

---

# Termini fondamentali

## Processo

- Insieme di attività modulari e scomponibili che, organizzate in sequenza e con precise responsabilità, producono risultati verificabili.
- Ogni processo ha input, output, precondizioni e postcondizioni (ISO 12207).

## Progetto

- Istanziamento concreta di uno o più processi di ciclo di vita per raggiungere un obiettivo specifico.
- Comprende attività pianificate, risorse, ruoli e risultati controllabili nel tempo.
- È sempre **collaborativo** e con vincoli di costo, tempo e qualità.

---

# Fasi e Milestone

## Fasi

- Segmenti temporali del ciclo di vita con caratteristiche coerenti e obiettivi precisi.
- Corrispondono a stati del ciclo di vita (es. analisi, progettazione, codifica, collaudo).
- Sono scandite “a orologio”: l’avanzamento segue pre- e post-condizioni definite.

## Milestone

- Eventi di rilievo a medio termine che segnano il completamento di una o più fasi.
- Ogni milestone rappresenta un punto di verifica e di validazione: consente di misurare avanzamento e qualità.

---

# Baseline e loro scopo

## Baseline

- Versione “congelata” di un insieme di prodotti o documenti (requisiti, architettura, codice, test).
- Funziona come **riferimento stabile** per tutto il team.
- Contiene le **milestone principali** e permette il confronto tra stato pianificato e stato effettivo.
- Ogni deviazione da una baseline deve essere approvata formalmente (controllo di configurazione).

---

# Le tre baseline del progetto

Baseline	Significato	Obiettivi principali
RTB – Requirements and Technology Baseline	Congelamento dell’analisi dei requisiti e dello studio di fattibilità	- Produrre la <b>demo/PoC</b> - Analizzare a fondo i requisiti- Comprendere e modellare <b>UML</b>
PB – Product Baseline	Congelamento dell’architettura e della realizzazione dell’MVP	- Prodotto funzionante- Architettura consolidata- Fine dei casi d'uso, inizio del design
CA – Customer Acceptance (opzionale)	Validazione finale con il committente	- Collaudo ufficiale- Dimostrazione al cliente con team e docente

---

# Connessione con il ciclo di vita del software

- Ogni **baseline** corrisponde a una fase nel **modello di sviluppo** (a cascata, incrementale, evolutivo, agile).
- **RTB** → **Concezione e Analisi dei requisiti**
- **PB** → **Progettazione e Realizzazione**
- **CA** → **Collaudo e Rilascio**
- Le fasi possono essere sequenziali (cascata) o iterativo-incrementali (agile, evolutivo).

---

## Ciclo di vita e pianificazione

- Il **ciclo di vita del software** va da **concezione** → **sviluppo** → **uso** → **ritiro**.
  - Il progetto gestisce solo una parte di questo ciclo (concezione–sviluppo).
  - Ogni fase produce **deliverable** e si conclude con **verifica e validazione**.
  - Gli strumenti tipici di pianificazione sono:
    - **Gantt** (sequenze e durate temporali)
    - **PERT** (dipendenze e cammini critici)
    - **WBS** (scomposizione gerarchica delle attività)
- 

## Gestione di progetto: ruoli chiave

Ruolo	Responsabilità principali
Analista	Studia il dominio del problema, redige e chiarisce i requisiti
Progettista	Definisce architettura e design tecnico
Programmatore	Implementa e mantiene il prodotto
Verificatore	Controlla qualità, coerenza e conformità ai processi
Responsabile di progetto	Coordina risorse, pianifica e rappresenta il team
Amministratore	Gestisce ambiente, versioni, configurazioni

---

## Analisi dei requisiti e RTB

- Nasce dal **capitolato** del committente (requisiti utente).
  - Obiettivi:
    - Comprendere il **dominio d’uso**
    - Definire e classificare i requisiti (**obbligatori, desiderabili, opzionali**)
    - Tracciare ogni requisito nel tempo (**tracciabilità**)
  - Tecniche: interviste, use case, prototipazione, brainstorming
  - La qualità dei requisiti (IEEE 830):
    - **Corretti, completi, consistenti, verificabili, tracciabili**
- 

## Dal RTB al PB

**RTB → PB = passaggio dall’analisi al design**

- Divisione dei requisiti in **sotto-sistemi** e definizione delle interfacce.
  - Si stabilisce **l’architettura complessiva** e si pianificano gli incrementi.
  - Da qui nasce il **Minimum Viable Product (MVP)**.
  - Riferimento diretto alle fasi “Design architetturale” e “Design di dettaglio” (ISO 12207).
- 

## Customer Acceptance (CA)

- Fase finale opzionale ma strategica.
  - **Scopo:** validazione del sistema da parte del cliente e del docente.
  - **Attività:** dimostrazione funzionale, confronto con i requisiti tracciati, accettazione formale.
  - Chiude il ciclo con il collaudo e porta allo stato **Fulfilled** (SEMAT).
- 

## Collegamento ai modelli di sviluppo

Modello	Caratteristiche	Gestione Baseline
Cascata	Fasi rigide e sequenziali	RTB → PB → CA lineari
Incrementale	Rilasci successivi e progressivi	Baseline per ogni incremento
Evolutivo	Versioni parallele in evoluzione	Baseline multiple per rami
Agile (Scrum)	Iterazioni brevi, incremento di valore	Baseline “leggere” e dinamiche

---

## Conclusione

- Le **baseline** sono strumenti di **controllo** e **comunicazione** tra fasi.
  - Consentono di gestire un progetto in modo **misurabile, tracciabile e ripetibile**.
  - RTB, PB e CA rappresentano la **spina dorsale del progetto software**:
    - Analisi e validazione dei requisiti (RTB)
    - Costruzione dell'architettura e rilascio del MVP (PB)
    - Verifica finale e accettazione (CA)
-