



# Ingegneria dei Requisiti

Cosa? - difficoltà, tecniche e processi

Andrea Polini

Ingegneria del Software  
Corso di Laurea in Informatica

# Sommario

- 1 Requisiti
  - Generalità
  - Requisiti Utente e di Sistema
  - Requisiti funzionali
  - Requisiti qualitativi
  - Vincoli
  - Requisiti di dominio
- 2 Attività dell'ingegneria dei requisiti
  - Studio di fattibilità
  - Elicitazione
    - Documentazione
    - Negoziazione e prioritizzazione
  - Validazione
  - Gestione

# Sommario

## 1 Requisiti

- Generalità
- Requisiti Utente e di Sistema
- Requisiti funzionali
- Requisiti qualitativi
- Vincoli
- Requisiti di dominio

## 2 Attività dell'ingegneria dei requisiti

- Studio di fattibilità
- Elicitazione
  - Documentazione
  - Negoziazione e prioritizzazione
- Validazione
- Gestione

# Sommario

- 1 Requisiti
  - Generalità
  - Requisiti Utente e di Sistema
  - Requisiti funzionali
  - Requisiti qualitativi
  - Vincoli
  - Requisiti di dominio
  
- 2 Attività dell'ingegneria dei requisiti
  - Studio di fattibilità
  - Elicitazione
    - Documentazione
    - Negoziazione e prioritizzazione
  - Validazione
  - Gestione

# Ingegneria dei Requisiti

Disciplina che si occupa di definire **cosa un sistema debba fare**, le sue **proprietà essenziali** ed i vincoli a cui deve rispondere. Scoprire, analizzare, documentare e validare i requisiti sono attività investigate dalla disciplina dell'**ingegneria dei requisiti**

Attività che presentano forte **interazione e comunicazione con il cliente**. Dunque non soltanto attività dagli aspetti tecnici ma forti **implicazioni socio-antropologiche** (ci si riferisce a questa attività come “comunicazione”)

# Software Intensive Systems

Il nostro interesse riguarda i “**Software Intensive systems**” ovvero quei sistemi che per cui larga parte delle funzionalità offerte dipendono o sono realizzate da software.

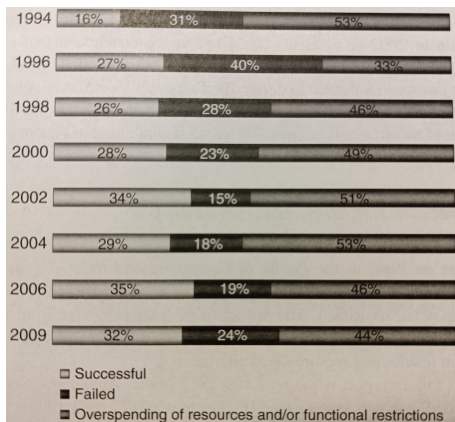
- **Information systems**: collezionano, Immagazzinano, trasformano, trasmettono, e/o processano dati o informazioni. Fornire le informazioni di cui gli utenti hanno bisogno nel posto giusto al momento giusto. **Software che viene eseguito su calcolatori general purpose**
- **Embedded Software-intensive Systems**: software è solo un parte di un sistema complesso che serve ad abilitare funzionalità innovative. **Il software è strettamente legato a dispositivi HW con il quale si stabiliscono complesse interazioni**. Il software viene eseguito su dispositivi specifici per lo scopo.

# Sfide nello sviluppo di SIS

- Software come motore di innovazione - faremo esperienza con Design Thinking
- Complessità sempre crescente
- Riduzione dei costi
- Riduzione del time to market
- Aumento della qualità

Costante necessità di nuovi metodi e approcci allo sviluppo, dove gli aspetti legati alle attività di “comunicazione” (Requisiti) rappresentano sempre più un fattore critico

# Stato della disciplina dell'IdS



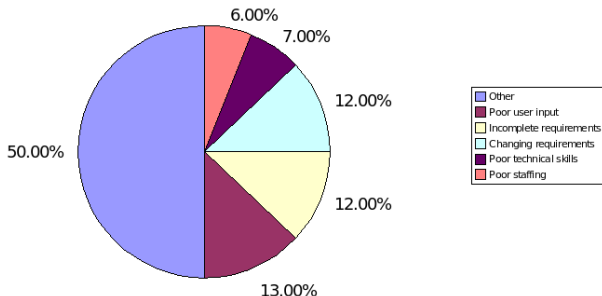
Percentuale di successo tra il 1994 e il 2009 - Standish group

da K. Pohl - Requirements Engineering - Springer Verlag



# Rilevanza dei Requisiti

- La fase di gestione dei requisiti è probabilmente la più critica
- Problemi inseriti in questa fase dello sviluppo sono i più costosi da rimuovere.
- Studi rivelano che circa il 48% dei problemi, nello sviluppo di sistemi software “challenging”, sono relativi alla attività della comunicazione



# Tipici problemi

- **La visione e lo scopo del progetto non sono state chiaramente definiti**
- I clienti erano troppo impegnati per lavorare con gli analisti e gli sviluppatori
- Il team di sviluppo non ha potuto mai interagire con gli utenti
- I clienti hanno dichiarato tutti i requisiti come critici, nessuna prioritizzazione
- Gli sviluppatori hanno incontrato ambiguità durante lo sviluppo
- La comunicazione si è focalizzata su come le interfacce dovessero essere rappresentate e non sulle funzionalità
- Il cliente non ha mai approvato i requisiti
- L'ambito del progetto è continuamente aumentato mentre lo sviluppo progrediva
- Richieste di modifiche ai requisiti sono state ignorate
- ...

# Tipici problemi

- La visione e lo scopo del progetto non sono state chiaramente definiti
- I clienti erano troppo impegnati per lavorare con gli analisti e gli sviluppatori
- Il team di sviluppo non ha potuto mai interagire con gli utenti
- I clienti hanno dichiarato tutti i requisiti come critici, nessuna prioritizzazione
- Gli sviluppatori hanno incontrato ambiguità durante lo sviluppo
- La comunicazione si è focalizzata su come le interfacce dovessero essere rappresentate e non sulle funzionalità
- Il cliente non ha mai approvato i requisiti
- L'ambito del progetto è continuamente aumentato mentre lo sviluppo progrediva
- Richieste di modifiche ai requisiti sono state ignorate
- ...

# Tipici problemi

- La visione e lo scopo del progetto non sono state chiaramente definiti
- I clienti erano troppo impegnati per lavorare con gli analisti e gli sviluppatori
- Il team di sviluppo non ha potuto mai interagire con gli utenti
- I clienti hanno dichiarato tutti i requisiti come critici, nessuna prioritizzazione
- Gli sviluppatori hanno incontrato ambiguità durante lo sviluppo
- La comunicazione si è focalizzata su come le interfacce dovessero essere rappresentate e non sulle funzionalità
- Il cliente non ha mai approvato i requisiti
- L'ambito del progetto è continuamente aumentato mentre lo sviluppo progrediva
- Richieste di modifiche ai requisiti sono state ignorate
- ...

# Tipici problemi

- La visione e lo scopo del progetto non sono state chiaramente definiti
- I clienti erano troppo impegnati per lavorare con gli analisti e gli sviluppatori
- Il team di sviluppo non ha potuto mai interagire con gli utenti
- I clienti hanno dichiarato tutti i requisiti come critici, nessuna prioritizzazione
- Gli sviluppatori hanno incontrato ambiguità durante lo sviluppo
- La comunicazione si è focalizzata su come le interfacce dovessero essere rappresentate e non sulle funzionalità
- Il cliente non ha mai approvato i requisiti
- L'ambito del progetto è continuamente aumentato mentre lo sviluppo progrediva
- Richieste di modifiche ai requisiti sono state ignorate
- ...

# Tipici problemi

- La visione e lo scopo del progetto non sono state chiaramente definiti
- I clienti erano troppo impegnati per lavorare con gli analisti e gli sviluppatori
- Il team di sviluppo non ha potuto mai interagire con gli utenti
- I clienti hanno dichiarato tutti i requisiti come critici, nessuna prioritizzazione
- Gli sviluppatori hanno incontrato ambiguità durante lo sviluppo
- La comunicazione si è focalizzata su come le interfacce dovessero essere rappresentate e non sulle funzionalità
- Il cliente non ha mai approvato i requisiti
- L'ambito del progetto è continuamente aumentato mentre lo sviluppo progrediva
- Richieste di modifiche ai requisiti sono state ignorate
- ...

# Tipici problemi

- La visione e lo scopo del progetto non sono state chiaramente definiti
- I clienti erano troppo impegnati per lavorare con gli analisti e gli sviluppatori
- Il team di sviluppo non ha potuto mai interagire con gli utenti
- I clienti hanno dichiarato tutti i requisiti come critici, nessuna prioritizzazione
- Gli sviluppatori hanno incontrato ambiguità durante lo sviluppo
- La comunicazione si è focalizzata su come le interfacce dovessero essere rappresentate e non sulle funzionalità
- Il cliente non ha mai approvato i requisiti
- L'ambito del progetto è continuamente aumentato mentre lo sviluppo progrediva
- Richieste di modifiche ai requisiti sono state ignorate
- ...

# Tipici problemi

- La visione e lo scopo del progetto non sono state chiaramente definiti
- I clienti erano troppo impegnati per lavorare con gli analisti e gli sviluppatori
- Il team di sviluppo non ha potuto mai interagire con gli utenti
- I clienti hanno dichiarato tutti i requisiti come critici, nessuna prioritizzazione
- Gli sviluppatori hanno incontrato ambiguità durante lo sviluppo
- La comunicazione si è focalizzata su come le interfacce dovessero essere rappresentate e non sulle funzionalità
- Il cliente non ha mai approvato i requisiti
- L'ambito del progetto è continuamente aumentato mentre lo sviluppo progrediva
- Richieste di modifiche ai requisiti sono state ignorate
- ...



# Tipici problemi

- La visione e lo scopo del progetto non sono state chiaramente definiti
- I clienti erano troppo impegnati per lavorare con gli analisti e gli sviluppatori
- Il team di sviluppo non ha potuto mai interagire con gli utenti
- I clienti hanno dichiarato tutti i requisiti come critici, nessuna prioritizzazione
- Gli sviluppatori hanno incontrato ambiguità durante lo sviluppo
- La comunicazione si è focalizzata su come le interfacce dovessero essere rappresentate e non sulle funzionalità
- Il cliente non ha mai approvato i requisiti
- L'ambito del progetto è continuamente aumentato mentre lo sviluppo progrediva
- Richieste di modifiche ai requisiti sono state ignorate
- ...

# Tipici problemi

- La visione e lo scopo del progetto non sono state chiaramente definiti
- I clienti erano troppo impegnati per lavorare con gli analisti e gli sviluppatori
- Il team di sviluppo non ha potuto mai interagire con gli utenti
- I clienti hanno dichiarato tutti i requisiti come critici, nessuna prioritizzazione
- Gli sviluppatori hanno incontrato ambiguità durante lo sviluppo
- La comunicazione si è focalizzata su come le interfacce dovessero essere rappresentate e non sulle funzionalità
- Il cliente non ha mai approvato i requisiti
- L'ambito del progetto è continuamente aumentato mentre lo sviluppo progrediva
- Richieste di modifiche ai requisiti sono state ignorate
- ...

# Requisito

## Definizione

- ▶ A condition or capability needed by a user to solve a problem or achieve an objective.
- ▶ A condition or capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification, or other formally imposed documents.
- ▶ A documented representation of a condition or capability as in (1) or (2).

[IEEE std. 610.12-1990]

# Tipi di requisiti

I requisiti vengono classificati in molti modi a seconda dell'obiettivo che ci si pone:

- Se ci si concentra su chi sono i **destinatari** del requisito:
  - **Requisiti utente**
  - **Requisiti di sistema**
- Ortogonalmente se ci focalizza sul **carattere** del requisito, si distingue in:
  - Requisiti funzionali
  - Requisiti qualitativi
  - Vincoli
- Altra categoria interessante riguarda l'**origine** dei requisiti:
  - Requisiti di dominio

# Tipi di requisiti

I requisiti vengono classificati in molti modi a seconda dell'obiettivo che ci si pone:

- Se ci si concentra su chi sono i **destinatari** del requisito:
  - **Requisiti utente**
  - **Requisiti di sistema**
- Ortogonalmente se ci focalizza sul **carattere** del requisito, si distingue in:
  - **Requisiti funzionali**
  - **Requisiti qualitativi**
  - **Vincoli**
- Altra categoria interessante riguarda l'origine dei requisiti:
  - **Requisiti di dominio**

# Tipi di requisiti

I requisiti vengono classificati in molti modi a seconda dell'obiettivo che ci si pone:

- Se ci si concentra su chi sono i **destinatari** del requisito:
  - **Requisiti utente**
  - **Requisiti di sistema**
- Ortogonalmente se ci focalizza sul **carattere** del requisito, si distingue in:
  - **Requisiti funzionali**
  - **Requisiti qualitativi**
  - **Vincoli**
- Altra categoria interessante riguarda l'**origine** dei requisiti:
  - **Requisiti di dominio**

# Sommario

- 1 **Requisiti**
  - Generalità
  - **Requisiti Utente e di Sistema**
  - Requisiti funzionali
  - Requisiti qualitativi
  - Vincoli
  - Requisiti di dominio
- 2 **Attività dell'ingegneria dei requisiti**
  - Studio di fattibilità
  - Elicitazione
    - Documentazione
    - Negoziazione e prioritizzazione
  - Validazione
  - Gestione

# Requisiti Utente vs. Requisiti di Sistema

## Requisiti Utente

- si rivolgono principalmente all'utente
- alto livello di astrazione
- usano linguaggio naturale e diagrammi

## Requisiti di Sistema

- si rivolgono principalmente ai progettisti/sviluppatori
- alto livello di dettaglio e precisione



# Requisiti di sistema

Aggiungono dettagli per capire come gli obiettivi specificati dai requisiti utente possono essere effettivamente raggiunti dal sistema. Anche questi si dovrebbero **limitare al comportamento osservabile e non contenere scelte che dovrebbero competere alle attività di design**. Ma ...

- Potreste aver bisogno di definire un'architettura iniziale per strutturare i requisiti.
- In molti casi il sistema interagirà con sistemi pre-esistenti che dunque in un certo qual modo forzano scelte progettuali  
*e.g. avete bisogno di certificare il sistema rispetto a norme di safety*  
*il sistema interagisce con un sistema che utilizza formati XML*

# Requisiti di sistema

Aggiungono dettagli per capire come gli obiettivi specificati dai requisiti utente possono essere effettivamente raggiunti dal sistema. Anche questi si dovrebbero **limitare al comportamento osservabile e non contenere scelte che dovrebbero competere alle attività di design**. Ma ...

- Potreste aver bisogno di definire un'architettura iniziale per strutturare i requisiti.
- In molti casi il sistema interagirà con sistemi pre-esistenti che dunque in un certo qual modo forzano scelte progettuali  
*e.g. avete bisogno di certificare il sistema rispetto a norme di safety*  
*il sistema interagisce con un sistema che utilizza formati XML*

# Come specificare i requisiti

Differenti tecniche possibili ovviamente non esclusive (requisiti possono essere definiti e poi raffinati con diversi stili):

- **Informali**: usano tipicamente linguaggi naturali
- **Semi formali**: usano notazioni grafiche per cui la semantica non è sempre precisamente definita
- **Formali**: attraverso modelli matematici

L'uso di una tecnica piuttosto che di un'altra dipende dal metodo usato e dal contesto

**Esempio**: si consideri la specifica di un sistema di controllo apertura sbarra passaggio a livello

# Esempio: sistema di gestione di una biblioteca

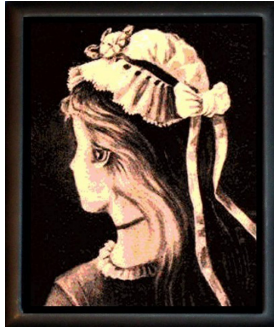
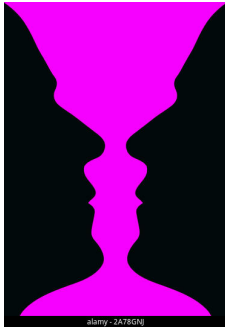
## Definizione di requisito utente:

- *Il sistema deve tener traccia di tutti i dati richiesti dalla normativa sul copyright*

## Requisiti di Sistema

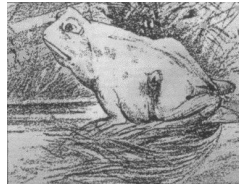
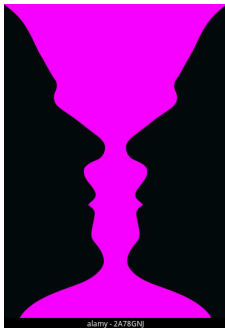
- *Il sistema deve prevedere un meccanismo di autenticazione per gli utenti*
- *il sistema deve memorizzare ogni richiesta per almeno 5 anni*
- *Il sistema deve mantenere traccia di tutte le richieste fatte*
- *Il sistema deve memorizzare informazioni sugli utenti sul materiale oggetto della richiesta, e sul personale che ha gestito la richiesta (??)*
- *Il sistema deve produrre resoconti mensili da inviare alle agenzie di controllo.*

# Ambiguità



“Figure” ambigue sono molto facili da creare con il linguaggio naturale

# Ambiguità



“Figure” ambigue sono molto facili da creare con il linguaggio naturale

# Ambiguità nel linguaggio naturale

Esistono diverse possibili sorgenti di ambiguità nell'uso del linguaggio naturale che possono impattare la specifica di requisiti:

- **Lessicale:** I termini usati sono polisemici, ovvero possono avere più significati.
  - Al di fuori di un cane, un libro è il miglior amico dell'uomo; da dentro è troppo difficile leggere (Groucho Marx)
- **Sintattica:** la frase ha più di un albero sintattico
  - Chiara ha visto Luca in giardino con il canocchiale
  - Una vecchia legge la regola
- **Semantica:** l'equivalente della frase nella logica dei predicati ha più di un'interpretazione
  - Tutti i linguisti preferiscono una teoria
- **Pragmatica:** l'interpretazione dipende dal contesto - "l'entrata è sulla destra"

# Requisiti di sistema

Problemi nell'uso di linguaggio naturale:

- si basa sulla comune comprensione dei concetti nel sistema
- troppo flessibile
- difficile modularizzare requisiti scritti con linguaggio naturale

Uso di notazioni semi-formali o formali

- Linguaggio Naturale Strutturato
- Linguaggi di Descrizione Progettuale
- Notazioni grafiche
- Specifiche Matematiche



# Scelte tecnologiche

Si considerino i seguenti requisiti:

- Il sistema deve permettere autenticazione degli utenti che accedono al prestito
- Il sistema deve permettere autenticazione degli utenti tramite login/password per accesso al prestito
- il sistema deve autenticare l'accesso dell'utente al prestito tramite un meccanismo di login e password

Quali sono le conseguenze di ognuno di questi requisiti? Come possono essere interpretati? Livello di astrazione e scelte tecnologiche?

# Requisiti utente

Specificano il comportamento del sistema in modo comprensibile al cliente. Si occupano del **comportamento osservabile** del sistema (input, output, eccezioni) per l'utente e non dovrebbero contenere specifiche di design.

Tipici problemi:

- Mancanza di chiarezza - **verbosità vs. precisione**
- Confusione - Le diverse tipologie di **requisiti sono mischiati** tra loro
- Accorpamento - Molti requisiti vengono specificati come un singolo requisito

# Requisiti utente

esempio

**Griglia di supporto:** nell'assistere l'utente nel posizionamento delle entità in un diagramma, l'utente può attivare una griglia, che fornisca sia i centimetri che i pollici, attraverso un'opzione nel pannello di controllo. Inizialmente la griglia è disattivata. La griglia può essere attivata/disattivata in qualsiasi momento. Un'opzione griglia verrà fornita nella vista adatta-a-dimensioni ma il numero di linee mostrate sarà ridotto per evitare di riempire diagrammi più piccoli con linee di griglia.

Problemi?

# Requisiti utente

## esempio

Molti differenti tipi di requisiti sono mischiati nella prima frase:

- funzionali: la griglia
- non funzionali: centimetri / pollici
- non funzionale: dove il meccanismo si trova

Nota: Troppi dettagli tecnici limitano il raggio di azione degli sviluppatori che invece potrebbero fornire soluzioni innovative.

# Requisiti utente

esempio

L'editor deve fornire una funzionalità griglia dove una **matrice di linee orizzontali e verticali** vengano visualizzate come **background** della vista nella finestra dell'editor. La griglia dovrebbe essere **passiva** e l'allineamento alla griglia deve essere fatto su iniziativa dell'utente.

*Motivazione: una griglia aiuta l'utente nella creazione di un diagramma più pulito con entità ben spaziate. Una griglia attiva può essere utile ma può creare effetti indesiderati come posizionamenti imprecisi. L'utente è la persona più appropriata a decidere il posizionamento*

Sorgente: Micky Mouse

# Requisiti utente

## esercizio

Il sistema di gestione della biblioteca intende fornire un supporto alla gestione dei “conti” che in particolare riporti tutti i pagamenti fatti dagli utenti del sistema. I gestori del sistema devono poter configurare il sistema in modo da poter accordare sconti ad utenti regolari.

Problemi?

# Sommario

- 1 **Requisiti**
  - Generalità
  - Requisiti Utente e di Sistema
  - **Requisiti funzionali**
  - Requisiti qualitativi
  - Vincoli
  - Requisiti di dominio
- 2 **Attività dell'ingegneria dei requisiti**
  - Studio di fattibilità
  - Elicitazione
    - Documentazione
    - Negoziazione e prioritizzazione
  - Validazione
  - Gestione

# Requisiti funzionali

## Definizione

Statements of services the system **should provide**, how the system **should react** to particular inputs and how the system **should behave** in particular situations. In some cases, the functional requirements may also state **what the system should not do** . . .

When expressed as user requirements, the requirements are usually described in a **fairly abstract way**. However, functional system requirements describe the system function in detail **its input and outputs, exceptions, and so on**.

[Sommerville - Software Engineering - Addison Wesley]



# Sommario

- 1 **Requisiti**
  - Generalità
  - Requisiti Utente e di Sistema
  - Requisiti funzionali
  - **Requisiti qualitativi**
  - Vincoli
  - Requisiti di dominio
- 2 Attività dell'ingegneria dei requisiti
  - Studio di fattibilità
  - Elicitazione
    - Documentazione
    - Negoziazione e prioritizzazione
  - Validazione
  - Gestione

# Requisiti Qualitativi

## Definizione

Un requisito qualitativo definisce una proprietà qualitativa dell'intero sistema o di un suo componente, servizio o funzione

[Pohl - Requirements Engineering - Springer]

Generalmente tutti i termini che rientrano nello schema \*-ability:

- Availability
- Efficiency
- Flexibility
- Integrity
- Interoperability
- Reliability
- Robustness
- Usability
- Maintainability
- Portability
- Reusability
- Testability
- Understandability
- ...

Dedicheremo una lezione a questo tipo di requisiti che risultano essere fondamentali nello sviluppo software

# Requisiti Qualitativi

## Definizione

Un requisito qualitativo definisce una proprietà qualitativa dell'intero sistema o di un suo componente, servizio o funzione

[Pohl - Requirements Engineering - Springer]

Generalmente tutti i termini che rientrano nello schema \*-ability:

- Availability
- Efficiency
- Flexibility
- Integrity
- Interoperability
- Reliability
- Robustness
- Usability
- Maintainability
- Portability
- Reusability
- Testability
- Understandability
- ...

Dedicheremo una lezione a questo tipo di requisiti che risultano essere fondamentali nello sviluppo software

# Sommario

- 1 **Requisiti**
  - Generalità
  - Requisiti Utente e di Sistema
  - Requisiti funzionali
  - Requisiti qualitativi
  - **Vincoli**
  - Requisiti di dominio
- 2 Attività dell'ingegneria dei requisiti
  - Studio di fattibilità
  - Elicitazione
    - Documentazione
    - Negoziazione e prioritizzazione
  - Validazione
  - Gestione

# Vincoli

## Definizione

Un vincolo è un requisito organizzativo o tecnologico che restringe il modo in cui il sistema deve essere sviluppato

[Robertson & Roberston - Mastering Requirements Eng.]

# Sommario

- 1 **Requisiti**
  - Generalità
  - Requisiti Utente e di Sistema
  - Requisiti funzionali
  - Requisiti qualitativi
  - Vincoli
  - **Requisiti di dominio**
- 2 Attività dell'ingegneria dei requisiti
  - Studio di fattibilità
  - Elicitazione
    - Documentazione
    - Negoziazione e prioritizzazione
  - Validazione
  - Gestione

# Requisiti di dominio

## Definition

I requisiti di dominio riguardano quei requisiti che derivano direttamente dallo specifico **dominio** e o **contesto applicativo**

Sono difficili da identificare perché ovvi al committente e spesso ignorati e non riportati

# Sommario

- 1 Requisiti
  - Generalità
  - Requisiti Utente e di Sistema
  - Requisiti funzionali
  - Requisiti qualitativi
  - Vincoli
  - Requisiti di dominio
- 2 Attività dell'ingegneria dei requisiti
  - Studio di fattibilità
  - Elicitazione
    - Documentazione
    - Negoziazione e prioritizzazione
  - Validazione
  - Gestione



# Processo di Ingegneria dei requisiti

Non esiste processo definitivo, attività tipicamente parte di un processo di ingegneria dei requisiti:

- Studio di fattibilità
- Elicitazione ed analisi dei requisiti
  - Scoperta dei requisiti
  - Classificazione ed organizzazione dei requisiti
  - Prioritizzazione dei requisiti e negoziazione
  - Documentazione dei Requisiti
- Validazione
- Gestione

Anche in questo caso le varie attività possono essere organizzate in diverse maniere. E.g. Iterativo - il peso delle varie attività dunque varierà nelle varie fasi.

# Sommario

- 1 Requisiti
  - Generalità
  - Requisiti Utente e di Sistema
  - Requisiti funzionali
  - Requisiti qualitativi
  - Vincoli
  - Requisiti di dominio
- 2 Attività dell'ingegneria dei requisiti
  - **Studio di fattibilità**
  - Elicitazione
    - Documentazione
    - Negoziazione e prioritizzazione
  - Validazione
  - Gestione

# Studio di fattibilità

**Studio preliminare** sulle implicazioni che il sistema avrà una volta costruito e sulla sua convenienza. Risultato di questa fase sarà una **raccomandazione sul continuare o meno lo sviluppo**. Può essere considerata una sorta di attività preliminare rispetto alle altre e in qualche modo le include.

Le domande a cui tipicamente uno studio di fattibilità dovrà rispondere sono:

- Il sistema contribuisce al **raggiungimento degli obiettivi** dell'organizzazione a cui è rivolto? Qual'è il suo impatto? (**Chiaramente questione di business**)
- Può il sistema essere implementato con le **tecnologie correnti** e con costi e tempi "prevedibili"?
- Può il sistema essere integrato con sistemi pre-esistenti?

# Studio di fattibilità

Nella raccolta delle informazioni sarà necessario interagire con il “cliente”. Alcune domande a cui dovrete cercare risposta sono:

- Come l'organizzazione risolverebbe il problema se non fosse possibile implementare il sistema?
- Quali sono i problemi con i processi attuali e come il sistema potrà risolverli?
- Quale contributo il sistema apporterà al raggiungimento degli obiettivi?
- Le informazioni possono essere **trasferite verso o da altre organizzazioni**?
- Il sistema richiederà l'introduzione di **nuove tecnologie**?
- Quali attività il sistema dovrà supportare e **cosa potrà essere lasciato fuori**?

# Sommario

- 1 Requisiti
  - Generalità
  - Requisiti Utente e di Sistema
  - Requisiti funzionali
  - Requisiti qualitativi
  - Vincoli
  - Requisiti di dominio
- 2 Attività dell'ingegneria dei requisiti
  - Studio di fattibilità
  - **Elicitazione**
    - Documentazione
    - Negoziazione e prioritizzazione
  - Validazione
  - Gestione

# Elicitazione ed analisi dei requisiti

## Difficoltà?

Primo passo è l'individuazione degli “*stakeholders*” (attori).

L'elicitazione dei requisiti è resa difficile da alcuni problemi “inevitabili”:

- attori non hanno piena **coscienza di ciò di cui hanno bisogno**. Hanno difficoltà ad identificare i limiti del sistema di cui hanno bisogno. Possono fornire dettagli che confondono e rendono difficile la focalizzazione sull'obiettivo principale. Possono trovare difficile esprimersi o possono richiedere sistemi inattuabili (dati anche corrispondenti costi)
- Uso di **linguaggio tecnico** del dominio applicativo
- **Stesso requisito** può essere espresso diversamente da differenti persone
- Requisiti aggiunti al fine di poter raggiungere obiettivi personali
- L'ambiente è tipicamente dinamico e le **condizioni possono mutare** anche repentinamente

# Elicitazione ed analisi dei requisiti

Difficoltà?

Elicitazione influenzata dalle caratteristiche dei processi cognitivi umani. In particolare attività di elicitazione devono considerare i processi mentali di:

- rimozione
- distorsione
- generalizzazione

Particolare attenzione ai termini: tutto, ogni, sempre, mai, nessuno, niente

# Scoperta dei requisiti

## punti di vista

Punti di vista permettono di **classificare gli attori**. Questo permette di avere un'idea della copertura ottenuta sui possibili requisiti. *Meglio intervistare 3 attori da gruppi differenti piuttosto che 10 da uno stesso gruppo.*

Tipicamente si distingue tra:

- **Punto di vista diretto**: chi interagisce direttamente con il sistema
- **Punto di vista indiretto**: chi non interagisce con il sistema ma è interessato al suo comportamento
- **Punto di vista di dominio**: attori esperti del dominio applicativo



# Confine del Sistema e Attori

Attore rappresenta il **ruolo** che un'entità esterna assume quando interagisce con il sistema. La stessa entità potrà ricoprire **più ruoli**. **Sorgenti e destinazioni delle informazioni** da e per il sistema devono essere presi sotto esame:

- **Persone**
- **Altri sistemi esterni**
- **Sensori e Attuatori**

Gli attori interagiscono con il sistema per mezzo di specifiche interfacce, **la cui specifica può essere pre-esistente**:

- **Interfacce Uomo-Macchina (HMI)**
- **Interfacce Software (API) e protocolli**
- **Interfacce Hardware**

# Individuazione del Confine del Sistema e degli Attori

## Individuazione degli attori:

- Chi o cosa usa il sistema? Chi installa il sistema?
- Chi partecipa alle varie fasi del ciclo di vita del sistema (avvio, manutenzione, dismissione, ...)?
- Chi ottiene informazioni dal sistema e a chi ne fornisce?
- Funzioni o azioni che vengono eseguite ad intervalli prestabiliti?

# Classificazione per gli attori

## Dove cercare gli attori

- **Attore primario**: utilizza direttamente il sistema
- **Attore finale**: vuole che il sistema sia utilizzato affinché vengano realizzati dei suoi obiettivi
- **Attore di supporto**: offre un servizio al sistema
- **Attore fuori scena**: ha un interesse nel comportamento del caso d'uso. Non rientra però tra gli attori menzionati

## Esempio: attori di un sistema bancomat (ATM)

Gli attori che sono interessati al funzionamento di un sistema sono vari. Non bisogna limitarsi soltanto ai più ovvi:

- Correntisti
- Altre banche

## Esempio: attori di un sistema bancomat (ATM)

Gli attori che sono interessati al funzionamento di un sistema sono vari. Non bisogna limitarsi soltanto ai più ovvi:

- Correntisti

- Altre banche

- Manager di filiali

## Esempio: attori di un sistema bancomat (ATM)

Gli attori che sono interessati al funzionamento di un sistema sono vari. Non bisogna limitarsi soltanto ai più ovvi:

- Correntisti
- Altre banche

● Manager di filiali

● Bancari

## Esempio: attori di un sistema bancomat (ATM)

Gli attori che sono interessati al funzionamento di un sistema sono vari. Non bisogna limitarsi soltanto ai più ovvi:

- Correntisti
- Altre banche
- Manager di filiali
- Bancari
- Amministratori di database
- Negozianti

## Esempio: attori di un sistema bancomat (ATM)

Gli attori che sono interessati al funzionamento di un sistema sono vari. Non bisogna limitarsi soltanto ai più ovvi:

- Correntisti
- Altre banche
- Manager di filiali
- Bancari
- Amministratori di database
- Manager della sicurezza



## Esempio: attori di un sistema bancomat (ATM)

Gli attori che sono interessati al funzionamento di un sistema sono vari. Non bisogna limitarsi soltanto ai più ovvi:

- Correntisti
- Altre banche
- Manager di filiali
- Bancari
- Amministratori di database
- Manager della sicurezza
- La divisione marketing

## Esempio: attori di un sistema bancomat (ATM)

Gli attori che sono interessati al funzionamento di un sistema sono vari. Non bisogna limitarsi soltanto ai più ovvi:

- Correntisti
- Altre banche
- Manager di filiali
- Bancari
- Amministratori di database
- Manager della sicurezza
- La divisione marketing
- I manutentori dello hardware e del software

## Esempio: attori di un sistema bancomat (ATM)

Gli attori che sono interessati al funzionamento di un sistema sono vari. Non bisogna limitarsi soltanto ai più ovvi:

- Correntisti
- Altre banche
- Manager di filiali
- Bancari
- Amministratori di database
- Manager della sicurezza
- La divisione marketing
- I manutentori dello hardware e del software
- Le autorità nazionali

## Esempio: attori di un sistema bancomat (ATM)

Gli attori che sono interessati al funzionamento di un sistema sono vari. Non bisogna limitarsi soltanto ai più ovvi:

- Correntisti
- Altre banche
- Manager di filiali
- Bancari
- Amministratori di database
- Manager della sicurezza
- La divisione marketing
- I manutentori dello hardware e del software

● Le autorità nazionali

● Il dispositivo ATM e i suoi componenti

## Esempio: attori di un sistema bancomat (ATM)

Gli attori che sono interessati al funzionamento di un sistema sono vari. Non bisogna limitarsi soltanto ai più ovvi:

- Correntisti
- Altre banche
- Manager di filiali
- Bancari
- Amministratori di database
- Manager della sicurezza
- La divisione marketing
- I manutentori dello hardware e del software
- Le autorità nazionali
- Il dispositivo ATM e i suoi componenti
- ...

## Esempio: attori di un sistema bancomat (ATM)

Gli attori che sono interessati al funzionamento di un sistema sono vari. Non bisogna limitarsi soltanto ai più ovvi:

- Correntisti
- Altre banche
- Manager di filiali
- Bancari
- Amministratori di database
- Manager della sicurezza
- La divisione marketing
- I manutentori dello hardware e del software
- Le autorità nazionali
- Il dispositivo ATM e i suoi componenti

● ...

## Esempio: attori di un sistema bancomat (ATM)

Gli attori che sono interessati al funzionamento di un sistema sono vari. Non bisogna limitarsi soltanto ai più ovvi:

- Correntisti
- Altre banche
- Manager di filiali
- Bancari
- Amministratori di database
- Manager della sicurezza
- La divisione marketing
- I manutentori dello hardware e del software
- Le autorità nazionali
- Il dispositivo ATM e i suoi componenti
- ...

# Scoperta dei requisiti: tecniche

- **Interviste:** maggiori dettagli nel seguito
- **Workshops:** maggiori dettagli nel seguito
- **Focus groups:** al fine di chiarire specifici aspetti del sistema un gruppo di persone vengono identificate e incaricate di studiare l'argomento. Al termine un documento potrà essere prodotto e utilizzato per altre attività
- **Osservazione o etnografia:** al fine di chiarire le reali necessità e possibili usi del sistema si procede all'osservazione dei potenziali utenti del sistema nello svolgimento regolare delle loro mansioni
- **Questionari:** vengono definite delle liste di domande e distribuite a specifici stakeholder al fine di chiarire specifici aspetti
- **Perspective-based reading:** si identificano specifici documenti considerati di rilievo per lo sviluppo del sistema e si procede alla loro lettura approfondita



# Tecniche accessorie

- **Brainstorming**: generalmente associato all'esecuzione di WKS serve a far emergere idee innovative e visionarie
- **KJ Method**: forma di brainstorming particolarmente utile in contesti di partecipanti eterogenei
- **Prototyping**
- **Mind mapping**: costruzione di mappe concettuali nel contesto del sistema
- **Elicitation Checklist**: lista di tipiche caratteristiche/aspetti/funzionalità/tipologie/. . . in un particolare contesto applicativo

# Aspetti da definire per le varie tecniche

La chiara descrizione delle tecniche richiede di fornire dettagli in relazione ai seguenti aspetti:

- Preparazione
- Esecuzione
- Follow-Up
- Benefici
- Complessità
- Fattori di successo

# L'intervista

Meeting nel quale si ha interazione con i vari attori. Obiettivo è mettere l'attore in una condizione di **massimo agio** in modo che possa esprimersi nel modo che più sente naturale rispetto ai requisiti del sistema, senza remore.

- **interviste "standardised"**: l'analista prepara domande a cui l'intervistato potrà rispondere fornendo il suo punto di vista.
- **interviste esplorative**: in questo caso l'analista intende meglio comprendere specifici aspetti del sistema. Partendo da domande congegnate dall'analista il discorso può spaziare al fine di chiarire gli aspetti oggetto dell'intervista
- **interviste non strutturate** (Non strutturate): si procede ad una discussione che è principalmente guidata dall'intervistato

Interviste sono un buon strumento per raggiungere una **comprensione generale** su cosa il sistema debba fare, ma forniscono scarsa comprensione del dominio applicativo e dettagli specifici. Si possono usare domande aperte e chiuse a seconda delle situazioni.

# L'intervista

Il risultato dell'intervista chiaramente dipende dall'intervistatore:

- ottime capacità di relazione
- ascoltare
- no preconcetti

# L'intervista - preparazione

- **Definire l'obiettivo dell'intervista:** ad esempio chiarire le necessità di specifici attori, o il comportamento del sistema in relazione a specifiche richieste
- **Selezione ed invito dei partecipanti:** è bene informarli sull'obiettivo dell'intervista
- **Selezione del luogo dell'intervista:** luogo di lavoro dell'intervistato rende accessibili facilmente delle risorse utili ma può comportare molte distrazioni e perdita di focus
- **Definizione delle domande**

In generale è utile aver informazioni sull'intervistato.

# L'intervista - esecuzione

- **Apertura:** è bene introdurre obiettivi e motivazioni
- **Conduzione:** è comodo aiutarsi con modelli, è importante dare feedback, attenzione alla comunicazione non verbale, fare pause, e comunque cercare sempre di riportare il fuoco sull'obiettivo dell'intervista
- **Chiusura:** fare un breve sommario finale di quanto scoperto e dei punti salienti, e ovviamente ringraziare.

# L'intervista - follow-up

A seguito dell'intervista:

- **Rielaborazione**: il materiale va riorganizzato e vanno definiti nel dettaglio i requisiti, gli scenari e i modelli che definiscono il sistema.
- **Identificazione di gaps**: aspetti del sistema che risultano essere ancora poco chiari e esplorati
- **Comunicazione agli intervistati** al fine di confermare i risultati

# L'intervista - benefici e sforzo richiesto

Le interviste sono efficaci per ottenere le **necessità principali** dei committenti. **Non sono lo strumento principale per la definizione di requisiti nuovi e innovativi.** D'altro canto lo **sforzo può essere tra il medio e l'alto** in dipendenza del numero di diversi attori e dalle metodiche di intervista adottate.



# L'intervista - fattori critici di successo

Tali fattori si riferiscono principalmente alle caratteristiche dell'intervistatore che deve essere estremamente aperto e cercare di cogliere al meglio gli aspetti socio-antropologici dell'attività e dunque mettere in atto tutte le strategie per far emergere le necessità principali e innovative.

# Workshops

In un WKS un gruppo di stakeholder sviluppano i requisiti per il sistema. In questo caso si tratta di un lavoro di gruppo che può portare a risultati eccellenti.

Utilizzo delle tecniche accessorie è certamente necessario, in particolare

- Brainstorming
- KJ Method
- definizione iterativa di scenari
- ...

# Workshop - preparazione

- Definizione degli obiettivi
- Definire tecniche da applicare e risultati attesi: Brainstorming, KJ Method, Discussion, definizione iterativa di scenari, organizzazione in sottogruppi, ...
- Scelta dei partecipanti, invito e accordo sugli obiettivi
- Scelta del luogo
- Identificazione di un moderatore
- Identificazione del “Minute-taker” (segretario per la redazione del verbale)

# Workshop - esecuzione

- **Apertura:** si descrivono gli obiettivi, le tecniche che saranno adottate, l'agenda, le regole di interazione
- **Conduzione:** il moderatore gestisce l'applicazione delle tecniche scelte e delle regole. Il "minute-taker" (segretario) prende nota di tutto quanto ritenuto rilevante
- **Chiusura:** vengono raccolti i risultati e illustrati sommariamente. Si definiscono anche possibili attività da dover svolgere successivamente al fine di chiarire le necessità

# Workshop - follow-up

Le minute vengono riorganizzate dal **segretario** e **fatte circolare** tra i **partecipanti** che possono **richiedere modifiche**. Le possibili problematiche evidenziate al termine dell'esecuzione vengono gestite così come eventuali gap.

# Workshop - benefici e sforzo richiesto

Tecnica molto efficace per poter identificare tutte le tipologie di requisiti e in particolare per identificare aspetti innovativi. D'altra parte lo sforzo richiesto è generalmente alto, se non molto alto, vista la partecipazione di molti stakeholder.

# Workshop - fattori critici di successo

Il successo del workshop dipende fortemente da diversi fattori quali in particolar modo:

- scelta dei giusti partecipanti
- capacità del moderatore di far rispettare le regole e applicare la tecnica scelta
- motivazione e interesse dei partecipanti
- scelta del luogo

# Tecnica accessoria - KJ Method

Permette di far emergere requisiti a partire da un gruppo di persone allo stesso tempo. Consta di due fasi principali:

- Riflessione individuale
- Lavoro di gruppo



# KJ Method - esecuzione

- **Apertura:** si presentano gli obiettivi dell'attività
- **Conduzione**
  - **Riflessione individuale e scrittura delle carte:** ogn partecipante singolarmente riflette e identifica caratteristiche ritenute rilevanti. Tali caratteristiche vengono riportate su di una carta (post-it)
  - **Presentazione e discussione delle carte:** Le carte vengono prese dal moderatore e lette ad alta voce. I partecipanti possono porre domande e richiedere chiarimenti
  - **Raggruppamento e sintesi:** le carte ritenute rilevanti vengono raggruppate considerando la possibilità che si riferiscano a tematiche omogenee del sistema
- **Chiusura:** il gruppo definisce come procedere con ulteriori attività per la rielaborazione dei risultati

## KJ method - follow up, sforzo e fattori critici

- Definizione del verbale e sua condivisione e accettazione
- generalmente richiede uno sforzo limitato
- Il successo dipende dalla partecipazione di un numero non troppo elevato di stakeholder, dalla selezione dei partecipanti e dalla chiara definizione degli obiettivi

Particolarmente efficace quanto i **partecipanti sono eterogenei** e non si conoscono tra di loro. In tal caso potrebbero avere **difficoltà ad esprimersi apertamente durante una discussione aperta**

# Formato definizione requisiti

Informazioni minime da riportare in un formato strutturato per la specificazione di requisiti:

**ID:** “Identificativo unico - scegliete formato utile agli scopi”

**Nome:** “Nome Mnemonico tipicamente azione nome”

Il “Sistema o parte di esso” Deve/Dovrebbe/Può/Potrebbe “descrizione funzionalità”

**Descrizione:** fornisce ulteriori indicazioni che servono a migliorare la comprensione del requisito

**Sorgente:** chi o cosa ha originato il requisito?

# Documentazione dei Requisiti

## raccomandazioni

- Definite un **formato standard** per la definizione dei requisiti
- Utilizzate linguaggio consistentemente - **attenzione alle parole** “deve”, “dovrebbe”
- **The MoSCoW principle**
- Utilizzate meccanismi di **evidenziazione del testo**
- Per requisiti utente: non usare, per quanto possibile, **gergo tecnico del dominio informatico**.

# Formato VOLERE<sup>©</sup> Atlantic Systems Guild

<b>ID</b>	<i>Identificativo unico</i>
<b>Tipo</b>	<i>Tipologia del requisito</i>
<b>Evento/CU correlato</b>	<i>A quale CU o evento si riferisce</i>
<b>Descrizione</b>	<i>Una frase che caratterizzi il sistema</i>
<b>Motivazione</b>	<i>Breve descrizione a contestualizzare il requisito</i>
<b>Sorgente</b>	<i>Chi ha richiesto inserimento requisito</i>
<b>Criterio di Valutazione</b>	<i>Come valutare il soddisfacimento del requisito</i>
<b>Soddisfazione Cliente</b>	<i>Grado di soddisf. se requisito sarà implementato</i>
<b>Insoddisfazione Cliente</b>	<i>Grado di insoddisf. se requisito non sarà implementato</i>
<b>Conflitti</b>	<i>conflitti con altri requisiti</i>
<b>Priorità</b>	<i>importanza per il cliente</i>
<b>Materiale di supporto</b>	<i>documentazione che può migliorare comprensione</i>
<b>Storia</b>	<i>Creazione, modifiche</i>

# Scoperta dei requisiti

## Descrizione di scenari

Attori trovano più semplice dire **come intendono utilizzare il sistema o come credono debba essere utilizzato**. È più semplice criticare l'uso del sistema che un singolo requisito.

Elicitazione di requisiti tramite **descrizione di scenari d'uso**

Nella forma più generale uno scenario comprende:

- Cosa ci si aspetta quando lo scenario parte
- La descrizione del flusso normale dello scenario
- Descrizione di cosa può andar male nell'esecuzione del flusso normale
- Informazione su attività che potrebbero svolgersi in parallelo
- Una descrizione dello stato del sistema alla fine

# Esempio di scenario

il sistema elettronico di biblioteca

**Assunzioni iniziali:** L'utente si è autenticato ed ha localizzato il link al documento che vuole scaricare

**Flusso Normale:** l'utente seleziona il documento. Il sistema richiede di fornire dettagli di pagamento. Pagamento può essere fatto con CC o con numero di conto da addebitare.

Viene richiesto all'utente di riempire un form di copyright che viene sottoposto al sistema.

Se transazione approvata il PDF del documento viene reso disponibile e l'utente viene informato. Nel caso di documento "print-only" si chiede di scegliere una stampante.

# Esempio di scenario

il sistema elettronico di biblioteca

**Cosa può andar storto:** Copyright form riempito scorrettamente. Si informa il cliente e si chiede di riempire nuovamente il form. Nel caso di errore la transazione viene rifiutata.

Il pagamento non va a buon fine. La transazione viene rifiutata. La stampa può fallire. Nel caso di documento “print-only” la transazione viene abortita ed il cliente viene riaccreditato dell'ammontare corrispondente.

**Attività in parallelo:** molti utenti possono essere connessi al sistema e potrebbero richiedere il download. Lo stesso utente potrebbe tener aperte più sessioni.

**Stato del sistema alla fine:** L'utente viene riportato ad una pagina di benvenuto, l'articolo è stato stampato e nel caso di “print-only” è stato eliminato da eventuali aree disco temporanee.



# Limitazioni dello strumento degli Scenari

Molto efficaci per raccogliere requisiti da punti di vista diretti

Non adatto a rappresentare requisiti derivanti da punti di vista **indiretti** o di **dominio** e a definire requisiti **extra-funzionali** (caratteristiche globali)

# Specifica dei punti di interazione

Praticamente tutti i sistemi software si trovano ad interagire con altri sistemi software. Le interfacce di interazione devono essere definite formalmente:

- Application Programming Interface (API)
- Strutture dati
- Rappresentazione dei dati

# Documento dei requisiti software

Il documento dei requisiti software è **ciò che deve essere implementato dagli sviluppatori**. Contiene generalmente **sia requisiti utente che di sistema**.

Differenti utenti . . . differenti “requisiti” . . . differenti formati

Formato dipendente anche da **processo adottato!**

Metodologie agili in molti casi suggeriscono di non derivare un documento dei requisiti ma di annotarli su apposite cards che verranno poi priorizzate.

# IEEE/ANSI 830-1998

Suggerisce la seguente struttura:

- Introduction
  - ① Scopo del documento dei requisiti
  - ② Scopo del prodotto
  - ③ Definizione, acronimi ed abbreviazioni
  - ④ Riferimenti
  - ⑤ Overview dell'intero documento
- Descrizione generale
  - ① Prospettive sul prodotto
  - ② Funzioni del prodotto
  - ③ Caratteristiche degli utenti
  - ④ vincoli generali
  - ⑤ Assunzioni e dipendenze
- Requisiti specifici
- Appendici
- Indici

# Contenuto generale

- 1 Prefazione
- 2 Introduzione
- 3 Glossary
- 4 Definizione dei requisiti utente
- 5 Architettura del sistema
- 6 Definizione dei requisiti di sistema
- 7 Modelli del sistema
- 8 Evoluzione del sistema
- 9 Appendici
- 10 Indici

# Negoziazione e prioritizzazione

- Sono attività che richiedono la gestione dei requisiti in particolare in caso di conflitti.
- necessità di prioritizzare per poter organizzare le attività e poter identificare gli aspetti più critici

# Sommario

- 1 Requisiti
  - Generalità
  - Requisiti Utente e di Sistema
  - Requisiti funzionali
  - Requisiti qualitativi
  - Vincoli
  - Requisiti di dominio
- 2 Attività dell'ingegneria dei requisiti
  - Studio di fattibilità
  - Elicitazione
    - Documentazione
    - Negoziazione e prioritizzazione
  - **Validazione**
  - Gestione

# Validazione dei requisiti

La fase di validazione dei requisiti cerca di rimuovere possibili problemi nella specifica dei requisiti. Possibili verifiche sono:

- **Controllo di validità:** verificare che ciò che è stato specificato coincide effettivamente con quanto necessario all'utente
- **Controllo di consistenza:** i requisiti non devono essere contraddittori
- **Controllo di completezza:** i requisiti dovrebbero specificare tutte le possibili funzionalità
- **Controllo di concretezza:** verificare che il requisito richieda qualcosa che effettivamente possa essere implementato date anche le tecnologie adottate, i costi e le scadenze imposte
- **Verificabilità:** requisiti devono essere scritti in modo da poter verificare la loro soddisfazione.



# Validazione dei requisiti

## tecniche

Tecniche che si sono rivelate utili nella validazione dei requisiti sono:

- **Revisione dei requisiti:** processo **manuale** che coinvolge team misti cliente/contractor. Può essere **formale** o **informale**. Verifiche che potrebbero essere fatte includono:
  - Verificabilità
  - Comprensibilità
  - Tracciabilità
  - Adattabilità
- **Prototipizzazione**
- **Generazione di casi di test**

# Sommario

- 1 Requisiti
  - Generalità
  - Requisiti Utente e di Sistema
  - Requisiti funzionali
  - Requisiti qualitativi
  - Vincoli
  - Requisiti di dominio
- 2 Attività dell'ingegneria dei requisiti
  - Studio di fattibilità
  - Elicitazione
    - Documentazione
    - Negoziazione e prioritizzazione
  - Validazione
  - **Gestione**

# Gestione dei requisiti

Requisiti sono costantemente sottoposti a **spinte di cambiamento**. Il requisito una volta definito non è fissato per sempre, anche considerando non solo le fasi di post-rilascio. Molte motivazioni per questo:

- Comunità estesa di utenti con richieste differenti ed anche conflittuali
- Acquirenti ed utenti diretti spesso non sono la stessa entità.
- L'ambiente di esecuzione cambia velocemente.

L'attività di gestione dei requisiti si occupa di far **emergere, permettere e gestire** modifiche ai requisiti

# Gestione dei requisiti

...di cosa c'è bisogno?

- Identificazione dei requisiti tramite ad esempio definizione di ID
- Definire un processo di modifica dei requisiti: tutte le modifiche sono trattate egualmente e consistentemente
- Definire meccanismi di tracciabilità
- uso e supporto da parte di CASE tool/environment (database, fogli elettronici etc. possono essere sufficienti)

Requisiti stabili vs. requisiti volatili

# Tracciabilità dei requisiti

Differenti tipi di tracciabilità tipicamente si immagazzinano le informazione in una matrice:

- **Sorgente**: Attore x Requisito
- **Relazioni con altri requisiti**: Requisito x Requisito
- **Design**: Sottosistema x Requisito

Matrici possono diventare particolarmente estese e poco gestibili. Uso di CASE (database) di supporto alle varie fasi:

- Immagazzinamento
- Gestione delle modifiche
- Gestione della tracciabilità

# Attività di gestione delle modifiche

L'attività di gestione della modifica dei requisiti sarà strutturata su sotto-attività quali:

- Analisi del problema e specifica della modifica
- analisi del cambiamento e valutazione del costo
- implementazione della modifica