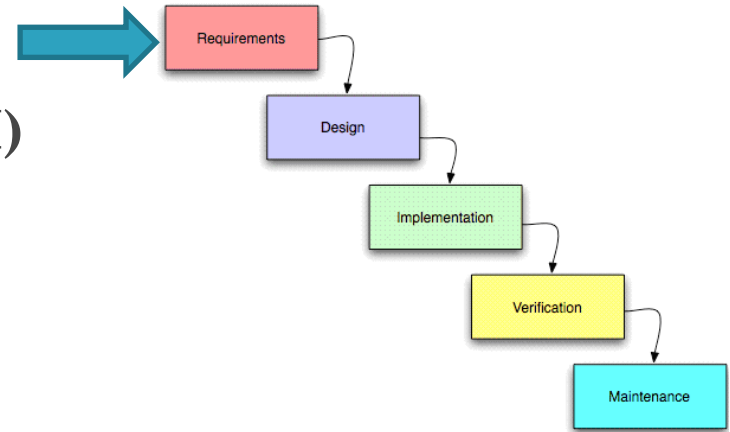




# **INGEGNERIA DEI REQUISITI**

**Ingegneria del Software 2023-2024**

# AGENDA (LEZIONE DI OGGI)



- Cosa è un **requisito SW**?
- Importanza dei requisiti
- Requisiti utente vs. di sistema
- Requisiti funzionali e non-funzionali
- Cosa è il Requirements Engineering?
  - Elicitation, Analisi, Stesura dei requisiti, Validazione
- Proprietà dei requisiti
- La figura dell'**analista software**
  - Esempio di intervista



# COSA È UN REQUISITO SW?

*Glossario terminologia SE*

## Requirement SW (requisito)

Def. IEEE Std 610.12

- difficile da definire precisamente ...
- **describe ‘qualcosa’ che il sistema dovrà fare (una funzionalità) o un vincolo a cui deve sottostare ...**
- **diversi livelli di astrazione:** può variare da una descrizione **astratta** ed **imprecisa** del sistema, fino a una descrizione **dettagliata** e **matematica** dello stesso

**Che cosa il sistema farà e non come!**

### Es. ATM

- Il sistema ATM dovrà controllare la validità della carta BANCOMAT inserita
- Il sistema ATM non dovrà erogare più di 250 euro per ogni carta, in un unico periodo di 24 ore



*La parte più complessa nella realizzazione di un sistema software consiste nel **decidere cosa realizzare***

*Nessuna parte del lavoro pregiudica maggiormente il risultato se viene eseguita in modo errato*

*Nessuna altra parte è più difficile da correggere successivamente*

**Fred Brooks**



**Turing Award 1999**

Brooks, F. 1975 (1995). *Mythical Man-month: essays on software engineering*, 20th anniversary edition. Addison-Wesley Professional.

*'Aggiungere personale ad un progetto SW in ritardo significa solo aumentare il ritardo'*

# IMPORTANZA DEI REQUISITI

- **Motivi di fallimento** di un progetto software  
(Standish survey, 1995)

	Motivi di fallimento	Percentuale
➡	Incompletezza dei requisiti	13.1%
➡	Mancato coinvolgimento degli utenti	12.4%
	Carenza di risorse	10.6%
➡	Aspettative non realistiche	9.9%
	Mancato supporto dal management	9.3%
➡	Cambiamento di requisiti e specifiche	8.7%
	Carenza di pianificazione	8.1%
	Cambiamento di obiettivi	7.5%
	Carenza di gestione del settore IT	6.2%
	Incompetenza tecnologica	4.3%
	Altro	9.9%

**= 44.1%**

*Nessuna parte del lavoro pregiudica maggiormente il risultato  
(=fallimento del progetto) se viene eseguita in modo errato*

# COSTO RIMOZIONE DIFETTI

## Costo relativo della rimozione dei difetti

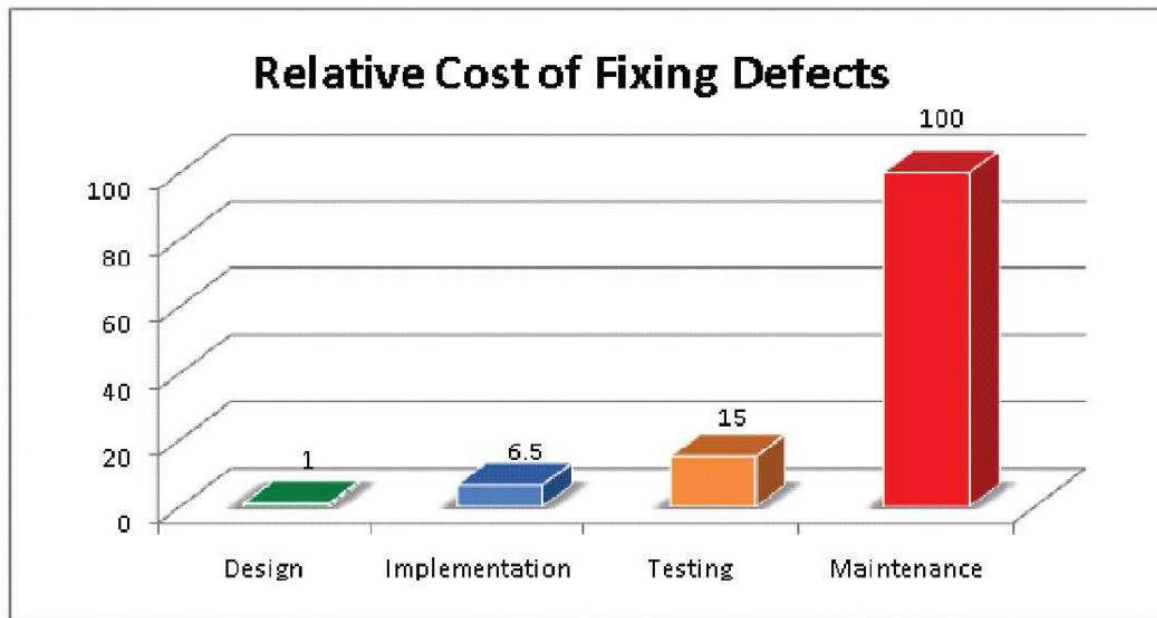


Figure 3: IBM System Science Institute Relative Cost of Fixing Defects

Se rimuovere un difetto (originato nei requisiti) durante il design costa **1**, rimuovere lo stesso difetto durante la fase di testing costa **15** ....

*I difetti dovrebbero essere scoperti il più presto possibile, ovvero a livello dei requisiti*

***Nessuna altra parte è più difficile da correggere successivamente ...***

# DIVERSI LIVELLI DI DETTAGLIO

Alan Mark Davis

## ○ Requisiti utente

- Descrizione in **linguaggio naturale** delle funzionalità che il sistema dovrà fornire e dei vincoli operativi

bisogni/necessità  
del cliente/utente  
(*user wish / needs*)

## ○ Requisiti di sistema

- Specificati mediante la stesura di un documento che **descrive in modo dettagliato** le funzionalità che il sistema dovrà fornire

cosa farà il  
sistema

# DIVERSI LIVELLI DI DETTAGLIO

Alan Mark Davis

## ○ Requisiti utente

- Descrizione in **linguaggio naturale** delle funzionalità che il sistema dovrà fornire e dei vincoli operativi
- **Sono scritti per (e con) il cliente**
  - Non servono conoscenze tecniche per comprenderli
  - Usati anche come forma di contratto

bisogni/necessità  
del cliente/utente  
(*user wish / needs*)

## ○ Requisiti di sistema

- Specificati mediante la stesura di un documento che **descrive in modo dettagliato** le funzionalità che il sistema dovrà fornire
- **Sono scritti per gli sviluppatori**
  - Sono usati come base per il progetto software
- Possono essere espressi facendo uso di notazioni differenti (anche formali)
  - Es. Linguaggio naturale strutturato (form/template), specifiche algebriche (linguaggio Z), UML

cosa farà il  
sistema



# DIVERSI LIVELLI DI DETTAGLIO

Alan Mark Davis

## ○ Requisiti utente

- Descrizione in **linguaggio naturale** delle funzionalità che il sistema dovrà fornire e dei vincoli operativi
- **Sono scritti per (e con) il cliente**
  - Non servono conoscenze tecniche per comprenderli
  - Usati anche come forma di contratto

bisogni/necessità  
del cliente/utente  
(*user wish / needs*)

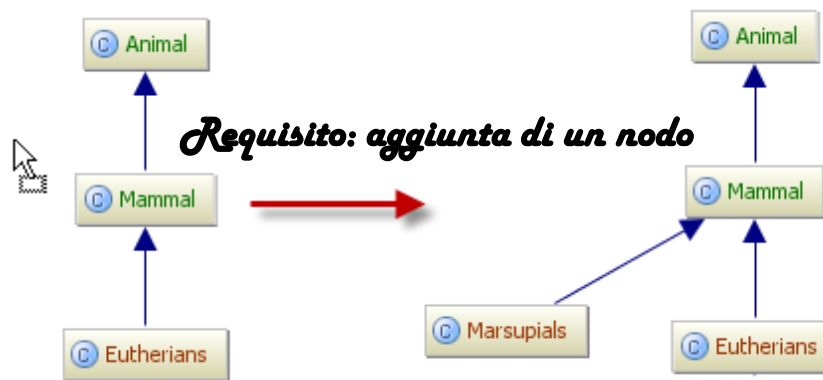
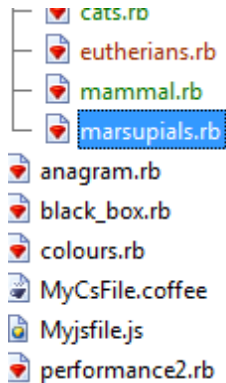
## ○ Requisiti di sistema

- Specificano il comportamento che **descrive in modo dettagliato** le funzionalità che il sistema dovrà fornire
- **Sono documenti questi devono però essere allineati**
- Possono essere espressi facendo uso di notazioni differenti (anche formali)
  - Es. Linguaggio naturale strutturato (form/template), specifiche algebriche (linguaggio Z), UML

cosa farà il  
sistema

# ESEMPIO DI REQUISITO UTENTE

Supponiamo di dover progettare un **editor grafico** che ci permette di produrre un **modello di design**



Rettangolo = classe SW  
freccia = relazione di specializzazione

## 3.5.1 Adding nodes to a design

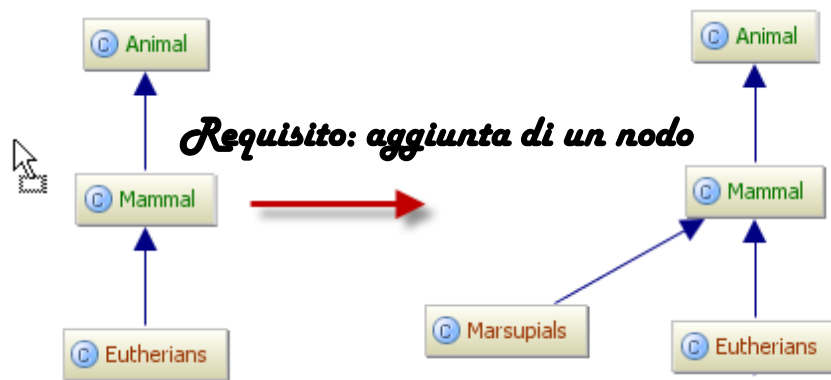
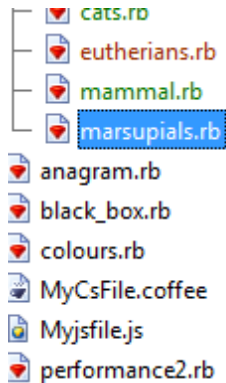
3.5.1.1 **The editor shall provide a facility for users to add nodes of a specified type to their design.**

3.5.1.2 The sequence of actions to add a node should be as follows:

1. The user should select the type of node to be added.
2. The user should move the cursor to the approximate node position in the diagram and indicate that the node symbol should be added at that point.
3. The user should then drag the node symbol to its final position.

# ESEMPIO DI REQUISITO UTENTE

Supponiamo di dover progettare un **editor grafico** che ci permette di produrre un **modello di design**



Rettangolo = classe SW  
freccia = relazione di specializzazione

## 3.5.1 Adding nodes to a design

3.5.1.1 The editor shall provide a facility for users to add nodes of a specified type to their design.

3.5.1.2 The sequence of actions to add a node should be as follows:

# Scritta per il Cliente/Utente!

position in the diagram and indicate that the node symbol should be added at that point.

3. The user should then drag the node symbol to its final position.

# ESEMPIO DI REQUISITO DI SISTEMA

- basato su *form* in linguaggio naturale

Form/Template



*ECLIPSE/Workstation/Tools/DE/FS/3.5.1*

<b>Function</b>	Add node
<b>Description</b>	Adds a node to an existing design. The user selects the type of node, and its position. When added to the design, the node becomes the current selection. The user chooses the node position by moving the cursor to the area where the node is added.
<b>Inputs</b>	Node type, Node position, Design identifier.
<b>Source</b>	Node type and Node position are input by the user, Design identifier from the database.
<b>Outputs</b>	Design identifier.
<b>Destination</b>	The design database. The design is committed to the database on completion of the operation.
<b>Requires</b>	Design graph rooted at input design identifier.
<b>Pre-condition</b>	The design is open and displayed on the user's screen.
<b>Post-condition</b>	The design is unchanged apart from the addition of a node of the specified type at the given position.
<b>Side-effects</b>	None

**Definition:** ECLIPSE/Workstation/Tools/DE/RD/3.5.1

[Link a requisito utente](#)

# ESEMPIO DI REQUISITO DI SISTEMA

- basato su *form* in linguaggio naturale

Form/Template

*ECLIPSE/Workstation/Tools/DE/FS/3.5.1*

<b>Function</b>	Add node
<b>Description</b>	Adds a node to an existing design. The user selects the type of node, and its position. When added to the design, the node becomes the current selection. The user chooses the node position by moving the cursor to the area where the node is added.
<b>Inputs</b>	Node type, Node position, Design identifier.
<b>Source</b>	Node type and Node position are input by the user, Design identifier from the database.
<b>Outputs</b>	Design identifier.
<b>Destination</b>	The design database. The design is committed to the database on completion of the operation.
<b>Requires</b>	
<b>Pre-condition</b>	The design is open and displayed on the user's screen.
<b>Post-condition</b>	The design is unchanged apart from the addition of a node of the specified type at the given position.
<b>Side-effects</b>	None

**Scritta per lo sviluppatore!**

**Definition:** ECLIPSE/Workstation/Tools/DE/RD/3.5.1

**Link a requisito utente**

# DIVERSI MODI DI RAPPRESENTARE I REQUISITI

- **Non scriverli, partire subito con l'implementazione**

badilanti!!!  
(code and fix)



- Mediante linguaggi formali

- Linguaggio Z

- Notazioni Visuali

- UML

- Testo strutturato

- Form/template

- Testo non strutturato  
(linguaggio naturale)

- Use cases

- Prossima lezione!

- User stories

metodi plan-driven

metodi agili



# ESEMPIO COMPLETO DI REQUISITI (USER STORIES)



*Bisogni/necessità del cliente espressi in linguaggio naturale*

# REQUISITI FUNZIONALI VS. NON-FUNZIONALI

## Requisiti funzionali

descrivono le **funzionalità ed i servizi** che saranno forniti dal sistema

- ciò che il sistema dovrà fare!
- indipendenti dall'implementazione di una soluzione
  - no 'come' ma 'cosa'!

### Es. Sistema bancario:

- *il sistema dovrà permettere la consultazione del saldo*
- *il sistema dovrà convertire il saldo (espresso in euro) in dollari a richiesta del cliente*

## Requisiti non-funzionali

non sono collegati direttamente con le **funzionalità implementate** dal sistema, ma piuttosto alle modalità operative, di gestione, ...

- **definiscono vincoli sul sistema** e sullo sviluppo del sistema
- in generale riguardano la scelta di linguaggi, piattaforme, strumenti (tools), tecniche d'implementazione, ma anche: prestazioni, questioni etiche, ...

### Es. Sistema bancario:

- *i computer devono essere dei **PC IBM 5150***
- *la risposta ad un'interrogazione deve avvenire **entro tre secondi***
- *i documenti devono essere registrati in formato **PDF***



# ESEMPIO: BANCOMAT



*Si consideri un sistema bancomat:*

- *Il sistema deve mettere a disposizione le funzioni di **prelievo**, **saldo** e **estratto conto***
- *Il sistema deve essere disponibile a persone portatori di Handicap, deve garantire un tempo di risposta inferiore al minuto, e deve essere sviluppato su architettura X86 con sistema operativo compatibile con quello della Banca*
- *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? NON FUNZIONALI?

- *Si consideri un sistema bancomat. Il sistema deve mettere a disposizione le funzioni di prelievo, saldo, estratto conto. Il sistema deve essere disponibile a persone portatori di Handicap, deve garantire un tempo di risposta inferiore al minuto, e deve essere sviluppato su architettura X86 con sistema operativo compatibile con quello della Banca. 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. ....*



# ESEMPIO: BANCOMAT

- Si consideri un sistema bancomat. **Il sistema deve mettere a disposizione le funzioni di prelievo, saldo, estratto conto.** Il sistema deve essere disponibile a persone portatori di Handicap, deve garantire un tempo di risposta inferiore al minuto, e deve essere sviluppato su architettura X86 con sistema operativo compatibile con quello della Banca. **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**



# ESEMPIO: BANCOMAT

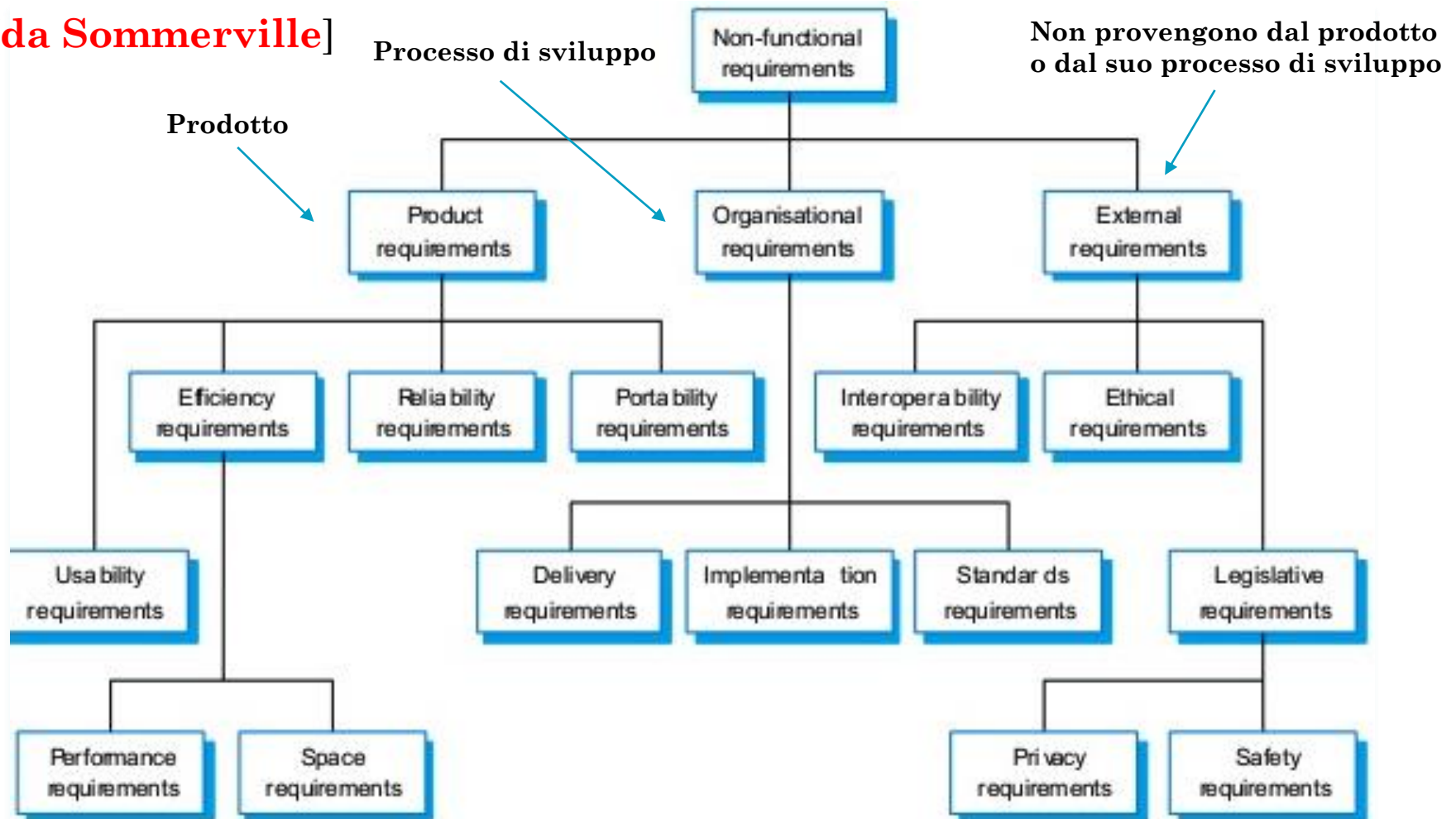
- *Si consideri un sistema bancomat. Il servizio deve mettere a disposizione le funzioni di prelievo, saldo, estratto conto. Il sistema deve essere disponibile a persone portatori di Handicap, deve garantire un tempo di risposta inferiore al minuto, e deve essere sviluppato su architettura X86 con sistema operativo compatibile con quello della Banca. 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 non  
funzionali*



# REQUISITI NON FUNZIONALI

[da Sommerville]

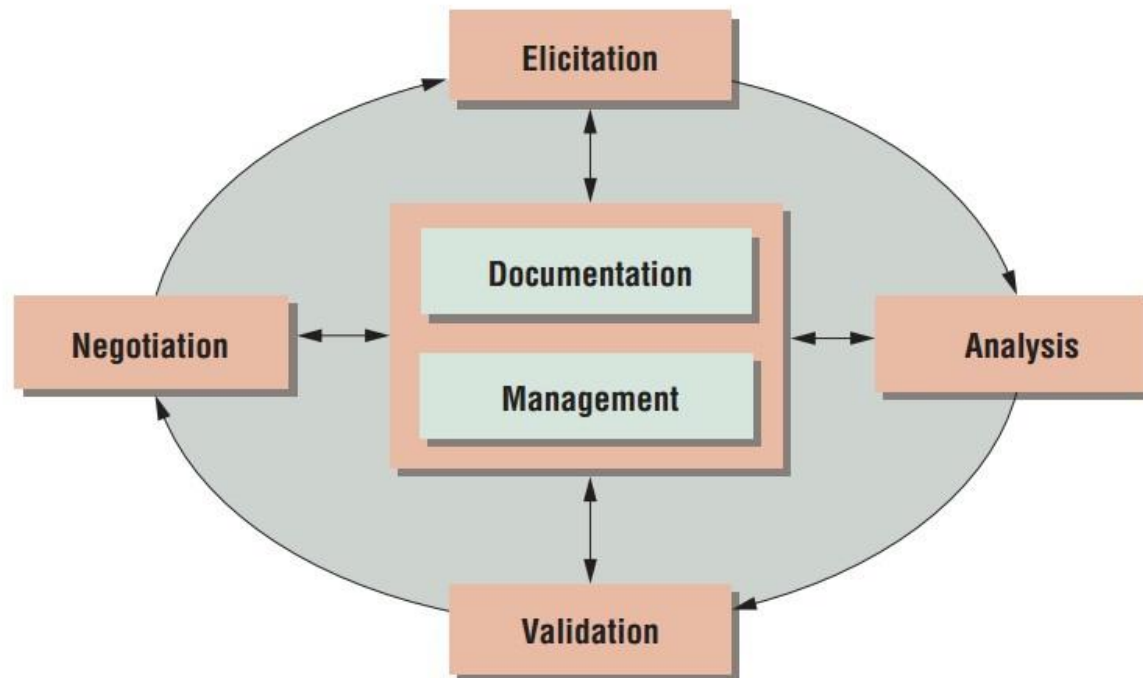


‘Velocità di esecuzione del sistema e memoria occupata’

**Requisito etico:** nella realizzazione dell’applicazione verranno utilizzati solo strumenti e servizi ‘non proprietari’ (es. no Microsoft!)

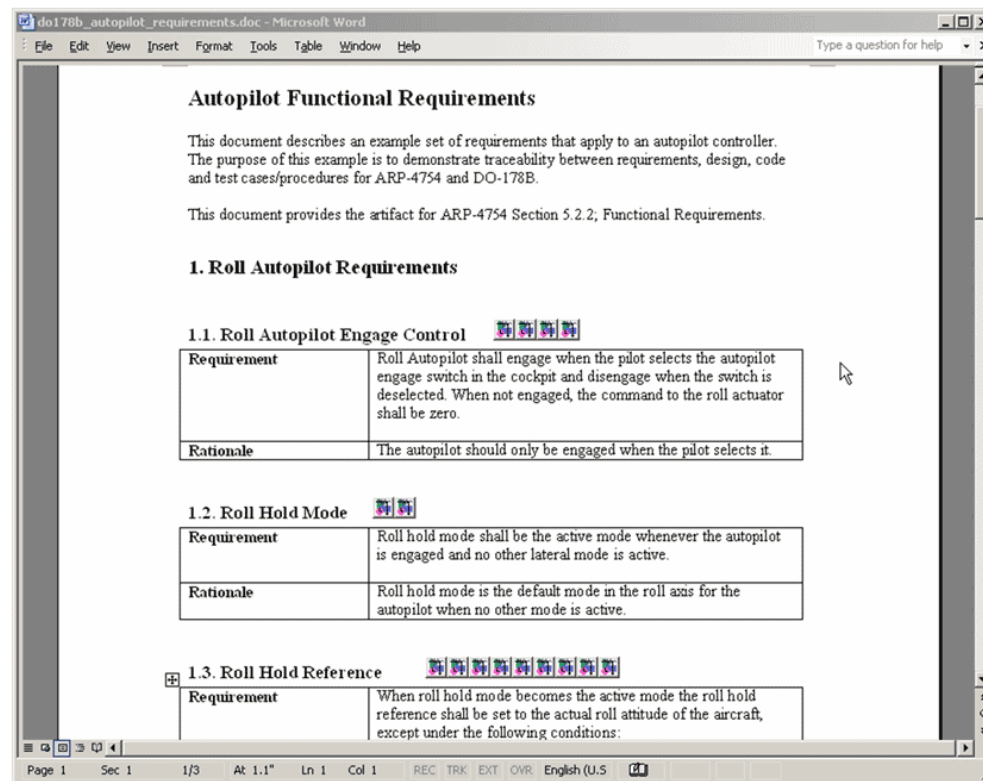
# REQUIREMENTS ENGINEERING

- **Requirements Engineering** (RE) è il termine usato per descrivere **le attività** necessarie per **raccogliere, documentare e tenere aggiornato** l'insieme dei requisiti di un sistema software



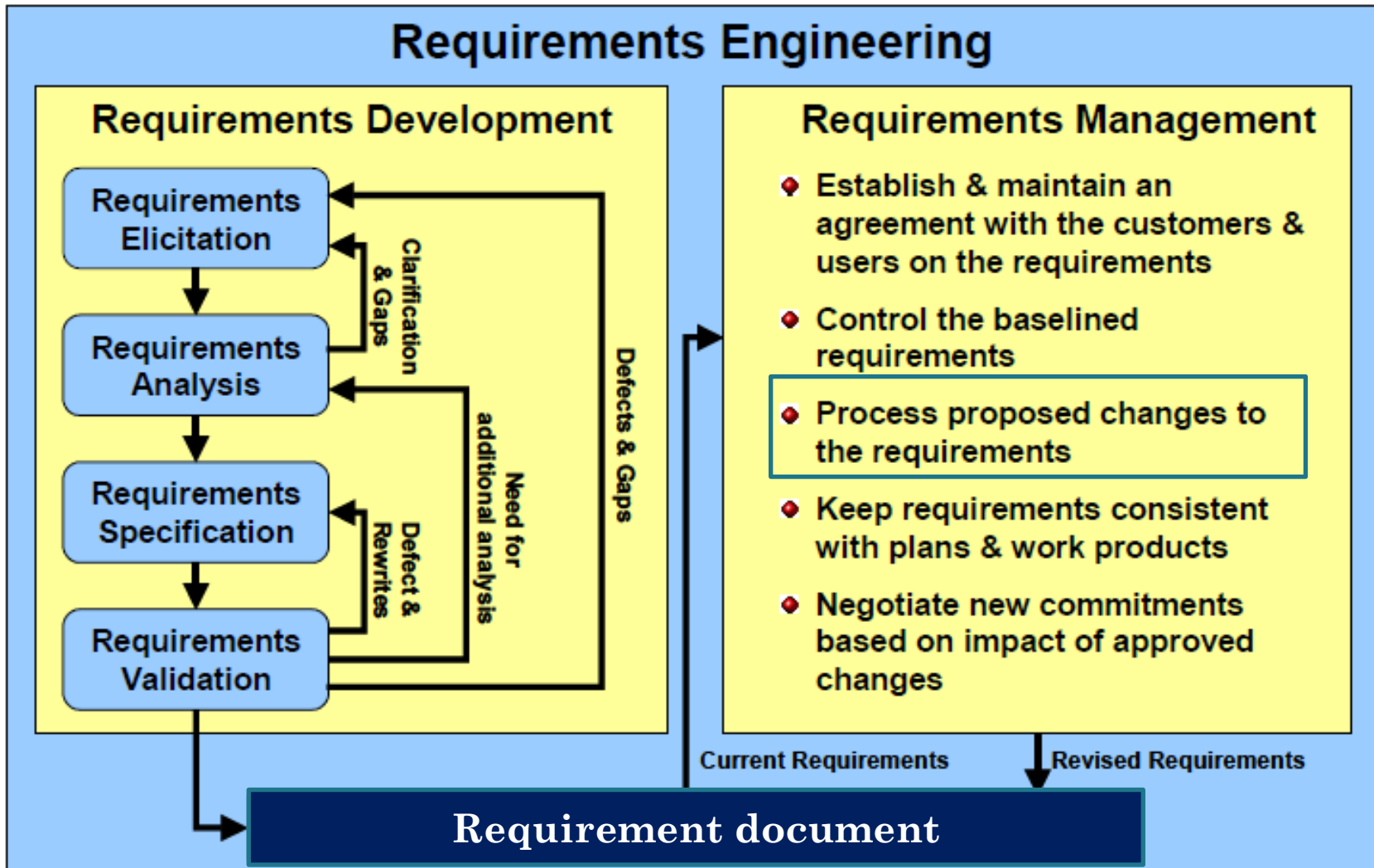
# SCOPO DEL REQUIREMENTS ENGINEERING?

- **Scopo primario** del RE è la produzione di un documento (il *requirement document*) che definisca le funzionalità e i servizi offerti dal sistema da realizzare
  - Anche tenerlo aggiornato ...



# RE = PROCESSO ITERATIVO

*Due processi: sviluppo e gestione ....*







# TOOL DI GESTIONE DEI REQUISITI

**Rich Text Requirement Description**

*RF è spesso supportato da tool!*

# ELICITATION (CATTURA/ESTRAZIONE)

- Ottenere, estrarre, ricavare, tirar fuori [*elìcere*, letteralmente “**strappare da**” / “**tirare fuori**”] i requisiti dal cliente e da altri partecipanti
- Il primo passo è identificare gli **stakeholders**
- Possono essere utilizzate diverse tecniche:
  - **Interviste**
  - Osservazione sul luogo di lavoro
  - Questionari
  - Analisi dei prodotti dei competitors
  - Workshop (brainstorming)
- Spesso include anche:
  - studio/analisi di leggi e regolamenti, help-desk reports, ‘change requests’ di prodotti analoghi, ‘lessons learned’ in progetti simili, ...



# ANALISI DEI REQUISITI

- I bisogni (*user needs*) degli stakeholders raccolti durante la fase di elicitation sono **analizzati e raffinati**
  - i requisiti prendono forma ...
- I requisiti vengono analizzati per stabilirne la fattibilità e la correttezza
  - Si cerca di capire se i requisiti sono corretti
    - Rispetto al volere del cliente
  - Si cercano di identificare i “**missing requirements**” (Gaps in Fig.)
  - Si identificano requisiti poco chiari (Clarification in Fig.)
  - Si risolvono i requisiti “contraddittori o in conflitto”



- Viene stabilita anche la priorità (**prioritizzazione**)

# CONFLITTI TRA REQUISITI: ESEMPIO



## Sistema hardware 'XYZ'

- **Req 12.** Al fine di minimizzare il consumo di energia, si deve **minimizzare il numero di chip** utilizzati preferendo quelli a **basso consumo**
- **Req 23.** Si devono garantire tempi di risposta molto rapidi

**Problema:** pochi chip a basso consumo non possono garantire tempi di risposta rapidi....

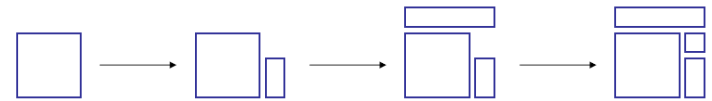
**Soluzione:** negoziazione con gli stakeholder e rimozione di uno dei requisiti in conflitto

# PRIORITÀ DEI REQUISITI

- Per sapere cosa “tagliare” se non tutti potranno essere realizzati

- Per stabilire cosa includere in un incremento

- **Metodi di sviluppo incrementali**



- Due scale:

- scala numerica
  - scala MoSCoW



**MoSCOW**

## - Requisiti necessari (MUST)

devono assolutamente essere implementati

## - Requisiti desiderabili (SHOULD)

sarebbe meglio implementarli, ma non sono strettamente necessari

## - Requisiti possibili (COULD)

da considerare solo nel caso di tempo e risorse disponibili

# DEFINIZIONE E SPECIFICA

*L'attività di stesura vera e propria dei documenti*

- I documenti:

- **Definizione (dei requisiti)** **Requisiti utente**
  - per cliente (manager, tecnici, utenti finali) e anche architetti di sistema, manager della ditta contraente
  - costituisce un **contratto** fra le parti
- **Specifica (dei requisiti)** **Requisiti di sistema**
  - per sviluppatori software e anche tecnici del cliente, architetti di sistema
  - più tecnico (e quindi preciso) anche espressa mediante linguaggi di specifica formali
  - costituisce “**starting point**” per la fase di design!

# VALIDAZIONE



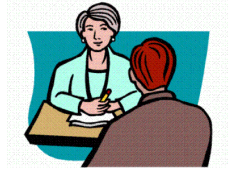
- **Esame della definizione/specifica** dei requisiti per valutarne la qualità
  - Sono ben scritti? Sono chiari? Sono ambigui?
  - Sono completi? **Soddisfano i bisogni dei clienti?**

**Vedere proprietà dei requisiti (dopo)!**

- Di solito la **convalida** o **validazione** si effettua mediante ***formal peer reviews***
  - Processo eseguito da professionisti non coinvolti nella stesura dei documenti che mira ad identificarne aree di sviluppo o miglioramento
- Altri modi sono:
  - scrivere dei casi di test a partire dai requisiti
  - sviluppare un prototipo (es. waterfall con prototipo)

# REQUIREMENTS MANAGEMENT

- Gestione dei requisiti nelle fasi successive allo sviluppo ...



- **Attività:**

- Approvazione di alcune richieste di cambio dei requisiti
- Implementazione dei cambi richiesti
- Negoziazione con il cliente
- ***Impact analysis*** per i cambi richiesti
  - Se cambio il requisito xyz quali altri requisiti dovrò modificare?
- Tenere allineati i requisiti e il codice (e casi di test)
- Tracciare il progresso di un progetto
- ...

*Numero di requisiti implementati  
sul totale dei requisiti*





# PROPRIETÀ DEI REQUISITI (1)

## Validità-correttezza

- sono veramente i requisiti voluti?
- non ce ne sono di inutili o sbagliati ...

Il documento dei requisiti “è conforme agli user needs”?

## Consistenza

- non ci sono requisiti contraddittori



## Completezza

- tutti gli aspetti che il cliente vuole sono coperti nei requisiti

[in assoluto è praticamente impossibile]



## Realismo

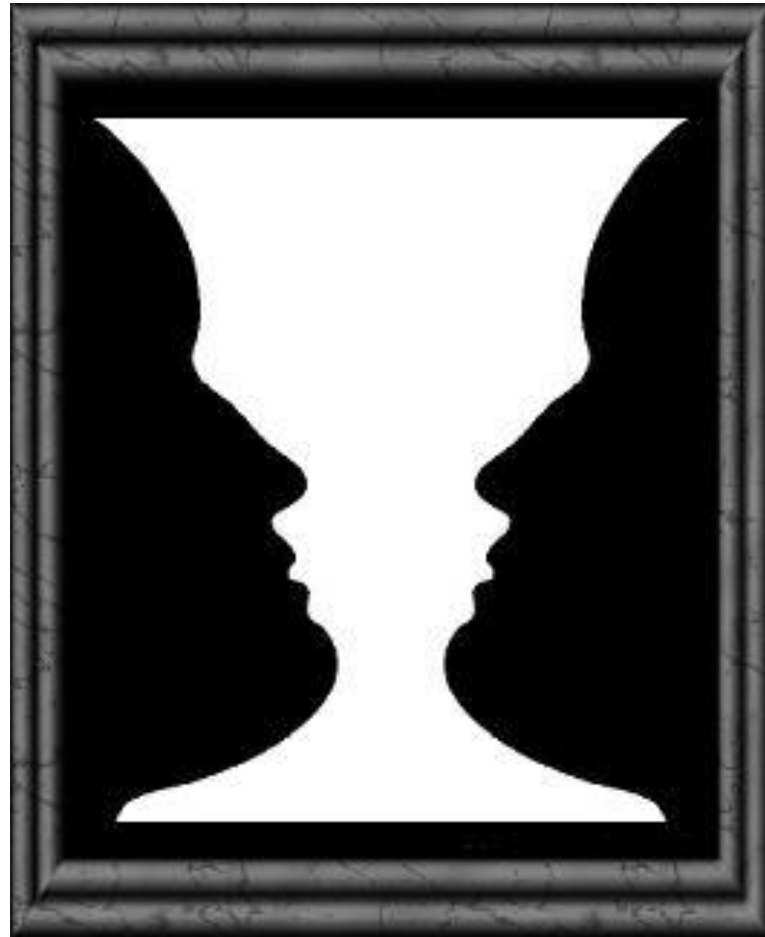
- non si chiede l'impossibile



# PROBLEMA TIPICO DEI REQUISITI ....

**Ambiguità!**

Cosa  
vedete?

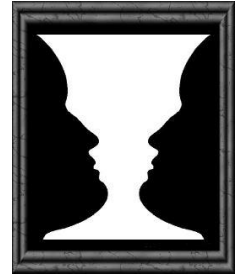


*Es. Il reboot del sistema viene fatto alle 12 ...*

# PROPRIETÀ DEI REQUISITI (2)

## Inequivocabilità (Unambiguous)

- Ogni requisito dovrebbe avere solo un'interpretazione



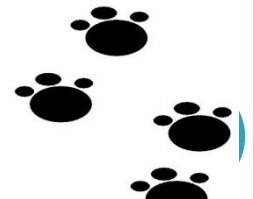
## Verificabilità

- Occorre sempre formulare ed esprimere test per “verificare” i requisiti
- I requisiti vanno espressi in modo che siano **testabili** (es: “immediatamente” deve diventare “in **X** secondi”)



## Tracciabilità

- Ogni funzionalità implementata nel sistema deve poter essere fatta risalire a dei requisiti in modo semplice
- Ogni requisito nella **requirement specification** deve corrispondere ad uno nella **requirement definition**
- ....



# MATRICE DI TRACCIABILITÀ

*Documento di design*



Requirement	Functionality	Internal design	Code	Tests
Restaurant has two ordering stations	Mgmt screen #2	Page 45	Line 12485	34, 57, 63
A waiter may order from any station	Order screen	Page 19	Line 6215	12, 14, 34, 57, 92
Any customer at a table may request a separate check	Order screen	Page 39	Line 2391	113, 85
A customer may get checks from more than one station	Check printing	Page 138	Lines 49234, 61423	74, 104

# STRUTTURA E CONTENUTI DI UN DOCUMENTO DEI REQUISITI

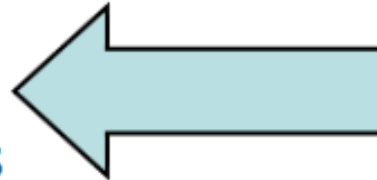
## IEEE Std 830-1998

- 1. Introduction
  - 1.1 Purpose
  - 1.2 Scope
  - 1.3 Definitions, acronyms, and abbreviations
  - 1.4 References
  - 1.5 Overview
- 2. Overall description
  - 2.1 Product perspective
  - 2.2 Product functions
  - 2.3 User characteristics
  - 2.4 Constraints
  - 2.5 Assumptions and dependencies
- 3. Specific requirements
- Appendixes
- Index

# SPECIFIC REQUIREMENTS

## 3 Specific requirements

- 3.1 External interface requirements
  - 3.1.1 User interfaces
  - 3.1.2 Hardware interfaces
  - 3.1.3 Software interfaces
  - 3.1.4 Communications interfaces
- 3.2 Functional requirements
  - *Qui i singoli requisiti ordinati*
- 3.3 Performance requirements
- 3.4 Design constraints
- 3.5 Software system attributes
  - Reliability
  - Maintainability
  - Etc.
- 3.6 Other requirements





# TEMPLATE/SCHEMA DEI REQUISITI

*Molto semplice se non si usano Form più dettagliate*

Conviene attenersi a questo schema

**<id> il <sistema> deve <funzione>**



- 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 il sistema deve elencare quali sono gli articoli da reintegrare in base alle vendite

***Testo non strutturato (linguaggio naturale)***

# LA FIGURA DELL'ANALISTA SOFTWARE

## Non esiste solo la figura dello sviluppatore!

- L'**analista software** o **di sistema** è la persona che:
  - si occupa dell'elicitazione dei requisiti (es. intervistando i clienti)
  - analizza i requisiti
  - scrive il documento dei requisiti (definizione e/o specifica)
  - Comunica/spiega i requisiti a sviluppatori e altri stakeholder...



- Arte della negoziazione
- Stabilire una strategia (problem solving)
- Giusta capacità di imporsi
- Ascoltare attentamente
- Dono della sintesi
- Padronanza del linguaggio naturale
- Buona conoscenza del dominio

*Alcune*  
**"competenze"**



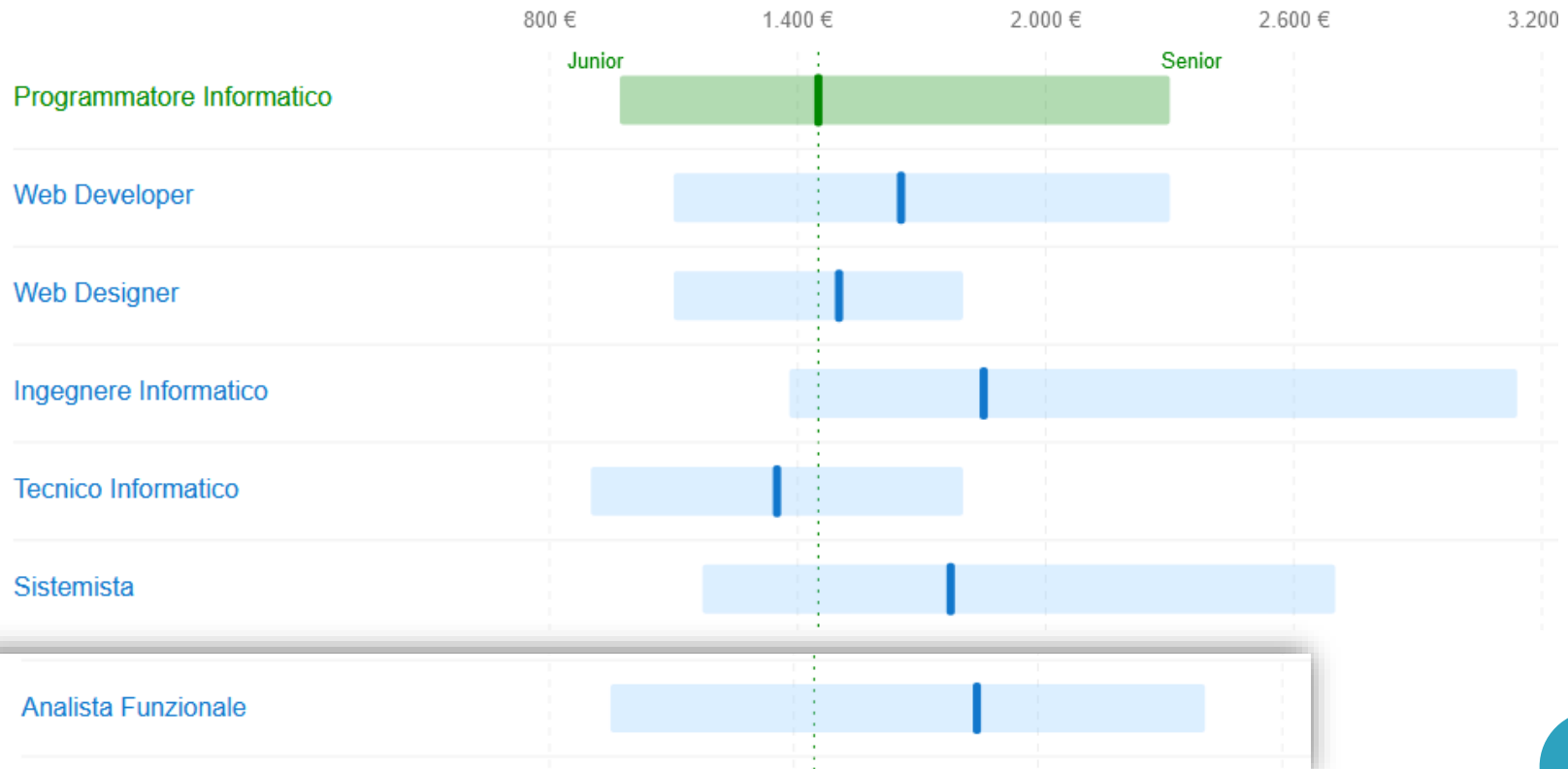
## LAVORI TECH PIÙ RICHIESTI DEL 2021

- System administrator
- System analyst
- Digital Marketing Specialist
- Specialista in Cyber Security
- Developer



# STIPENDI (2021)

## Programmatore Informatico - Stipendio di Professioni Correlate



# ESEMPIO DI INTERVISTA (ANALISTA)



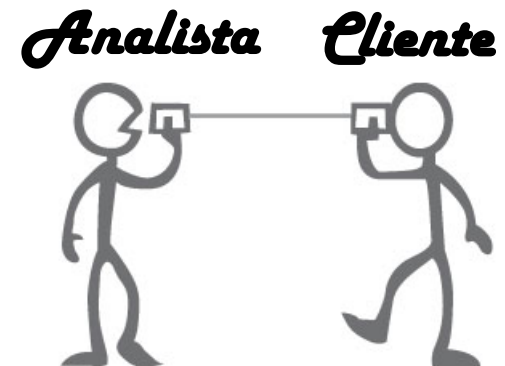
## Consigli:

1. Fare molte domande!
2. Ascoltare bene
3. Mettere in discussione i quantificatori universali: ‘tutto, ogni, sempre, ...’
4. Annotare tutte le risposte

- **Stakeholder:** “Utilizzano il sistema per prendere in prestito i libri”
- **Analista:** Chi utilizza il sistema per prendere in prestito i libri?
- **Risposta:** gli utenti della biblioteca, le altre biblioteche e i bibliotecari
  
- **Stakeholder:** Tutti devono avere una tessera per prendere in prestito i libri
- **Analista:** Non esiste un utente del sistema che può operare senza tessera?
- **Risposta:** Sì, le altre biblioteche ....

# SULL'IMPORTANZA DELLA COMUNICAZIONE

- Elicitation = Attività molto delicata perchè mette in comunicazione **due o più persone** di realtà anche **molto diverse**
- Frequenti incomprensioni, che si ripercuotono sulla qualità dei requisiti

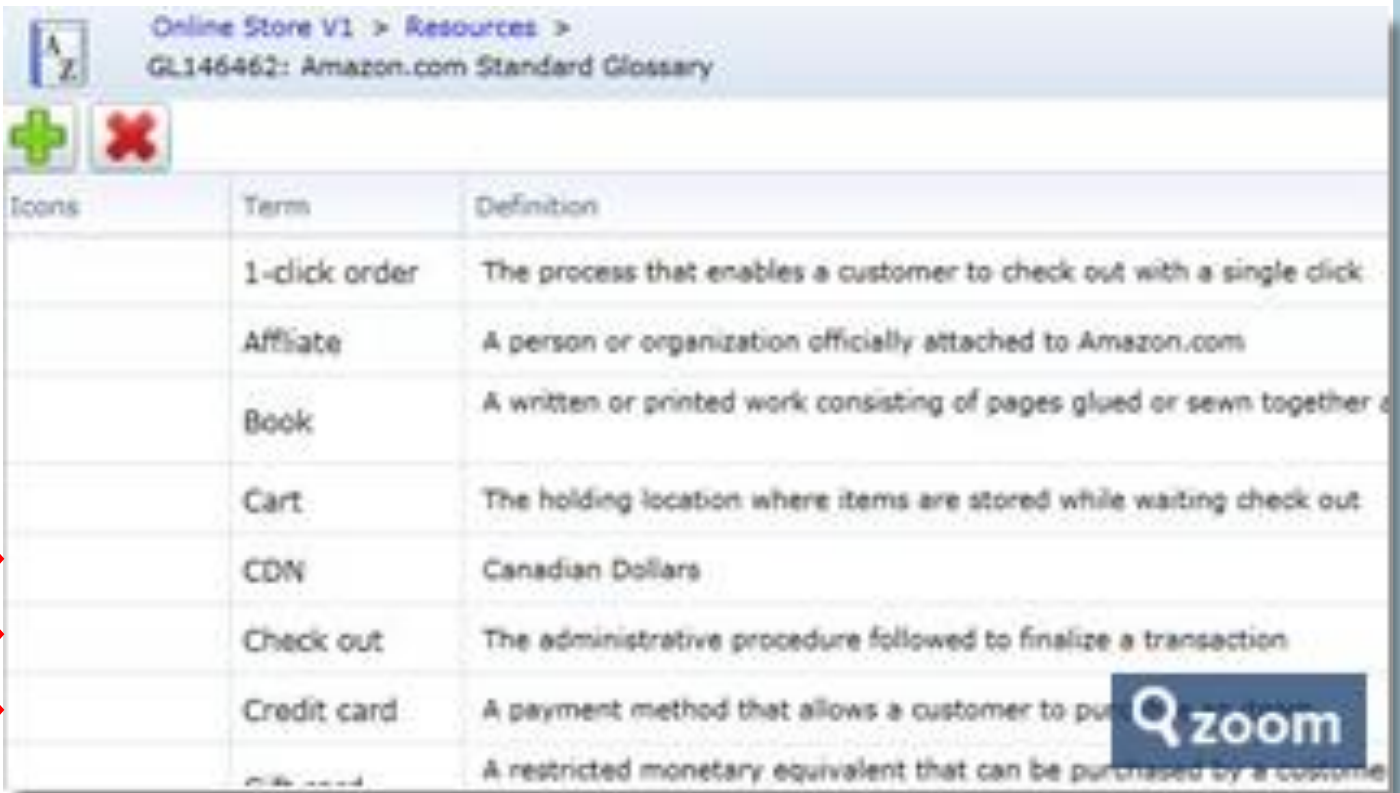


Occorre fare molta attenzione a:

- Diversità di significato che si attribuisce ai termini:  
possibile soluzione **definizione del glossario** (vedi dopo)
  - Termini tecnici (sia informatici che specifici del dominio)
- Assunzioni (Hidden assumptions)
- Verbosità
- Mancanza di chiarezza/precisione

## IL GLOSSARIO

- per la spiegazione dei termini tecnici
- per ridurre l'ambiguità dei termini usati
- per “espandere” gli acronimi



Icons	Term	Definition
	1-click order	The process that enables a customer to check out with a single click
	Affiliate	A person or organization officially attached to Amazon.com
	Book	A written or printed work consisting of pages glued or sewn together
	Cart	The holding location where items are stored while waiting check out
	CDN	Canadian Dollars
	Check out	The administrative procedure followed to finalize a transaction
	Credit card	A payment method that allows a customer to purchase items
	Gift card	A restricted monetary equivalent that can be purchased by a customer

# PROBLEMI TIPICI DI UN DOCUMENTO DEI REQUISITI

- Mancanza di chiarezza
- Verbosità
- Utilizzo di termini tecnici
  - Informatici e non
- Ambiguità
- Hidden assumptions
- Requisiti troppo generici o troppo dettagliati
  - **In medio stat virtus ....**
- Parti di soluzione (design) inserite nei requisiti
- Due o più requisiti sono specificati come un singolo requisito
- Tipologie diverse di requisiti ‘mescolate’ tra loro
  - Es. funzionali e non funzionali



Tipologie diverse di requisiti 'mescolate'

# PROBLEMI??

Più requisiti in uno ...

- *Si consideri un sistema bancomat. Il servizio deve mettere a disposizione le funzioni di prelievo, saldo, estratto conto. Il sistema deve essere disponibile a persone portatori di Handicap, deve garantire un tempo di risposta inferiore al minuto, e deve essere sviluppato su architettura X86 con sistema operativo compatibile con quello della Banca. 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 non funzionali

Requisiti funzionali

# CONSIGLI

- PER PRODURRE BUONI REQUISITI!
- PER GESTIRE AL MEGLIO I REQUISITI!

- Riutilizzo di (parte di) requisiti
  - Non solo riutilizzo di software!
- Utilizzo di un **glossario** comune tra clienti, utenti e analisti
- Utilizzo di un 'buon' template/form
- Utilizzo di un software per la gestione/raccolta e analisi dei requisiti



La lista è lunghissima ....

TOOL	Produttore
Caliber	Borland
DOORS	Rational (IBM)
Requirement Specification Tool	Visual Paradigm
OSRMT	Progetto open source
Requisite Pro	Rational (IBM)
ReqView	Eccam s.r.o.



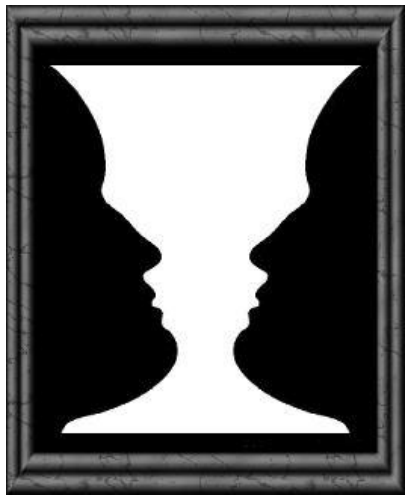
# RIASSUMENDO ...

## Cosa è un requisito?

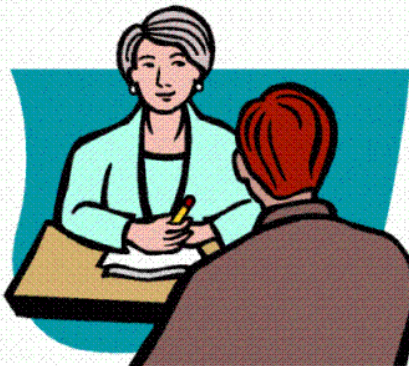
Requisiti Utente ↔ Requisiti di Sistema

Requisiti Funzionali ↔ Non Funzionali

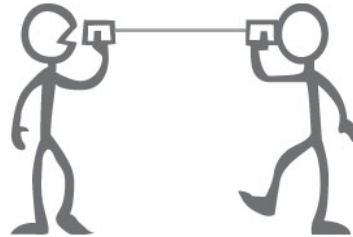
Proprietà dei requisiti



Analista Software



Comunicazione



Consigli



Verbosità



## IMPORTANZA DEI REQUISITI

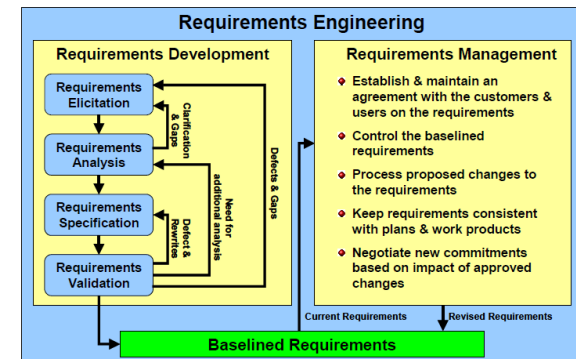
- Motivi di insuccesso di un progetto software (Standish survey, 1995)

Motivi di fallimento	Percentuale
Incompletezza dei requisiti	13.1%
Mancato coinvolgimento degli utenti	12.4%
Carenza di risorse	10.6%
Aspettative non realistiche	9.9%
Mancato supporto dal management	9.3%
Cambiamento di requisiti e specifiche	8.7%
Carenza di pianificazione	8.1%
Cambiamento di obiettivi	7.5%
Carenza di gestione del settore IT	6.2%
Incompetenza tecnologica	4.3%
Altro	9.9%

= 44.1%

Nessuna parte del lavoro pregiudica maggiormente il risultato se viene eseguita in modo errato

5



# APPROFONDIMENTI ...

- Articolo: Software Requirements Engineering: What, Why, Who, When, and How By Linda Westfall
- Articolo: *Exploring User Requirements* through Mind Mapping by K. HIRANABE
- Standard: <http://homepages.dcc.ufmg.br/~rodolfo/es-1-03/IEEE-Std-830-1998.pdf>
- Lista di 'requirement tools':  
[http://www.jiludwig.com/Requirements\\_Management\\_Tools.html](http://www.jiludwig.com/Requirements_Management_Tools.html)
- Dispense:  
[http://didattica.agentgroup.unimore.it/wiki/images/d/d7/IEEE830\\_ITA.pdf](http://didattica.agentgroup.unimore.it/wiki/images/d/d7/IEEE830_ITA.pdf)
- Esempio documento dei requisiti:  
<https://www.slideshare.net/indrisrozas/example-requirements-specification>