

# Il processo di *ingegneria dei requisiti*

- Il processo di **ingegneria dei requisiti** (*requirements engineering*) varia in base al dominio applicativo, alle persone coinvolte ed all'organizzazione che sviluppa il sistema software
- Si può però individuare un insieme di attività generiche comuni a tutti i processi:
  - studio di fattibilità (*feasibility study*)
  - identificazione e analisi dei requisiti (*req. elicitation and analysis*)
  - specifica dei requisiti (*req. specification*)
  - convalida dei requisiti (*req. validation*)
  - gestione dei requisiti (*req. management*)

# Studio di fattibilità

- Fase iniziale del processo di ingegneria dei requisiti
- Si basa su una **descrizione sommaria** del sistema software e delle necessità utente
- Le informazioni necessarie per lo studio di fattibilità vengono raccolte da **colloqui** con:
  - **client manager**
  - **ingegneri del software** con esperienza nello specifico *dominio applicativo*
  - **esperti delle tecnologie** da utilizzare
  - **utenti finali** del sistema

# Report di fattibilità

- Lo studio di fattibilità produce come risultato un **report** che **stabilisce l'opportunità o meno di procedere allo sviluppo** del sistema software
- **Domande tipiche** dello studio di fattibilità:
  - In che termini il sistema software contribuisce al raggiungimento degli **obiettivi strategici** del cliente?
  - Può il sistema software essere sviluppato usando le **tecnologie** correnti e rispettando i **vincoli** di durata e costo complessivo?
  - Può il sistema software essere **integrato** con altri sistemi già in uso?

# Attività di identificazione e analisi dei requisiti

- Il team di sviluppo incontra il cliente e gli utenti finali al fine di **identificare** l'insieme dei requisiti utente, dalla cui **analisi** si generano i requisiti di sistema (specifiche)
- L'identificazione dei requisiti può coinvolgere personale che copre **vari ruoli** sia all'interno dell'organizzazione del cliente che in altre organizzazioni o tra gli utenti finali
- Il termine *stakeholder* viene usato per identificare tutti coloro che hanno un interesse diretto o indiretto sui requisiti del sistema software da sviluppare

# Identificazione e analisi dei requisiti

## Task

- **Comprensione del dominio**: l'analista deve acquisire conoscenze sul dominio applicativo (es. se il sistema software deve supportare il lavoro di un ufficio postale, l'analista deve comprendere il funzionamento di un ufficio postale)
- **Raccolta dei requisiti**: mediante interazione con gli stakeholder si identificano i requisiti utente
- **Classificazione**: l'insieme dei requisiti raccolti viene suddiviso in sotto-insiemi coerenti di requisiti
- **Risoluzione dei conflitti**: eventuali contraddizioni e/o conflitti tra requisiti vanno identificati e risolti
- **Assegnazione delle priorità**: mediante interazione con gli stakeholder, ad ogni requisito o sotto-insiemi di requisiti va assegnata una classe di priorità
- **Verifica dei requisiti**: i requisiti vengono controllati per verificarne completezza e consistenza, in accordo a quanto richiesto dagli stakeholder

# Identificazione e analisi dei requisiti (2)

- Tecniche di **identificazione dei requisiti**
  - Ethnography
  - Casi d'uso (basati su *scenari*)
  - Prototipazione
- Tecniche di **analisi (e specifica) dei requisiti**
  - **semi-formali**, basate su *modelli del sistema* e usate dai metodi di *analisi strutturata* o *analisi orientata agli oggetti*
  - **formali** (basate su Petri Net, FSM, Z, etc.)

# Convalida dei requisiti

- La convalida dei requisiti è finalizzata ad accertare se il documento dei requisiti, ottenuto come risultato della fase di analisi, descrive realmente il sistema software che il cliente si aspetta
- La scoperta di errori in questa fase è fondamentale per evitare costosi *rework* in fasi più avanzate del ciclo di vita
- I **controlli** da effettuare includono:
  - validità
  - consistenza
  - completezza
  - realizzabilità
  - verificabilità

# Tecniche di convalida dei requisiti

Le *tecniche di convalida* dei requisiti includono:

- revisioni informali
- revisioni formali
  - *walkthrough*
  - *ispezioni*
- prototipazione
- generazione dei test-case
- analisi di consistenza automatizzata (per requisiti formali)



# Gestione dei requisiti

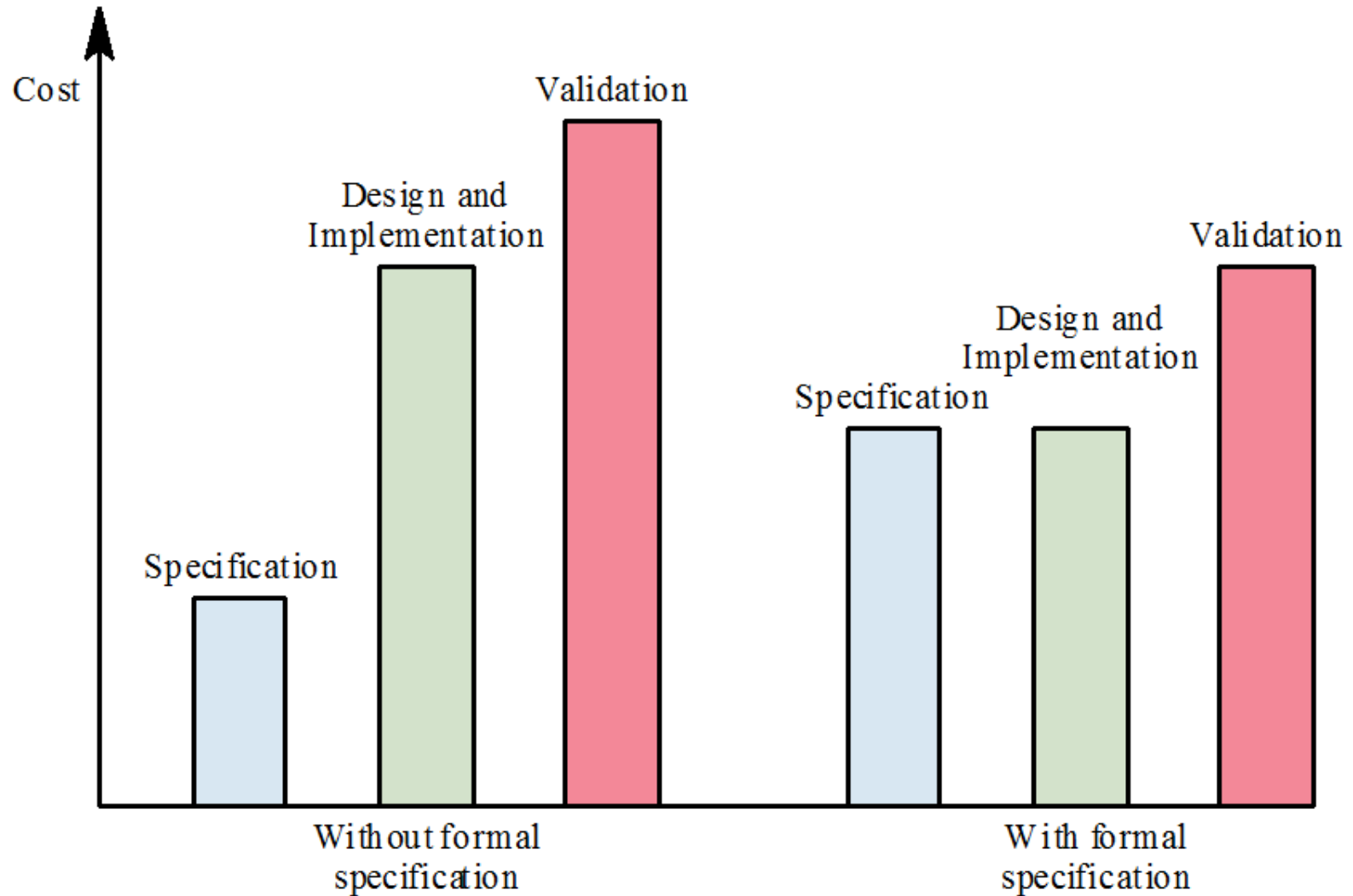
- Processo di identificazione e **controllo delle modifiche** subite dai requisiti di un sistema software lungo il ciclo di vita
- I requisiti di un sistema software possono essere classificati in termini della loro evoluzione come:
  - requisiti **stabili** (probabilità minima di essere modificati nel tempo)
  - requisiti **volatili** (probabilità elevata di essere modificati nel tempo):
    - **mutabili** (modifiche legate a cambiamenti nell'ambiente operativo)
    - **emergenti** (modifiche causate da una migliore comprensione del sistema software)
    - **consequenziali** (modifiche legate all'introduzione di sistemi informatici nel flusso di lavoro)
    - **di compatibilità** (modifiche legate a cambiamenti nei sistemi e nei processi aziendali)

# Gestione delle modifiche di requisiti

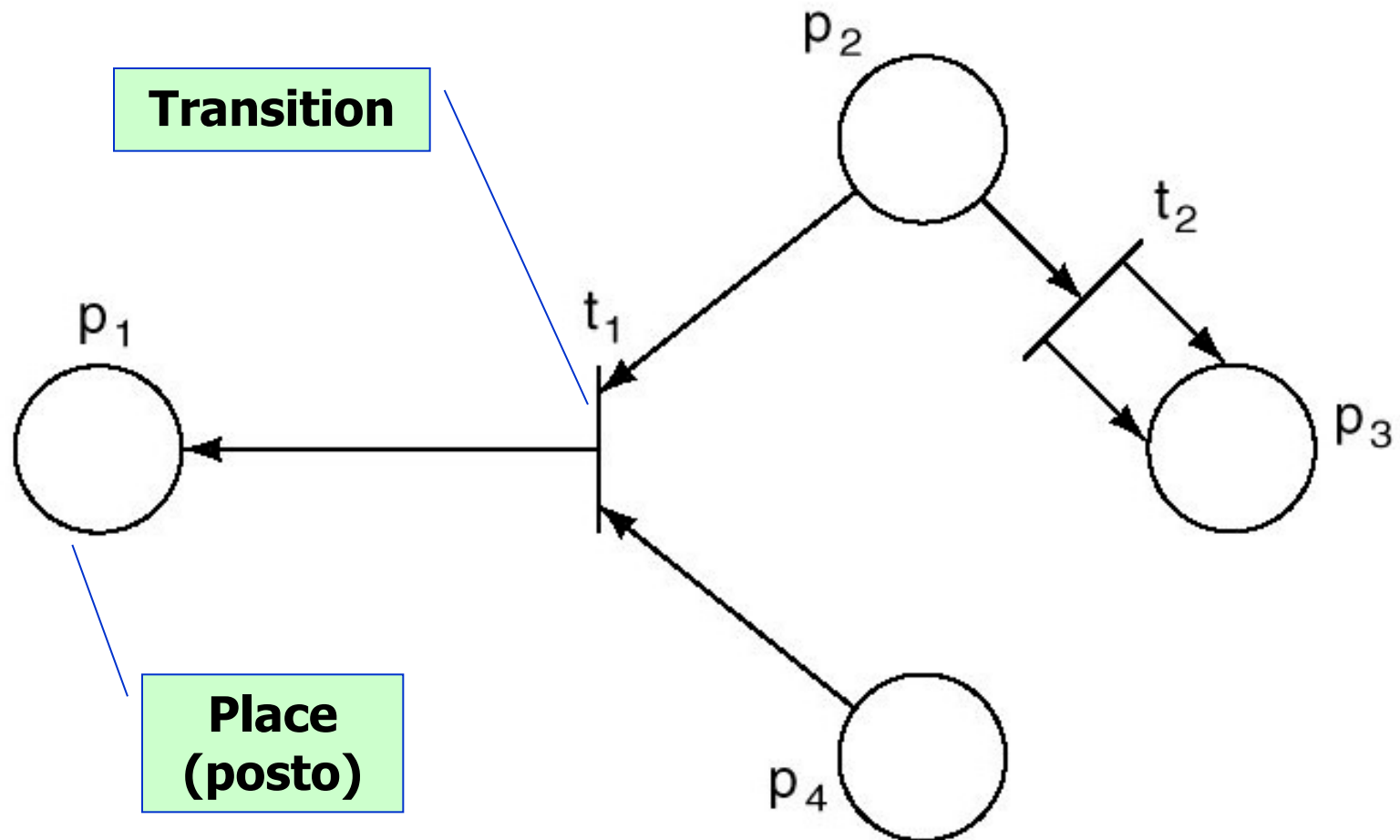
Le modifiche dei requisiti vanno opportunamente pianificate mediante:

- identificazione univoca dei requisiti
- gestione delle modifiche
  - analisi dei costi
  - analisi dell'impatto
  - analisi della realizzazione
- politiche di *tracciabilità* (relazioni tra requisiti e tra requisiti e progetto del sistema software)
- uso di tool CASE per il supporto alle modifiche

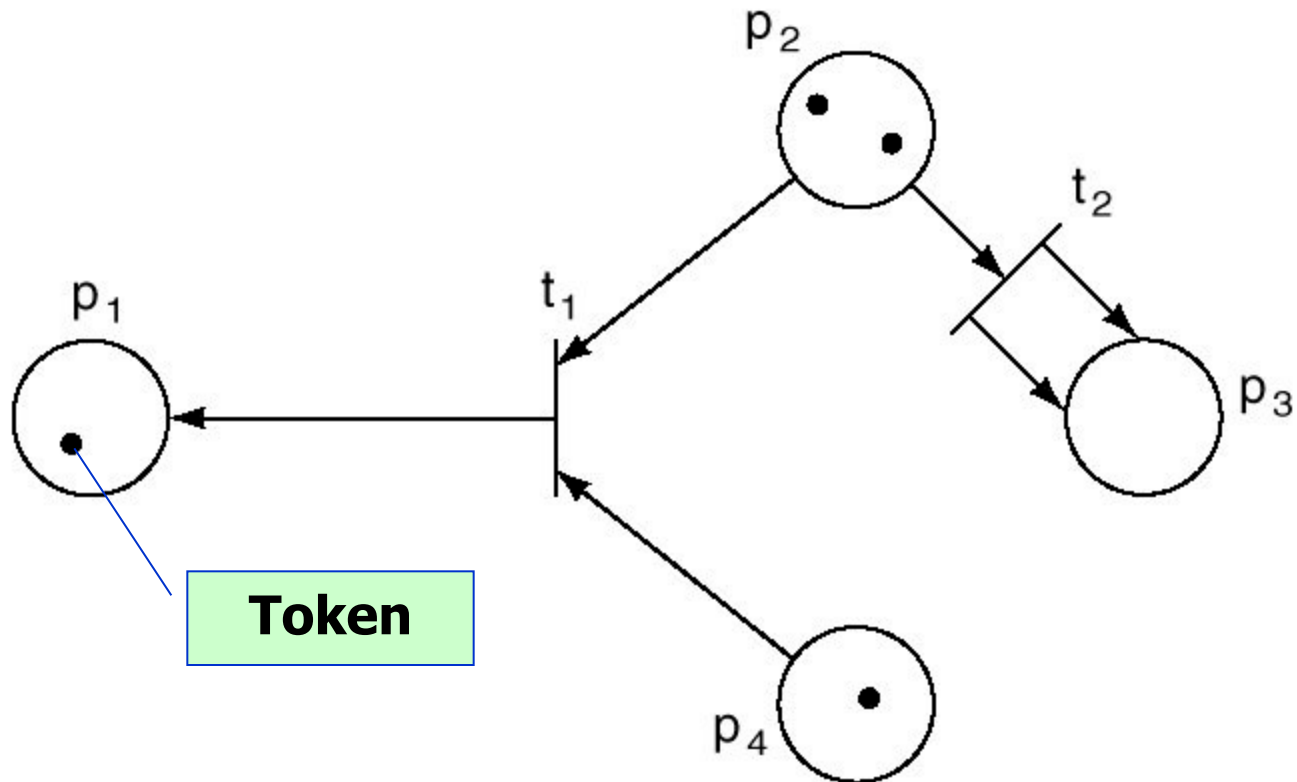
# Specifiche formali vs. informali



# Specifiche formali con Petri Net

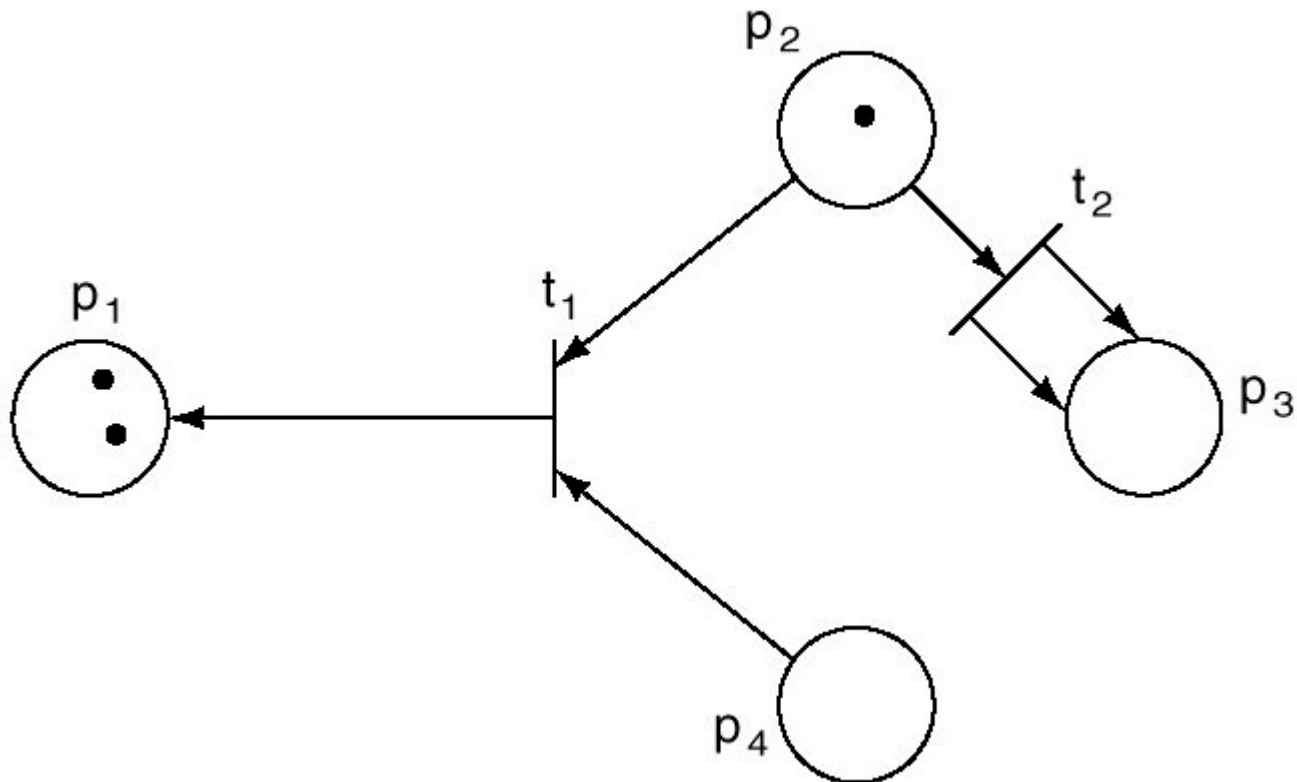


# Marked Petri Net (1)



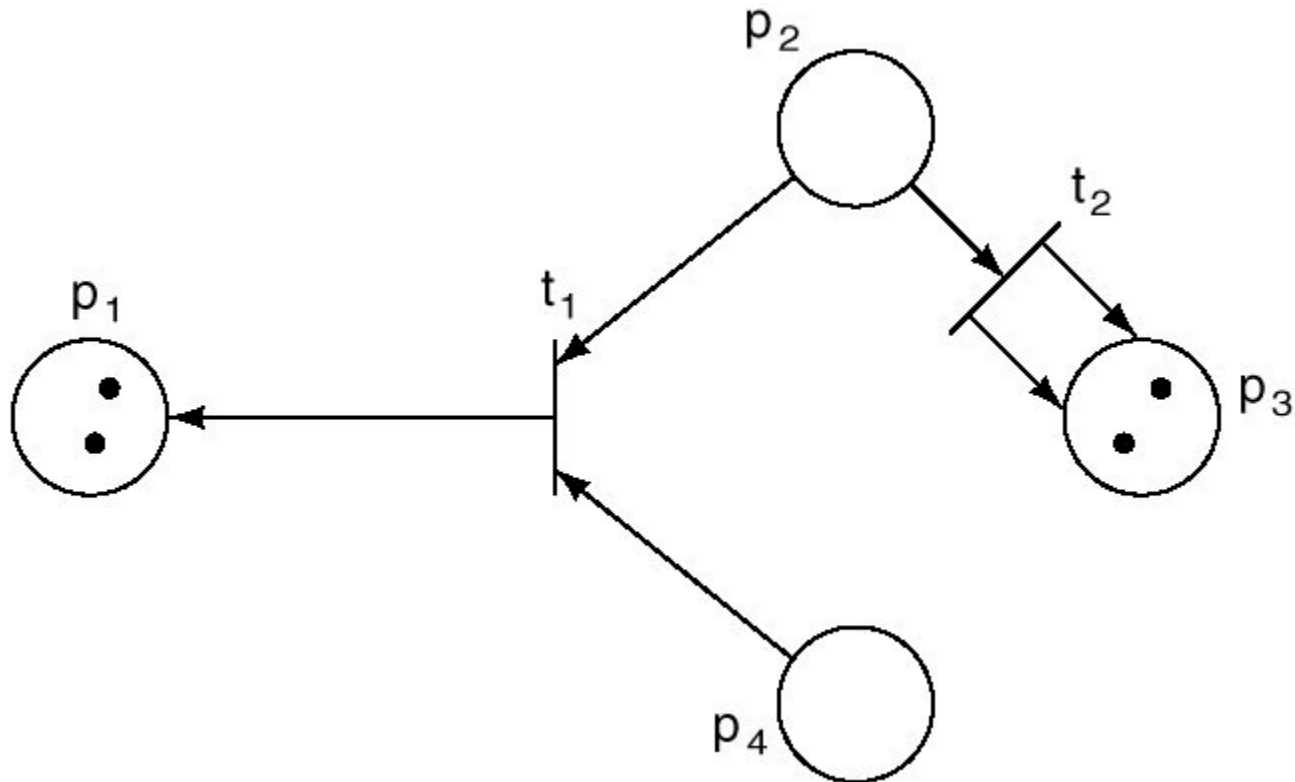
# Marked Petri Net (2)

- after firing transition  $t_1$



# Marked Petri Net (3)

- after firing transition  $t_2$



# Petri Net: esempio

