

Corso di Laurea in Ingegneria Informatica

Corso di Ingegneria del Software

**Analisi e Specifica
dei Requisiti**

Sommario

- L' ingegneria dei requisiti
- Tipologie di requisiti
- Caratteristiche dei requisiti
- Analisi e specifica dei requisiti
- Il documento di specifica dei requisiti

Ingegneria dei Requisiti

Il processo condotto per stabilire le funzionalità/servizi e le caratteristiche di qualità che il cliente richiede al sistema, i vincoli di sviluppo e i vincoli operativi, e per la gestione dei requisiti in tutto il ciclo di vita

I requisiti sono la descrizione di tali funzionalità e caratteristiche e dei relativi vincoli

Importanza dell' ingegneria dei requisiti

L'ingegneria dei requisiti (Requirements Engineering) sviluppa metodi per **raccogliere, documentare, classificare, analizzare e gestire** i requisiti

- ◆ È di importanza fondamentale, in quanto errori in questa fase **si propagano** nei passi successivi e *il costo per porvi rimedio cresce nel tempo*
- È stato stimato che correggere errori nei requisiti **costa 100 volte in più** che correggere errori di programmazione; correggerli in fase operazionale **costa dalle 60 alle 200 volte in più** che in fase di analisi e specifica dei requisiti

Importanza dell' ingegneria dei requisiti

- ❖ Requisiti non chiari, incorretti o incompleti possono portare a:
 - Consegne ritardate
 - Costi maggiori
 - Insoddisfazione dell' utente
 - Comportamenti errati e imprevisti
 - Prodotto software corretto (realizzato conformemente alle specifiche) ma che non soddisfa i bisogni dell' utente (con eventuali contestazioni, mancati pagamenti, etc.)
 - Alti costi di manutenzione

Cos' è un Requisito?

Requisito: una frase in linguaggio naturale che descrive qualcosa che il sistema dovrà fare (**requisito funzionale**), o una proprietà o vincolo che si desidera per il sistema (**requisito non funzionale**), e che uno o più **stakeholders** (portatori di interesse) richiedono al sistema stesso.

Una frase breve che descrive **cosa**, ma non la soluzione (**come**).

Ogni requisito deve essere riferito ad almeno uno **stakeholder**

Requisiti

Un requisito può variare da una descrizione astratta, ad alto livello, di qualcosa che il sistema farà, di una proprietà o vincolo, ad una specifica formale di una proprietà desiderata

Ciò è inevitabile, perchè un requisito può avere vari scopi, ad es.:

- Può essere la base per una gara fra fornitori concorrenti
- Può essere la base del contratto stesso - quindi deve essere definito in dettaglio

Esempio: descrizione iniziale dei requisiti

- ❖ *Si vuole sviluppare un sistema software per il noleggio di testi gestiti dalle biblioteche di una università.*
- ❖ Il sistema deve consentire agli studenti di noleggiare una copia di uno dei testi consigliati nell' ambito dei corsi previsti dal proprio piano di studi. Ciascuno studente frequenta più corsi del suo piano di studi. Nell' ambito di ciascun corso sono consigliati uno o più testi. Un' università presenta una o più strutture adibite a biblioteca identificate da un codice identificativo e dal nome dell' edificio in cui essa è ubicata. Presso tali strutture possono essere disponibili una o più copie dei testi. Ogni copia di un testo è caratterizzata da un identificativo e dal prezzo.
....
- ❖ Il sistema deve permettere ad uno studente di prenotare il noleggio di una copia dei testi consigliati nell' ambito dei corsi del proprio piano di studi. ...Il gestore della biblioteca ha il compito di consolidare i noleggi, ossia ...il sistema deve consentire agli amministratori di aggiungere copie di testi adottati da un corso di studio erogato dall' università

Necessità di diversi livelli di astrazione nei requisiti (Davis)

“Se una compagnia vuole dare in appalto un grande progetto di sviluppo software, deve definire le sue esigenze in modo abbastanza astratto, da non predefinire alcuna soluzione.

I requisiti devono essere scritti in modo da consentire a vari appaltatori di interpretare le esigenze e proporre vari metodi per soddisfarle.

Quando l' appalto è stato assegnato, l' appaltatore deve scrivere per il cliente una descrizione del sistema molto dettagliata, in modo che il cliente possa capire e validare ciò che il sistema farà.

Entrambi i documenti si chiameranno documenti dei requisiti.”

Tipi di Requisiti

◆ **Requisiti Utente**

Frasi in linguaggio naturale (e diagrammi) ad alto livello, relativi a funzionalità/servizi che il sistema deve fornire e i suoi vincoli operativi. Scritti per i clienti.

◆ **Requisiti di Sistema**

Un documento strutturato che fornisce una descrizione dettagliata di funzionalità/servizi/vincoli operativi del sistema. Può essere parte del contratto fra cliente e sviluppatore.

Tipi di Requisiti

◆ **Esempio di definizione di un requisito utente**

Il software deve permettere la gestione del magazzino delle scorte del ristorante

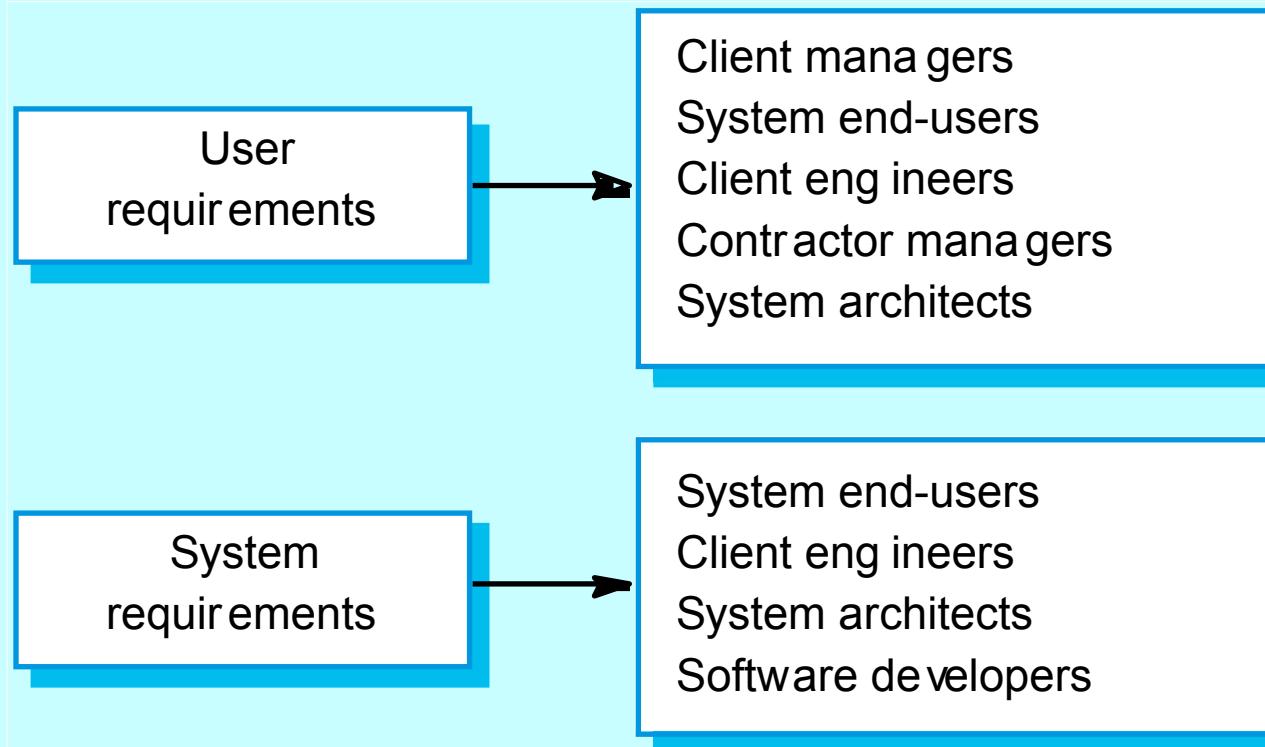
◆ **Esempio di Specifiche dei requisiti di sistema**

- 1 Il magazziniere deve registrare le merci consegnate dai fornitori
- 2 Per ogni articolo consegnato si dovrà registrare il nome, il produttore, la quantità, la data di scadenza
- 3 Il magazziniere deve registrare le uscite degli articoli usati per preparare le pietanze
- 4 Il software deve controllare automaticamente che in magazzino non siano presenti articoli scaduti
- 5 Il software deve riordinare automaticamente gli articoli le cui scorte siano in esaurimento
- 6 ...

Un altro esempio

- ◆ Es. di Definizione di un **requisito utente**:
 - Il software ***DOCTORS*** deve generare mensilmente dei report sui costi dei farmaci prescritti dai medici in quel mese.
- ◆ Es. di Specifiche dei **requisiti di sistema**
 - Nell' ultimo giorno lavorativo del mese, deve essere generato un report con la sintesi di farmaci prescritti, i loro costi, ed i medici prescriventi.
 - Il sistema deve generare automaticamente il report di stampa dopo le 17:30 dell' ultimo giorno del mese
 - Deve essere generato un report per ciascun medico che elencherà i nomi dei farmaci, il totale delle prescrizioni, il numero di dosi prescritte, ed il costo totale delle dosi prescritte.
 - L' accesso ai report dei costi sarà riservato a personale autorizzato appartenente ad una lista per controllo accessi.

I lettori dei vari tipi di requisiti



Tipi di Requisiti

◆ **Requisiti Funzionali**

Descrivono funzionalità/servizi del sistema (cosa si richiede che faccia)

◆ **Requisiti Non Funzionali**

- Caratteristiche di qualità
- Vincoli sul sistema/ambiente
- Vincoli sul processo di sviluppo
- Vincoli tecnologici

◆ **Requisiti di dominio**

- Requisiti che derivano dal dominio applicativo
 - Es.: norme e standard di dominio, per es. ferroviario, avionico, etc.

Requisiti Funzionali

- ◆ Descrivono le funzionalità/servizi offerti
- ◆ Anche in questo caso, due livelli di astrazione:
 - Requisiti funzionali utente: frasi ad alto livello su ciò che il sistema farà
 - Requisiti funzionali di sistema: descrizioni dettagliate dei servizi

Requisiti Funzionali

- ◆ Quali input il sistema dovrà accettare
- ◆ Quali output il sistema dovrà produrre
- ◆ Quali dati il sistema dovrà immagazzinare che potranno essere usati da altri sistemi
- ◆ Quali elaborazioni il sistema dovrà svolgere
- ◆ Requisiti di tempificazione e di sincronizzazione delle elaborazioni
- ◆

Un esempio: Il sistema DOCTORS

- ♦ È un sistema per gestire informazioni relative a pazienti in trattamento medico per problemi di salute mentale.

Esempi di requisiti funzionali utente

1. Un utente potrà cercare le liste di appuntamenti di tutti i medici
2. Il sistema dovrà generare ogni giorno, per ciascun medico, un elenco dei pazienti attesi in quel giorno.
3. Tutti i dipendenti della clinica che usano il sistema saranno identificati dalla propria matricola del personale a 8 cifre.

Notare i diversi livelli di dettaglio fra requisito 1 e 3.

Imprecisioni nei requisiti

- ◆ Requisiti imprecisi o ambigui possono essere interpretati in modi diversi da sviluppatori ed utenti.
- ◆ Es.: il requisito 1 richiede una ricerca degli appuntamenti di tutti i medici
 - L'utente intendeva poter cercare tra gli appuntamenti di tutti i medici (senza conoscere il nome del medico)
 - Lo sviluppatore potrebbe intendere che la ricerca verrà fatta specificando prima il nome del medico.

Completezza e consistenza dei requisiti

- ♦ I requisiti devono essere completi e consistenti
- ♦ **Completezza**
 - Tutti i requisiti richiesti devono essere presenti
- ♦ **Consistenza**
 - Non ci devono essere definizioni contraddittorie
- ♦ Nella realtà è facile commettere errori o omissioni, soprattutto per sistemi complessi

Requisiti Non Funzionali

- ◆ Definiscono o limitano le **proprietà** del sistema (es. Affidabilità, tempi di risposta, memoria occupata)
- ◆ Possono vincolare anche il processo di sviluppo da adottare (es. Uso di particolari standard per la documentazione, su sistemi CASE, linguaggi di codifica, metodi di sviluppo)
- ◆ Possono essere più critici dei req. funzionali: se non sono soddisfatti, il sistema è inutile.
 - Es. Affidabilità di un sistema aerospaziale, o performance di un sistema di controllo embedded

Tipi di requisiti non-funzionali

◆ Requisiti del prodotto

- Specificano il comportamento del prodotto (usabilità, efficienza, affidabilità, portabilità).

◆ Requisiti organizzativi

- Derivano dalle politiche e procedure dell'organizzazione del cliente e dello sviluppatore (es. Regole organizzative del dominio, standard di processo da usare, piattaforme, requisiti di consegna, etc.)

◆ Requisiti esterni

- Derivano da fattori esterni al sistema e al suo processo di sviluppo (come i requisiti di interoperabilità, legislativi, etici, etc.)

Esempi di requisiti non funzionali

◆ Requisiti del Prodotto

- la disponibilità del sistema DOCTORS deve essere garantita durante I giorni feriali dalle 08:30 alle 17:30. Eventuali tempi di down non devono superare I 5 secondi al giorno

◆ Requisiti organizzativi

- Gli utenti si autenticheranno attraverso la propria carta di identità sanitaria.

◆ Requisiti esterni

- Il sistema deve implementare le politiche sulla privacy dei pazienti, come stabilito nell'apposito documento sulla privacy XYZ.

Ancora sui Requisiti Non Funzionali

1. *Usability, efficiency, reliability, maintainability e reusability*

- *Tempi di risposta*
- *Throughput*
- *Uso delle risorse*
- *Affidabilità*
- *Disponibilità*
- *Recovery from failure*
- *Manutenibilità*
- *Riusabilità*

Ancora sui Requisiti Non Funzionali

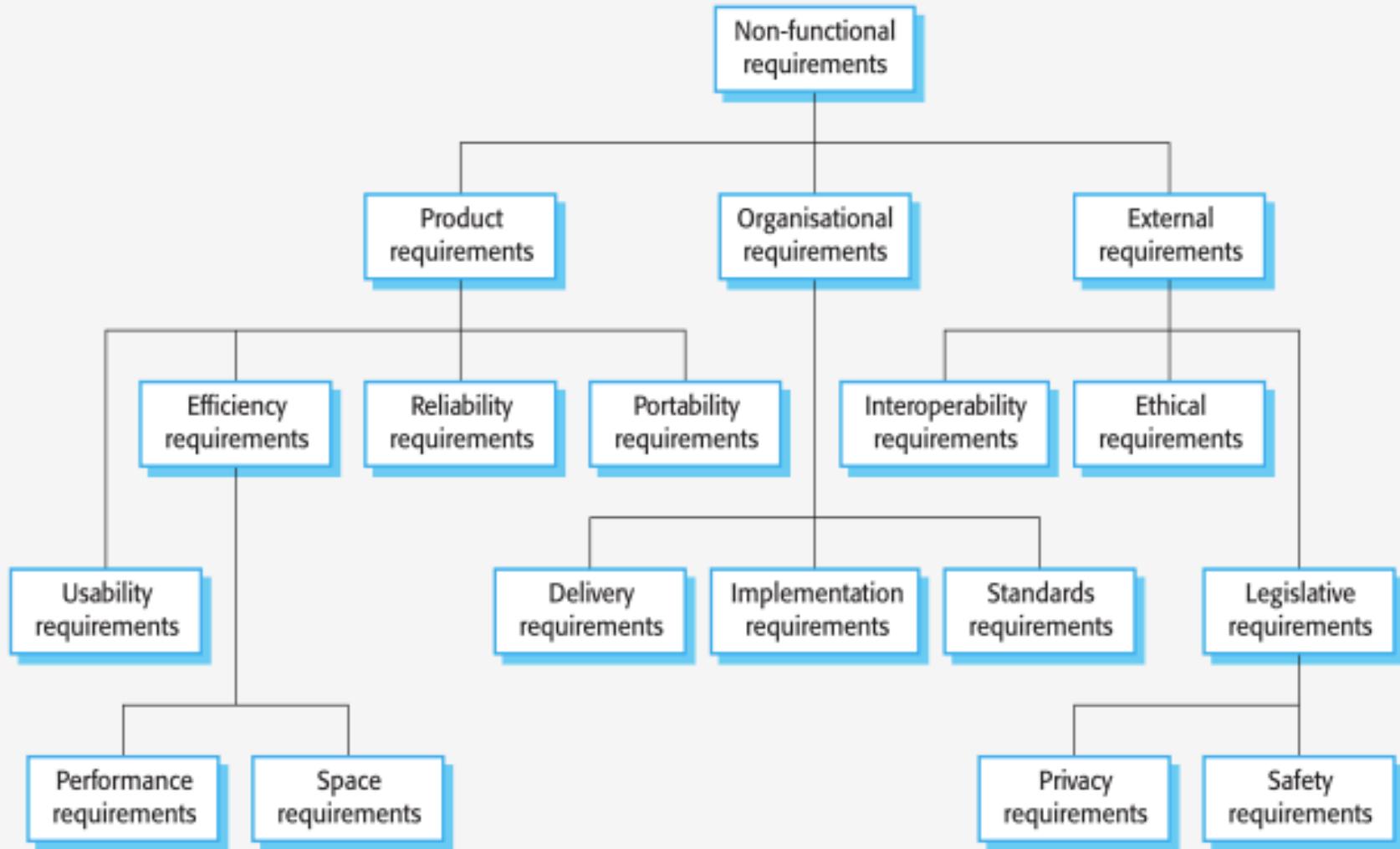
2. Vincoli sull' ambiente di funzionamento e sulle tecniche del sistema.

- *Piattaforma*
- *Tecnologie di sviluppo*
- Es.: “*il sistema deve essere accessibile non con un client ad hoc, ma attraverso un browser web*”; oppure: “*il sistema deve essere accessibile da PC e da smartphone*”

3. Vincoli su *project plan* e metodi di sviluppo

- *Processo e metodologie da adottare*
- *Procedure per il controllo di qualità*
- *Costi e date di consegna*

Una tassonomia di requisiti non funzionali



Verificabilità dei requisiti non funzionali

- ♦ I requisiti non funzionali possono essere difficili da definire precisamente, e quindi difficili da verificare.
- ♦ L'utente li specifica come obiettivi
 - Es. La facilità d'uso
- ♦ Requisito verificabile
 - Una frase che contiene qualche misura che potrà essere oggettivamente verificata.

Verificabilità dei requisiti non funzionali

- ♦ “Il sistema dovrebbe essere usato facilmente dallo staff medico e dovrebbe minimizzare gli errori.”
- ♦ Requisito non-funzionale verificabile:
 - Lo staff medico dovrà essere capace di usare tutte le sue funzioni dopo due ore di addestramento. Dopo tale addestramento, il numero medio di errori fatti dagli utenti non dovrà essere superiore a due errori al giorno

Contraddizioni fra i requisiti

- ◆ I diversi requisiti non-funzionali possono essere in contraddizione fra loro, soprattutto in sistemi complessi (es. Tempi di risposta e portabilità)
- ◆ Es.: Sistemi aerospaziali
 - Per minimizzare il peso, il numero di chips a bordo dovrebbe essere minimo e la memoria complessiva non superiore a 4Mbyte
 - Un vincolo potrebbe richiedere di usare il linguaggio ADA (adatto a sistemi critici real time)
 - Purtroppo può accadere che non si possano avere tutte le funzioni richieste in 4 Mbyte. Quale requisito dovrà essere soddisfatto fra I due?

Requisiti di Dominio

- ◆ Derivano dal dominio di applicazione, piuttosto che da necessità degli utenti.
- ◆ I requisiti di dominio possono essere nuovi requisiti funzionali, vincoli su altri requisiti funzionali, o possono delineare calcoli da effettuarsi.
- ◆ Se non sono soddisfatti, il sistema potrebbe essere inutilizzabile.

Es.: Requisiti di dominio per il sistema Libreria

- ◆ Deve esserci una interfaccia utente uniforme per tutti i database basata sullo standard Z39.50.
- ◆ A causa delle restrizioni sul copyright, alcuni documenti devono essere cancellati immediatamente dopo l'arrivo. A seconda delle richieste dell'utente, questi documenti potranno o essere stampati localmente sul server del sistema per l'invio manuale all'utente, oppure devono essere inoltrati presso una stampante di rete.

Es.: Il sistema di protezione dei treni

- La decelerazione del treno viene calcolata come:

- $D_{train} = D_{control} + D_{gradient}$

dove $D_{gradient}$ vale $9.81\text{ms}^2 * \text{compensated gradient/alpha}$ e dove i valori di $9.81\text{ms}^2 / \text{alpha}$ sono noti per diversi tipi di treno.

Problemi dei requisiti di dominio

♦ Comprendibilità

- I requisiti sono espressi nel linguaggio del dominio;
- Tale linguaggio potrebbe non essere immediatamente comprensibile per gli ingegneri del software.

♦ Esplicitazione

- Gli specialisti del dominio conoscono così bene il dominio stesso, da lasciare fuori dai requisiti informazioni che sembrano loro ovvie (e invece non lo sono per l'ingegnere del software...)
- L'analista deve rendere esplicativi eventuali requisiti di dominio impliciti

Analisi e Specifica dei Requisiti

Analisi dei Requisiti

Processo di comprendere del dominio del problema al fine di:

- ❖ stabilire le funzionalità, i servizi, le interfacce, le prestazioni, i fattori di qualità che il committente richiede al sistema da sviluppare, ed i vincoli di utilizzo e sviluppo;
- ❖ produrre un **Documento di Specifica dei Requisiti Software**, che sia completo, consistente, non ambiguo e comprendibile;
- ❖ predisporre un piano di test.

Definisce **Cosa** il sistema deve fare, non il **Come**

Analisi, Specifica e Validazione



Analisi

Comprensione dei bisogni (es.: cosa deve fare il sistema software)

Specifiche dei requisiti

Trasformazione dei requisiti in un documento di specifiche tecniche e funzionali caratterizzanti il sistema

Validazione delle specifiche

Le specifiche, formalizzate, vengono riviste con l'utente/committente

Analisi vs Specifica



Analisi dei requisiti

- Processo di valutazione delle necessità del committente, con redazione di un documento di analisi dei requisiti, di solito in linguaggio naturale
- La raccolta dei requisiti avviene tramite di interviste svolte dagli analisti con gli *stakeholder* (committente, utente, installatore, ...), lo studio nelle norme di dominio, etc.

Specificazione dei requisiti

- Processo di schematizzazione dei requisiti di un sistema, con redazione di un documento secondo un modello (*template*) prefissato
- È la base del contratto di fornitura

Alcune linee guida

- ◆ Adottare un formato standard (interno all' azienda/organizzazione) per scrivere i requisiti
- ◆ Adoperare frasi semplici e brevi
- ◆ Evitare mescolanza di requisiti
 - Una frase non deve esprimere più requisiti
- ◆ Usare il linguaggio naturale in maniera coerente. Es. distinguere i requisiti **obbligatori** (indicati con “deve”) da quelli **desiderati** (indicati con “dovrebbe”)
- ◆ Usare diversi stili di testo per evidenziare i punti chiave del requisito
- ◆ Evitare il gergo informatico

Esempio di formato per la descrizione di req. utente

- ◆ *Nome requisito:*
- ◆ *Descrizione funzionalità:*
- ◆ *Motivazione:*
- ◆ *Specifiche dettagliate:*
- ◆ *Sorgente:*

Esempio

- **Nome requisito:** **Funzionalità della griglia**
- **Descrizione funzionalità:**
 - L'editor deve fornire una funzione griglia dove una matrice di linee orizzontali e verticali fornisce uno sfondo alla finestra dell'editor.
- **Motivazione:**
 - La griglia aiuta l'utente a costruire diagrammi più precisi
- **Specifiche:**
 - ECLIPSE/WS/Tools/FS Sezione 5.6
- **Sorgente:**
 - Mario Rossi, Ufficio di Roma

Specifiche dei Requisiti di Sistema

- ◆ Sono specifiche più dettagliate dei requisiti utente che serviranno come base per la progettazione, ma non sono le specifiche di **come** verrà sviluppato.
- ◆ Spesso sono inclusi nel contratto fra cliente e sviluppatore.
- ◆ In alcuni casi potrebbero però includere già riferimenti all'architettura che il sistema avrà
 - per poter strutturare meglio i requisiti relativamente ai sotto-sistemi,
 - per soddisfare specifici requisiti non-funzionali,
 - per esigenze di inter-operabilità con altri sistemi.

Linguaggi per la specifica

- ◆ Spesso anche i requisiti di sistema vengono espressi in linguaggio naturale (NL).
- ◆ Alcune alternative al NL:
 - Linguaggio naturale strutturato e stilizzato (con uso di Moduli/Template standard)
 - Modelli grafici (come gli **use case** e **sequence diagrams**)
 - Specifiche matematiche formali (con **macchine a stati**, notazioni insiemistiche, o altro)

Specifiche in linguaggio strutturato

- ◆ La libertà di chi scrive i requisiti è limitata usando un '**modello**' predefinito e standard per tutti i requisiti.
- ◆ Viene limitata anche la terminologia usata per descrivere il requisito.
- ◆ Il **vantaggio** è che si conserva l'espressività del linguaggio naturale, ma al tempo stesso si impone un grado di uniformità per descrivere le specifiche.

Specifiche basate su un Template

- ◆ Definizione della **funzione** o entità.
- ◆ Descrizione degli **input** e relative sorgenti.
- ◆ Descrizione degli **output** e loro destinazione.
- ◆ **Informazioni** richieste per l'elaborazione o altre entità del sistema **usate** (ossia richieste).
- ◆ *Pre-condizioni e Post-condizioni*
- ◆ Eventuali **effetti collaterali** della funzione.

Esempio di Modello/ Template per la specifica

| |
|---------------------|
| Nome Requisito: |
| Funzione |
| Descrizione |
| Input |
| Sorgente |
| Output |
| Destinazione |
| Informazioni |
| Richieste |
| Elaborazione |
| Pre-condizioni |
| Post-Condizioni |
| Effetti Collaterali |

Esempio: il requisito del calcolo della dose di Insulina

Insulin Pump/Control Software/SRS/3.3.2

Function Compute insulin dose: Safe sugar level

Description Computes the dose of insulin to be delivered when the current measured sugar level is in the safe zone between 3 and 7 units.

Inputs Current sugar reading (r2), the previous two readings (r0 and r1)

Source Current sugar reading from sensor. Other readings from memory.

Outputs CompDose – the dose in insulin to be delivered

Destination Main control loop

Action: CompDose is zero if the sugar level is stable or falling or if the level is increasing but the rate of increase is decreasing. If the level is increasing and the rate of increase is increasing, then CompDose is computed by dividing the difference between the current sugar level and the previous level by 4 and rounding the result. If the result, is rounded to zero then CompDose is set to the minimum dose that can be delivered.

Requires Two previous readings so that the rate of change of sugar level can be computed.

Pre-condition The insulin reservoir contains at least the maximum allowed single dose of insulin..

Post-condition r0 is replaced by r1 then r1 is replaced by r2

Side-effects None

Specifica Tabellare

| Condizione | Azione |
|---|--|
| Livello zuccheri in discesa ($r_2 < r_1$) | CompDose = 0 |
| Livello zuccheri stabile ($r_2 = r_1$) | CompDose = 0 |
| Livello zuccheri crescente e tasso di crescita in diminuzione ($(r_2 - r_1) < (r_1 - r_0)$) | CompDose = 0 |
| Livello zuccheri crescente e tasso di crescita in aumento ($(r_2 - r_1) \geq (r_1 - r_0)$) | CompDose = round $((r_2 - r_1)/4)$ If rounded result = 0 then CompDose = MinimumDose |

Ingegneria dei Requisiti

- ♦ Fasi principali dell' Ingegneria dei Requisiti
 - *Elicitation*
 - Analisi e Negoziazione
 - Documentazione
 - Validazione
 - Gestione

Elicitation

Acquisizione dei requisiti tramite consultazioni/interviste con gli **stakeholders**, documenti, conoscenza del dominio e studi di mercato

Avvio - Porre domande per stabilire ...

- *Una conoscenza di base del problema*
- *Le persone che si aspettano una soluzione (clienti/ utenti/etc.)*
- *La natura della soluzione desiderata*
- *L'efficacia della comunicazione/collaborazione tra stakeholder e analista*

Deduzione - Raccogliere i requisiti dai vari stakeholders tramite interviste, descrizione di scenari, prototipi, raccolta “collaborativa”

Elicitation

Una tecnica utilizzata è la **Raccolta Collaborativa dei requisiti**

Basata su riunioni strutturate con gli stakeholders

- ◆ Regole precise di preparazione e conduzione della riunione
- ◆ Uso di una agenda
- ◆ Un moderatore coordina la riunione
- ◆ Viene scelto un meccanismo di definizione (fogli di lavoro, schede, fogli, adesivi, *chat room* o *virtual forum*)

Scopi:

- ◆ Identificare il problema e proporre elementi della soluzione
- ◆ Accordarsi sui diversi possibili approcci
- ◆ Specificare un insieme preliminare di requisiti

Analisi e Negoziazione

Valutazione dei requisiti rispetto a: **completezza, conflitti** (tra viste diverse dei vari *stakeholders*), **compatibilità**

Checklist: per verificare se i requisiti hanno le caratteristiche desiderate

Requirements prioritization

Interaction matrix: per identificare conflitti

Risk Analysis

Elaborazione - Creare un modello di analisi che definisce il dominio informativo, funzionale e comportamentale del problema (ad es., UML)

Negoziazione - accordarsi su una definizione finale condivisa con gli *stakeholders*

Documentazione (specifica)

Specifica - un documento tecnico che descrive funzioni, prestazioni e vincoli del sistema.

Varie forme di specifica:

- Un documento testuale
- Un insieme di modelli
- Un modello matematico formale
- Un insieme di scenari utente (*use cases*)
- Un prototipo

Si può distinguere secondo gli stili di specifica tra:

- **Specific Informale**, esprime i requisiti in linguaggio naturale, (con anche tabelle, figure, ecc.) possibilmente in modo strutturato
- **Specific Formale**, ad es. automi a stati finiti, reti di Petri
- **Specific Semi-formale**, ad es. utilizzando notazioni semi-formali, come UML

Documentazione (specifiche)

Vari modelli/formalismi per problemi o classi di applicazioni diverse

- ✓ *Modelli orientati all'elaborazione dati (data-flow)*
- ✓ *Modelli basati su classificazione* (modelli ad oggetti)
- ✓ *Modelli basati su risposte a stimoli (real-time)*
- ✓ *Modelli orientati a processi* (reti di Petri)
- ✓ *Modelli basati sul comportamento* (diagr. di stato)
- ✓ ...

Il Documento di Specifica dei Requisiti (SRS)

- ◆ È una dichiarazione ufficiale di ciò che gli sviluppatori dovrebbero implementare (attenzione: il COSA ma non il COME)
- ◆ Deve includere sia i requisiti utente (RU), sia una specifica dettagliata dei requisiti di sistema (RS).
- ◆ Talora RU e RS sono contenuti nello stesso documento, talora in documenti separati.

SRS: Specifica dei Requisiti Software

Il documento di Specifica dei Requisiti Software (SRS) costituisce il punto di convergenza di tre diversi punti di vista: **cliente, utente, sviluppatore**

Un SRS serve a:

Costituire la base per la successiva progettazione

Costituire la base per la validazione del prodotto finale

Costituire la base per il contratto

- Evitando controversie, avvocati e tribunali...

Preparazione Congiunta dell' SRS

- ♦ Di solito né il fornitore né il committente sono in grado, da soli, di scrivere un buon SRS
 - Il **committente** non ha la necessaria conoscenza dei processi software per scrivere un buon SRS
 - Il **fornitore** non ha una conoscenza perfetta del dominio applicativo, e non può esprimere le effettive necessità
- ♦ Il processo di raccolta dei requisiti è un processo di confronto fra necessità e possibilità

Effetti di un errore nell' SRS

- ◆ Un SRS di qualità è il pre-requisito per un software di alta qualità
 - un errore nell' SRS produrrà errori nel sistema finale
- ◆ Un SRS di qualità riduce i costi di sviluppo
 - correggere un errore dell' SRS dopo lo sviluppo costa 100 volte più che correggerlo durante la fase di Analisi...
- ◆ Costi per la correzione di un errore nell' SRS, al variare della fase di sviluppo in cui è scoperto.

| FASE | COSTO (in ore uomo) |
|--------------------------|---------------------|
| Analisi dei Requisiti | 2 |
| Progettazione | 5 |
| Codifica | 15 |
| Test di accettazione | 50 |
| Esercizio e manutenzione | 150 |

Lo standard IEEE 830 -1998

- ◆ Definisce la struttura generica di un documento dei requisiti, che deve essere istanziata per ogni specifico sistema.
- ◆ Contiene più capitoli:
 1. Introduzione.
 2. Descrizione Generale.
 3. Requisiti Specifici.
 4. Appendici.
 5. Indice.

Struttura dell' SRS (Std IEEE 830 1998)

Table of contents

1. Introduction

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

2. General Description

- 2.1. Product Perspective (Relazione del prodotto con l'esterno)
- 2.2. Product Functions
- 2.3. User Characteristics
- 2.4. General Constraints
- 2.5. Assumptions and Dependencies

Struttura dell' SRS (Std IEEE 830 1998)

3. Specific Requirements

3.1. Functional Requirements

3.1.1. Functional Requirement 1

3.1.1.1. Introduction

3.1.1.2. Inputs

3.1.1.3. Processing

3.1.1.4. Output

3.1.2. Functional Requirement 2

...

3.1.3. Functional Requirement n

...

3.2. External Interface Requirements

- 3.2.1. User Interfaces
- 3.2.2. Hardware Interfaces
- 3.2.3. Software Interfaces
- 3.2.4. Communications Interfaces

3.3. Performances Requirements

3.4. Design Constraints

- 3.4.1 Standard Compliance
- 3.4.2. Hardware Limitation

...

3.5. Software System Attributes

- 3.5.1. Security
- 3.5.2. Maintainability

...

3.6. Other Requirements

- 3.6.1. Logical Database Requirements
- 3.6.2. Operations
- 3.6.3. Site Adaptation requirements

...

Formato per l' elenco dei requisiti (Cap. 2 dell' SRS)

- ◆ <id> Il sistema deve <funzione>

- ◆ Esempio:
 - RF1: Il sistema deve gestire tutti i registratori di cassa del negozio
 - RF2: Il sistema deve stampare la sintesi degli incassi della giornata
 - RF3: a fine giornata il sistema deve elencare gli articoli da riordinare, in base alle vendite
 - R4: il tempo di risposta di ogni interrogazione deve essere inferiore a 10 sec.

Esempio di Specifica di Funzionalità (Cap. 3 dell' SRS)

Ciascuna Funzionalità deve essere specificata indicando gli input, l' output e l' azione elaborativa. Ad esempio:

3.1.1 Inserimento articolo in archivio

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

Input: codice articolo, descrizione, unità di misura

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

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

Un'altra possibile struttura di SRS

- ◆ Prefazione
- ◆ Introduzione
- ◆ Glossario
- ◆ Definizione dei requisiti utente
- ◆ Architettura del sistema
- ◆ Specifica dei Requisiti di Sistema
- ◆ Modelli di sistema
- ◆ Evoluzione del sistema
- ◆ Appendici
- ◆ Indice

Le caratteristiche di qualità di un buon SRS ...

♦ Corretto

- se ogni requisito presente nell' SRS è realmente richiesto/ necessario per gli utenti del sistema finale

♦ Completo

- se ogni funzione richiesta al software ed il suo comportamento rispetto ad ogni possibile input è specificato

♦ Non Ambiguo

- ogni requisito ha una sola interpretazione (formale vs. informale)

♦ Verificabile

- se è possibile verificare che il sistema realizzi ogni requisito (richiede la non ambiguità dei requisiti)

... le caratteristiche di un buon SRS

◆ Consistente

- se nessun requisito è in contraddizione con gli altri

◆ Modificabile

- se la struttura e lo stile dell' SRS sono tali da consentire facili modifiche, preservando consistenza e completezza (un SRS con ridondanze non si modifica facilmente)

◆ Tracciabile

- se l' origine di ciascun requisito è chiara e può essere referenziata nello sviluppo futuro (def. IEEE)
- *forward traceability*: requisito collegabile a qualche elemento del progetto e del codice
- *backward traceability*: dal progetto e dal codice è possibile risalire al requisito corrispondente

Non ambiguità

- ♦ Ogni requisito deve avere sempre una sola interpretazione
 - Per ogni caratteristica usare sempre e solo lo stesso termine
 - Se un termine può avere diversi significati, in base al contesto, specificarlo in un Glossario
- ♦ È utile costruire un Data Dictionary, che dia una interpretazione univoca ai termini usati

Non ambiguità (esempio)

- ◆ Contesto: Sistema per la gestione di un ristorante
- ◆ RF1: Ogni ordine deve essere evaso dal responsabile
- ◆ Significati possibili:
 - Ordine per la cucina
 - Ordine per il fornitore
 - Ogni ordine ha un suo responsabile (cameriere?)
 - Tutti gli ordini hanno lo stesso responsabile
- ◆ Il linguaggio naturale è ambiguo
 - Sarebbe preferibile usare un Linguaggio Formale ...
 - ... ma non è per nulla facile!

Verificabilità

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

Tracciabilità

- ◆ Deve essere chiara l' origine di ogni requisito (tracciatura all' indietro)
- ◆ Ogni requisito ha un nome e un numero (tracciatura in avanti)
- ◆ Quando un requisito è conseguenza di un altro, i due devono essere tracciabili in avanti ed indietro
 - ◆ Es. Il requisito dei tempi di risposta della base dati è conseguenza di quello sui tempi di risposta del sistema complessivo
- ◆ Quando un requisito deriva da legislazioni, norme, ... si dovrebbero riportare i loro estremi

Validazione delle specifiche

Validazione: Revisione delle specifiche al fine di trovare

- ◆ Errori - Aree poco chiare
- ◆ Informazioni mancanti, inconsistenze
- ◆ Requisiti contrastanti o irrealizzabili

Nel caso dell' SRS, l' obiettivo è assicurare che esso rifletta accuratamente e con chiarezza i requisiti richiesti al software

Tipi di errori riscontrabili in un SRS:

- ◆ Omissione (mancata presenza di un requisito)
- ◆ Inconsistenza
- ◆ Incorrettezza (fatti non corretti nell' SRS)
- ◆ Ambiguità (requisiti con significati multipli)

SRS: Serve? Si fa ancora?

- ◆ In certi settori (ferroviario, avionico, *automotive*, spazio, elettromedicali, etc..) la specifica così descritta è **assolutamente obbligatoria**
- ◆ In questi settori esistono Enti che promulgano specifici Standard (ISO, IEEE, CELENEC, NIST, IEC, UNI, ...)
- ◆ Gli Standard definiscono il tipo di documenti da produrre (tra cui l' SRS)

Cosa fare in ambienti meno restrittivi?

- ◆ All' inizio dell' analisi conviene sempre produrre un SRS
- ◆ Poi i Casi d' Uso
- ◆ Poi incrociare i requisiti ed i casi d' uso

Validazione delle specifiche

- ❖ Revisioni ed ispezioni manuali
- ❖ Uso di Scenari relativi alle varie modalità operative del sistema
- ❖ Prototipazione
- ❖ Test dei requisiti
- ❖ Bozza Manuale Utente (utile a identificare possibili *mismatch*)

Gestione dei requisiti

- ◆ Man mano che il processo di sviluppo avanza, è probabile che vi siano cambiamenti/raffinamenti nei requisiti
- ◆ ***Requirements Management***: il processo di gestione di cambiamenti nei requisiti di sistema
 - Controllo dei cambiamenti:
 - ◆ Stima dell'impatto dei cambiamenti sugli altri requisiti e sulle altre attività

Punti chiave

- ◆ I Requisiti definiscono ciò che il sistema dovrebbe fare ed i vincoli sulla sua operatività ed implementazione.
- ◆ I requisiti funzionali descrivono funzionalità/servizi.
- ◆ I requisiti non-funzionali vincolano il sistema o il suo processo di sviluppo.
- ◆ I requisiti utente sono descrizioni ad alto livello di ciò che il sistema farà.
- ◆ I requisiti di sistema descrivono i dettaglio le funzioni che il software svolgerà.
- ◆ Un documento dei requisiti software è un documento di accordo sui requisiti del sistema.
- ◆ Lo standard IEEE è un utile punto di partenza per definire standard più dettagliati per la specifica dei requisiti.

