

03. Analisi dei requisiti

IS 2024-2025



Laura Semini, Jacopo Soldani

Corso di Laurea in Informatica

Dipartimento di Informatica, Università of Pisa

STUDIO DI FATTIBILITÀ (PRIMA DEI REQUISITI)

Fase **preliminare** per stabilire l'**opportunità** (o meno) di realizzare un sistema software

Si basa su:

- **Descrizione sommaria** del sistema software e delle necessità dei suoi utenti
- Analisi di **mercato**
 - Mercato attuale vs. mercato futuro
 - Costo di produzione
 - Redditività (attesa) dell'investimento
- Analisi (tecnica) di **realizzabilità**
 - Strumenti per la realizzazione (software, librerie, ecc.)
 - Soluzioni algoritmiche e architetturali
 - Hardware
 - Processo (prototipazione, progetti esplorativi, ricerca, ecc.)



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 le scarpe online?

- 1999, Nick Swinmurn voleva fare un negozio online di scarpe
- È 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

Risultato?

- Comprato da Amazon nel 2009 per 880 milioni di dollari!



ANALISI DEI REQUISITI

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**

Studiare e definire il problema da risolvere per

- **capire cosa** deve essere realizzato,
- **documentare cosa** deve essere realizzato e
- consentire **negoziazioni** tra committente e fornitore



COSA fare
NON COME farlo

COSA FARE, NON COME

- Identificare i confini del sistema software e il modo in cui deve interagire con il suo ambiente
- Definire e analizzare i requisiti di qualità
- Negoziare le priorità con gli stakeholder
- Elaborare i requisiti del sistema 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
- Individuare e risolvere eventuali difetti nei requisiti (p.e. ambiguità o conflitti)

PERCHÉ È IMPORTANTE?

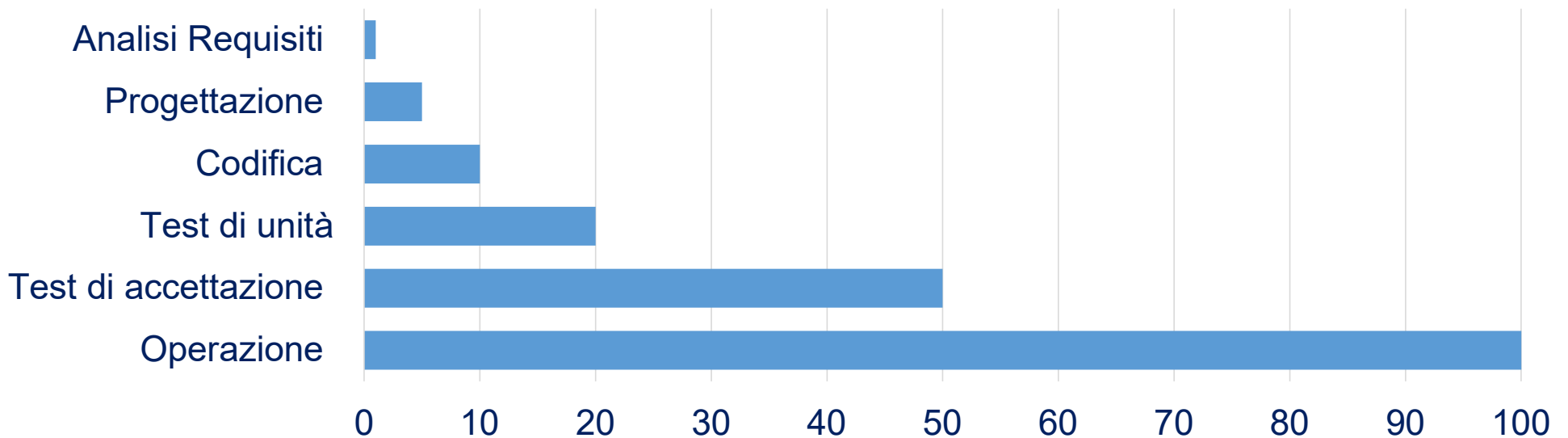
Le cause di abbandono secondo lo Standish Group // *repetita iuvant*

Project Challenged Factors	% of Responses
1. Lack of user input	12.8
2. Incomplete requirements & specifications	12.3
3. Changing requirements & specifications	11.8
4. Lack of executive support	7.5
5. Technology incompetence	7.0
6. Lack of resources	6.4
7. Unrealistic expectations	5.9
8. Unclear objectives	5.3
9. Unrealistic time frames	4.3
10. New technologies	3.7
Other	23

PERCHÉ È IMPORTANTE? (CONT.)

«56% of errors in installed systems were due to poor communication between user and analyst in defining requirements»¹

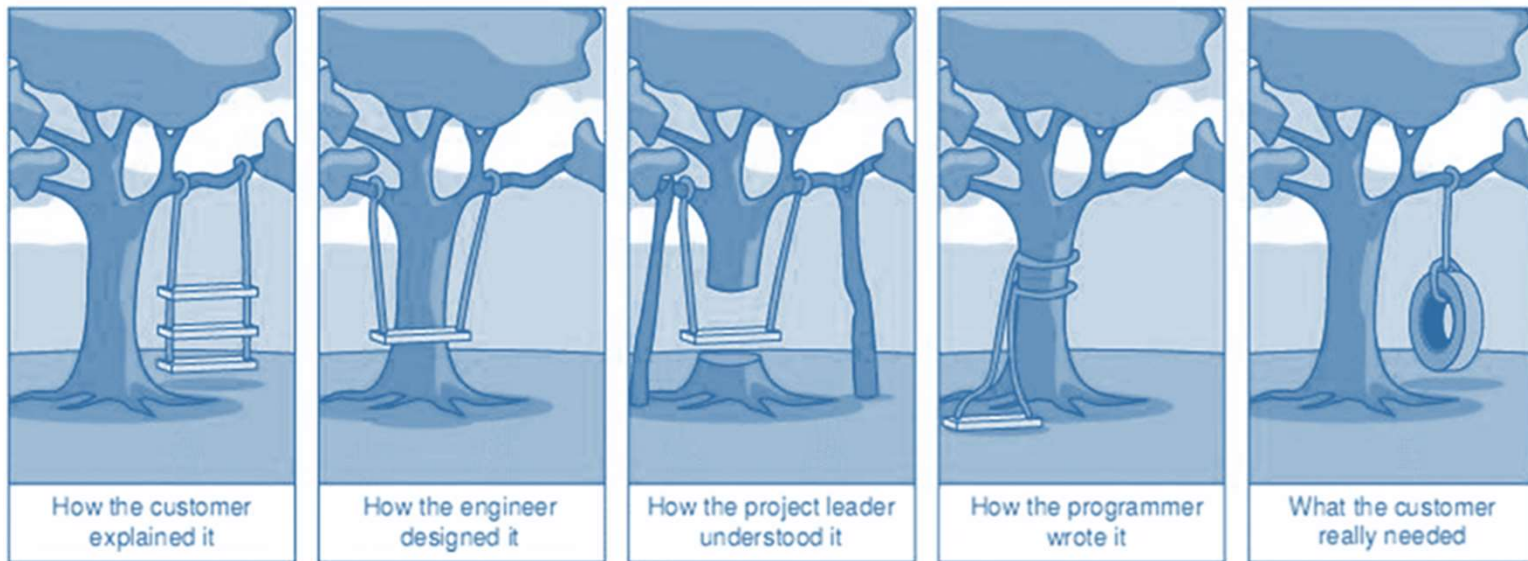
Ma **quanto costa** correggere un errore nell'analisi dei requisiti?²



1. V. Goodrich, L. Olfman. **An Experimental Evaluation of Task and Methodology Variables for Requirements Definition Phase Success**. 23rd Annual Hawaii International Conference on System Sciences, 201-209. IEEE, January 1990.

2. B.W. Bohem. **Software Engineering Economics**. Prentice Hall, 1981.

PERCHÉ È IMPORTANTE? (CONT.)



<https://www.cms-garden.org/de/file/treeswing-illustration>

PRODOTTO DELL'ANALISI DEI REQUISITI



Documento e/o modello che descrive

- il **dominio** del sistema da realizzare
- i **requisiti** del sistema (per l'utente)

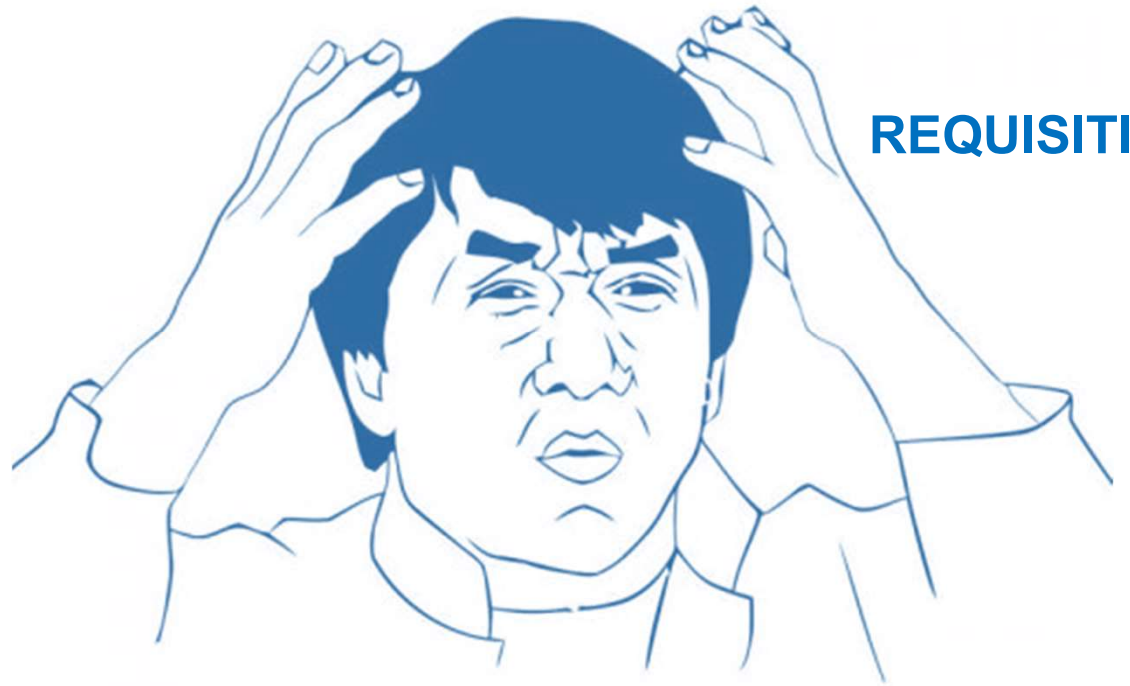
DOMINIO?



Opzionalmente, anche

- manuale utente
- casi di test

REQUISITI?



DOMINIO

Per acquisire conoscenza e definire il dominio



si costruisce un glossario

- collezione di definizioni di termini rilevanti in un dominio specifico
- il team di analisti lo costruisce mentre studia il dominio
- si arricchisce via via che si incontrano nuovi termini
- può essere riutilizzato in progetti successivi nello stesso dominio

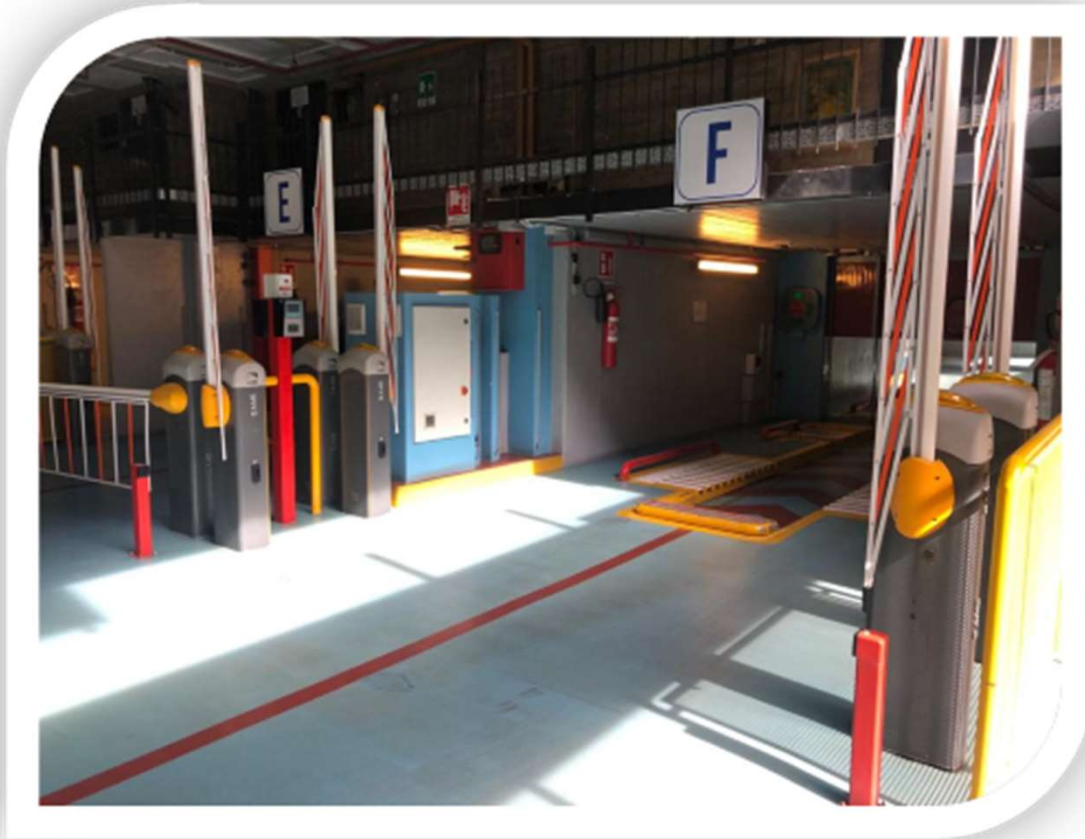


si definiscono due modelli

- modello statico
- modello dinamico

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

ESEMPIO: HOUSE OF CARS



ESEMPIO: HOUSE OF CARS (CONT.)

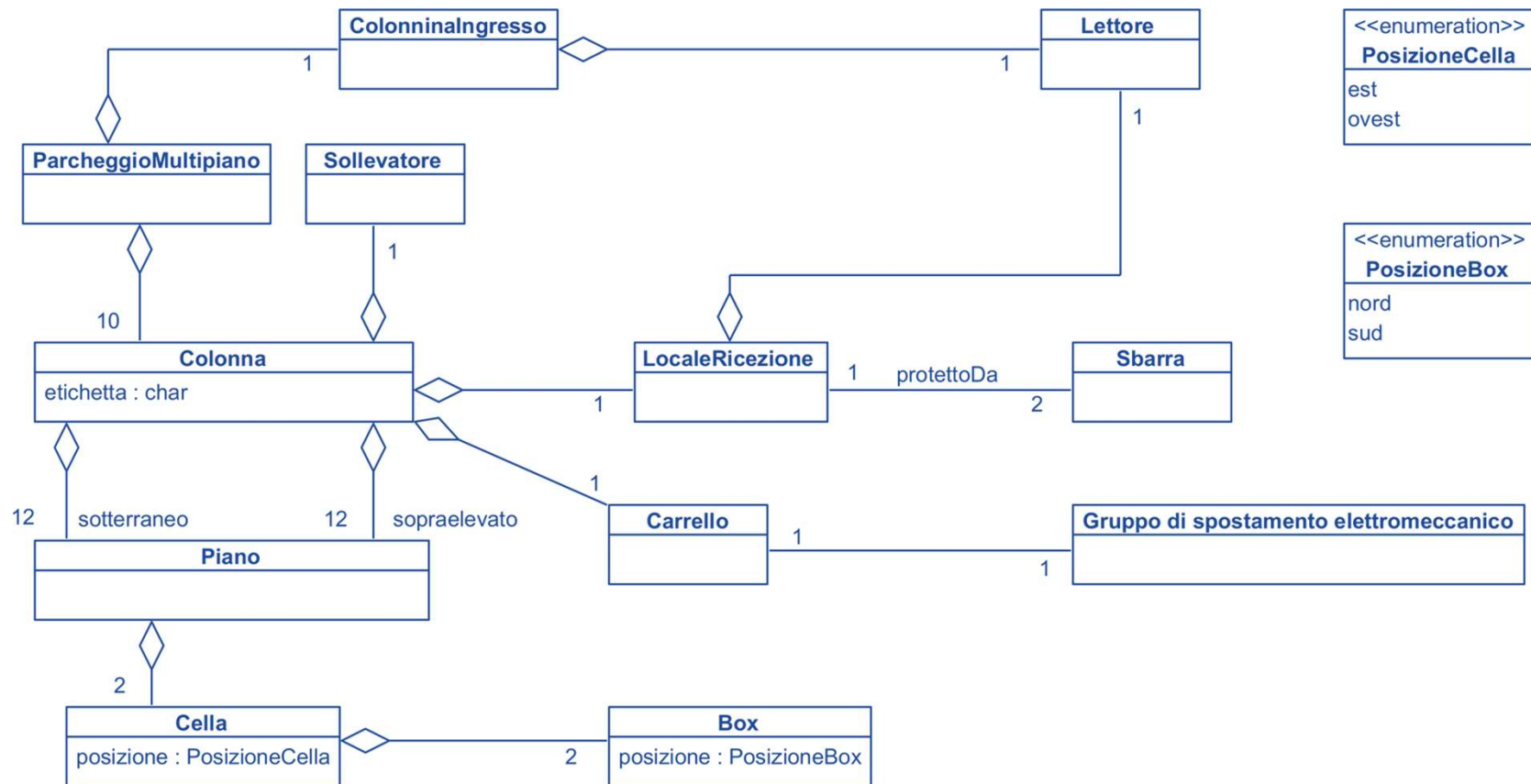
House of Cars è un parcheggio verticale multipiano, formato da 10 colonne e 24 piani per colonna, 12 sotto al livello strada e 12 sopra

Glossario

- **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; è 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
- ...

ESEMPIO: HOUSE OF CARS (CONT.)

Modello **statico** (usando UML)



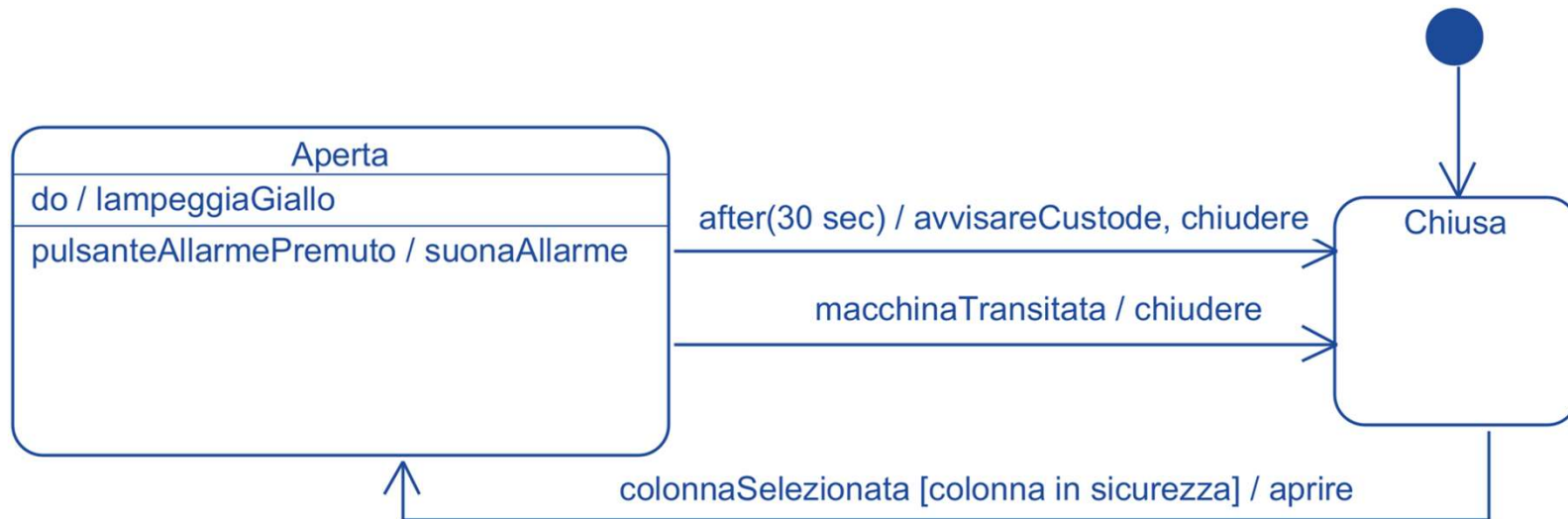
ESEMPIO: HOUSE OF CARS (CONT.)

Modello **dinamico** (descrizione testuale):

- 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
- ...

ESEMPIO: HOUSE OF CARS (CONT.)

Modello **dinamico** (usando UML):



Fino a qui giovedì 19

DOMINIO: ISTRUZIONI PER L'USO

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



E I REQUISITI?

Software **requirements** are **capabilities** that the system **must deliver**
(dal libro di testo)

Secondo il glossario IEEE,¹ un **requisito** è

- Una condizione (o capacità) necessaria a un utente per risolvere un problema o raggiungere un obiettivo
- Una condizione (capacità) che deve essere soddisfatta/posseduta da un sistema per soddisfare un contratto, uno standard, una specifica, o altri documenti formali

In altre parole, si tratta di una proprietà che deve essere garantita dal sistema per soddisfare una necessità di un utente (funzionalità, qualità, ecc)

1. <https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=159342>

ESEMPIO: HOUSE OF CARS (CONT.)

Esempi di requisiti

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

REQUISITI FUNZIONALI // CATEGORIE DI REQ.

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, e
- come si comporta in situazioni particolari

Esempi

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

Tradizionalmente, i requisiti funzionali sono quelli a cui è dato maggior valore



REQUISITI NON-FUNZIONALI // CATEGORIE DI REQ.

Descrivono le **proprietà** del sistema software

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

REQUISITI NON-FUNZIONALI // CATEGORIE DI REQ.

Esempi di requisiti non-funzionali

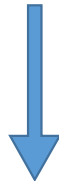
- Il tempo di risposta del sistema all'inserimento della password utente deve essere inferiore a 10 sec
- I documenti di progetto (deliverable) devono essere conformi allo standard XYZ-ABC-12345
- Il sistema software non deve rilasciare ai suoi operatori alcuna informazione personale relativa ai clienti,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

REQUISITI FUNZIONALI E NON-FUNZIONALI

Meglio 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.

Requisiti funzionali?



Requisiti non-funzionali?

FUN FACT

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, Wired 10/01/2022)

DESCRIVERE I REQUISITI DI UN SISTEMA

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



Descrizione basata su **linguaggio naturale**

- **Glossario**
- **Specifica** dei requisiti (funzionali e non-funzionali)



Descrizione basata su **linguaggi grafici**

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

Sono utilizzati anche linguaggi e metodi formali

che vedrete alla magistrale
(come logiche temporali, abstract state machines, o statecharts)

o che avete visto al primo anno
(come logica del primo ordine e triple di Hoare)

IN QUESTA LEZIONE



Descrizione basata su **linguaggio naturale**

- **Glossario**
- **Specifica** dei requisiti (funzionali e non-funzionali)



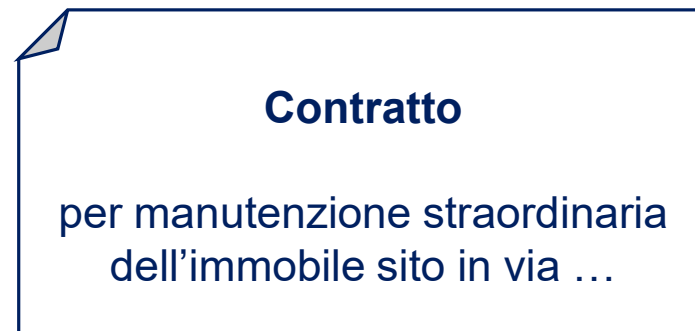
KEEP
CALM

AND START
FROM NATURAL
LANGUAGE DESCRIPTIONS

DOCUMENTO DEI REQUISITI

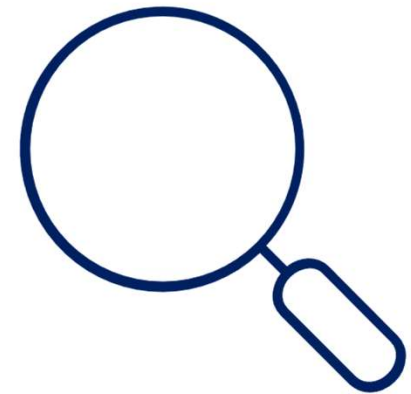
Elenca i requisiti del sistema da realizzare

- Specifica **cosa deve fare** e **quali vincoli** deve soddisfare
- È un **contratto** tra lo sviluppatore e il committente
- Specifica (in genere) una **scadenza** per la consegna del prodotto



ANALISI DEI REQUISITI IN 5 PASSI

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



ACQUISIZIONE MEDIANTE 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

(Inoltre è difficile condurre una buona intervista)

Come analista, ho
bisogno di sapere:
cosa vuoi?



ACQUISIZIONE MEDIANTE INTERVISTE (CONT.) 😊

Come analista, ho
bisogno di sapere:
cosa vuoi?



Voglio che tu
progetti il software
per me



Ma cosa vuoi farci
con il software che
mi chiedi?



Non so rispondere,
se prima non mi dici
cosa può fare



Bene, posso
progettare software
in grado fare tutto



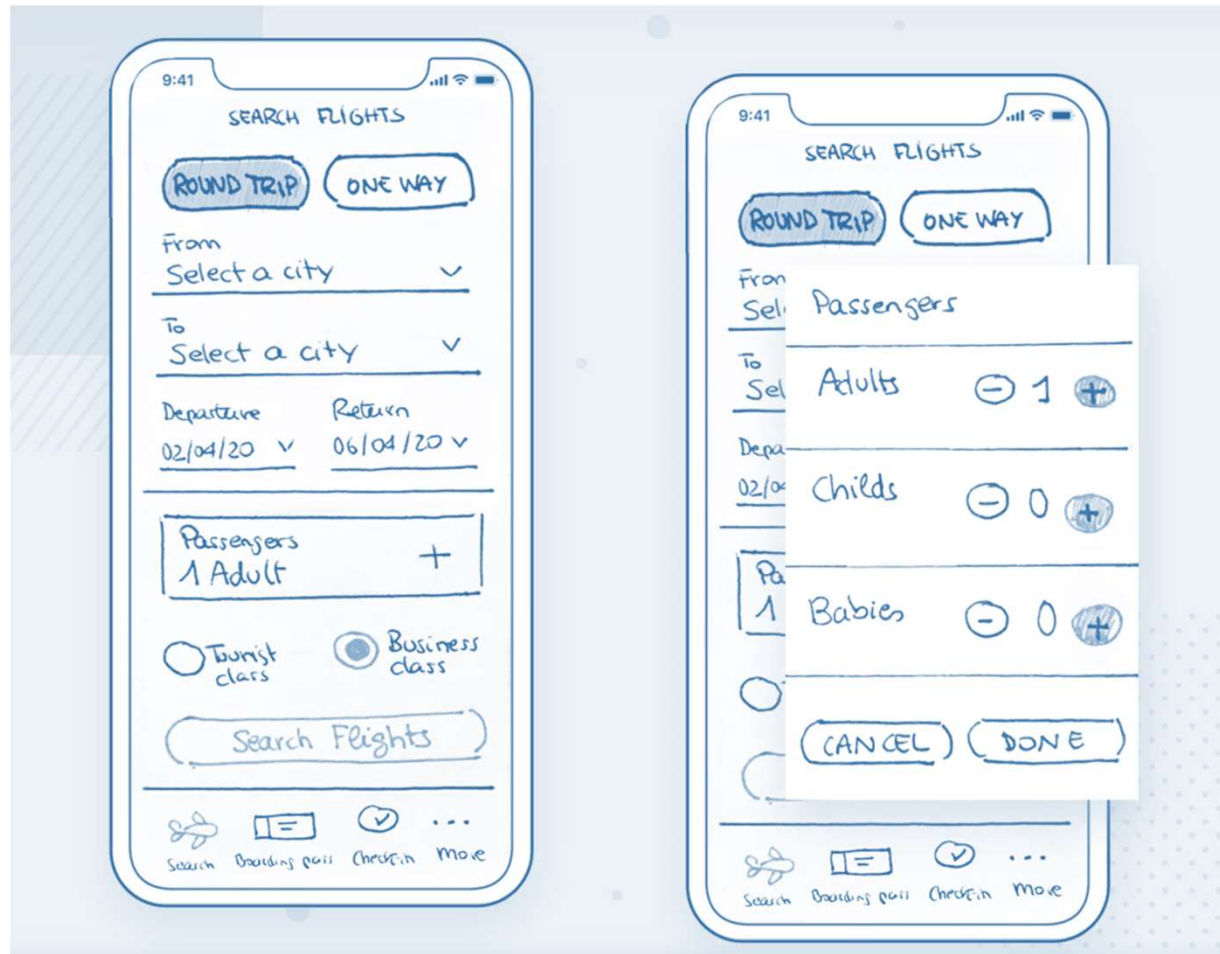
Puoi farne uno che
sia in grado di dirti i
miei requisiti?



ACQUISIZIONE: ALTERNATIVE

- 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

Durante la fase di elaborazione

- I requisiti vengono espansi e raffinati
- Viene scritta la prima bozza del documento dei requisiti

STRUTTURA DEL DOCUMENTO DEI REQUISITI¹

1. **Introduzione**
perché il sistema è desiderabile e come si inquadra negli obiettivi più generali del cliente
2. **Glossario**
termini e concetti tecnici usati
3. **Definizione dei requisiti funzionali**
servizi richiesti
4. **Definizione dei requisiti non-funzionali**
vincoli operativi del sistema e del processo di sviluppo
5. **Architettura**
strutturazione in sottosistemi (cui riferire i requisiti)

1. Proposta da Ian Sommerville e ispirata allo standard IEEE/ANSI 830-1993 su *requirements document*

STRUTTURA DEL DOCUMENTO DEI REQUISITI (CONT.)

6. **Specifica di requisiti del software e del sistema**
specifica dettagliata dei requisiti funzionali
7. **Modelli astratti del sistema**
modelli formali o semi-formali (ciascuno illustra un solo punto di vista: controllo, dati, funzioni)
8. **Evoluzione del sistema**
previsione di successivi cambiamenti (es. cambiamenti HW o di requisiti)
9. **Appendici**
individuazione ed eventuale descrizione della piattaforma hardware, requisiti di database, manuale utente, piani di test
10. **Indici**
lemmario (lista di termini, con puntatori ai requisiti che li usano)

REQUISITI BEN POSTI

Usare la forma assertiva

Il <sistema> deve <funzionalità/proprietà>

(non dovete scrivere un poema in versi 😊)

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

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 medesimo significato e termini uguali con differenti significati
- Presenza di **dettagli tecnici**
- **Ridondanza** // può esserci ridondanza, ma solo tra sezioni diverse

INCOMPLETEZZE

Esempio: *Il sistema deve essere in grado di gestire prenotazioni di tavoli*



Incompleto?

Incompleto! Non specifica in dettaglio come il sistema dovrebbe gestire le prenotazioni

- Come gli utenti effettueranno le prenotazioni (online, telefono, in loco)?
- Quali informazioni saranno richieste durante il processo di prenotazione (nome, numero di persone, data e ora, preferenze)?
- Come verranno confermate o annullate le prenotazioni?
- Si fa overbooking?
- ...

AMBIGUITÀ, VAGHEZZE

Il linguaggio naturale è per sua natura **impreciso**

*Il sistema deve mostrare **TUTTE** le strade **VICINO** alla destinazione*



Quale delle due?



AMBIGUITÀ: QUANTIFICATORI

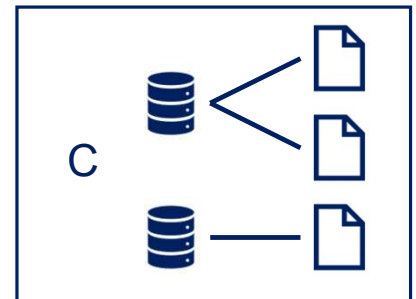
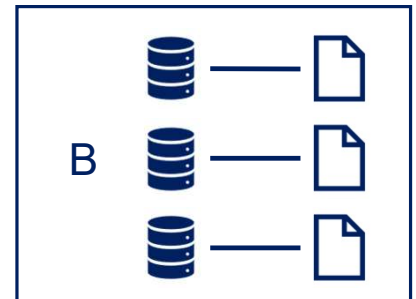
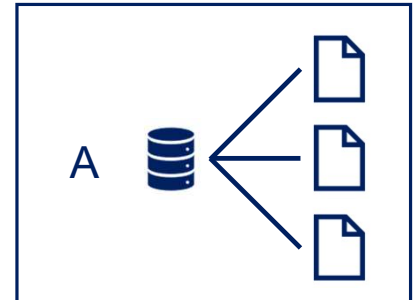
Esempio: **TUTTI** i file sono controllati da **UN** blocco di controllo

Possibili interpretazioni:

- A. Un unico blocco di controllo controlla tutti i file
- B. Ogni file ha un suo specifico blocco di controllo
- C. 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

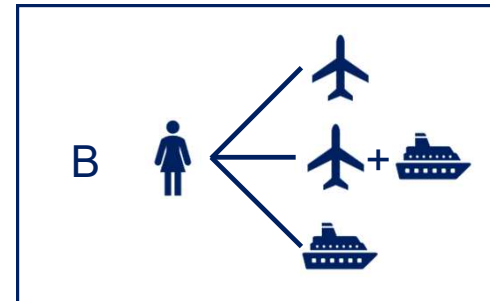
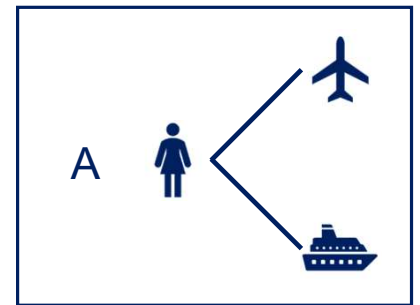


AMBIGUITÀ: DISGIUNZIONI

Esempio: *Viaggerò in aereo o in traghetto*

Possibili interpretazioni:

- A. Viaggerò in aereo XOR viaggerò in traghetto
(XOR => mutuamente esclusivi)
- B. Viaggerò in aereo OR viaggerò in traghetto
(OR debole, interpretabile come e/o)

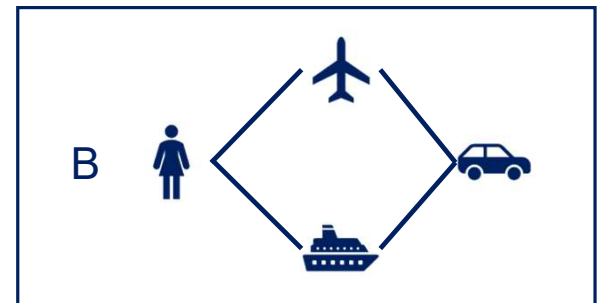
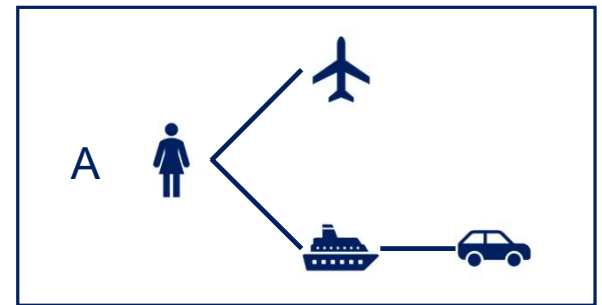


AMBIGUITÀ DI COORDINAZIONE

Esempio: *Viaggerò in aereo o in traghetto e in macchina*

Possibili interpretazioni:

- A. Viaggerò in aereo o viaggerò (in traghetto e in macchina)
(associazione su e)
- B. Viaggerò (in aereo o in traghetto) e in macchina
(associazione su o)



AMBIGUITÀ REFERENZIALE

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

Possibili interpretazioni:

A. Non sapere di avere bambini



B. Non sapere che ci sia un asilo



Cercare i **pronomi**: Un requisito contiene un'ambiguità referenziale quando non è chiaro a **cosa si riferisca un pronome**

AMBIGUITÀ REFERENZIALE: ANAFORA

Esempio di requisito con anafora: *La procedura convertirà l'immagine a 24 bit in un'immagine a 8 bit, quindi **la** mostrerà in una finestra dinamica*

Possibili interpretazioni:

- A. L'immagine a 24 bit
- B. L'immagine a 8 bit



Cercare i **pronomi**: Un requisito contiene un'ambiguità anaforica quando un elemento di una frase riferisce (e dipende da) un altro elemento, antecedente, e **non è chiaro a quale antecedente si riferisca**

AMBIGUITÀ: VAGHEZZE

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



Appropriato?

Cercare **aggettivi** e **avverbi**: Un requisito contiene una vaghezza quando ci sono un **aggettivo qualificativo** o un **avverbio non misurabile** (abbondante, vantaggioso, accessibile, veloce, male, brevemente, ecc.)

VERBI DEBOLI E FORME PASSIVE

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



Si attivano sempre?
Chi li attiva?

Cercare **verbi deboli**: ad esempio, potere

Cercare **forme passive** senza
complemento di agente



NON USARE (DOPPIE) NEGAZIONI

Esempio (negazione singola): *Gli utenti con tre o più account **non** devono essere migrati*



Senza negazione: *Il sistema deve migrare solo gli utenti con meno di tre account*

Esempio (con doppia negazione): *I premi e i punti **non** saranno visibili agli utenti che **non** partecipano al concorso*



Senza negazione: *I premi e i punti saranno visibili solo agli utenti che partecipano al concorso*

A CACCIA DI DIFETTI NEI REQUISITI



<https://www.modernanalyst.com/Resources/Articles/tabid/115/ID/2076/Ambiguity-Uncertainty-or-Both.aspx>

TECNICHE DI CONVALIDA DEI REQUISITI

Validazione di un documento già strutturato

- **Deskcheck**
 - **Walkthrough**: lettura «sequenziale» dei documenti
 - **Ispezione**: lettura «strutturata» dei documenti, ad esempio
 - Tecnica del lemmario
 - Ricerca di rimozioni, distorsioni, generalizzazione (N. Chomsky, 1975)
- Uso di strumenti di **analisi del linguaggio naturale**
- **Prototipi**

CONVALIDA DEI REQUISITI: ESEMPIO

*R1: 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 ...

*R2: 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 (dal punto di vista dell'utente)

*R3: Ad ogni nuovo ordine deve essere **associato un identificatore unico (Order_ID)***

- Requisito di basso livello che introduce vincoli implementativi: è necessario/utile all'utente?

CONVALIDA MEDIANTE TECNICA DEL LEMMARIO

Utilizzo di **termini del glossario** con **puntatori ai requisiti** che li nominano

Facilità la ricerca di

- Inconsistenze
- Sinonimi
- Omonimi
- Ridondanze



CONVALIDA MEDIANTE NLP¹

Utilizzo di strumenti di analisi lessicale e sintattica del linguaggio naturale



1. NLP = Natural Language Processing

CONVALIDA MEDIANTE NLP (CONT.)

The screenshot shows the QVscribe application window. On the left, a list of requirements (R1 to R18) is displayed. On the right, the 'Quality Analysis' panel is open, showing a detailed analysis of requirement R16: 'R16 After delivery, the system may enable the users to enter their reviews...'. The analysis includes a 'Quality Score' and 'Problem Phrases'.

Analizzano la lista di requisiti in cerca di termini

- ambigui,
- vaghi o
- sottospecificati

Quality Score	Problem Phrases
Has no imperatives	
Contains vague words	enable
Contains optional escape clauses	may
Contains cross-referencing pronouns	their
Contains non-specific temporal words	After

CONVALIDA MEDIANTE NLP (CONT.)

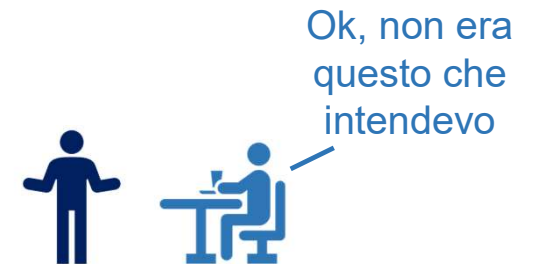
Alcuni strumenti NLP per l'analisi dei requisiti

- 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 MEDIANTE PROTOTIPI

Quindi gli utenti
possono **solo**
eliminare quando si
loggano?



E DOPO CHE HO TROVATO I DIFETTI?

Ambiguità, incompletezze, ecc. vanno
sempre risolte incontrando il **committente**

(meglio non inventarsi –interpretazioni di– requisiti da soli 😊)

ESERCIZIO: CONVALIDARE E MIGLIORARE

Requisiti:

- R1: Il sistema deve gestire tutti i registratori di cassa del negozio (non più di 20)
- R2: Il sistema deve stampare la sintesi degli incassi della giornata
- R3: A fine giornata devono essere elencati gli articoli da reintegrare in base alle vendite

ESERCIZIO: CONVALIDARE E MIGLIORARE (CONT.)

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:

- R1: Il sistema deve gestire tutti i registratori di cassa del negozio
- R2: A fine giornata il sistema deve stampare la sintesi degli incassi della giornata
- R3: Il sistema mantiene un valore di scorta minima per ogni prodotto
- R4: 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 e tempi di produzione

Le priorità si usano per decidere se alcuni requisiti possono essere

- cancellati o
- sviluppati in un secondo momento (ciclo di vita incrementale)



NEGOZIAZIONE: MOSCOW

MoSCoW – tecnica per dare priorità ai requisiti, suddividendoli in 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 differibili // contrattabili per successive versioni

ANALISI DEI REQUISITI IN 5 PASSI



1. Acquisizione



2. Elaborazione



3. Convalida



4. Negoziazione



5. Gestione

GESTIONE: IDENTIFICAZIONE DEI REQUISITI

Nella **gestione** dei requisiti è fondamentale

Assegnare ad ogni requisito un **identificatore unico**

L'identificatore può essere, ad esempio

- un numero sequenziale (1, 2, 3, ...)
- un numero basato sulla struttura del documento (2.4.7)
- una coppia <categoria,numero>

GESTIONE: UTILIZZO DI ATTRIBUTI

Assegnare a ogni requisito **attributi** che ne indicano

- **Stato** (proposto, approvato, rifiutato, incorporato)
- **Priorità** (importanza, tipo MoSCoW)
- «**Effort**» 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 CASE¹ per la gestione dei requisiti

1. CASE = Computer-Aided Software Engineering

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

ANALISI DEI REQUISITI IN 5 PASSI

✓ 1. Acquisizione

✓ 2. Elaborazione

✓ 3. Convalida

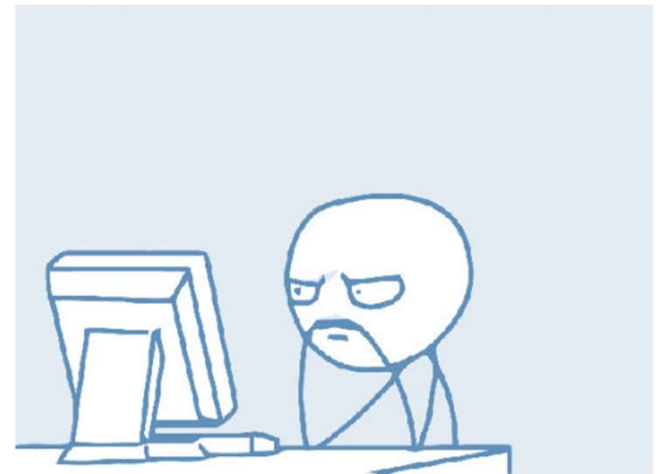
✓ 4. Negoziazione

✓ 5. Gestione

LAST, BUT NOT LEAST

Acquisire e descrivere i requisiti (funzionali) mediante

- **casi d'uso**
- **user stories**



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

- la sequenza di eventi corretta e
- eventuali comportamenti inattesi (le eccezioni)

USER STORIES

La tecnica delle **user stories** viene usata nei processi Agile

I requisiti sono descritti con un template predefinito

	As a <user role>
	I want <goal>
	so that <benefit>

Esempio: *In qualità di cliente voglio cancellare la mia prenotazione in hotel, per avere un rimborso*

USER STORY CARDS

La user story viene spesso scritta su una scheda di carta formato A6

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



The diagram illustrates a User Story Card template. It features a blue border and a white background. At the top, there are three small white boxes with blue borders, labeled 'Story ID:', 'Size:', and 'Prio:'. Below these, there are three larger white boxes with blue borders, labeled 'As a:', 'I want:', and 'So that:'. The boxes are arranged vertically, with the 'As a:' box at the top, followed by the 'I want:' box, and then the 'So that:' box at the bottom.

USER STORIES: LIMITAZIONI E UTILIZZO

- 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 in 5 passi
- Descrizione (e modello) del dominio
- Descrizione dei requisiti nel documento dei requisiti
- Requisiti funzionali e non funzionali
- Importanza di scrivere bene i requisiti

HOMEWORK

- Scrivere il documento dei requisiti del sistema *Fino a qui tutto bene* (trovate il testo online) eliminando ogni ambiguità o altro difetto evidente e usando un formato standard
- In caso di dubbio, evidenziarlo con un (?),
// Utilizzare i (?) per fare domande al committente, invece di decidere al posto suo
- Riscrivere 3 tre requisiti anche usando il modello delle user stories.



RIFERIMENTI

Contenuti

- **Capitolo 4** di "Software Engineering" (G. C. Kung, 2023)

Approfondimenti

- A. Fantechi, S. Gnesi, L. Passaro and L. Semini. *Inconsistency Detection in Natural Language Requirements using ChatGPT: a Preliminary Evaluation*. 2023 IEEE 31st International Requirements Engineering Conference (RE), Hannover, Germany, 2023, pp. 335-340, doi: 10.1109/RE57278.2023.00045