progetto, è necessario rivedere regolarmente il piano per riflettere i requisiti, la pianificazione e i cambiamenti dovuti ai rischi.
- ~ Cambiando gli obiettivi aziendali si modificano anche i piani di progetto. Con il mutare degli obiettivi di business, questo potrebbe interessare tutti i progetti, che dovranno poi essere rielaborati.

# Processo di pianificazione di un progetto





# Processo di pianificazione di un progetto

## ~ Definire i vincoli:

· & Data di consegna del progetto

· & il personale del team di sviluppo

· & il budget

· & gli strumenti disponibili

· & altro

## ~ Oltre ai vincoli

· & identificare anche i traguardi raggiunti e le consegne effettuate

~ i traguardi sono punti del piano rispetto ai quali si può valutare il progresso del progetto (per esempio un sistema è pronto per i test)

~ le consegne sono prodotti che possono essere passati al clienti (per esempio un documento sui requisiti del sistema)



# Ipotesi di pianificazione

- ~ Quando si definisce un piano di progetto si dovrebbero fare ipotesi **realistiche** piuttosto che ottimistiche.
- ~ Problemi di qualche tipo si presentano sempre durante un progetto e ciò causa ritardi nel progetto.
- ~ Le ipotesi iniziali e la programmazione dovrebbero quindi tenere conto di problemi imprevisti.
- ~ Si dovrebbero fare ipotesi pessimistiche. Il piano dovrebbe prevedere alcune **strategie di emergenza** da adottare quando qualcosa va storto.

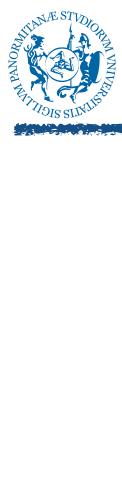
# Riduzione dei rischi

- ~ Se ci sono gravi problemi durante lo sviluppo che possono causare ritardi significativi, è necessario avviare azioni di riduzione del rischio per ridurre i rischi di fallimento del progetto.
- ~ In concomitanza con queste azioni, è necessario anche pianificare nuovamente il progetto
  - se il progetto è grande e dura anni si può anche decidere per l'annullamento
- ~ Ciò può comportare la rinegoziazione dei vincoli e dei risultati del progetto con il cliente. È inoltre necessario stabilire e concordare con il committente un nuovo calendario di completamento dei lavori.



# Tempistica dei progetti

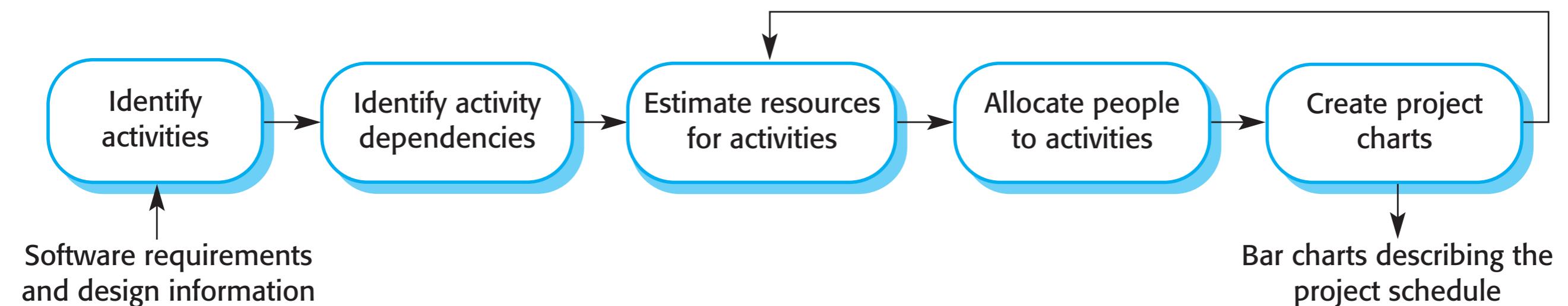
- ~ La tempistica di un progetto, o scheduling, è il processo per decidere come il lavoro in un progetto sarà organizzato come attività separate, quando e come queste attività saranno eseguite.
- ~ Si stima il tempo necessario per completare ogni attività, lo sforzo richiesto e chi lavorerà sulle attività che sono state identificate.
- ~ È inoltre necessario stimare le risorse necessarie (ed il tempo di utilizzo) per completare ogni attività
  - esempio lo spazio su disco richiesto su un server, il tempo richiesto su hardware specializzato, come un simulatore, e quale sarà il budget di viaggio.

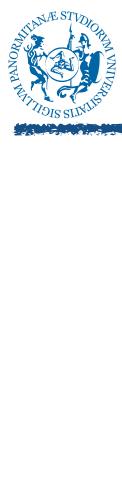


# Tempistica di progetto - attività

- ~ Dividere il progetto in attività e stimare il tempo e le risorse necessarie per completare ogni attività
  - da una settimana a due mesi
  - tempi più piccoli rendono più complessi la ripianificazione e l'aggiornamento del progetto
- ~ Organizzare le attività contemporaneamente per rendere ottimale il ricorso alla forza lavoro.
- ~ Ridurre al minimo le dipendenze delle attività per evitare ritardi causati da un' attività in attesa che un'altra sia completata.
- ~ Dipende dall'intuizione e dall'esperienza dei project manager.

# Definizione della tempistica di un progetto





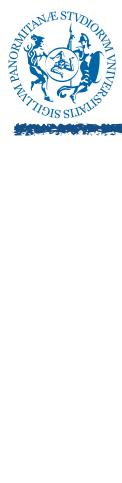
# Tempistica di progetto - attività

- ~ Evitare situazioni in cui l'intero progetto subisce ritardi
  - per esempio compito critico non ancora ultimato
- ~ Nella definizione della tempistica si deve mettere in conto le varie possibilità che qualcosa possa andar storto
  - le persone possono sbagliare
  - le persone possono andarsene
  - l'hardware potrebbe guastarsi
  - hw e sw di supporto potrebbero arrivare in ritardo
- ~ Una buona prassi:
  - fare stime come se tutto dovesse andar bene e poi aumentare le stime tenendo conto dei problemi prevedibili e delle potenziali emergenze



# Problemi dello scheduling

- ~ valutare la difficoltà dei problemi e quindi il costo dello sviluppo di una soluzione è difficile.
- ~ La produttività non è proporzionale al numero di persone che lavorano in un'attività.
- ~ Aggiungere persone a un progetto tardivo lo fa ritardare di più a causa delle spese generali di comunicazione.
- ~ L'imprevisto accade sempre. Consentire sempre la pianificazione in caso di emergenza.



# Presentazione dello scheduling

- ~ Le notazioni grafiche sono normalmente utilizzate per illustrare il programma del progetto.
- ~ Questi mostrano la suddivisione del progetto in attività. I compiti non dovrebbero essere troppo piccoli. Dovrebbero richiedere circa una settimana o due.
- ~ calendar-based
  - i grafici a barre sono la rappresentazione più comunemente utilizzata per i piani di progetto. Essi mostrano il programma come attività o risorse rispetto al tempo.
- ~ Reti di attività
  - Mostra dipendenze attività



# Le attività di progetto

- ~ Le attività di progetto (compiti) sono l' elemento di base della pianificazione. Ogni attività ha:
  - una durata in giorni o mesi di calendario,
  - una stima dello sforzo, che indica il numero di giorni/ persona o di mesi-persona per completare il lavoro,
  - un termine entro il quale l' attività deve essere completata,
  - un punto finale definito, che può essere un documento, lo svolgimento di una riunione di riesame, la corretta esecuzione di tutte le prove, ecc.



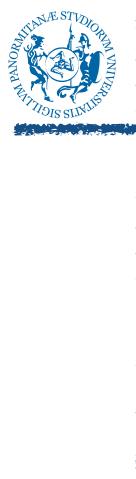
# Milestones e deliverables

- ~ Pietre miliari (milestones) sono punti nel calendario rispetto ai quali è possibile valutare i progressi, ad esempio, la consegna del sistema per il test.
- ~ i prodotti forniti (deliverables) sono prodotti di lavoro che vengono consegnati al cliente, ad esempio un documento dei requisiti del sistema.



# Presentazione della Tempistica

- ~ Si può presentare la tempistica di un progetto tramite
  - Tabella o foglio elettronico
- ~ Obiettivo
  - mostrare i compiti le attività stimate la durata e la dipendenza tra i compiti
- ~ Difficile presentare le relazioni e le dipendenze tra le varie attività
- ~ Sono state sviluppate grafiche alternative



# Task, durata e dipendenze

Task	Effort (person-days)	Duration (days)	Dependencies
T1	15	10	
T2	8	15	
T3	20	15	T1 (M1)
T4	5	10	
T5	5	10	T2, T4 (M3)
T6	10	5	T1, T2 (M4)
T7	25	20	T1 (M1)
T8	75	25	T4 (M2)
T9	10	15	T3, T6 (M5)
T10	20	15	T7, T8 (M6)
T11	10	10	T9 (M7)
T12	20	10	T10, T11 (M8)



# Presentazione della Tempistica

- ~ Due tipi di rappresentazioni grafiche più comuni
  - Diagrammi a barre basati sul calendario
    - ~ mostrano persone, durate, tempo di inizio e fine per ogni task
    - ~ inventati da Henry Gantt (1937)
  - Reti di attività
    - ~ mostrano le dipendenze tra le varie attività di un progetto



# Presentazione della Tempistica

~ Ogni attività ha:

- durata espressa in giorni (o mesi di calendario)
- stima dell'impegno delle persone espressa in numeri di giorni-persona o mesi-persona
- una scadenza entro la quale l'attività deve essere completata
- obiettivo/i, per esempio scrittura di un documento, svolgimento di una riunione, superamento di test, etc.

# Qualche elemento in più sulle minestrone

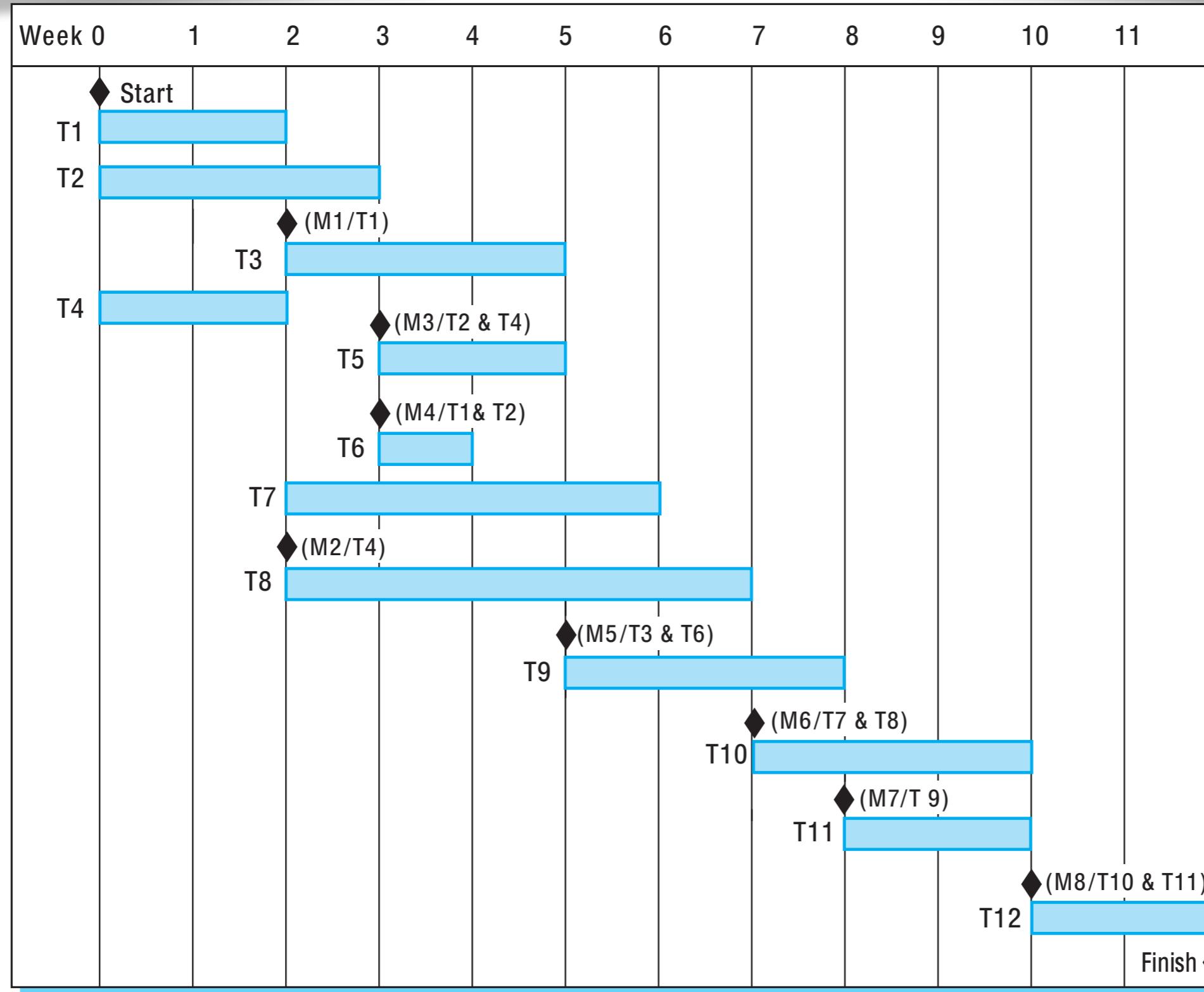
- ~ quando si pianifica un progetto si possono definire i traguardi (milestones)
- ~ un traguardo di un progetto è la fine logica di una delle sue fasi
  - in corrispondenza può essere rivisto l'avanzamento del lavoro
  - deve essere documentato da un breve rapporto (per esempio una mail)
  - riassume il lavoro svolto e il se il lavoro è stato completato come previsto nel piano del progetto



# Traguardi vs consegne

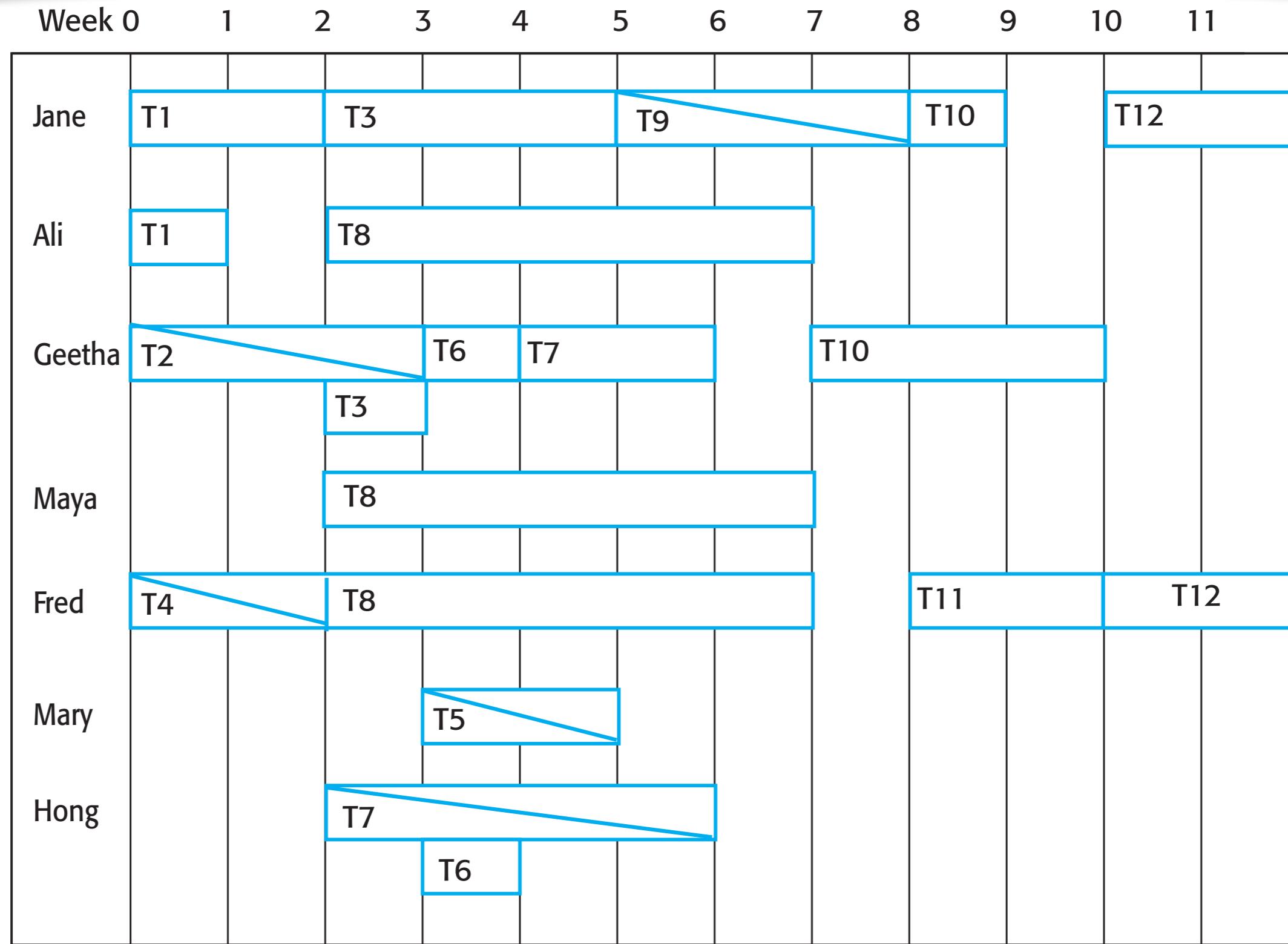
- ~ Le consegne del progetto sono gli output che devono essere consegnati al cliente
- ~ Le consegne sono esplicitamente specificate nel contratto
- ~ Il cliente giudica l'avanzamento del progetto dalle consegne
- ~ In breve, i traguardi sono brevi sintesi (anche solo interne) sull'avanzamento del lavoro mentre le consegne identificano un risultato più "sostanzioso"
  - e. traguardi - email, report, note di riunioni
  - e. consegne - documento dei requisiti, implementazione iniziale del sistema

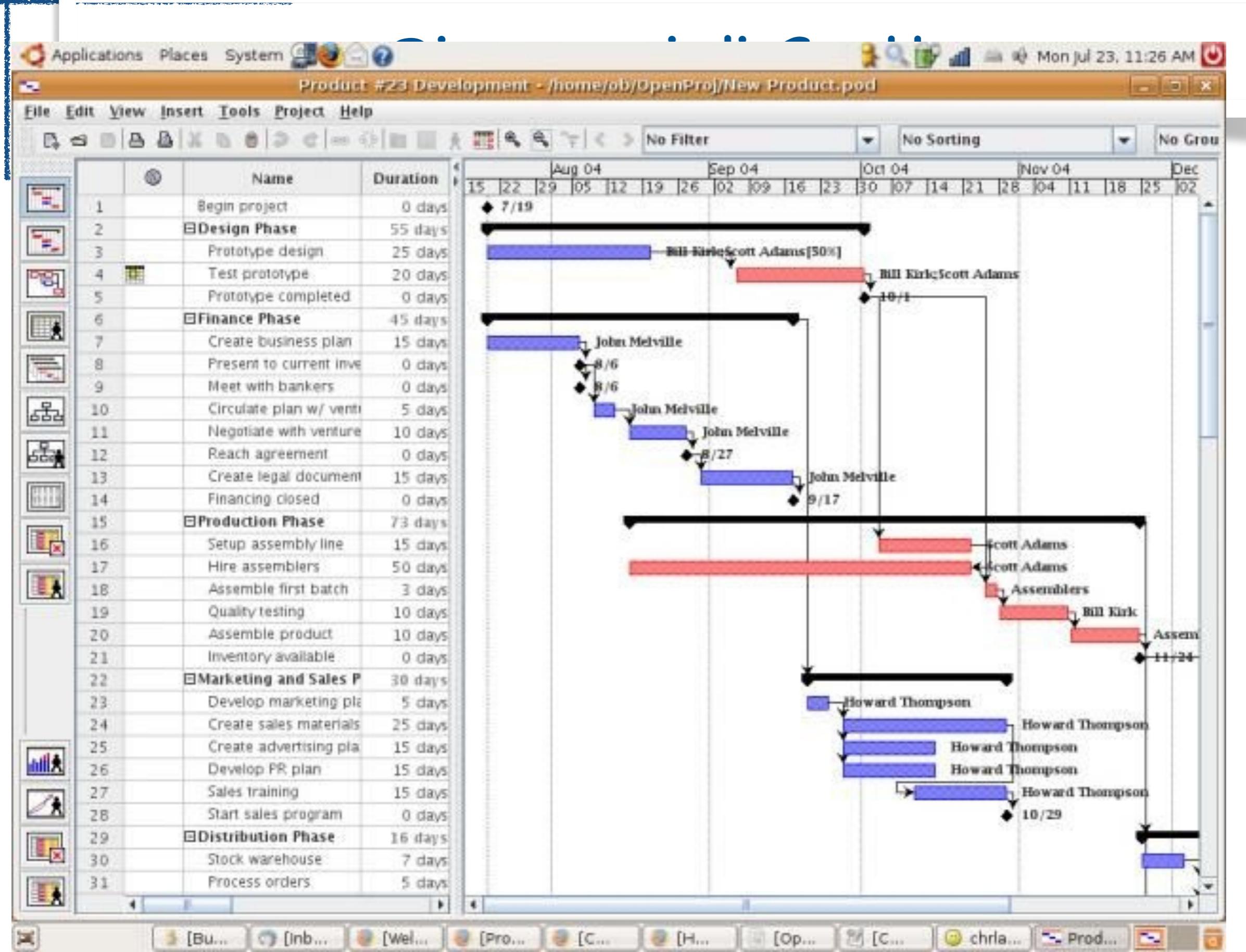
# Diagramma a barre delle attività





# Diagramma di allocazione delle risorse





This Gantt chart illustrates the project timeline and dependencies across four main phases (M1, M2, M3, M4) and 14 sub-phases (M1 to M41).

- Phase 1 (M1):** Work packages 1.1 to 1.3 are completed by M1.
- Phase 2 (M2):** Work packages 2.1 to 2.8 are completed by M2.
- Phase 3 (M3):** Work packages 3.1 to 3.6 are completed by M3.
- Phase 4 (M4):** Work packages 4.1 to 4.11 are completed by M4.
- Phase 5 (M5):** Work packages 5.1 to 5.8 are completed by M5.
- Phase 6 (M6):** Work packages 6.1 to 6.4 are completed by M6.
- Phase 7 (M7):** Work packages 7.1 to 7.3 are completed by M7.
- Phase 8 (M8):** Work packages 8.1 to 8.2 are completed by M8.
- Phase 9 (M9):** Work packages 9.1 to 9.6 are completed by M9.
- Phase 10 (M10):** Work package 10.1 is completed by M10.

Key milestones include the release of MS1 (M1), MS2 (M2), MS3 (M3), MS4 (M4), MS5 (M5), MS6 (M6), MS7 (M7), MS8 (M8), and MS11 (M11). The chart also shows the availability of monitoring (MS4) and decision process (MS5) modules, as well as the final demonstration of the system (MS14).

	RESILIENCE		Leader	Year 1												Year 2												Year 3												Year 4							
				M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12	M13	M14	M15	M16	M17	M18	M19	M20	M21	M22	M23	M24	M25	M26	M27	M28	M29	M30	M31	M32	M33	M34	M35	M36	M37	M38	M39	M40	M41	M42		
<b>WP1</b> Project Management		UNIPA																																													
Task 1.1 - Project Management		UNIPA		D1.4		D1.1																																									
Task 1.2 - Administrative, Financial and Legal Management		UNIPA																																													
Task 1.3 - Ethical, Gender and Risk Management and Data Management		UNIPA						D1.2	D1.3																																						
<b>WP2</b> Analysis, Design and Software Architecture Definition		UNIPA															D2.1																														
Task 2.1 - User Requirements and System Models Definition		UNIPA																D2.2																													
Task 2.2 - Clinical Protocol Identification and Definition		UNIPA																D2.3																													
Task 2.3 - Identifying Biometric and Clinical Data Type		UNIPA																	D2.4																												
Task 2.4 - Defining Human-System Interfaces		UNIPA																	D2.5																												
Task 2.5 - Identifying Software Architecture		CNR																		D2.6																											
Task 2.6 - Definition of Smart Components Interfaces		CNR																		D2.7																											
Task 2.7 - Energy Sustainability Analysis for Smart Ambient		UBFC																		D2.8																											
Task 2.8 - Privacy, Legal and Ethical Principles		UNIPA																		D2.9																											
Task 2.9 - Tests, Pilots and Validation Definition		UNIPA																			D2.9																										
Task 2.10 - Continuous Integration - Iterative Project Refinement		UNIPA																				D2.9																									
<b>WP3</b> Human-System Interaction: Monitoring Module		POLIMI																				D3.1																									
Task 3.1 - Robots and IoMT: Identifying Sensors and Robots for Monitoring		POLIMI																				D3.1																									
Task 3.2 - Clinical Digital Phenotyping and Activities Identification		UP8																				D3.1																									
Task 3.3 - Patient, Environment and Energy Monitoring		UBFC																				D3.1																									
Task 3.4 - Data Standardization and Integration		UNIPA																				D3.2																									
Task 3.5 - User Centric Human-System Interfaces		ENT																				D3.3																									
<b>WP4</b> Decision Making Processes		UCBM																			D4.1																										
Task 4.1 - Medical Body of Knowledge Development		UCBM																			D4.1																										
Task 4.2 - Trends for Physical, Cognitive and Behavioural Monitoring		POLIMI																				D4.2																									
Task 4.3 - Experience-Driven Decision Support		UCBM																				D4.3																									
Task 4.4 - Data-Driven Decision Support		LNU																				D4.4																									
Task 4.5 - Decision Support based on Radiomics and Deep Learning		UNIPA																				D4.5																									
Task 4.6 - Systemic Decision Support		UCBM																				D4.6																									
Task 4.7 - Smart Power Management		UBFC																				D4.7																									
Task 4.8 - Designing Data Visualization Module		UVEG																																													
Task 4.9 - Identity and Access Control		NASK																				D4.8																									
Task 4.10 - Secure Data Traceability		NASK																				D4.9																									
<b>WP5</b> Human - System Interaction: Supporting Module		UP8																				D5.1																									
Task 5.1 - User Interfaces to Patients		UP8																				D5.1																									
Task 5.2 - A Medical User Interface for Patient's Active Management		UP8																																													
Task 5.3 - A Semantic Data Representation for Physicians Supporting		UVEG																																													
Task 5.4 - Handling Changed Behavior		POLIMI																				D5.3																									
Task 5.5 - From Therapy to Executable Workflows		CNR																				D5.4																									
Task 5.6 - Designing and Implementing Human-System Interactions Employing Robots		UNIPA																				D5.5																									
Task 5.7 - IoT Platform Development and Integration		UCBM																				D5.6																									
<b>WP6</b> IoT Edge Intelligence Framework and Networks		CNR																																													
Task 6.1 - Development of a Middleware for Medical																																															



# Il diagramma di Gantt

- ~ un diagramma di Gantt rappresenta il piano di lavoro delle attività da svolgere e la durata complessiva
  - quali attività e quanto durano
- ~ come individuiamo le attività (un passo indietro)
  - Work Breakdown Structure
  - individuare i deliverable e le attività per ottenerli

# Il diagramma di Gantt



# Il diagramma di Gantt

- ~ Dopo aver identificato le attività:
  - stimo tempi, costi e risorse
  - dipendenze e ritardi tra le attività
- ~ Passaggi pratici
  - WBS, reticolo di progetto (Pert), Gantt

# Diagramma di PERT - Program Evaluation Review

