

# INGEGNERIA DEL SOFTWARE - 2025-26

---

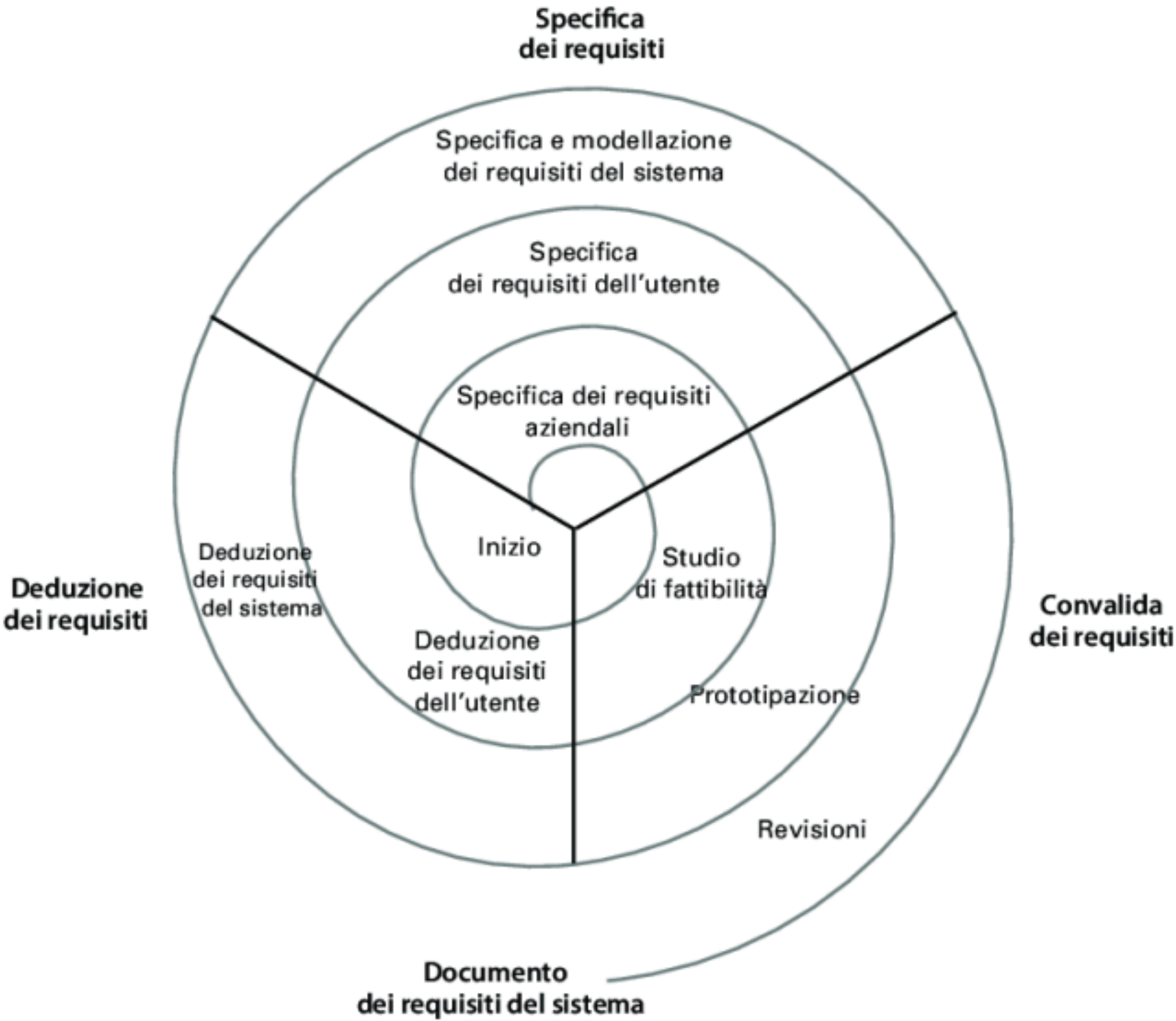
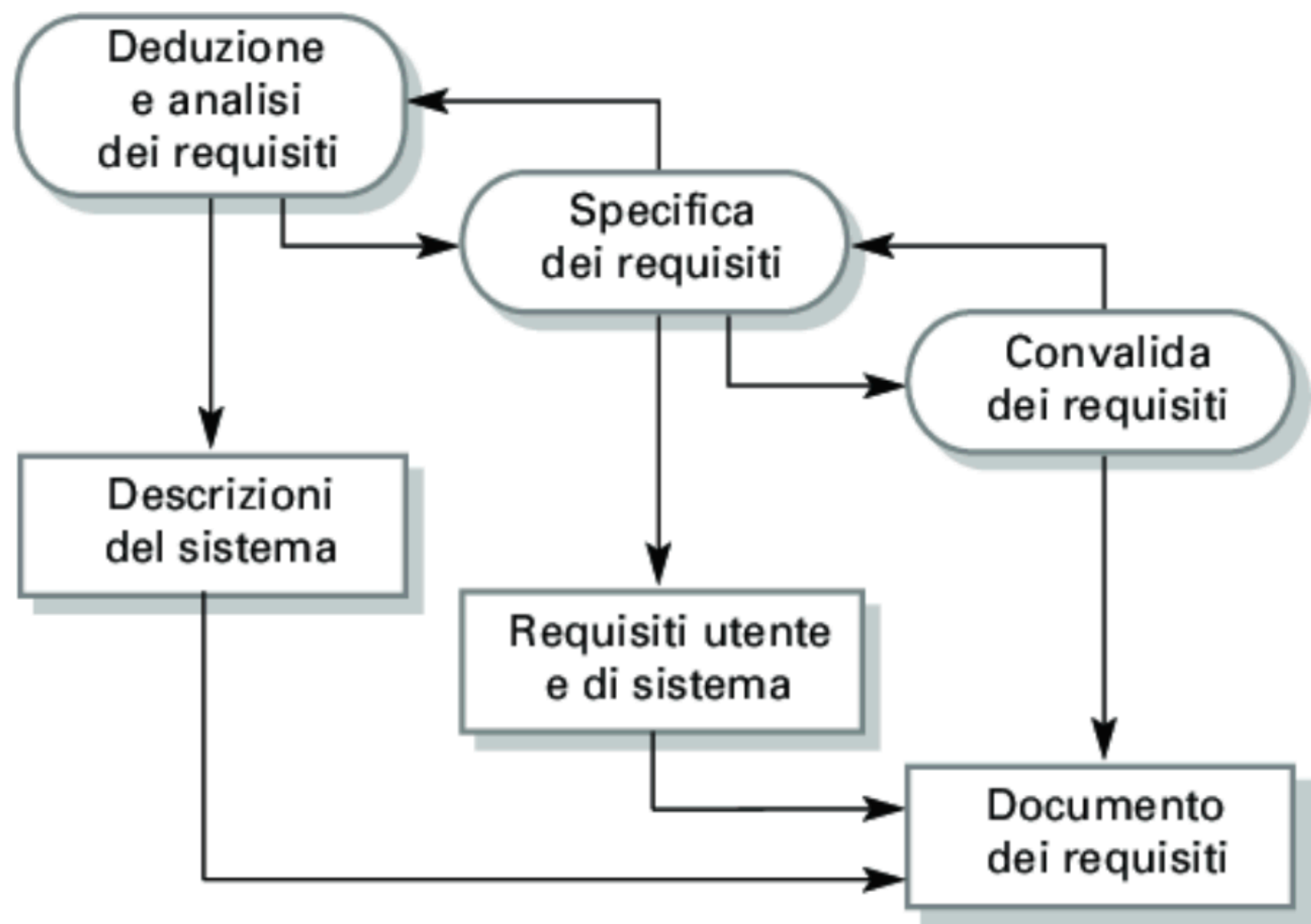
## ANALISI, SPECIFICA E CONVALIDA DEI REQUISITI

LEZIONE 8  
22/10/2025  
VINCENZO RICCIO

# RIFERIMENTI

- Sommerville - Paragrafi 4.3, 4.4, 4.5







# DEDUZIONE E ANALISI DEI REQUISITI



# STAKEHOLDERS

- Uno **stakeholder** è qualsiasi individuo, gruppo o organizzazione che ha un legittimo interesse nel progetto di sviluppo software
- Gli stakeholder possono includere i clienti, gli utenti finali, i membri del team di sviluppo, i finanziatori, i responsabili delle decisioni aziendali e altre parti che potrebbero essere impattate dal successo o dal fallimento del progetto





# DEDUZIONE E ANALISI DEI REQUISITI

- Capire l'ambiente operativo e come gli stakeholder potrebbero utilizzare il software
- Gli ingegneri dei requisiti devono lavorare con gli stakeholder per scoprire informazioni sul dominio applicativo, sulle funzionalità che il sistema dovrebbe fornire loro, sulle prestazioni richieste ed altri vincoli operativi





- Di solito né il fornitore né il committente sono in grado, da soli, di estrarre in maniera efficace i requisiti di un sistema
  - Il committente non ha la necessaria conoscenza dei processi software per definire in maniera efficace i requisiti, può non avere un'idea chiara sui requisiti, può usare termini propri del dominio di appartenenza
  - Differenti stakeholder potrebbero avere requisiti contrastanti
  - Il fornitore non ha una conoscenza perfetta del dominio applicativo, e non può esprimere le effettive necessità

# DEDUZIONE E ANALISI DEI REQUISITI: PROCESSO



- Processo iterativo che termina quando il documento dei requisiti è "completo"
- La comprensione dei requisiti da parte dell'ingegnere del software migliora ad ogni iterazione



# DEDUZIONE E ANALISI DEI REQUISITI: PROCESSO



- **Scoperta e comprensione:** gli analisti interagiscono con gli stakeholder per scoprire i loro requisiti

# DEDUZIONE E ANALISI DEI REQUISITI: PROCESSO



- **Classificazione e organizzazione:** i requisiti scoperti sono una raccolta non strutturata
- I requisiti tra loro correlati vanno raggruppati in gruppi coerenti (eliminando duplicati)

# DEDUZIONE E ANALISI DEI REQUISITI: PROCESSO



- Quando sono coinvolti più stakeholder, i requisiti possono essere in conflitto
- **Negoziazione e priorità:** dare una priorità ai requisiti, trovare e risolverne i conflitti attraverso la negoziazione;



# DEDUZIONE E ANALISI DEI REQUISITI: PROCESSO



- **Documentazione:** requisiti vengono documentati e diventano l'input della successiva iterazione
- Diversi livelli di documentazione a seconda del processo: bozze di documenti dei requisiti software, oppure informalmente su lavagne, wiki o altri spazi condivisi

# TECNICHE PER ESTRARRE I REQUISITI



- **Interviste:** sia formali che informali, a risposta chiusa o aperta
- **Etnografia:** osservare e analizzare le persone nell'ambiente operativo
- **Storie Utente e Scenari:** testi narrativi che descrivono scenari pratici di utilizzo del software

# TECNICHE PER ESTRARRE I REQUISITI: INTERVISTE



- I team di ingegneria fanno domande (chiuse e/o aperte) agli stakeholder sul sistema che usano e sul sistema che deve essere sviluppato; dalle loro risposte ottengono i requisiti
- Comprensione delle necessità degli stakeholder (ad es., come interagirebbero con il sistema e le difficoltà che affrontano con i sistemi già esistenti)





# SUGGERIMENTI PER INTERVISTE



- Un prototipo può aiutare ad avere requisiti specifici e dettagliati
- Gli specialisti del dominio potrebbero usare terminologia specifica o omettere dettagli che considerano ovvi
- Alcuni dettagli organizzativi o politici potrebbero essere non rivelati a degli estranei
- L'intervistatore deve essere *open-minded*, evitando di avere preconcetti durante l'intervista
- Evitare domande aperte troppo generiche





# TECNICHE PER ESTRARRE I REQUISITI: ETNOGRAFIA



- Un analista si immerge nell'ambiente di lavoro in cui il sistema sarà usato; osserva il lavoro quotidiano, prendendo nota dei compiti in cui i partecipanti sono coinvolti
- Scopre requisiti impliciti che riflettono processi reali (invece delle descrizioni di processo aziendale)
- Poiché si focalizza su utenti finali, andrebbe arricchita con altri metodi per scoprire requisiti ad alto livello





# MOTIVAZIONI ETNOGRAFIA



- I sistemi software non sono mai isolati; vengono utilizzati in un contesto sociale e organizzativo che influenza l'utilizzo pratico del sistema
- Alcuni requisiti impliciti possono essere molto differenti dai processi formali definiti dall'organizzazione e comunicati nelle interviste ufficiali
- Per alcune persone può essere difficile descrivere i dettagli del proprio lavoro quotidiano in relazione al lavoro degli altri





# TECNICHE PER ESTRARRE I REQUISITI: STORIE E SCENARI



- Descrizioni di come il sistema può essere utilizzato per svolgere particolari compiti
- Descrivono che cosa fanno le persone, quali informazioni utilizzano e producono, e quali sistemi possono utilizzare in questo processo
- **Storie:** Testi narrativi che presentano una descrizione di alto livello del modo in cui il sistema è utilizzato
- **Scenari:** informazioni specifiche e spesso strutturate raccolte come input, output e flusso di eventi durante un'interazione con il sistema (possono dettagliare parti delle storie)



# TECNICHE PER ESTRARRE I REQUISITI: STORIE E SCENARI

- Più persone possono facilmente mettersi in relazione con storie e scenari, permettendo la raccolta di informazioni da un pubblico più vasto
- Le persone in genere trovano più semplice riferirsi a esempi di vita reale, piuttosto che a descrizioni generali e astratte
- Più semplice spiegare come si gestirebbe un contesto di lavoro che descrivere un requisito





## Condivisione di fotografie nella classe

Jack è un maestro elementare a Ullapool (un villaggio nel nord della Scozia). Ha deciso che un progetto per la classe potrebbe riguardare l'industria della pesca in quell'area, descrivendone la storia, lo sviluppo e l'impatto economico. Come parte di questo progetto, agli alunni viene chiesto di acquisire ricordi dai parenti, usare archivi di giornali e raccogliere vecchie fotografie relative alla pesca e alle comunità di pescatori in quell'area. Gli alunni usano un wiki di iLearn per raccogliere le storie sulla pesca e SCRAN (un sito di risorse storiche) per accedere agli archivi di giornali e fotografie. Jack ha anche bisogno di un sito di condivisione per le foto, perché vuole che ogni alunno possa guardare e commentare le foto degli altri alunni e possa caricare le scansioni di vecchie foto che potrebbe trovare a casa sua.

Jack invia una e-mail a un gruppo di maestri, di cui è membro, per vedere se qualcuno possa consigliare un sistema appropriato. Due maestri rispondono, ed entrambi suggeriscono che usano KidsTakePics, un sito di condivisione di foto che permette ai maestri di controllarne il contenuto. Poiché KidsTakePics non è integrato con il servizio di autenticazione di iLearn, Jack prepara un account per un maestro e la classe. Usa il servizio di configurazione di iLearn per aggiungere KidsTakePics ai servizi visti dagli alunni nella sua classe, in modo che quando si collegano (login), possono immediatamente utilizzare il sistema per caricare foto dalle loro unità mobili o dai computer della classe.





## Caricare le fotografie su KidsTakePics

**Ipotesi iniziale:** un utente o un gruppo di utenti hanno una o più foto digitali da caricare nel sito di condivisione delle foto.

Queste foto sono salvate su un tablet o su computer. Gli utenti si sono collegati con successo al sistema KidsTakePics.

**Normale:** l'utente sceglie di caricare le foto; il sistema gli chiede di selezionare le foto da caricare e il nome del progetto sotto il quale le foto dovranno essere memorizzate. L'utente dovrà avere anche la possibilità di inserire le parole chiave da associare a ogni foto caricata. A ogni foto caricata viene assegnato il nome formato dal nome dell'utente seguito dal nome del file della foto nel computer.

Dopo aver caricato le foto, il sistema invia automaticamente una e-mail al moderatore del progetto, per chiedergli di controllare il nuovo contenuto; poi visualizza un messaggio sullo schermo dell'utente per avvisarlo che il controllo è stato fatto.

**Che cosa può andare male:** nessun moderatore è associato al progetto selezionato. Il sistema genera automaticamente una e-mail per il preside della scuola, per chiedergli di nominare un moderatore di quel progetto. Gli utenti devono essere informati su un possibile ritardo nella visualizzazione delle loro foto.

Le foto con lo stesso nome sono state già caricate dallo stesso utente. Il sistema dovrà chiedere all'utente se desidera ricaricare le foto con lo stesso nome, rinominare le foto o annullare l'operazione. Se l'utente sceglie di ricaricare le foto, quelle esistenti vengono sovrascritte; se sceglie di rinominare le foto, viene generato automaticamente un nuovo nome ad ogni foto aggiungendo un numero al nome del file esistente.

**Altre attività:** il moderatore può essere collegato al sistema e può approvare le foto quando vengono caricate.

**Stato del sistema al completamento:** un utente è collegato. Le foto caricate ricevono lo stato "in attesa del moderatore". Le foto sono visibili al moderatore e all'utente che le ha caricate.



# SPECIFICA DEI REQUISITI



- Processo di descrizione dei requisiti utente e di sistema in un documento
- I requisiti **utente** sono quasi sempre scritti in linguaggio naturale. Devono essere comprensibili da utenti e clienti senza background tecnico. Il documento dei requisiti può includere tabelle e diagrammi appropriati
- I requisiti **di sistema** possono essere scritti nel linguaggio naturale, ma possono essere utilizzate altre notazioni basate su moduli, grafici o modelli matematici. Sono più dettagliati e possono contenere informazioni tecniche



# SPECIFICA DEI REQUISITI UTENTE



- I requisiti utente dovrebbero essere comprensibili dagli utenti del sistema che non hanno conoscenze tecniche speciali
- Teoricamente, questi requisiti dovrebbero specificare soltanto il comportamento del sistema visto dall'esterno e i suoi vincoli
- La specifica dei requisiti utente non dovrebbe includere dettagli sull'architettura o sulla progettazione del sistema
- Solitamente scritti in linguaggio naturale, con semplici tabelle, moduli e diagrammi intuitivi
- Non contengono gergo tecnico, notazioni strutturate o formali



# SPECIFICA DEI REQUISITI DI SISTEMA



- I requisiti di sistema sono versioni "espansive" dei requisiti utente: aggiungono dettagli sul comportamento esterno del sistema e i suoi vincoli
- Utilizzati dagli ingegneri del software come base di partenza per la progettazione del sistema
- Possono essere usati come parte del contratto fra cliente e sviluppatore e quindi dovrebbero essere una specifica completa e dettagliata dell'intero sistema, contenendo potenzialmente anche alcuni riferimenti all'architettura del sistema



# SPECIFICA DEI REQUISITI VS PROGETTO



- In teoria:
  - ◎ la specifica dei requisiti non dovrebbe contenere informazioni sulla progettazione o l'implementazione del sistema *[cosa e non come]*
  - ◎ Il progetto deve descrivere come i requisiti sono realizzati
- Nella pratica, requisiti e progetto sono inseparabili. Non è possibile né auspicabile escludere tutte le informazioni sulla progettazione
- Es 1: l'utilizzo di una specifica architettura può essere necessario per soddisfare requisiti non funzionali
- Es 2: è possibile strutturare i requisiti per ogni sotto-sistema per renderli più comprensibili

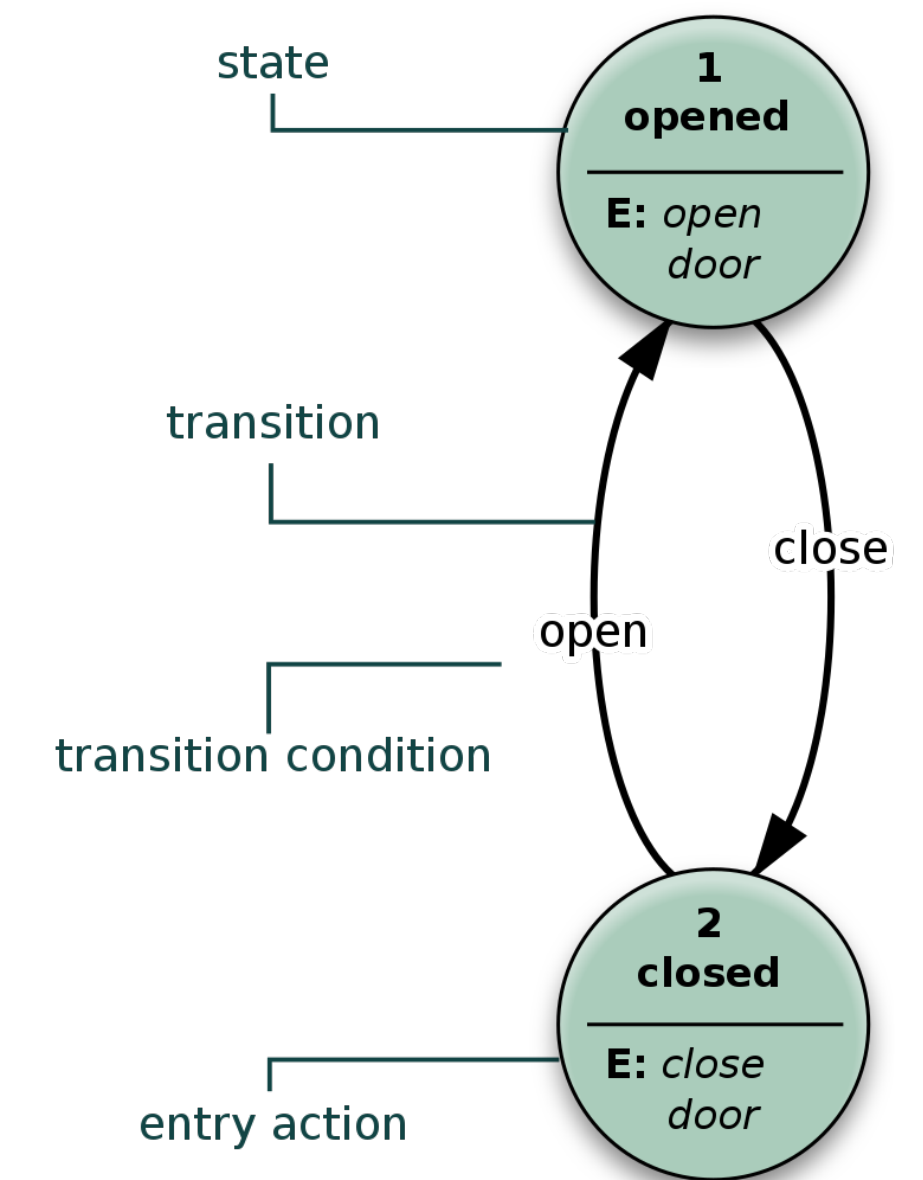
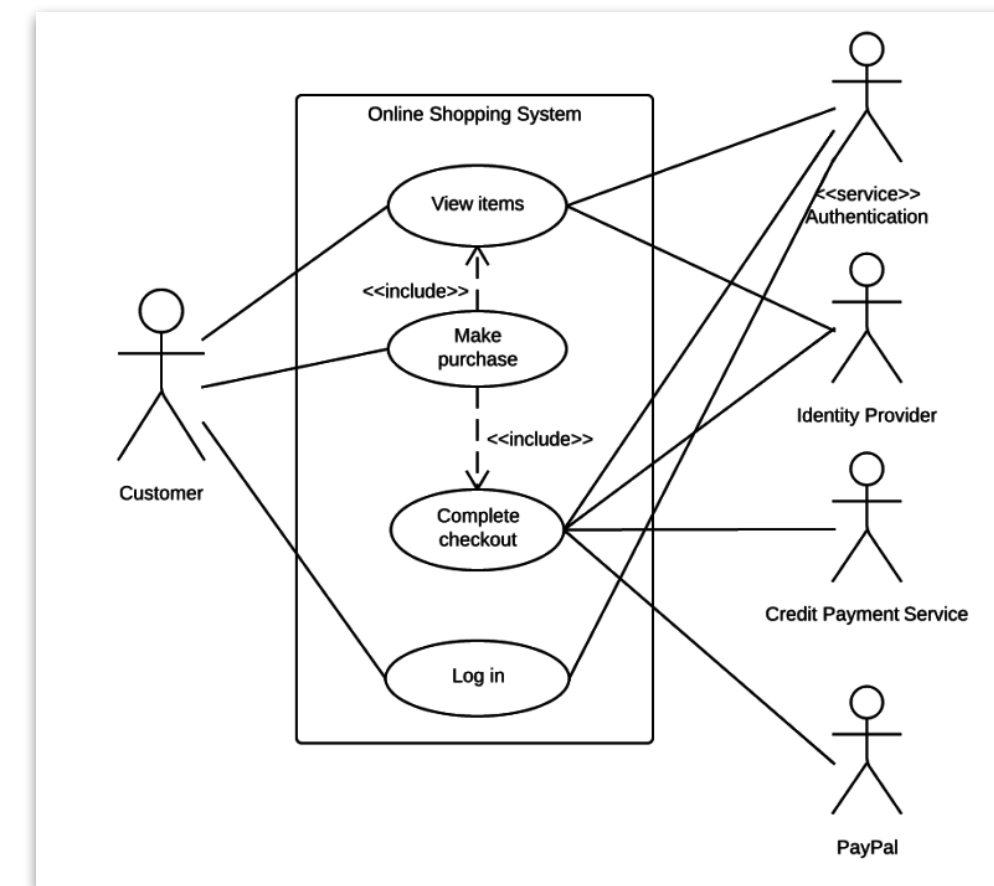


# LINGUAGGI PER LA SPECIFICA



- I requisiti sono spesso espressi in linguaggio naturale (NL)
- Alcune alternative al NL:
  - Linguaggio naturale strutturato o semi-strutturato
  - Modelli grafici (vedi use case e sequence diagrams)
  - Specifiche formali (vedi macchine a stati finiti)

ID	Requirement description
C1	For a department d and a period p, in which courses most of students might obtain a (low, regular, or high) grade?
C2	For a professor o and a period p, which are o's (weak or outstanding) issues according most students' opinion?
C3	For a department d and a period p, how (easy, regular, or difficult) are the courses, in terms of difficulty level, available resources, correspondence to number of credits, and grade expected for course c?





- I requisiti possono essere scritti in linguaggio naturale, integrato con diagrammi e tabelle
- **Pro:**
  - Espressivo, intuitivo e universale: può essere compreso anche da utenti e clienti
- **Contro:**
  - Mancanza di chiarezza: difficile usare il linguaggio in modo conciso e, allo stesso tempo, preciso e non ambiguo
  - Confusione: difficile distinguere le varie tipologie di requisiti (ad es. funzionali e non funzionali). Inoltre, diversi requisiti potrebbero essere espressi in una singola frase



- Formato standard coerente e conciso, riduce il rischio di omissioni e semplifica il controllo dei requisiti
- Utilizzo coerente del linguaggio: "deve" per requisiti obbligatori, "dovrebbe" se desiderabili ma non obbligatori
- Formattazione coerente del testo (grassetto, corsivo o i colori) per evidenziare i punti chiave di un requisito
- Evitare l'utilizzo del linguaggio tecnico
- Spiegare perché un requisito è necessario e chi lo ha proposto (*logica del requisito*), in modo da sapere chi consultare se il requisito dovesse essere modificato





Req 3.2: Il sistema deve misurare il livello degli zuccheri nel sangue e rilasciare l'insulina, ogni 10 minuti (*variazioni degli zuccheri nel sangue sono relativamente lente, quindi non sono necessarie misure più frequenti; misure meno frequenti potrebbero portare a livelli di zuccheri inutilmente elevati*).

- Conciso
- Usa deve
- Motiva il requisito
- Usa un formato standard evidenziando le diverse parti del requisito



- Utilizzo di linguaggio naturale con una struttura predefinita standard per tutti i requisiti, che garantisce maggiore uniformità
- Ciascun elemento della struttura fornisce informazioni su un aspetto del requisito
- Limita la libertà di chi scrive i requisiti, permettendo di scriverli in maniera guidata
- Si possono usare costrutti del linguaggio di programmazione (IF, FOR)
- Si può usare formattazione (grassetto, corsivo o i colori) per evidenziare i punti chiave di un requisito

# SPECIFICHE STRUTTURATE IN TEMPLATE



Nome Requisito
Descrizione funzionalità
Motivazione
Specifica dettagliata
Sorgente



# SPECIFICHE STRUTTURATE IN TEMPLATE – ESEMPIO



Nome Requisito	Funzionalità della griglia
Descrizione funzionalità	L'editor grafico deve fornire una funzione griglia ove una matrice di linee orizzontali e verticali fornisce uno sfondo all'editor
Motivazione	Aiutare l'utente a costruire diagrammi precisi
Specifica dettagliata	Contenuta in Tools/FS/Sezione 5.6
Sorgente	Mario Rossi, Ufficio di Sappada

# TEMPLATE DI ROBERTSON E ROBERTSON



## Pompa di insulina/ Software di controllo/SRS/ 3.3.2

<b>Funzione</b>	Calcolo della dose di insulina: livello sicuro degli zuccheri.
<b>Descrizione</b>	Calcola la dose di insulina che deve essere fornita quando il livello corrente degli zuccheri è nella zona sicura tra 3 e 7 unità.
<b>Input</b>	La misura corrente degli zuccheri (r2) e le due misure precedenti (r0 e r1).
<b>Sorgente</b>	La misura corrente degli zuccheri fatta dal sensore. Altre misure memorizzate.
<b>Output</b>	CompDose: la dose di insulina che deve essere fornita.
<b>Destinazione</b>	Ciclo di controllo principale.
<b>Azione</b>	CompDose è zero se il livello degli zuccheri è stabile o sta scendendo, o se il livello sta crescendo ma il tasso di crescita è in diminuzione. Se il livello sta crescendo e anche il tasso di crescita sta aumentando, allora CompDose è calcolato dividendo per 4 la differenza tra il livello corrente degli zuccheri e il precedente livello, e arrotondando il risultato. Se il risultato è arrotondato a zero, allora CompDose viene impostato alla minima dose che può essere fornita. (Si veda la Figura 4.14.)
<b>Richiede</b>	Due precedenti misure in modo da poter calcolare il tasso di variazione del livello degli zuccheri.
<b>Precondizione</b>	La riserva di insulina deve contenere almeno la quantità massima permessa per una singola dose.
<b>Postcondizione</b>	r0 viene sostituito da r1, poi r1 viene sostituito da r2.
<b>Effetti collaterali</b>	Nessuno.



- Quando bisogna specificare calcoli complessi, è difficile non introdurre ambiguità
- Possibile aggiungere informazioni supplementari al linguaggio naturale, per esempio tabelle o modelli grafici del sistema
- Utili per descrivere situazioni alternative e le azioni da intraprendere in ciascuna situazione

Condizione	Azione
Livello zuccheri in discesa ( $r2 < r1$ )	CompDose = 0
Livello zuccheri stabile ( $r2 = r1$ )	CompDose = 0
Livello zuccheri crescente e tasso di crescita in diminuzione ( $(r2-r1) < (r1-r0)$ )	CompDose = 0
Livello zuccheri crescente e tasso di crescita in aumento ( $(r2-r1) \geq (r1-r0)$ )	CompDose = round $((r2-r1)/4)$ If rounded result = 0 then CompDose = MinimumDose





## ▸ **Vantaggi:**

- Conserva l'espressività del linguaggio naturale
- Impone uniformità per descrivere le specifiche, riducendo la variabilità
- Organizza i requisiti in modo efficace

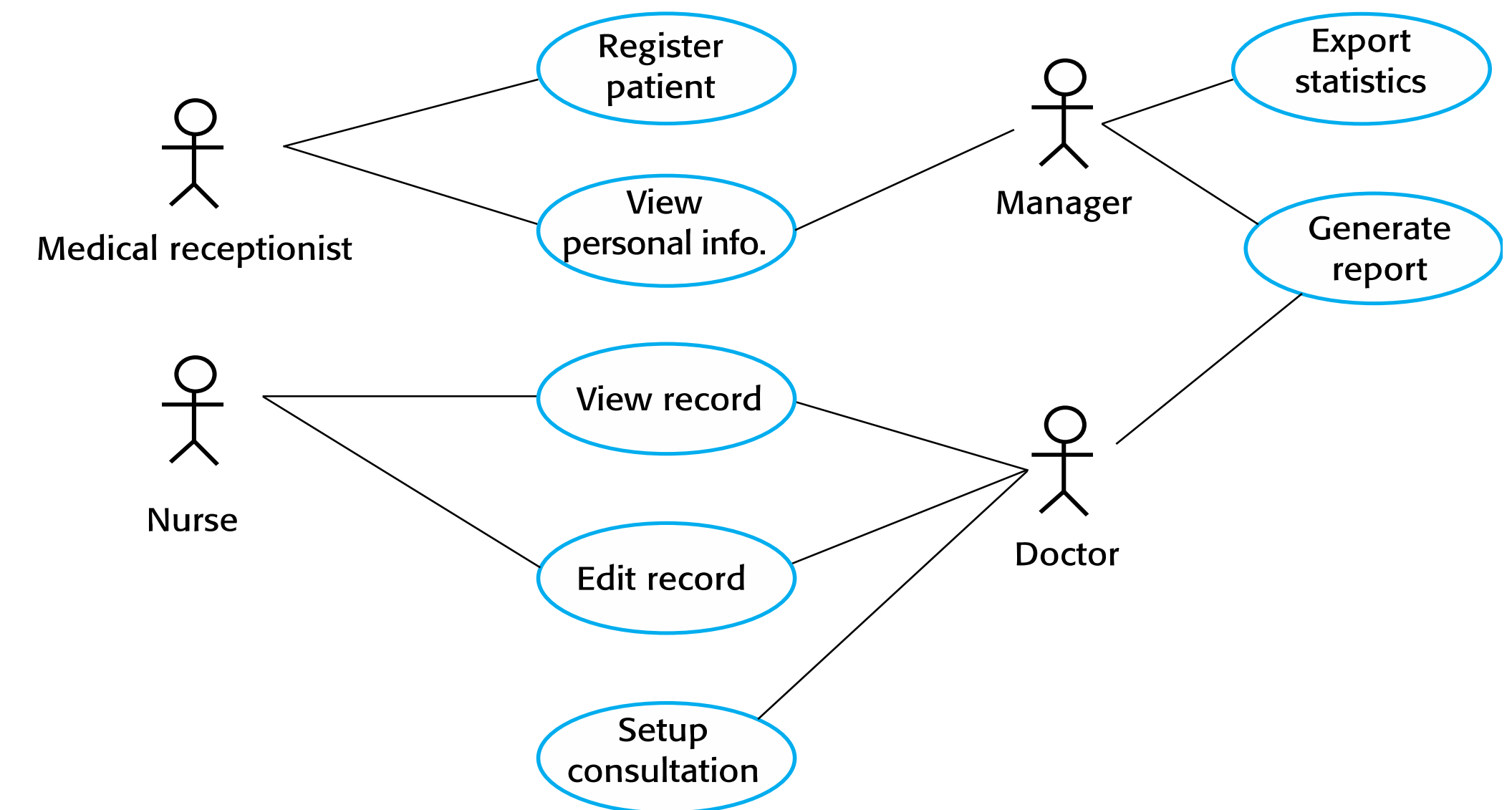
## ▸ **Svantaggi:**

- Troppo rigido per descrivere alcuni tipi di requisiti che richiedono descrizioni più libere ed espressive, ad es. requisiti di business
- Difficile scrivere i requisiti in modo non ambiguo quando sono molto complessi (ad es. calcoli)

# CASI D'USO (CENNI)



- Scenari definiti nel linguaggio standard UML
- Descrivono chi interagisce con il sistema (attori) e tutte le interazioni
- Sono arricchiti da informazioni dettagliate su ciascuna interazione
- I diagrammi di sequenza sono ulteriori diagrammi che possono descrivere in maniera più dettagliata le sequenze di eventi che occorrono durante l'utilizzo del sistema (alternativa a linguaggio naturale)



# IL DOCUMENTO DI SPECIFICA DEI REQUISITI DEL SOFTWARE (SRS)



- Definizione ufficiale dei requisiti del software
- Definisce ciò che gli sviluppatori dovrebbero implementare
- Deve includere sia i requisiti utente (RU) che i requisiti di sistema (RS)
- Diversi SRS potrebbero essere organizzati in modo diverso, ad es. potrebbero avere tutti i RU nell'introduzione oppure avere RU e RS integrati in un'unica descrizione
- Nota: si ricorda ancora che SRS non è un documento di progetto. Pertanto, dovrebbe (per quanto possibile) definire **cosa** il sistema dovrebbe fare, non il **come**





- Il documento di Specifica dei Requisiti Software (SRS) costituisce il punto di convergenza di tre diversi punti di vista: cliente, utente, sviluppatore
- Un SRS ha molteplici scopi:
  - Costituire la base per la successiva progettazione
  - Costituire la base per la validazione del prodotto finale
  - Costituire la base per il contratto
  - Ridurre i costi di sviluppo, evitando controversie o ritardi dovuti ad ambiguità

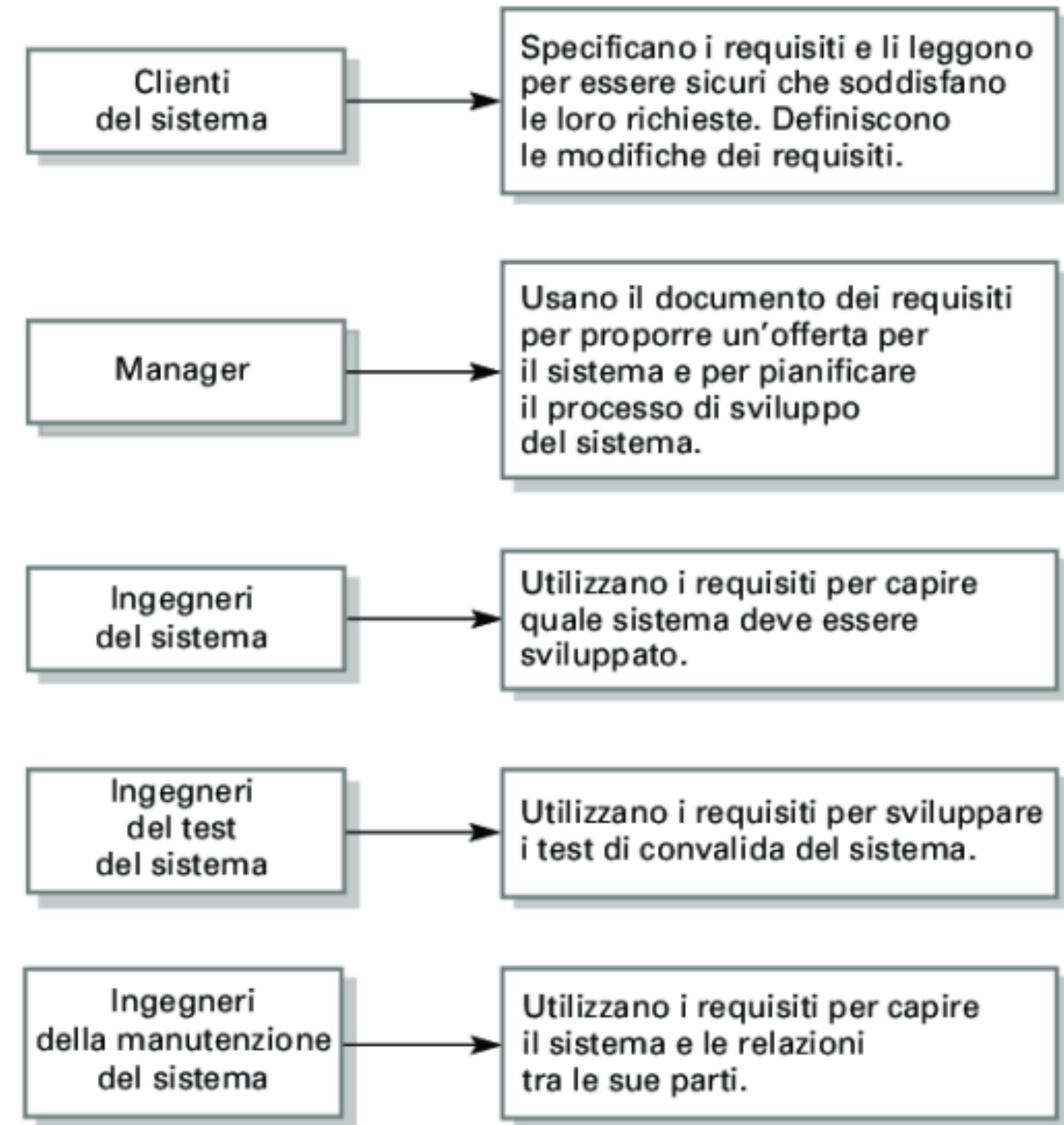
# IL DOCUMENTO DI SPECIFICA DEI REQUISITI DEL SOFTWARE (SRS)



- Per i sistemi critici servono requisiti dettagliati, in quanto la sicurezza e la protezione devono essere analizzate con cura per scovare eventuali errori nei requisiti. I documenti SRS saranno lunghi e dettagliati
- Quando il sistema viene sviluppato da una società esterna, le specifiche del sistema devono essere precise e dettagliate
- Se si usa uno sviluppo iterativo all'interno della società, il documento dei requisiti può essere meno dettagliato

# UTILIZZATORI DELL'SRS

- ▶ Molteplici tipologie di utenti che utilizzano l'SRS per diversi scopi
- ▶ Le informazioni contenute dipendono dal tipo di sistema e dall'approccio di sviluppo
- ▶ La varietà di utenti comporta un compromesso sul livello di dettaglio da adottare





# LO STANDARD IEEE/ANSI 830-1998



- <http://www.math.uaa.alaska.edu/~afkjm/cs401/IEEE830.pdf>
- Suggerisce una struttura generica per il documento SRS
- Utile come guida per sviluppare requisiti dettagliati per lo specifico sistema
- Particolarmente appropriato per i sistemi critici che hanno lunga durata e di solito sono sviluppati da un gruppo di organizzazioni
- Contiene più capitoli:
  - Introduzione
  - Descrizione Generale
  - Requisiti Specifici
  - Appendici
  - Indice



## **Table of contents**

### **1.Introduction**

- 1.1. Purpose (Scopo del documento)
- 1.2. Scope (Scopo del prodotto)
- 1.3. Definitions, Acronyms and Abbreviations
- 1.4. References
- 1.5. Overview (Descrizione generale del resto del documento)

### **2.General Description**

- 2.1. Product Perspective (Relazione del prodotto con l'esterno e altri prodotti)
- 2.2. Product Functions (Sommario delle attività)
- 2.3. User Characteristics (Caratteristiche degli utenti target ad es. esperienza)
- 2.4. General Constraints
- 2.5. Assumptions and Dependencies



## 3. Specific Requirements

### 3.1. Functional Requirements

#### 3.1.1. Functional Requirement 1

3.1.1.1. Introduction

3.1.1.2. Inputs

3.1.1.3. Processing

3.1.1.4. Output

#### 3.1.2. Functional Requirement 2

...

#### 3.1.3. Functional Requirement n

...



# ESEMPIO DI SPECIFICA DI REQUISITI FUNZIONALI (SEZ. 3)



## 3.1.1 Inserimento articolo in archivio

**Introduzione:** La funzionalità consente l'immissione dei dati relativi ad un nuovo articolo ed alla loro registrazione in un archivio

**Input:** codice articolo, descrizione, unità di misura

**Elaborazione:** visualizzazione di un form per l'immissione dei dati di input; lettura dei dati di input; verifica che l'unità di misura sia uno dei tre valori possibili e registrazione dei dati immessi nell'archivio Articoli

**Output:** dati di input registrati nell'archivio Articoli o messaggio di errore 'Unità di misura non ammessa'

- Ciascuna Funzionalità deve essere specificata indicando gli input, l'output e l'azione elaborativa
- Se necessario devono essere riportate anche le pre-condizioni e le post-condizioni

# LO STANDARD IEEE/ANSI 830-1998

- Il capitolo 3 definisce sia requisiti funzionali che non funzionali. Un paragrafo è dedicato alle interfacce
- I capitoli 4 e 5 contengono le appendici e l'indice, rispettivamente

## 3.2. External Interface Requirements

### 3.2.1. User Interfaces

### 3.2.2. Hardware Interfaces

### 3.2.3. Software Interfaces

## 3.3. Performance Requirements

## 3.4. Design Constraints

### 3.4.1 Standard Compliance

### 3.4.2. Hardware Limitation

...

## 3.5. Software System Attributes

### 3.5.1. Security

### 3.5.2. Maintainability

...

## 3.6. Other Requirements

...



# SRS SI UTILIZZA ANCORA?



- In certi settori (ferroviario, avionico, automotive, aerospaziale, medicale, etc..) la specifica descritta in SRS è assolutamente obbligatoria
- In questi settori esistono Enti che promulgano specifici Standard (ad es.: ISO, IEEE, CELENEC, NIST, IEC, UNI)
- Gli Standard impongono il tipo di documenti da produrre (tra cui l'SRS)



# SRS SI UTILIZZA ANCORA?



- Anche in ambienti meno restrittivi all'inizio dell'analisi conviene produrre un SRS, anche non seguendo pedissequamente un template
- In seguito sono definiti i **Casi d'Uso** (che non sono previsti da IEEE STD 830-1998)
- Infine, i requisiti possono essere "incrociati" con i casi d'uso

Nel nostro corso adotteremo questa strategia



# CONVALIDA DEI REQUISITI

- Processo che verifica se la specifica dei requisiti definisca il sistema realmente voluto dal cliente con accuratezza e chiarezza
- Gli errori nel documento SRS possono portare a costi molto alti se scoperti durante lo sviluppo o dopo la messa in servizio del sistema
- Una modifica dei requisiti di solito implica che anche la progettazione e l'implementazione del sistema devono cambiare







- **Corretto/Valido:** ogni requisito soddisfa i bisogni espressi dagli utenti
- **Completo:** tutte le funzionalità e vincoli sono inclusi nelle specifiche
- **Realistico:** i requisiti possono essere implementati dati le tecnologie ed il budget disponibile
- **Consistente/Non Ambiguo:** ogni requisito ha una sola interpretazione e non contraddice altri requisiti
- **Verificabile:** è possibile dimostrare che il sistema soddisfi ogni requisito (richiede scrittura di test)
- **Tracciabilità:** l'origine di ciascun requisito è chiara e può essere referenziata nello sviluppo futuro

# AMBIGUITÀ



- Ogni requisito deve avere sempre una sola interpretazione
- Per ogni caratteristica usare sempre e solo lo stesso termine
- Se un termine può avere diversi significati, in base al contesto, specificarlo in un Glossario
- Spesso dipendente dall'ambiguità del linguaggio naturale



# AMBIGUITÀ – ESEMPIO

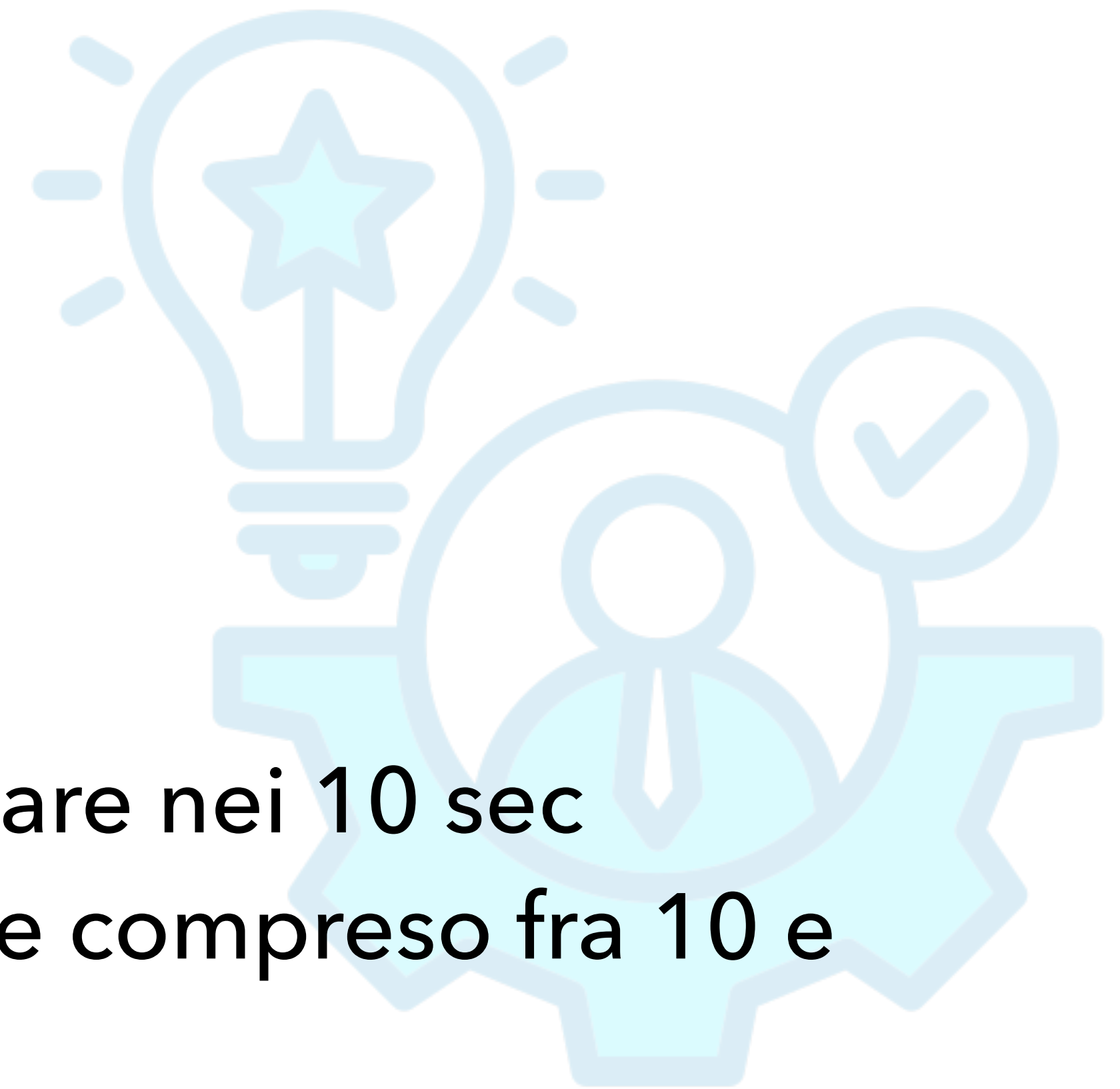


- Requisiti di un Sistema software di un Ristorante
- ***RF1: Ogni ordine deve essere evaso dal responsabile***
- Significati possibili:
  - Ordine per la cucina oppure Ordine per il fornitore
  - Ogni ordine ha un suo responsabile (cameriere?) oppure  
Tutti gli ordini hanno lo stesso responsabile degli ordini





- Un requisito è verificabile se esiste un procedimento che una persona o una macchina possono seguire per verificare se il software soddisfa tale requisito.
- Esempio di requisiti **non verificabili**:
  - A. Il software deve funzionare bene
  - B. Il tempo di risposta deve essere breve
- Esempio di requisiti **verificabili**
  - A. Il tempo di risposta all'evento X deve rientrare nei 10 sec
  - B. Il tempo di risposta all'evento X deve essere compreso fra 10 e 20 sec.







- Un SRS è tracciabile se è possibile:
  - ◎ **Tracciatura all'indietro (backward traceability)**
    - è chiara l'origine di ciascun requisito
    - dalla documentazione e dal codice è possibile risalire al requisito corrispondente
  - ◎ **Tracciatura in avanti (forward traceability)**
    - ogni requisito è collegabile ad un elemento del codice o del progetto o a requisiti futuri
    - Il requisito deve essere identificato univocamente da un nome e/o un numero



- Quando un requisito è conseguenza di un altro, essi devono essere tracciabili in avanti ed indietro
  - *Esempio:* Il requisito R3.5 sui tempi di risposta della base dati è conseguenza del requisito R1.5 sui tempi di risposta del sistema complessivo
  
- Quando un requisito deriva da legislazioni o norme si dovrebbero riportare i loro estremi
  - *Esempio:* Il requisito sul passaggio dal secondo al terzo anno di corso R3.1.1 deve fare riferimento all'ordinamento dell'Ateneo art. 2, comma 5, pag. 77



- **Revisione dei requisiti:** i requisiti vengono analizzati da un team di revisori, che controllano errori e incoerenze (ad es. attraverso ispezioni manuali)
- **Prototipazione:** gli utenti finali e i clienti provano un prototipo eseguibile del sistema per vedere se esso soddisfa le loro necessità
- **Generazione di test case:** sviluppo di test per verificare che i requisiti siano testabili. Lo sviluppo dei test svela problemi presenti nei requisiti. Inoltre, se un test è difficile o impossibile da progettare, significa che i requisiti saranno difficili da implementare e dovranno essere riconsiderati
- **Scenari:** chiariscono le modalità operative del sistema

# PROBLEMATICHE DELLA CONVALIDA DEI REQUISITI



- Difficile dimostrare che un insieme di requisiti soddisfi tutte le necessità dell'utente
- Gli utenti e gli sviluppatori potrebbero fare fatica a immaginare il sistema in funzione
- I problemi nei requisiti non rilevati dalle operazioni di convalida dovranno quindi essere risolti attraverso modifiche dopo che il documento dei requisiti è stato accettato dalle parti





# ESERCIZIO



- Trovare almeno quattro omissioni e ambiguità nella seguente descrizione dei requisiti dell'ipotetico sistema software "TicketMaster"

Un sistema automatico di emissione dei biglietti vende biglietti ferroviari. Gli utenti selezionano la loro destinazione e inseriscono la carta di credito e il codice personale. Il biglietto viene emesso e addebitato sulla carta di credito. Quando l'utente preme il pulsante di avvio, compare un menu che mostra le varie destinazioni possibili, insieme a un messaggio che chiede di selezionare una destinazione e il tipo di biglietto. Una volta selezionata la destinazione, sullo schermo appare il prezzo del biglietto e all'utente è chiesto di inserire la carta di credito. Viene verificata la sua validità, e all'utente viene chiesto di inserire il codice personale di identificazione (PIN). Quando la transazione economica è completata, il biglietto viene emesso.



Un sistema automatico di emissione dei biglietti vende biglietti ferroviari. Gli utenti selezionano la **loro destinazione** e inseriscono la carta di credito e il codice personale. Il biglietto viene emesso e addebitato sulla carta di credito. Quando l'utente preme il pulsante di avvio, compare un menu che mostra le varie destinazioni possibili, insieme a un messaggio che chiede di selezionare una destinazione e il tipo di biglietto. Una volta selezionata la destinazione, sullo schermo appare il prezzo del biglietto e all'utente è chiesto di inserire la carta di credito. Viene verificata la sua validità, e all'utente viene chiesto di inserire il codice personale di identificazione (PIN). Quando la transazione economica è completata, il biglietto viene emesso.

- Il sistema deve poter vendere solo biglietti che hanno come partenza la stazione in cui è situato il sistema?



Un sistema automatico di emissione dei biglietti vende biglietti ferroviari. Gli utenti selezionano la loro destinazione e inseriscono la carta di credito e il codice personale. Il biglietto viene emesso e addebitato sulla carta di credito. Quando l'utente preme il pulsante di avvio, compare un menu che mostra le varie destinazioni possibili, insieme a un messaggio che chiede di selezionare una destinazione e il tipo di biglietto. Una volta selezionata la destinazione, sullo schermo appare il prezzo del biglietto e all'utente è chiesto di inserire la carta di credito. **Viene verificata la sua validità, e all'utente viene chiesto di inserire il codice personale di identificazione (PIN).** Quando la transazione economica è completata, il biglietto viene emesso.

- Cosa accade se la carta di credito è invalida?
- Cosa accade se il PIN è errato?





Un sistema automatico di emissione dei biglietti vende biglietti ferroviari. Gli utenti selezionano la loro destinazione e inseriscono la carta di credito e il codice personale. Il biglietto viene emesso e addebitato sulla carta di credito. Quando l'utente preme il pulsante di avvio, compare un menu che mostra le varie destinazioni possibili, insieme a un messaggio che chiede di selezionare una destinazione e il tipo di biglietto. Una volta **selezionata la destinazione, sullo schermo appare il prezzo del biglietto** e all'utente è chiesto di inserire la carta di credito. Viene verificata la sua validità, e all'utente viene chiesto di inserire il codice personale di identificazione (PIN). Quando la transazione economica è completata, il biglietto viene emesso.

- Si può acquistare più di un biglietto alla volta per la stessa destinazione oppure bisogna acquistarne uno alla volta?