

Scrivere (e leggere) i requisiti

La stesura dei requisiti

- Stabilire **cosa** richiede il cliente ad un prodotto software
- (Non stabilire **come** il prodotto verrà costruito)

Obiettivi

- Cosa sono i requisiti di un prodotto software?
- Il processo di stesura dei requisiti
- Classificazione dei requisiti
- Notazioni per i requisiti

Ingegneria dei requisiti

- E' il processo di definizione dei:
 - **Servizi** che un cliente richiede ad un sistema
 - **Scenari** d'uso del sistema
 - **Vincoli di sviluppo** di un sistema
 - **Vincoli operativi** di un sistema

Che cos'è un requisito?

- Un requisito può essere una **descrizione** astratta e vaga di un servizio o vincolo di sistema, oppure una **specificata** dettagliata di una funzionalità del sistema
- I requisiti hanno infatti una doppia funzione:
 - Base per un *offerta di contratto*: devono essere aperti a varie interpretazioni
 - Base per un *contratto*: devono essere dettagliati e non soggetti ad interpretazioni arbitrarie

Domande utili per scrivere i requisiti

- Cosa deve fare il software?
- Che interfacce ha con i suoi utenti, con l'hw, con altro sw?
- Che prestazioni deve esibire il sw?
- Quali attributi (es. portabilità) dovrà avere?
- Quali vincoli dovrà soddisfare?

I requisiti sono “bifronti”

Se un'azienda desidera assegnare un contratto per lo sviluppo di un sistema software, scrive un documento per definire i suoi scopi in modo sufficientemente astratto da non predisporre una particolare soluzione. I requisiti devono essere scritti in modo che più contraenti possano rispondere al bando del contratto, offrendo magari diversi modi di realizzare la soluzione.

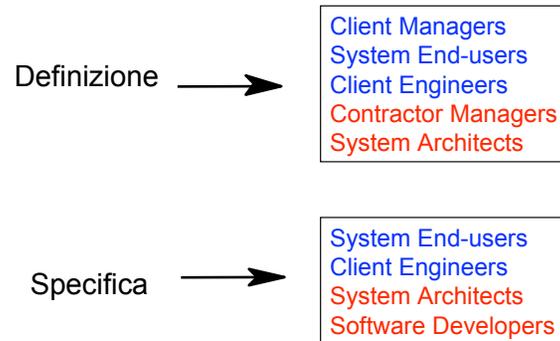
Dopo che il contratto è stato assegnato, il contraente finale deve scrivere per il cliente una definizione più dettagliata, in modo che il cliente possa comprendere e convalidare ciò che farà il sw

Entrambi questi documenti possono essere definiti “*documento dei requisiti*” del sistema da sviluppare

Specificata e definizione dei requisiti

- **Definizione dei requisiti**
 - Descrizione in linguaggio naturale dei servizi e dei vincoli operativi di un sistema
 - Si scrive per i clienti
- **Specificata dei requisiti**
 - Documento strutturato che dettaglia i servizi attesi dal sistema
 - Scritto come contratto tra cliente e contraente
 - Scritto per i contraenti e gli sviluppatori

Lettori dei requisiti



Inconsistenza dei requisiti

- E' difficile anticipare l'effetto di un nuovo servizio software sull'organizzazione che lo realizza
- Utenti diversi hanno priorità e requisiti diversi
- Gli utenti finali hanno aspettative diverse da quelle di chi finanzia lo sviluppo
- Per chiarire i requisiti è spesso utile l'uso di **prototipi**

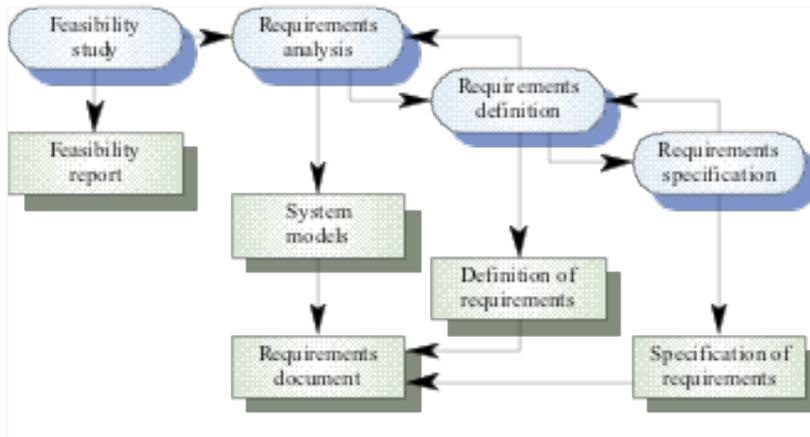
Scrivere requisiti col linguaggio naturale: problemi

- Quando si usa il linguaggio naturale, scrittori e lettori della specifica dovrebbero usare le stesse parole per lo stesso concetto
- Una specifica in linguaggio naturale è sempre soggetta a interpretazioni alternative
- I requisiti non vengono partizionati da strutture linguistiche specifiche

Alternative al linguaggio naturale

- Linguaggio naturale strutturato
- Notazioni visuali
- Specifiche formali

Il processo di stesura dei requisiti



Il documento dei requisiti

- Il documento dei requisiti contiene la definizione “ufficiale” di **cosa** viene richiesto agli sviluppatori
- Dovrebbe includere sia la definizione che la specifica dei requisiti
- **Non è un documento di progettazione**
- Per quanto possibile, dice **cosa** il sistema deve fare, non **come** lo farà

Requisiti dei documenti dei requisiti

- Deve specificare il comportamento “ai morsetti” del sistema (dal punto di vista esterno)
- Deve specificare i vincoli realizzativi
- Dev’essere facilmente modificabile
- Deve servire da riferimento per la manutenzione
- Deve predire possibili cambiamenti al sistema
- Deve definire le risposte del sistema ad eventi inaspettati

Struttura del documento dei requisiti

- Introduzione (definizione dei requisiti)
 - Descrivere perché si costruisce il sistema e come risponde ai bisogni dell’organizzazione
- Requisiti funzionali
 - Descrivere in dettaglio i servizi da fornire.
- Requisiti non-funzionali
 - Definire i vincoli sul sistema e sul suo processo di sviluppo
- Evoluzione del sistema
 - Definire le assunzioni critiche su cui si basa il sistema ed anticipare i possibili cambiamenti
- Glossario
 - Definire i termini tecnici che si usano
- Indici generale e analitico

La prima pagina del documento dei requisiti

- Nome del progetto o prodotto
- Data
- Numero di versione
- Autori
- Responsabilità di ciascun autore
- Cambiamenti principali dall'ultima versione

Classificazione dei requisiti

- I requisiti sono funzionali o non-funzionali
 - I **requisiti funzionali** descrivono i servizi o gli effetti di un sistema (eg. "Questa procedura stampa un documento")
 - I **requisiti non funzionali** sono vincoli sul sistema o sul suo processo di sviluppo (eg. "Questo documento dev'essere stampato su stampante a colori")
- Alcuni requisiti possono essere sia funzionali che non funzionali (eg. alcuni requisiti di sicurezza)

Definizione dei requisiti

- Specificare il *comportamento esterno* ("ai morsetti") del sistema
- I requisiti non dovrebbero indicare un *modello computazionale* o un'*architettura software* di riferimento

Esempi di requisiti funzionali

- *L'utente dovrà essere in grado di cercare o in tutti i database o in un loro sottoinsieme*
- *Il sistema fornirà i visualizzatori appropriati per permettere all'utente di leggere i documenti in memoria*
- *Ad ogni ordine verrà associato un identificatore unico (ORDER_ID) che l'utente potrà copiare nell'area di memoria permanente del suo conto*

Scrivere i requisiti

- In genere si usa per la definizione dei requisiti il **linguaggio naturale**
- Problemi col linguaggio naturale:
 - **Mancanza di chiarezza.** Più il documento è preciso, meno sarà leggibile
 - **Confusione.** I requisiti funzionali e non-funzionali vengono facilmente mescolati
 - **Amalgamazione.** Potranno essere espressi simultaneamente, in singole frasi, più requisiti

Formato per specifiche funzionali

- Definizione della funzione o entità
- Descrizione degli input e della loro provenienza
- Descrizione degli output e della loro destinazione
- Indicazione di altre entità richieste
- Pre e post condizioni (ove appropriato).

Requisiti con linguaggio naturale strutturato

- Una forma ristretta di linguaggio naturale si può usare per esprimere i requisiti
- Rimuove parte dei problemi di ambiguità e impone uniformità al documento di specifica
- Spesso si basa su una serie di documenti schematici (form)

Esempio di specifica funzionale in formato predefinito

ECLIPSE/Workstation/Tools/DE/FS/3.5.1	
Function	Add node
Description	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.
Inputs	Node type, Node position, Design identifier.
Source	Node type and Node position are input by the user, Design identifier from the database.
Outputs	Design identifier.
Destination	The design database. The design is committed to the database on completion of the operation.
Requires	Design graph rooted at input design identifier.
Pre-condition	The design is open and displayed on the user's screen.
Post-condition	The design is unchanged apart from the addition of a node of the specified type at the given position.
Side-effects	None
<i>Definition: ECLIPSE/Workstation/Tools/DE/RD/3.5.1</i>	

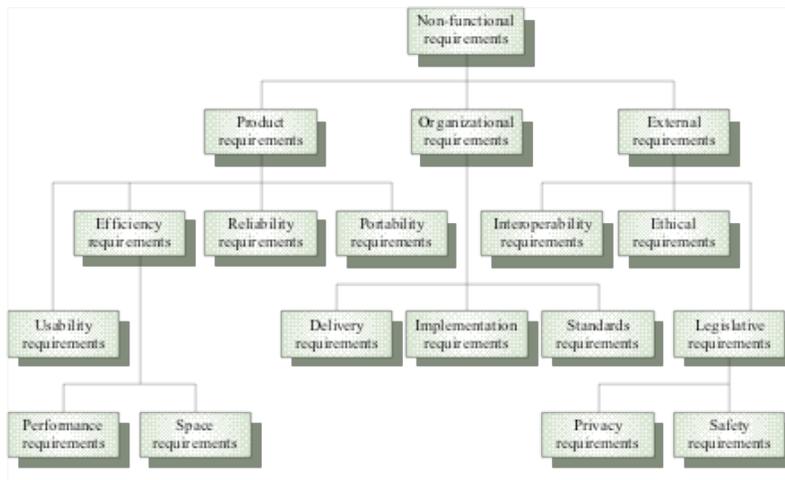
Spiegazione dei requisiti

- E' importante dare **spiegazioni** assieme ai requisiti
- Le spiegazioni aiutano gli sviluppatori a comprendere il dominio applicativo ed il perché di un requisito
- Le spiegazioni sono cruciali quando i requisiti debbono essere modificati: se ci sono si riduce la possibilità che il cambiamento abbia effetti inattesi

Requisiti non funzionali

- Definiscono proprietà e vincoli di sistema. Per esempio: affidabilità, tempo di risposta, dimensioni di memoria necessarie. Possibili vincoli sono: funzionalità di dispositivi I/O, relazioni con altri sistemi, ecc.
- Si possono anche specificare vincoli di processo, sul metodo di sviluppo, sul linguaggio di programmazione, sugli strumenti da usare
- I requisiti non funzionali possono essere più critici di quelli funzionali: se non vengono soddisfatti il sistema non verrà utilizzato

Tipi di requisiti non funzionali



Esempi di requisiti non funzionali

- Requisito di prodotto
 - 4.2.8 Per ogni comunicazione tra l'ambiente e l'utente dovrà essere possibile usare l'insieme di caratteri standard ASCII
- Requisito organizzativo
 - 9.3.2 Il processo di sviluppo ed i documenti consegnati saranno conformi al processo ed alle strutture di documento descritte nello standard ISO-XYWZ-2005
- Requisito esterno
 - 7.6.5 Il sistema non permetterà ai suoi operatori di conoscere alcuna informazione personale sui clienti, eccetto il nome ed il numero di riferimento interno

Requisiti d'utente

- I requisiti d'utente sono requisiti funzionali e non-funzionali comprensibili anche da utenti senza conoscenze tecniche
- Di solito si definiscono in linguaggio naturale e con l'aiuto di tabelle e diagrammi informali

Esempio: un requisito di un editor

2.6 Griglia Per aiutare a posizionare gli oggetti grafici su un diagramma, l'utente può attivare una griglia in centimetri o pollici, mediante un'opzione del pannello di controllo. Inizialmente la griglia è spenta. La griglia può essere attivata o disattivata in ogni momento durante una sessione di editing, così come può cambiare da centimetri a pollici o viceversa in qualsiasi momento. Esisterà un'opzione della vista "reduce-to-fit" per ridurre il numero di linee di griglia che viene visualizzato, allo scopo di non riempire il diagramma rimpicciolito con linee di griglia

Requisiti di sistema

- Sono requisiti sul sistema visto come entità unitaria
- Esempio
 - Il tempo richiesto per formare un operatore nell'uso efficace del sistema non deve superare i due giorni di lavoro
- Possono essere requisiti "emergenti", cioè non direttamente derivabili o conseguenti da alcun sottoinsieme dei requisiti

Discussione dell'esempio

- I requisiti dell'esempio sono sia funzionali che non funzionali e sono incompleti
- Facili da criticare, tuttavia difficili da scrivere in buona forma
- L'uso di un formato standard con campi predefiniti aiuta a inserire le informazioni necessarie senza dimenticarne qualcuna

Requisito griglia

2.6 Funzioni griglia

2.6.1 L'editor fornirà una funzione griglia, ovvero una matrice di linee orizzontali e verticali sullo sfondo della finestra di editing.

La griglia è passiva: l'allineamento delle entità grafiche sarà responsabilità dell'utente.

Spiegazione: Una griglia aiuta l'utente a creare un diagramma più elegante con entità ben allineate e separate. Potrebbe essere più utile una griglia attiva, dove le entità "saltano" sulle linee, ma il posizionamento risulterebbe impreciso comunque. E' meglio che sia l'utente a decidere dove vanno posizionate le entità.

2.6.2 In modalità 'reduce-to-fit' (vedi 2.1), la separazione tra le linee di griglia dev'essere aumentata.

Spiegazione: Se la spaziatura tra le linee non viene aumentata, lo sfondo si riempirà di linee di griglia.

Specifica: ECLIPSE/WS/Tools/DE/FS Sezione 5.6

Requisito di creazione di nodo

3.5.1 Aggiungere nodi ad un disegno

3.5.1.1 L'editor fornirà una funzione per cui un utente potrà aggiungere nodi di tipo specifico ad un disegno. I nodi vengono selezionati (vedi 3.4) quando vengono aggiunti al disegno.

3.5.1.2 Questa è la sequenza di azioni da compiere per aggiungere un nodo:

1. L'utente seleziona il tipo di nodo da aggiungere.
2. L'utente muove il cursore sulla posizione voluta nel diagramma e indica che il simbolo del nodo va aggiunto in quel punto.
3. Il simbolo viene posizionato al suo posto finale.

Spiegazione: L'utente è la persona più adatta a decidere dove posizionare il nodo nel diagramma. Questo approccio dà all'utente il controllo diretto sulla selezione del tipo ed il successivo posizionamento del nodo.

Specifica: ECLIPSE/WS/Tools/DE/FS. Section 3.5.1

Tracciabilità dei requisiti

- **Tracciare** i requisiti significa che i requisiti **correlati** vengono in qualche modo collegati ed magari collegati alla loro **fonte**
- La tracciabilità è una proprietà della specifica dei requisiti che permette di trovare facilmente i requisiti correlati

Tecniche di tracciabilità

- Assegnare un identificatore unico a ciascun requisito
- Costruire la lista dei riferimenti incrociati ai requisiti usando tale id
- Uso di link ipertestuali (es. HTML) per realizzare meccanismi di navigazione della tracciabilità

Verificabilità dei requisiti

- I requisiti vanno scritti in modo che possano essere **oggettivamente** verificati nel prodotto finale
- **Esempio:** *“Il sistema dovrebbe essere facile da usare da parte di operatori esperti ed organizzato in modo tale che gli errori degli utenti siano minimizzati”.*
- Questo requisito è vago: che significa “minimizzare gli errori”?
- Occorre **quantificare** il tasso degli errori.
 - *“Gli operatori esperti dovrebbero poter controllare le funzioni di sistema dopo due ore di formazione. Dopo tale formazione, il numero medio di errori degli operatori esperti non dovrebbe superare i due al giorno”*

Controllare i requisiti

- **Validità:** Il sistema fornisce le funzioni che soddisfano i bisogni del cliente?
- **Consistenza:** Ci sono conflitti tra i requisiti?
- **Completezza:** Sono incluse tutte le funzioni richieste dal cliente?
- **Realisticità:** I requisiti sono soddisfacibili dato il budget disponibile e la tecnologia corrente?

Validazione dei requisiti

- La dimostrazione che i requisiti definiscono il sistema davvero voluto dal cliente
- Il costo di un errore nei requisiti è molto alto, quindi la validazione è importante
 - La correzione di un errore nei requisiti dopo la consegna può costare oltre 100 volte il costo di correzione di un errore implementativo
- La tecnica più importante di validazione dei requisiti è la **prototipazione**

Revisioni dei requisiti

- Le revisioni dovrebbero essere fatte regolarmente durante il periodo di definizione dei requisiti
- Alle revisioni partecipano sia persone del cliente che del contraente
- Le revisioni possono essere formali (strutturate) o informali
- Le revisioni servono ad affrontare problemi o errori di specifica nella fase iniziale del processo

Controlli durante le revisioni

- **Verificabilità:** Il requisito è realisticamente controllabile?
- **Comprensibilità:** Il requisito si capisce correttamente?
- **Tracciabilità:** L'origine del requisito è definita correttamente?
- **Adattabilità:** Il requisito può essere modificato senza influenzare pesantemente gli altri requisiti?

Evoluzione dei requisiti

- I requisiti evolvono perché si capiscono meglio i bisogni dell'utente e gli obiettivi dell'organizzazione cambiano
- E' importante anticipare e pianificare il cambiamento dei requisiti mentre il sistema viene sviluppato ed usato

Letture strutturata

- È economica e rivela almeno il 60% degli errori [Boehm]
- **Esempio**
 - Analisi dei requisiti di CTC (Centralised Traffic Controller) delle ferrovie nordamericane - 1990
 - 10 gruppi di analisti in **parallelo**
 - Dei 92 difetti del documento dei requisiti
 - 77 vengono trovati durante l'ispezione dei requisiti
 - 15 nelle fasi successive
 - Ogni gruppo trova mediamente 'solo' 25 difetti.

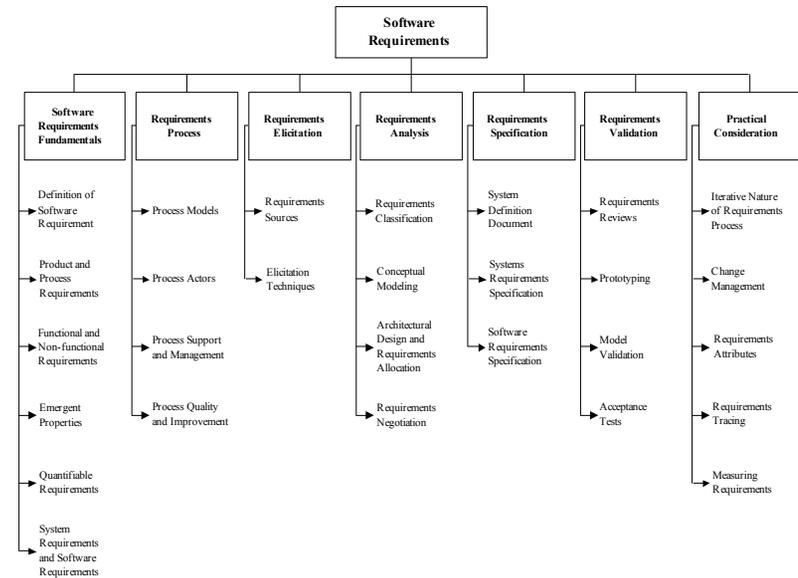
Classi di requisiti

- **Requisiti duraturi:** requisiti stabili, tipici del dominio dell'applicazione o dell'organizzazione che la userà
 - Esempio: regole di privacy negli ospedali
- **Requisiti effimeri:** requisiti che cambiano durante lo sviluppo o dopo che l'applicazione è divenuta operativa
 - Esempio: gestione di certe malattie negli ospedali

I requisiti nel SWEBOK

Moificare un documento dei requisiti

- Il documento dei requisiti dev'essere organizzato in modo che possa essere modificato senza bisogno di riscriverlo completamente
- Dev'essere modulare
- I riferimenti all'esterno del documento devono essere pochi, possibilmente nessuno (il documento dev'essere autosufficiente)



Riferimenti

- Capitolo 2 del SWEBOK. "Requirements Engineering"
- 830-1998 *Recommended Practice for Software Requirements Specifications* (ANSI/IEEE)
- "*Requirements Engineering. A good practice guide*", Ian Sommerville and Pete Sawyer, John Wiley and Sons, 1997
- "Managing Software Requirements; A Unified Approach," Dean Leffingwell and Don Widrig, Addison-Wesley, 2000
- sourceforge.net/projects/eclipse/srs/

Domande?

