

# Ingegneria del Software

## Requisiti e Specifiche

Obiettivi.

Affrontare i primi passi della produzione del software: la definizione dei requisiti ed il progetto architeturale che porta alla definizione delle specifiche.

Illustrare finalità e caratteristiche dello studio di fattibilità: l'analisi del bisogno e l'idea di soluzione.

Rispondere ad alcune domande: cos'è un requisito? Come scoprire i requisiti?

Presentare la fase di progettazione architeturale che porta alla produzione del documento delle specifiche.

Fulvio Sbroiavacca



# Studio di fattibilità (1)

- Il cliente normalmente presenta il suo problema al fornitore in modo molto informale e vago, il fornitore deve capire ciò che si aspetta il cliente, spiegare se è fattibile e con quanto sforzo
- Il fornitore raccoglie le informazioni necessarie per mezzo di interviste agli utenti, utilizzando questionari e check-list
  - Le check-list sono liste di quesiti importanti che non vengono sottoposti all'utente ma utilizzate come memorandum da parte dell'intervistatore per non dimenticare punti determinanti durante l'intervista
- Normalmente vi sono dei vincoli che il sistema deve soddisfare per essere utilizzabile
  - Protezione (security) delle informazioni da accessi indesiderati
  - Sicurezza (safeness) dei dati: non devono essere persi accidentalmente
  - Prestazioni (performance) del sistema (ad es.: vincoli sui tempi di risposta)

## Studio di fattibilità (2)

- Una considerazione sulle prestazioni
  - *Un utente abituato a cercare una informazione in un documento cercando per ore in vari scaffali, dopo l'utilizzo di un sistema di ricerca informatizzato, trova che il sistema è lento se deve aspettare 10 secondi*

## Studio di fattibilità (3)

- Lo studio di fattibilità è un documento che indica al committente come possono essere raggiunti gli obiettivi e con quali costi
  - Il linguaggio e la struttura del documento devono essere adeguati a colui che deve leggerlo, che normalmente ha poche conoscenze informatiche
- Le informazioni che fornisce riguardano i punti salienti del sistema da realizzare
  - Esistenza di un prodotto di mercato adeguato alle esigenze, eventualmente adeguabile con un insieme limitato di modifiche e/o integrazioni
  - Caratteristiche ed architetture dell'hardware necessario
  - Tempi di sviluppo del software
  - Impiego di risorse dell'utente per consentire la definizione del prodotto
  - Analisi dei costi/benefici (nel caso di progetti molto costosi, come ad esempio i progetti di Business Process Reengineering, è fondamentale individuare delle stime dei benefici che bilancino l'investimento iniziale)
- Quando il cliente accetta la proposta contenuta nello studio il prodotto verrà sviluppato: è l'inizio del suo ciclo di vita

# I requisiti

## Cosa sono i requisiti?

- *I requisiti sono le caratteristiche che utente e committente desiderano che siano presenti in un prodotto software da realizzare*
  - L'obiettivo di un progetto di sviluppo software, quando si realizza un nuovo sistema o si apportano modifiche ad un sistema già esistente, è sempre realizzare un prodotto che soddisfi i requisiti
  - Utente e committente valuteranno il risultato sulla base del fatto che soddisfi o meno i requisiti

# Analisi dei requisiti (1)

- Il committente esprime una serie di vincoli
- Il fornitore formula una o più ipotesi di soluzione, in grado di rispondere, in tutto o in parte, ai requisiti espressi
- Il committente sceglie tra le soluzioni proposte quella migliore (dal suo punto di vista) in termini di rapporto tra costi e benefici, e stipula un contratto con il committente
- Il fornitore produce un documento dei requisiti
  - in fase di collaudo la conformità ai requisiti concordati costituirà il criterio per l'accettazione del prodotto da parte del committente
    - deve essere leggibile e facilmente modificabile: è soggetto a modifiche durante la fase di progetto, tipicamente per l'aggiunta di funzionalità richieste dall'utente
    - deve essere completo e consistente: devono cioè essere considerate tutte le funzionalità richieste dal cliente e non vi devono essere requisiti contraddittori

## Analisi dei requisiti (2)

- Il documento dei requisiti deve essere verificabile: quando si descrive un vincolo oppure un comportamento si deve poter verificare se è soddisfatto
  - *“il sistema deve essere facile da usare facendo pochi errori”* la frase non è adeguata poiché descrive qualcosa di non verificabile
  - *“l’utente medio deve sapere usare il sistema dopo due ore di addestramento e non deve fare poi più di due errori al giorno”* è corretta perché descrive qualcosa di verificabile
- Deve definire un modello astratto del sistema finale
  - Comportamento funzionale del sistema (descrizione funzionale dei servizi forniti dal sistema, delle operazioni accessibili all’utente)
  - Vincoli: prestazioni, protezione, sicurezza, usabilità, ecc.
  - Descrizione dell’hardware necessario
  - Database prescelto
  - Informazioni per il mantenimento del sistema (ad esempio release periodici, manutenzione per malfunzionamenti)
  - Glossario per la spiegazione dei termini tecnici

# I principali fattori di insuccesso dei progetti

- I principali fattori di insuccesso dei progetti non riguardano gli aspetti tecnici
- Problemi manageriali
  - mancanza di supporto da parte della direzione
  - mancanza di risorse
  - mancanza di pianificazione
- Problemi sui requisiti (soprattutto)
  - requisiti incompleti
  - mancato coinvolgimento degli utenti
  - attese irrealistiche
  - cambiamento dei requisiti in corso d'opera
  - progetti cancellati perché non più utili



# Gestione dei requisiti (1)

- La gestione dei requisiti è una delle fasi più critiche del processo di sviluppo
  - comprende attività di: acquisizione, analisi, ingegnerizzazione, negoziazione, validazione
- Il committente esprime i requisiti del progetto attraverso una serie di bisogni da soddisfare e di obiettivi da raggiungere, rappresentando contemporaneamente dei vincoli
  - normalmente il committente ha chiari i propri obiettivi in termini di business ma non è in grado di stilare un elenco dettagliato di requisiti
- Il compito del progettista è comprendere ed acquisire i requisiti
  - il progettista deve interagire con il committente per arrivare ad un chiarimento progressivo delle varie caratteristiche del sistema finale

## Gestione dei requisiti (2)

- Si tratta di un processo iterativo, svolto attraverso:
  - interviste,
  - analisi di scenario
  - analisi di rischio
  - analisi costi/benefici

che comporta successive proposte da parte del progettista, per rendere il committente consapevole di diverse possibilità di soluzione, ed acquisire feedback sulle ipotesi prospettate

- *L'individuazione dei requisiti è un'attività faticosa, comporta una serie ripetuta di interazioni e discussioni tra committente e fornitore*

# I conflitti tra requisiti

- *Il committente desidera avere il sistema più completo ed efficiente possibile, rilasciato nel più breve tempo possibile, con i minori costi possibili*
- *Gli utenti desiderano avere il sistema più adatto alle loro necessità*
- Il progettista deve mettere in luce i conflitti tra i requisiti in modo da renderli evidenti al committente e trovare le relative soluzioni

# Il cambiamento dei requisiti

- I requisiti nascono e cambiano continuamente (anche mentre sono in corso le attività di realizzazione):
  - cambiamenti organizzativi
  - innovazioni legislative
  - modifica delle strategie aziendali
- A fronte di un cambiamento di requisiti in corso d'opera vi sono due possibili strade di soluzione:
  - rimandare la presa in carico dei nuovi requisiti ad una release successiva del sistema  
(non è sempre praticabile, determina comunque scontentezza nel committente)
  - rinegoziare i termini del progetto, rivedendone i contenuti, i costi ed i tempi  
(il progettista deve rendere costantemente partecipe il committente)
- I requisiti cambiano dopo che il sistema è stato rilasciato
  - il progettista deve presidiare l'evoluzione dei requisiti durante l'intero ciclo di vita del sistema

# Come raggiungere un accordo sui requisiti?

- Le richieste del committente (e degli utenti) vanno gestite in modo distinto dalle proposte di soluzione del progettista
- Attraverso la risoluzione dei conflitti sui requisiti si converge verso una serie di soluzioni che soddisfano le richieste del committente (e degli utenti)
- Il committente sceglie quella migliore, dal suo punto di vista, in termini di rapporto tra costi e benefici
- Viene stipulato un accordo o contratto per la realizzazione
- La soluzione sulla base della quale viene raggiunto l'accordo deve:
  - rispondere in modo adeguato ai requisiti espressi dal committente (e dagli utenti)
  - evidenziare costi e tempi di realizzazione
  - essere comprensibile nelle sue caratteristiche per tutte le parti in causa, senza ambiguità
  - evidenziare i benefici in rapporto ai costi (ad esempio miglioramenti organizzativi)
  - chiarire quali sono i requisiti effettivamente soddisfatti, eliminati (a causa di conflitti), rimandati ad un momento successivo (ad esempio per contenere tempi e costi)

# Prototipizzazione

- Quando utilizzare la prototipizzazione?
  - lo studio di fattibilità da solo non basta a decidere la messa in opera del progetto
  - il fornitore non ha le idee sufficientemente chiare
  - l'utente non è sicuro di ciò che il fornitore gli propone
- I due principali criteri per la realizzazione del prototipo:
  - *si deve poter costruire rapidamente*
  - *e a basso costo rispetto al prodotto finale*
- Per poter ottenere ciò  
si deve operare con standard di qualità  
più bassi rispetto a quelli del prodotto finito

# Caratteristiche e finalità del prototipo

- Le principali caratteristiche del prototipo
  - poco modulare (lontano quindi da un prodotto del SWE)
  - poco affidabile
  - poco efficiente
  - non facilmente modificabile
  - normalmente da gettare a fine ciclo
- Il prototipo è necessario per:
  - rilevare le incomprensioni tra fornitore e cliente  
(il fornitore fa solitamente una dimostrazione del prototipo al cliente)
  - scoprire incompletezze, cioè funzionalità mancanti
  - inquadrare i punti di difficile utilizzo
  - come strumento per redigere chiaramente i documenti dei requisiti e delle specifiche

# Quali funzionalità si realizzano nel prototipo?

- Nel prototipo vengono realizzate le funzionalità richieste per il sistema finale, ma vi sono comunque delle mancanze
  - Non vengono realizzati i vincoli non funzionali  
(ad esempio se è necessario che il database sia crittografato, non lo sarà nel prototipo)
  - E' meno performante del sistema finale  
ciò può risultare strano all'utente,  
al quale normalmente si presenta il prototipo  
in un ambiente che simula quello finale  
(ad esempio per un software che controlla la navigazione di un aeroplano si costruisce un simulatore dell'aeroplano)
- Nel prototipo non bisogna mai eliminare l'interfaccia utente
  - *per l'utente il sistema è l'interfaccia*  
se questa è fatta bene è ciò che impressiona l'utente  
(mentre si può costruire il prototipo simulando la presenza di un database)



# Come si fanno i prototipi?

- L'idea di base è programmare in modo veloce *anche se poco modulare*
- Il linguaggio utilizzato di norma è di alto livello per sollevare lo sviluppatore dal maggior numero possibile di dettagli
- Ci si concentra molto sulle interazioni dell'interfaccia utente
- Devono essere “colte” le funzionalità “chiave” cercando contemporaneamente di coprire lo spettro funzionale richiesto
- Normalmente va sempre privilegiato l'aspetto funzionale rispetto a quello tecnico

# Aspetti contrattuali

- Individuati i requisiti del prodotto, validati dal committente, si passa alla fase di progettazione che porta alla produzione del documento delle specifiche, funzionali e non funzionali
  - si tratta di tradurre il documento dei requisiti in un linguaggio preciso
- In generale a questo punto è già stato firmato un contratto
- Normalmente vi è un contratto per lo studio di fattibilità
- Nel caso si sviluppi un prototipo viene firmato di solito un contratto per il prototipo
  - A meno che non si tratti di attività svolte su iniziativa del produttore
- Infine viene firmato il contratto finale per il sistema finale

# Le specifiche

## Cosa sono le specifiche?

- *Le specifiche sono la riformulazione dei requisiti in modo formalizzato per l'utilizzo nel processo interno del fornitore, in modo da definire precisamente che cosa deve fare il software da produrre*
  - L'obiettivo di un processo di sviluppo è realizzare un prodotto che soddisfi le specifiche
  - Il risultato del processo di sviluppo verrà valutato sulla base del fatto che soddisfi o meno le specifiche

# Specifiche formali e informali

- Il documento delle specifiche descrive in modo formale ciò che farà il software e le sue parti/moduli
- Le specifiche informali, che sono le più diffuse, sono formulate in linguaggio naturale, anche con l'ausilio di tabelle e grafici
  - Per ottenere maggiore precisione si seguono degli standard ma non vi è una vera e propria simbologia matematica
- Le specifiche formali si basano su formalismi matematici, sono espresse in un linguaggio con sintassi e semantica definite formalmente
  - Il linguaggio naturale si utilizza per commentare e chiarire le notazioni utilizzate
  - Viene utilizzata una metodologia di definizione formale delle specifiche

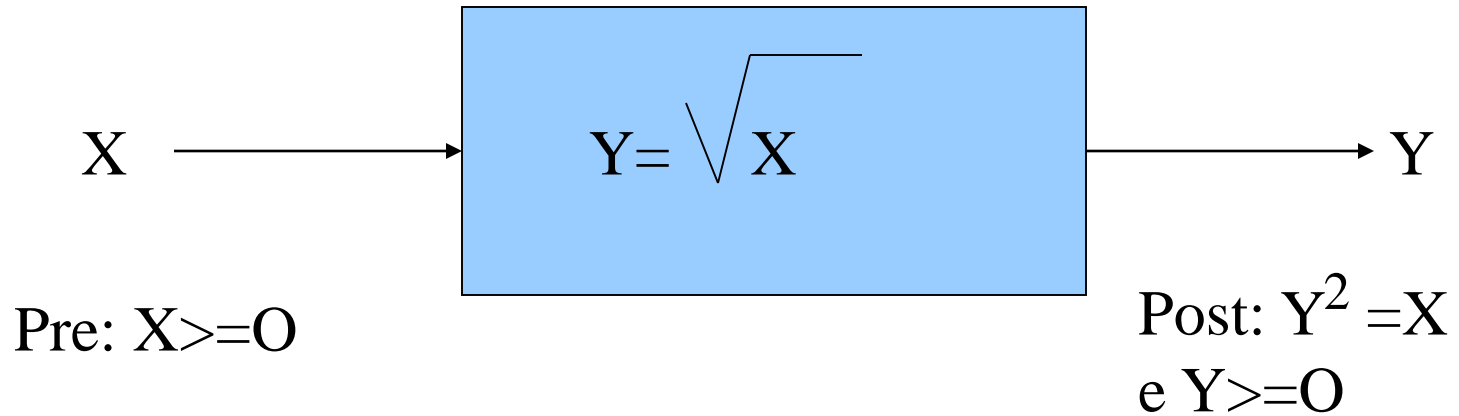
# Uso di specifiche formali

- Vantaggi delle specifiche formali
  - L'utilizzo di un linguaggio formale (assieme ad un opportuno software di supporto) permette di rilevare inconsistenze nelle specifiche e può aiutare l'analista a scoprire incompletezze
  - Se esiste una definizione formale della semantica del linguaggio di programmazione utilizzato per la codifica, si possono utilizzare strumenti automatici per effettuare la verifica formale del codice
  - Se si dispone di un linguaggio di specifiche formali computabili si può costruire un interprete per eseguire il documento delle specifiche: si ottiene così un prototipo del sistema da realizzare
- Per formulare specifiche formali sono necessari strumenti di supporto automatici
- Generalmente le specifiche formali sono poco utilizzate a causa di:
  - alti costi di utilizzo
  - scarse conoscenze degli sviluppatori
  - poco utilizzabili per diverse parti del sistema: ad esempio l'interfaccia utente o processi ed eventi esterni

# Un esempio di linguaggi di specifiche formali: precondizioni e postcondizioni

- La forma di specifica formale più semplice è la specifica assiomatica
- Il sistema viene rappresentato come un insieme di funzioni (senza stato interno) ed ogni funzione è descritta per mezzo di precondizioni e postcondizioni
- Nella pratica è utilizzabile per piccoli sistemi o parti di sistemi
- Come si opera?
  - Il sistema viene suddiviso in funzioni o blocchi
  - La granularità di un blocco può variare di molto: un'assegnazione, una procedura o l'intero sistema
  - Le precondizioni sono le condizioni che devono essere verificate affinché la funzione possa essere eseguita correttamente
    - Riguardano l'input del blocco, specificando in particolare quali input sono corretti
  - Le postcondizioni sono le condizioni che risultano verificate al termine di un'esecuzione corretta del blocco
    - Descrivono l'output prodotto dal blocco per un input corretto

## Esempio di pre- e post-condizioni

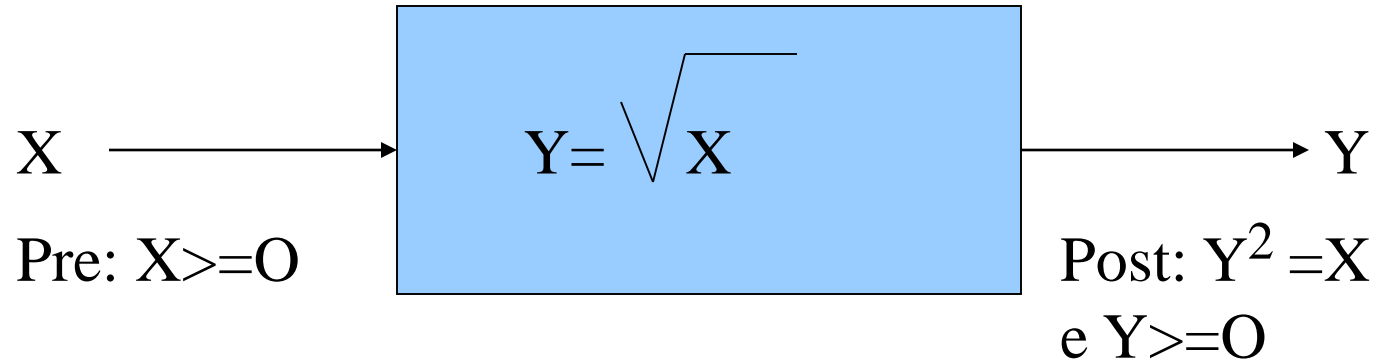


(1)  $\text{PRE}(x, f) \longrightarrow \text{POST}(x, f)$

(2)  $\text{PRE}(x, f) \longrightarrow \text{POST}(x, f)$

$\text{PRE}(x, f) \longrightarrow \text{ERROR}(x, f)$

## Esempio di pre- e post-condizioni



$$(1) \quad \text{PRE}(x, f) \longrightarrow \text{POST}(x, f)$$

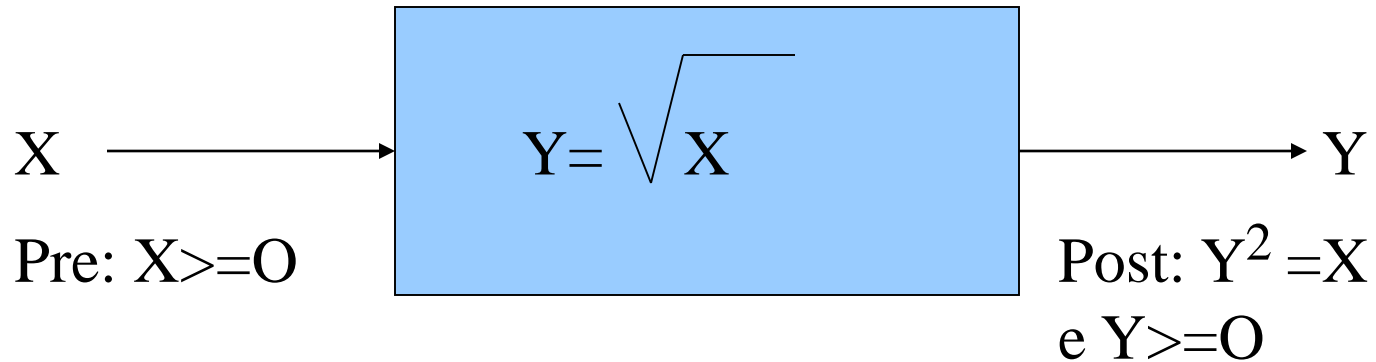
$$(2) \quad \text{PRE}(x, f) \longrightarrow \text{POST}(x, f)$$

$$\text{PRE}(x, f) \longrightarrow \text{ERROR}(x, f)$$

La preconditione definisce la classe degli input accettabili  
e la postcondizione fornisce un modo per verificare che l'output sia  
corretto



## Esempio di pre- e post-condizioni



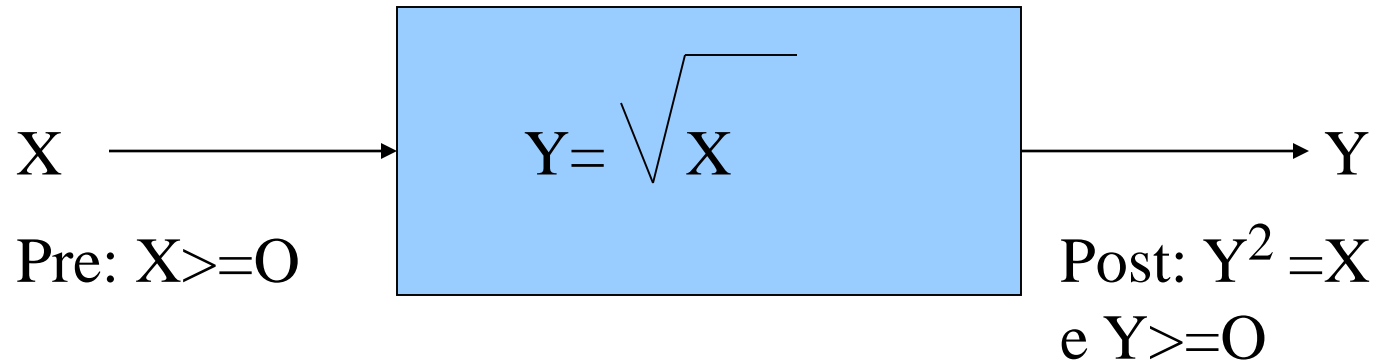
$$(1) \quad \text{PRE}(x, f) \longrightarrow \text{POST}(x, f)$$

$$(2) \quad \text{PRE}(x, f) \longrightarrow \text{POST}(x, f)$$

$$\text{PRE}(x, f) \longrightarrow \text{ERROR}(x, f)$$

Quando vale la preposizione (1), la postcondizione descrive l'output correttamente solamente quando l'input è corretto (correttezza parziale), non è applicabile quando l'input non sia corretto, cioè la preconditione non è soddisfatta

## Esempio di pre- e post-condizioni



$$(1) \quad \text{PRE}(x, f) \longrightarrow \text{POST}(x, f)$$

$$(2) \quad \text{PRE}(x, f) \longrightarrow \text{POST}(x, f)$$

$$\text{PRE}(x, f) \longrightarrow \text{ERROR}(x, f)$$

Devono essere gestite le condizioni di errore, che descrivono le caratteristiche dell'output nel caso di input errato

Nell'esempio avremo error:  $y = -1$ .

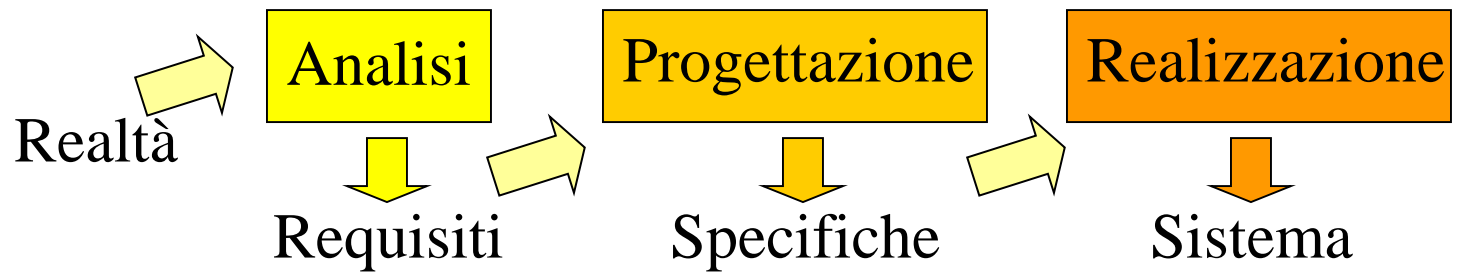
Una specifica fornita correttamente soddisfa la legge (2)

# Requisiti e Specifiche

## Documenti dei requisiti e delle specifiche

- Documento dei requisiti e documento delle specifiche possono sembrare simili, in realtà vi sono delle differenze sostanziali
- Il documento dei requisiti viene fornito all'utente
  - deve quindi essere comprensibile per l'utente
  - viene utilizzato al momento del collaudo
  - rappresenta uno strumento per il fornitore in caso di contese con il cliente
- Il documento delle specifiche viene utilizzato internamente dal produttore
  - non viene mostrato all'utente
  - può essere quindi molto tecnico

# Requisiti e Specifiche



- *Applichiamo le conoscenze acquisite*
- Il caso della gestione ordini