

---

# Analisi dei Requisiti

---



# Riassunto lezione precedente Outline della lezione

---

## Lezione precedente:

- Organizzazione del lavoro per realizzare un sistema sw
  - Processo sw
  - Modelli di ciclo di vita del sw

## Questa lezione

- Studio di fattibilità
- Analisi dei requisiti
  - Importanza dell'attività di analisi dei requisiti
  - Dominio
    - Comprensione e modellazione
  - Requisiti
    - Acquisizione e analisi

# Studio di fattibilità

---

Fase preliminare per stabilire l'opportunità o meno di realizzare il software

Si basa su:

- Una descrizione sommaria del sistema sw e delle necessità utente
- Un'analisi di mercato:
  - Confronto tra il mercato attuale e quello futuro
  - Costo della produzione, redditività dell'investimento
- Un'analisi tecnica per capire se è realizzabile
  - Strumenti per la realizzazione (software, librerie, ...)
  - Soluzioni algoritmiche e architetturali
  - Hardware
  - Processo ( prototipazione, progetti esplorativi, ricerca..)

# Dropbox ancora non esiteva, e...

---



Drew Houston  
...made a video

<https://www.youtube.com/watch?v=7QmCUDHpNzE>



“Our beta waiting list went from 5,000 people to 75,000 people literally overnight.”

# Qualcuno comprerà scarpe online?

---

- 1999, Nick Swinmurn voleva fare un negozio online di scarpe
- E' andato in un negozio vicino a casa, ha fotografato le scarpe e ha messo le foto in rete
- Se qualcuno comprava, lui andava al negozio a comprare e spediva
- Zero spese per iniziare
- Comprato da Amazon nel 2009 per 880 milioni di dollari



# Attività di analisi dei requisiti

---

Studiare e definire il problema da risolvere

- Per **capire cosa** deve essere realizzato
- Per **documentare cosa** deve essere realizzato
- Per **negoziazioni** committente/fornitore

# Che cos'è l'analisi dei requisiti

---

Il processo di studio e analisi delle esigenze del committente e dell'utente per giungere alla definizione del dominio del problema e dei requisiti del sistema.

## ■ Obiettivi

- Scoprire i confini del nuovo sistema (o software) e il modo in cui deve interagire con il suo ambiente
- Individuare e risolvere i conflitti tra i requisiti
- Negoziare le priorità degli stakeholders
- Definire le priorità dei requisiti
- Elaborare i requisiti del sistema, definiti nel documento dei requisiti, in modo che i manager possano fornire stime realistiche del progetto e gli sviluppatori possano progettare, implementare e testare.
- Classificare le informazioni sui requisiti in varie categorie e assegnare i requisiti ai sottosistemi.
- Valutare i requisiti di qualità desiderabile

# Implicazioni economiche e di qualità (repetita iuvant)

---

## Le cause di abbandono secondo lo Standish Group

1. Requisiti incompleti
2. Scarso coinvolgimento degli utenti
3. Mancanza di risorse
4. Attese irrealistiche
6. Modifiche a specifiche e requisiti
7. [.....]
8. [.....]
9. [.....]
10. Ignoranza tecnologica



# Quanto “costa” correggere un errore nei requisiti [Boehm]

---

<b>FASE</b>	<b>COSTO</b>
<b>Analisi Requisiti</b>	<b>1</b>
<b>Progettazione</b>	<b>5</b>
<b>Codifica</b>	<b>10</b>
<b>Test di unità</b>	<b>20</b>
<b>Test di accettazione</b>	<b>50</b>
<b>Operazione</b>	<b>100</b>

# Importanza dell'analisi dei requisiti

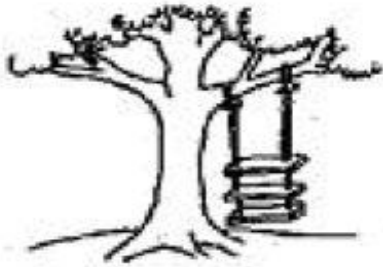
---

“Il 56% degli errori di un software possono essere riferiti ai requisiti”

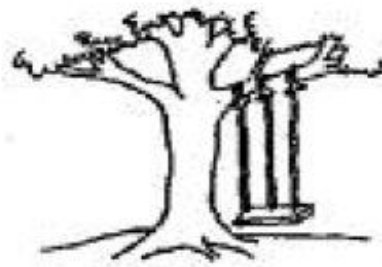
[Tom De Marco]

# L'altalena dei requisiti (una vignetta storica)

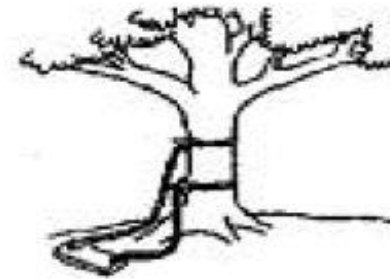
---



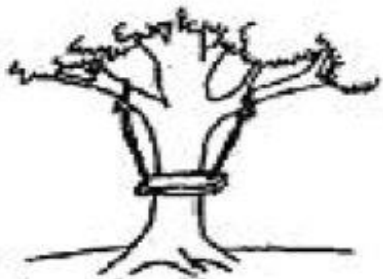
**As proposed  
by the project  
sponsor.**



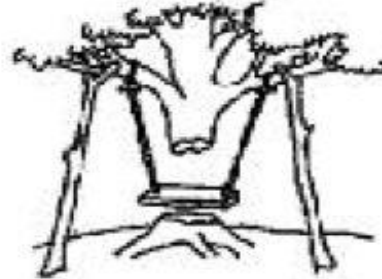
**As specified  
in the project  
request.**



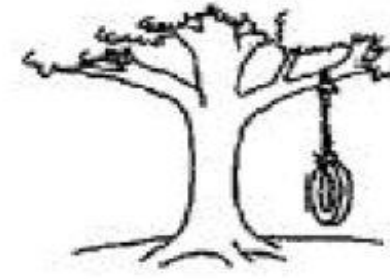
**As designed  
by the senior  
architect.**



**As produced  
by the  
engineers.**



**As installed at  
the user's  
site.**



**What the  
customer  
really wanted.**

Importante: si definisce

---

Cosa fare,  
non come farlo

# Prodotto dell'attività di analisi

---

Documento e/o modello che descrive:

- Il **dominio** del sistema da realizzare
- I **requisiti** utente

Opzionalmente anche:

- Manuale utente
- Casi di test

# Dominio

---

- Dominio
  - Il campo di applicazione del prodotto
- Prima ancora di incontrare i committenti il team di analisti deve acquisire conoscenza del dominio di applicazione
  - Per poter porre le domande giuste domande: pensate al software per transazioni finanziarie
  - Per capire la differenza sottile che può esserci tra termini solo apparentemente sinonimi: pensate a supporto, montante, sostegno, trave in ingegneria civile

# Dominio

---

Per acquisire conoscenza e definire il dominio:

- Si costruisce un **glossario**
  - Una **collezione di definizioni** di termini rilevanti in un dominio specifico
  - il team di analisti lo costruisce mentre studia il dominio e poi si arricchisce via via che si incontrano nuovi termini
  - Può essere riusato in progetti successivi nello stesso dominio
- e/o
- Si definiscono un **modello statico** e un **modello dinamico**

La conoscenza del dominio evolve dopo incontro con committenti, utenti, ecc

# Esempio: House of Cars

---





# Esempio di glossario

---

**House of Cars** è un parcheggio verticale multipiano, formato da 10 colonne e 24 piani per colonna, 12 sotto al livello strada e 12 sopra.

**Colonna:** colonna di posti auto dotata di un sollevatore centrale che raggiunge tutti i piani del parcheggio. Ha un proprio locale di ricezione auto. Comprende un carrello

**Carrello:** carrello per movimentare le vetture. E' dotato di "forchette" in corrispondenza delle ruote

**Cella:** formata da due box affiancati. Può quindi contenere due auto.

**Gruppo di spostamento elettromeccanico:** controlla il carrello che trasla la vettura nelle celle posizionate davanti e dietro al sollevatore.

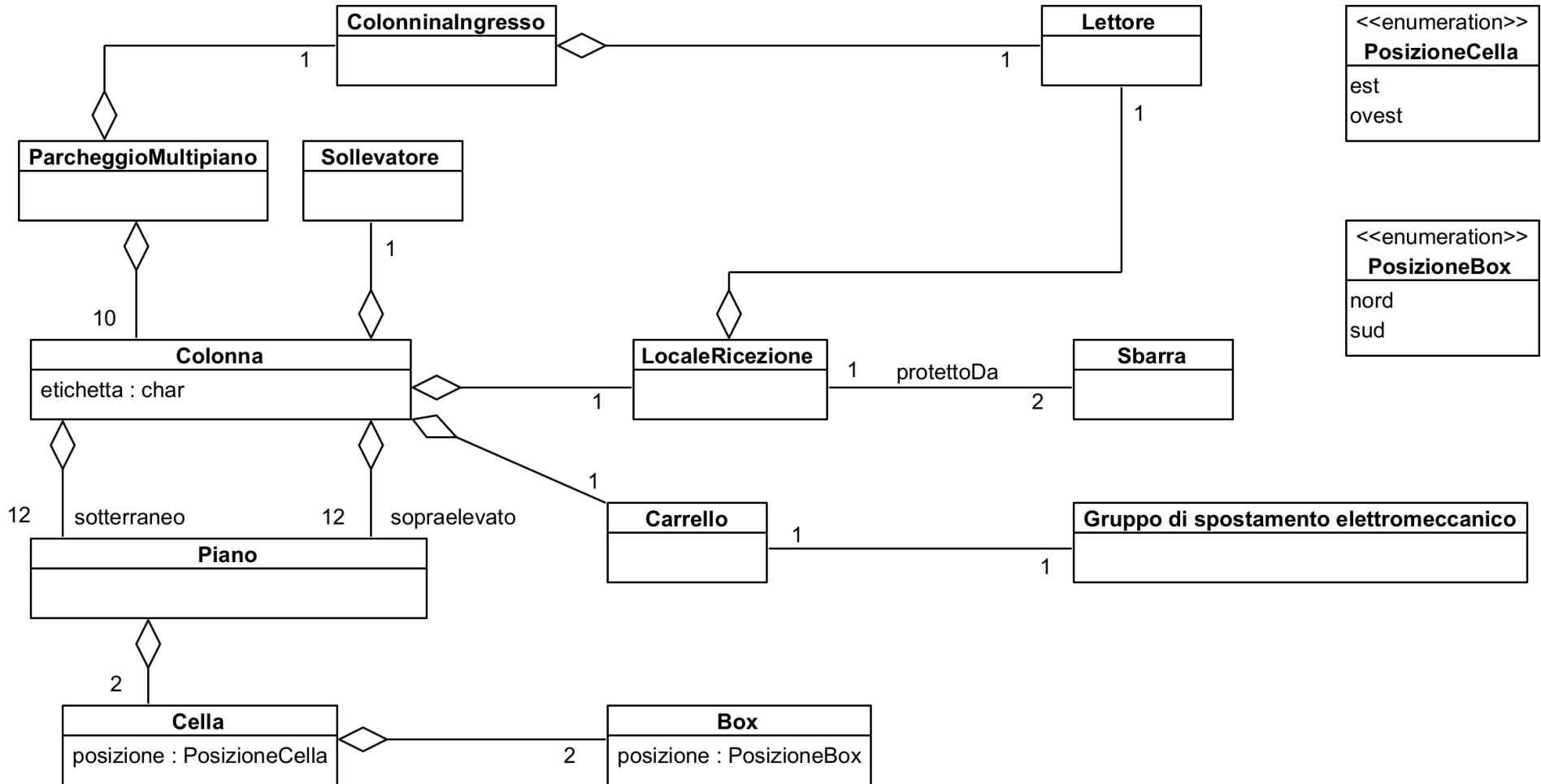
**Sollevatore:** trasporta la vettura al piano. Il sollevatore è dotato di videocamera

**Autovettura** = vettura

**Vettura:** dimensioni max: altezza 1,9 m, larghezza 2m, lunghezza 5m, peso 2,2 ton.

...

# Modello statico (usando UML)



# Modello dinamico, descrizione testuale

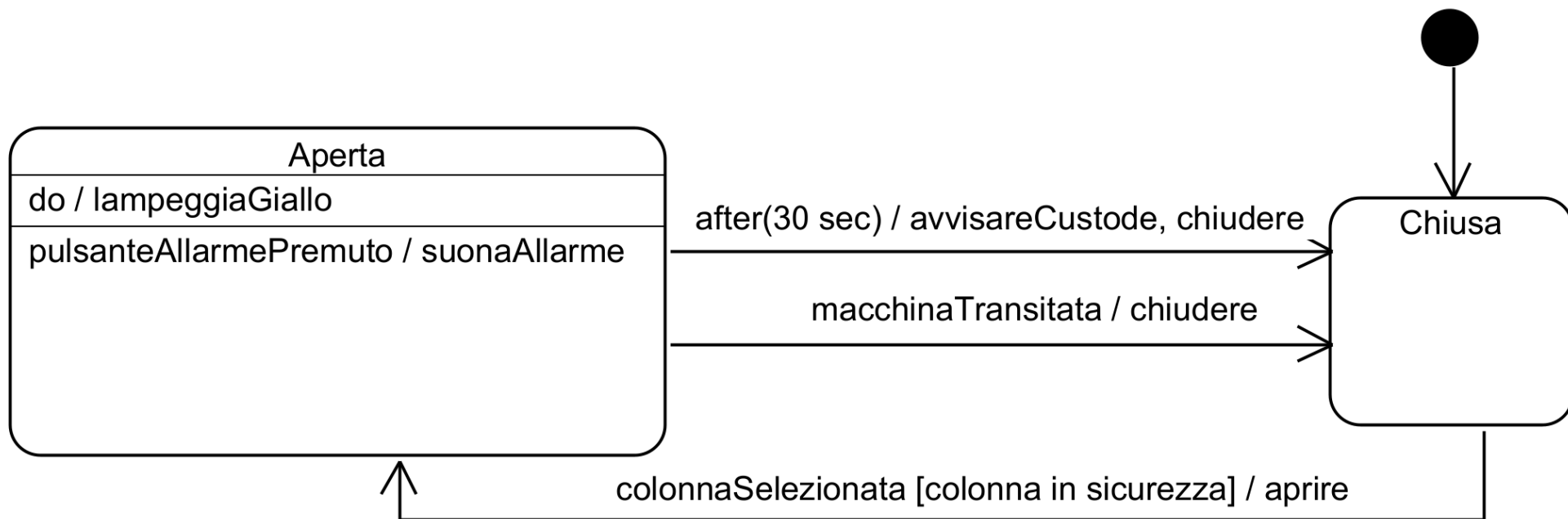
---

## Modello dinamico dall'autista non abbonato

1. Preleva il biglietto a banda magnetica
2. Segue le indicazioni visive e foniche che lo indirizzano alla colonna
3. Attende l'apertura della sbarra di sicurezza
4. Avanza e parcheggia correttamente la vettura di fronte alle porte di ingresso del sollevatore nel rispetto delle prescrizioni indicate
5. Spegne il motore, tira il freno a mano e lascia le chiavi nella vettura.
6. Esce e si allontana dall'auto
7. ...

# Modello dinamico (usando UML)

---



# Descrizione del dominio: hint

---

- Quando si descrive o modella il dominio di applicazione di un sistema software, occorre concentrarsi su
  - entità,
  - relazioni,
  - processi e
  - comportamenti

che **esistono indipendentemente dal sistema software che deve essere realizzato**

Il sistema software opera in un dominio e possibilmente lo controlla, ma il dominio esisteva anche prima

# Requisito: definizione

---

- Requisito
  - Una condizione o una capacità necessaria a un utente per risolvere un problema [..]
  - Una condizione (capacità) che deve essere soddisfatta (posseduta) [..] da un sistema [..] per soddisfare un contratto [..]
- [Glossario IEEE]
- In altre parole:
  - Una proprietà che deve essere garantita dal sistema per soddisfare una necessità di un utente
    - Funzionalità, qualità, ...

# Esempi di requisiti

---

1. Il sistema deve leggere il badge dell'utente abbonato
2. Il sistema, dopo aver letto il badge, deve selezionare la colonna con la cella libera più vicina al piano terra
3. Il sistema deve guidare l'utente che ha accesso al parcheggio mediante indicazioni visive e foniche alla colonna individuata per il parcheggio
4. Il sistema deve aprire le sbarre di sicurezza della colonna selezionata per consentire il passaggio dell'auto
5. ...

# Categorie di requisiti: Requisiti funzionali

---

Descrivono le funzionalità che il sistema deve realizzare

- In termini di:
  - azioni che il sistema deve compiere
  - come il sistema software reagisce a specifici tipi di input
  - come si comporta in situazioni particolari.

Tradizionalmente i requisiti a cui è dato maggior valore

## Esempi

1. Il sistema software deve fornire un appropriato visualizzatore per i documenti memorizzati
2. L'utente deve essere in grado di effettuare ricerche sia sull'intero insieme di basi di dati che su un loro sottoinsieme
3. Ad ogni nuovo ordine deve essere associato un identificatore unico (Order\_ID)



# Categorie di requisiti: Requisiti non funzionali

---

Descrivono le proprietà del sistema software in relazione a determinati servizi o funzioni e possono anche essere relativi al processo:

- caratteristiche di qualità quali:
  - efficienza, affidabilità, safety, usabilità, interfaccia, security, robustezza ecc.
- caratteristiche del processo di sviluppo:
  - standard di processo, uso di ambienti CASE, linguaggi di programmazione, metodi di sviluppo, ecc.
- caratteristiche esterne:
  - interoperabilità con sistemi di altre organizzazioni, vincoli legislativi, ecc.
- requisiti fisici (hardware, rete)

# Esempi di requisiti non funzionali

---

1. Il tempo di risposta del sistema all'inserimento della password utente deve essere inferiore a 10 sec
2. I documenti di progetto (deliverable) devono essere conformi allo standard XYZ-ABC-12345
3. Il sistema software non deve rilasciare ai suoi operatori alcuna informazione personale relativa ai clienti, tranne nominativo e identificatore

# Requisiti funzionali e non funzionali

---

E' bene tenerli separati.

Esempio:

- Il sistema deve validare il pin inserito dal cliente entro 3 secondi



- Il sistema deve validare il pin inserito dal cliente
- Il sistema deve completare la validazione del pin entro 3 secondi

# Esercizio: Bancomat

---

Si consideri un sistema per il prelievo automatizzato di denaro contante (Bancomat). Per poter utilizzare il sistema l'utente deve possedere una tessera magnetica. Il sistema deve mettere a disposizione le funzioni di prelievo, saldo, estratto conto. Il sistema deve essere disponibile a persone non vedenti, deve garantire un tempo di risposta inferiore al minuto, e deve essere sviluppato su architettura X86. Le operazioni di prelievo devono richiedere autenticazione tramite un codice segreto memorizzato sulla carta. Il sistema deve essere facilmente espandibile, e adattabile alle future esigenze bancarie.

# Curiosità

---

Si calcola che il **95% delle operazioni di un bancomat** venga eseguita tramite software scritto in **Cobol** e che in totale ci siano 200 miliardi di righe di codice che gestiscono ogni giorno 30 miliardi di transazioni.

[Antonio Dini su Wired 10-1-22]

# Descrizione dei requisiti

---

Approcci diversi che vedremo nel corso  
(non mutuamente esclusivi nella pratica)

Basata sul linguaggio naturale

- Glossario
- Specifica dei requisiti
  - funzionali
  - non funzionali

Basata su linguaggi grafici  
(UML)

- Modello del dominio
- Casi d'uso (per descrivere i requisiti funzionali)

# Descrizione formale dei requisiti

---

- Linguaggi formali che potrete vedere alla magistrale
  - Logiche temporali
  - Abstract state machines
  - Statecharts
- O che avete visto al primo anno
  - Logica al primo ordine
  - triple di Hoare

# In questa lezione

---

Approccio basato sulla produzione di documenti in linguaggio naturale

## Output

- Documento dei requisiti
  - Comprende il glossario



# Documento dei requisiti

---

- Elenca i requisiti del sistema da realizzare
  - Specifica cosa il prodotto deve fare e quali sono i vincoli che deve soddisfare
- E' un contratto tra lo sviluppatore e il committente.
- Specifica anche in genere una deadline per la consegna del prodotto

# Analisi dei requisiti in 5 passi

---

1. Acquisizione
2. Elaborazione
3. Convalida
4. Negoziazione
5. Gestione

# Acquisizione: Interviste

---

- Il team di analisti incontra i membri dell'organizzazione del committente
- Si procede con delle interviste che possono essere
  - strutturate
  - non strutturate
- Entrambe presentano vantaggi e rischi
- Difficile condurre una buona intervista



# Acquisizione ;-)

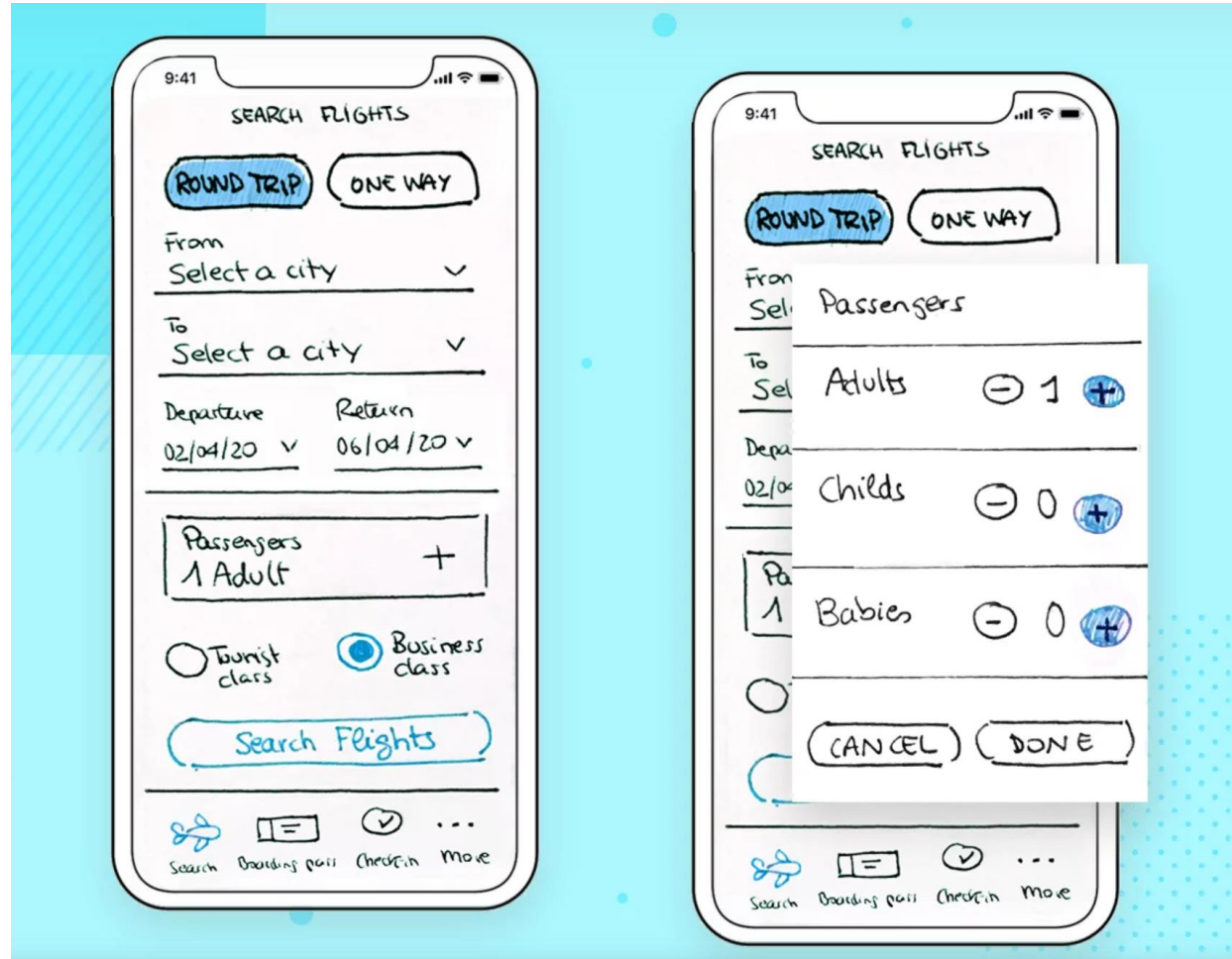


# Acquisizione: altre tecniche

---

- Questionari a risposte multiple a tutti (o quasi) i membri dell'organizzazione del committente
- Costruzione di prototipi
- Osservazione di futuri utenti al lavoro
- Studio di documenti

# Vanno bene anche prototipi su carta



# Analisi dei requisiti in 5 passi

---

1. Acquisizione
2. Elaborazione
3. Convalida
4. Negoziazione
5. Gestione

# Elaborazione

---

I requisiti sono espansi e raffinati

Viene scritta la prima bozza del documento dei requisiti



# Struttura del documento dei requisiti (1/2)

---

Proposta da Sommerville, e ispirata allo **standard IEEE/ANSI 830-1993** su *requirements documents*

## Introduzione

- Perché il sistema è desiderabile e come si inquadra negli obiettivi più generali del cliente

## Glossario

- I termini e i concetti tecnici usati

## *Definizione* dei Requisiti funzionali

- I servizi richiesti

## *Definizione* dei Requisiti non funzionali

- I vincoli operativi del sistema, e quelli sul processo di sviluppo

## Architettura

- La strutturazione in sottosistemi (cui riferire i requisiti)

# Struttura del documento dei requisiti (2/2)

---

## Specifica di requisiti del software e del sistema

- Specifica dettagliata dei requisiti funzionali

## Modelli astratti del sistema

- Modelli formali, o semi-formali: ciascuno illustra un solo punto di vista: controllo, dati, funzioni

## Evoluzione del sistema

- Previsione di successivi cambiamenti (p. es. di HW, o di requisiti)

## Appendici

- Individuazione ed eventuale descrizione della piattaforma hardware
- Requisiti di DataBase
- Manuale Utente, Piani di Test

## Indici

- Lemmario: lista di termini, con puntatori ai requisiti che li usano

# Requisiti ben posti

---

Non dovete scrivere un poema in versi, usate la forma assertiva:

**Il < sistema > deve < funzionalità > / < proprietà >**

Esempi:

- Il sistema deve gestire tutti i registratori di cassa del negozio
- Il sistema deve stampare la sintesi degli incassi della giornata
- A fine giornata il sistema deve elencare quali sono gli articoli da reintegrare in base alle vendite
- Il tempo di risposta del sistema a qualunque interrogazione deve essere inferiore a 3 secondi

# Analisi dei requisiti in 5 passi

---

1. Acquisizione
2. Elaborazione
3. Convalida
4. Negoziazione
5. Gestione

# Proprietà del documento dei requisiti

---

## Difetti da evitare

- Omissioni (incompletezze)
  - Mancata presenza di un requisito (incompletezza)
- Inconsistenze
  - Contraddizione fra i vari requisiti o dei requisiti rispetto all'ambiente operativo
- Ambiguità
  - Requisiti con significati multipli
- Sinonimi e omonimi
  - Termini diversi con il medesimo significato e termini uguali con differenti significati
- Presenza di dettagli tecnici
- Ridondanza
  - Può esserci ridondanza, ma solo tra sezioni diverse.

# Incompletezze

---

*Il sistema dovrà permettere agli utenti di fare delle ricerche per titolo, autore, o ISBN.*

*Che cosa significa per i CD-ROM?*

- Possono non avere un ISBN
- Vale solo per i libri
- Immaginate se realizzate il sistema con questo requisito senza considerare i CD-ROM.

Naturalmente non possiamo scrivere requisiti universali, ma possiamo tentare di essere il più possibile completi.

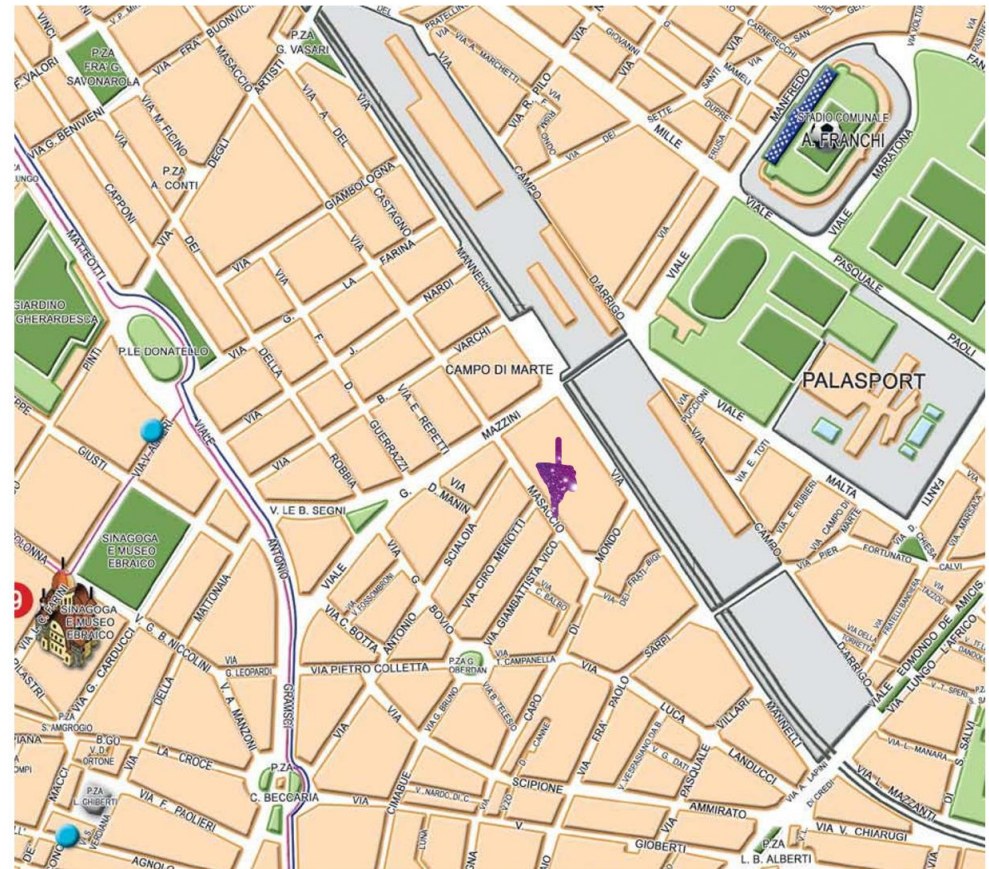
# Ambiguità, vaghezze, ...

Il linguaggio naturale è per sua natura impreciso

Il sistema deve mostrare

**tutte** le strade

**vicino** alla destinazione



# Ambiguità: Quantificatori

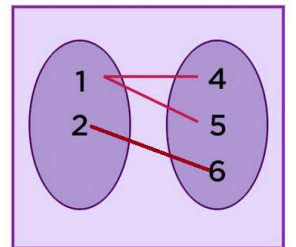
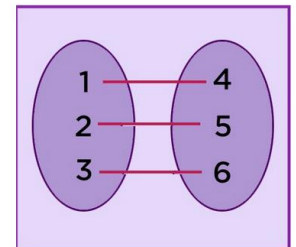
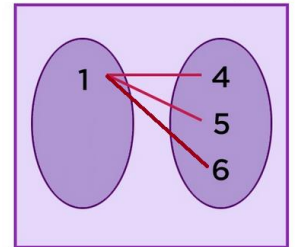
---

“**Tutti** i file sono controllati da **un** blocco di controllo”

Interpretazioni:

1. Un unico blocco di controllo controlla tutti i file
2. Ogni file ha un suo specifico blocco di controllo
3. Ogni file è controllato da uno specifico blocco di controllo; alcuni blocchi di controllo possono controllare più di un file

Cercare **i quantificatori**: Tutti, sempre, ogni, niente, ogni, qualsiasi





# Ambiguità di coordinazione

---

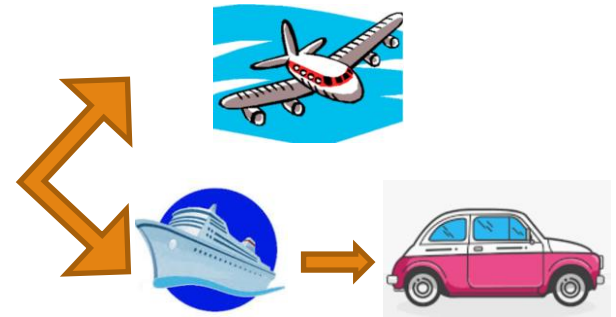
“Viaggerò in aereo **o** in traghetto **e** in macchina”

- “Viaggerò in aereo o viaggerò (in traghetto e in macchina)”

oppure

- “Viaggerò (in aereo o in traghetto) e in macchina”

Cercare **e o**



# Ambiguità referenziale

---

Per quelli di voi che hanno bambini e non **lo** sanno, abbiamo un asilo al piano di sotto

cercare i **pronomi**:

- Un requisito contiene un'ambiguità referenziale quando non è chiaro a cosa si riferisca un pronome



# Ambiguità referenziale: anafora

---

“La procedura convertirà l'immagine a 24 bit in un'immagine a 8 bit, quindi **la** mostrerà in una finestra dinamica”.

- Questa è un'anafora (accezione linguistica di anafora)

cercare i **pronomi**:

- Un requisito contiene un'ambiguità anaforica quando un elemento di una frase dipende per il suo riferimento da un altro elemento, **antecedente**, e non è chiaro a quale antecedente si riferisca

# Vaghezze

---

Il sistema software deve fornire un **appropriato** visualizzatore per i documenti memorizzati

- Appropriato è un termine vago

cercare aggettivi e avverbi:

- Un requisito contiene una vaghezza quando ci sono un **aggettivo qualificativo o un avverbio** non misurabile
- Abbondante, Vantaggioso, Accessibile, Bello, Veloce, Male, Brevemente, E così via

# Verbi deboli e forme passive

---

Il sistema deve includere allarmi sonori, che **possono essere attivati** quando viene rilevata una violazione della sicurezza da un contatto magnetico

Cercare verbi deboli: potere

Cercare forme passive senza complemento di agente



Non usare negazioni, soprattutto doppie negazioni

---

Gli utenti con tre o più account **non** devono essere migrati

- → Il sistema deve migrare solo gli utenti con meno di tre account

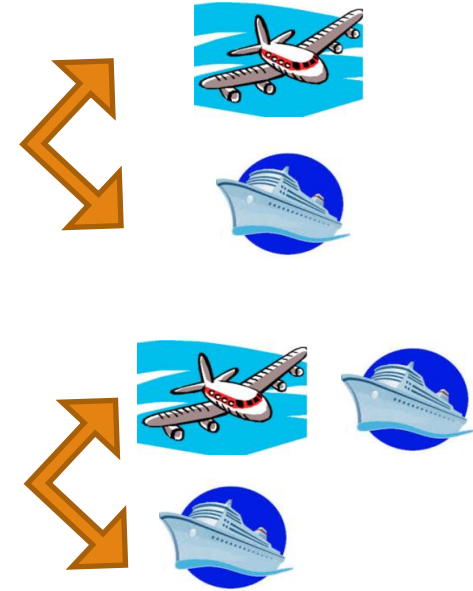
I premi e i punti **non** saranno visibili agli utenti che **non** partecipano al concorso

- → I premi e i punti saranno visibili solo agli utenti che partecipano al concorso

# Ambiguità: Disgiunzioni

Viaggerò in aereo **o** in traghetto

- Viaggerò in aereo xor viaggerò (mutualmente esclusivi)
- Viaggerò in aereo e/o in traghetto (or debole)



---

# A caccia di difetti nei requisiti





# Convalida dei requisiti: Tecniche

---

## Validazione di un documento già strutturato

- Deskcheck
  - Walkthrough
    - lettura “sequenziale” dei documenti
  - Ispezione
    - lettura “strutturata” dei documenti
    - Esempi:
      - tecnica del lemmario
      - Ricerca di rimozioni, distorsioni, generalizzazioni (Noam Chomsky 1975)
- Uso di strumenti di analisi del linguaggio naturale
- Prototipi

# Convalida dei requisiti: esempio

---

1. Il sistema software deve fornire un **appropriato** visualizzatore per i documenti memorizzati
  - Appropriato è un termine vago che genera ambiguità
  - Per il resto sembra un requisito ben posto, in forma assertiva: Il sistema deve ...
2. L'utente deve essere in grado di effettuare ricerche sia sull'intero insieme di basi di dati che su **un loro sottoinsieme**
  - Un loro sottoinsieme è ambiguo: uno solo, e in caso quale? Tutti?
  - Per il resto sembra un requisito ben posto, in stile caso d'uso (con gli occhi dell'utente)
3. **Ad ogni nuovo ordine deve essere associato un identificatore unico (Order\_ID)**
  - Requisito di basso livello: introduce vincoli implementativi: è necessario ?

# Convalida dei requisiti: Tecnica del lemmario

---

Termini del glossario, con puntatori ai requisiti che li nominano

Facilita la ricerca di

- Inconsistenze
- Sinonimi
- Omonimi
- Ridondanze

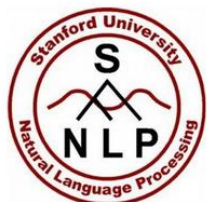
# Strumenti di elaborazione del linguaggio naturale

---

Strumenti di analisi lessicale e sintattica del linguaggio naturale

(NLP=Natural Language Processing)

spaCy flair 



NLTK  
Natural Language Toolkit  
Reference Guide

AllenNLP

NLP ARCHITECT

# Strumenti di elaborazione del linguaggio naturale per analizzare i requisiti

Salvataggio automatico eshop.docx - Salvato in questo PC monica arrabito MA

File Home Inserisci Progettazione Layout Riferimenti Lettere Revisione Visualizza Guida QVscribe Cerca

QVscribe QVscribe for Individuals

User: QRA Default(1)

Remove All Marks Toggle Color Background QVscribe Help EARS Export Requirements

R1 The system shall enable user to enter the search...  
R2 The system shall display all the matching products...  
R3 The system possibly notifies with a pop-up the...  
R4 The system shall allow a user to create his profile...  
R5 The system shall authenticate user credentials...  
R6 The system shall display the list of active and/c...  
R7 The system shall maintain customer email info...  
R8 The system shall send an order confirmation to...  
R9 The system shall allow an user to add and rem...  
R10 The system shall display various shipping met...  
R11 The order shall be shipped to the client address...  
R12 The system shall enable the user to select the...  
R13 The system may display the current tracking i...  
R14 The system shall display the available payment...  
R15 The system shall allow an user to select the...  
R16 After delivery, the system may enable the use...  
R17 In order to publish the feedback on the purch...  
R18 The "collect in-store" service excludes the tr...

QVscribe

Quality Analysis Consistency Similarity

Last Analyzed: settembre 30, 2019 at 4:53 Issues: All Viewing 18 of 18

Overall Quality Score 6 Quality Warnings

Requirement Score

R9 The system shall allow an user to add and remove products in the sho...  
R10 The system shall display various shipping methods.  
R11 The order shall be shipped to the client address or, if the "collect in-...  
R12 The system shall enable the user to select the shipping method.  
R13 The system shall display the current tracking information about the o...  
R14 The system shall display the available payment methods.  
R15 The system shall allow an user to select the payment method for or...  
R16 After delivery, the system may enable the users to enter their reviews...  
R17 In order to publish the feedback on the purchase ratings.  
R18 The "collect in-store" service excludes the tr...

Quality Warnings

Written in passive voice

Problem Phrases

shall be shipped

enable  
may  
their  
After

Pagina 1 di 1 258 parole

Analizzano la lista di requisiti in cerca di termini ambigui, vaghi o sottospecificati

R16 After delivery, the system may enable the users to enter their reviews...

## Quality Score

- Has no imperatives
- Contains vague words
- Contains optional escape clauses
- Contains cross-referencing pronouns
- Contains non-specific temporal words

## Problem Phrases

enable  
may  
their  
After

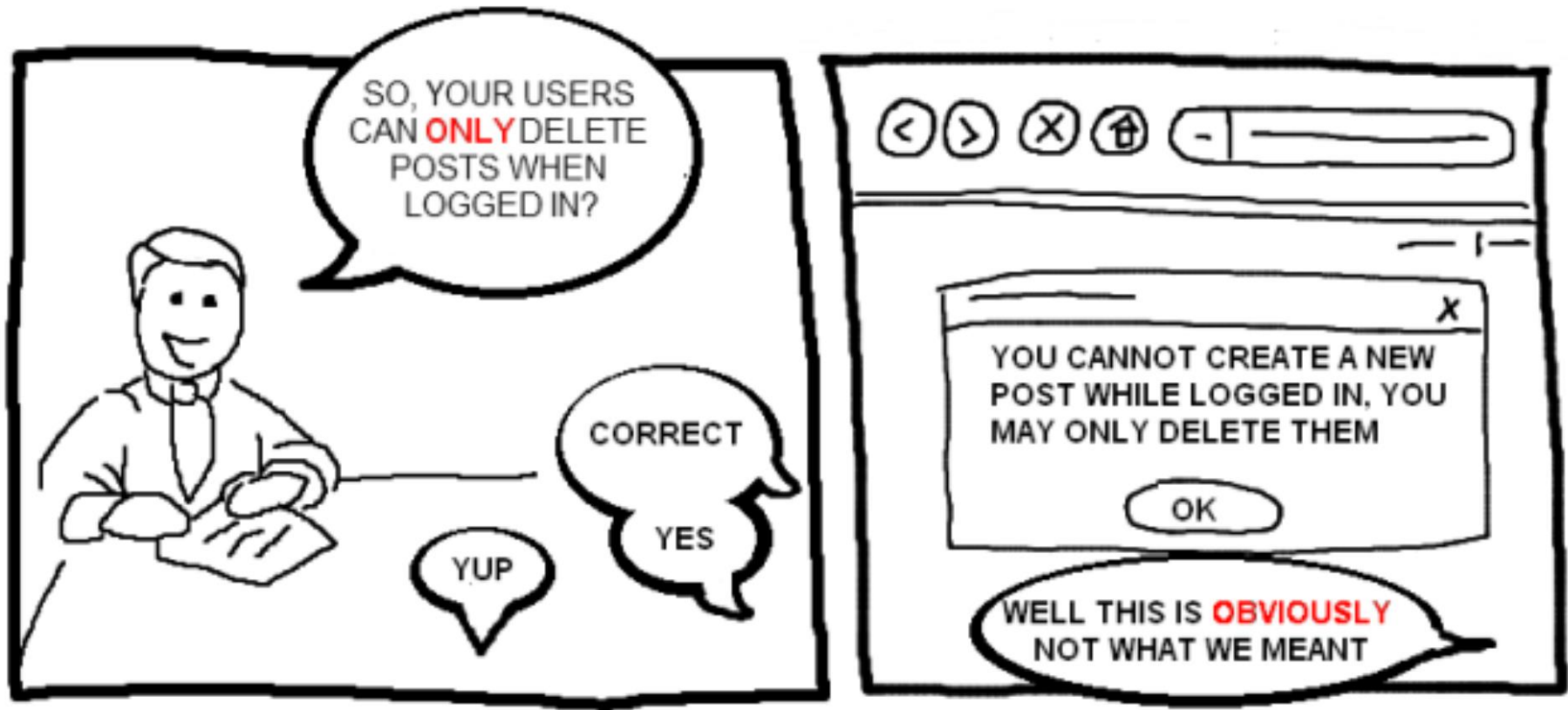
# Strumenti di elaborazione del linguaggio naturale per analizzare i requisiti

---

## Alcuni NLP tools:

- QuARs (ISTI-CNR pisa) → free
- TIGER-PRO (accademico) → free
- Requirements Scout developed by Qualicen GmbH
- QVscribe by QRA
- RAT by Reuse Company
- RQA by IBM

# Convalida: prototipi



# E dopo che ho trovato i difetti?

---

Ambiguità, incompletezze etc vanno **sempre** risolte incontrando il **committente**

Non inventate nulla da soli!



# Esercizio: convalidare e migliorare (1/2)

---

1. Il sistema deve gestire tutti i registratori di cassa del negozio (non più di 20)
2. Il sistema deve stampare la sintesi degli incassi della giornata
3. A fine giornata devono essere elencati gli articoli da reintegrare in base alle vendite

# Esercizio: convalidare e migliorare (2/2)

---

## Dominio

- Un negozio contiene al massimo 20 registratori di cassa
- Sintesi degli incassi = elenco dei totali incassati da ogni registratore di cassa a fine giornata

## Requisiti

1. Il sistema deve gestire tutti i registratori di cassa del negozio
2. Il sistema deve stampare la sintesi degli incassi della giornata
3. Il sistema mantiene un valore di scorta minima per ogni prodotto
4. A fine giornata il sistema deve elencare gli articoli sotto scorta minima

# Analisi dei requisiti in 5 passi

---

1. Acquisizione
2. Elaborazione
3. Convalida
4. **Negoziazione**
5. Gestione

# Negoziazione

---

- Nella fare di negoziazione si assegnano delle priorità ai requisiti in base a:
  - Esigenze del committente
  - Analisi costi, tempi di produzione
- Le priorità si usano per decidere se alcuni requisiti possono essere
  - Cancellati o
  - Sviluppate in un secondo tempo (ciclo di vita incrementale)

# Negoziazione: MoSCoW

---

MoSCoW: tecnica per dare priorità ai requisiti, che sono divisi nelle classi:

- **Must have (Requisiti obbligatori)**
  - Irrinunciabili per il cliente
- **Should have (Requisiti desiderabili)**
  - Non necessari, ma utili
- **Could have (Requisiti opzionali)**
  - Relativamente utili, da realizzare se c'è tempo
- **Want to have (Requisiti postponibili)**
  - Contrattabili per successive versioni

# Analisi dei requisiti in 5 passi

---

1. Acquisizione
2. Elaborazione
3. Convalida
4. Negoziazione
5. Gestione

# Gestione: identificazione

---

Assegnare a ogni requisito un identificatore unico

- Numero sequenziale (1,2,3,4....)
- Numero basato sulla struttura del documento (2.4.7)
- Coppia <CATEGORIA, NUMERO>

# Gestione: attributi

---

Assegnare a ogni requisito attributi che ne indicano:

Stato

- Proposto, approvato, rifiutato, incorporato

Priorità

- Importanza, tipo MoSCoW

Sforzo in gg/uomo

Rischio

- Valutazione della fattibilità tecnica

Stabilità

Versione destinazione

- Per lo sviluppo incrementale



# Gestione: tracciabilità

---

- La tracciabilità è la capacità di descrivere e seguire la vita di un requisito del processo di sviluppo
- Mappa tra requisiti e
  - componenti del sistema
  - codice
  - test
- Strumenti CASE per la gestione dei requisiti

# Requisiti e aspetti contrattuali

---

- Il documento dei requisiti normalmente precede la stipula del contratto, e ne è parte integrante
- Se alla stipula del contratto non è possibile avere un documento definitivo, è opportuno prevedere di rinegoziare il contratto.
- In caso di gara il documento dei requisiti può essere prodotto dal committente come parte di capitolato tecnico.

# Last but not least

---

Casi d'uso

User Stories

# Acquisire e descrivere i requisiti (funzionali): Casi d'uso

---

- I casi d'uso sono un altro modo per acquisire i requisiti
- Il caso d'uso è un modo in cui un utente può usare il prodotto
- Si prospetta al committente insieme al risultato atteso e si aspettano i commenti
- I casi d'uso devono includere non solo la sequenza di eventi corretta ma anche comportamenti inattesi: **le eccezioni.**

# Acquisire e descrivere i requisiti: User stories

---

Tecnica usata nei processi Agile

I requisiti sono descritti come:

Nel mio ruolo di ....., ho bisogno che il sistema ....., al fine di .....

As a <user role>

I want <goal>

so that <benefit>.

# Esempi di User Stories

---

- In qualità di cliente voglio cancellare la mia prenotazione in hotel, per poter avere un rimborso
- In quanto appassionata di fitness, voglio prenotare rapidamente le lezioni di fitness dal mio smartphone, in modo da poter includere più facilmente l'esercizio fisico nella mia vita.

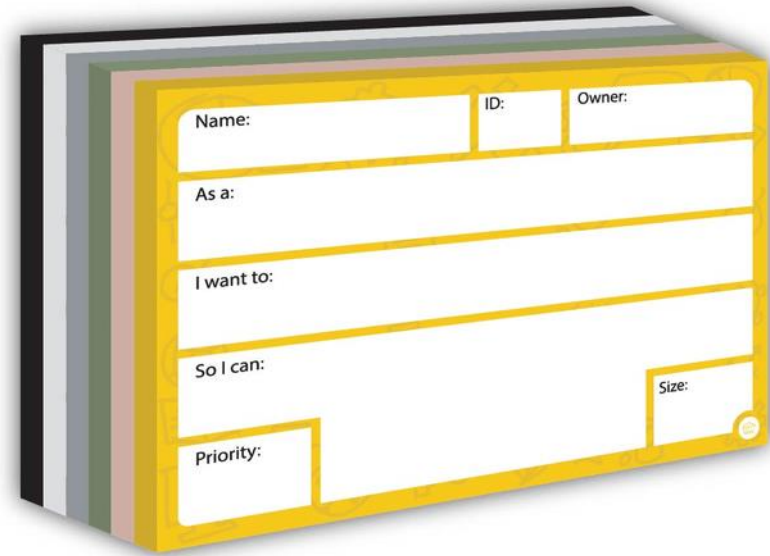
# User Story cards

---

- La user story viene spesso scritta su una scheda di carta
- Il formato piccolo favorisce la sintesi

La carta è

- facile da usare
- rende possibile raggruppare le card al muro o sul tavolo per poter valutare la consistenza, completezza e le connessioni tra diverse story
- favorisce la visibilità



# User Stories

---

- Scalabilità
  - scritte su piccole schede fisiche sono difficili da mantenere, difficili da scalare su grandi progetti e problematiche per i team distribuiti geograficamente.
- Vaghe, informali e incomplete: sono spunti di conversazione, non sono adatte a raggiungere accordi formali o a scrivere contratti legali.
  - informali, si prestano a molte interpretazioni.
  - brevi, non riportano tutti i dettagli necessari.
- Raramente includono dettagli sulle prestazioni o sui requisiti non funzionali.



# Riepilogo

---

- L'attività di analisi
  - 5 passi
- Descrizione del dominio
  - Modello del dominio
- Descrizione dei requisiti
  - Documento dei requisiti
- Requisiti funzionali e non funzionali
- Importanza di scrivere bene i requisiti

# Homework

---

- Scrivete il documento dei requisiti del sistema *La Piscina* (trovare il testo online) eliminando ogni ambiguità o altro difetto evidente e usando un formato standard.
  - In caso di dubbio, evidenziatelo con un (?), questo vi serve per fare domande al committente, non decidete voi al posto suo
- Riscrivere 3 tre requisiti anche usando il modello delle user stories.