

Copyright © 2004 Moreno Marzolla

This work is licensed under the Creative Commons Attribution-Noncommercial-Share Alike 2.5 Italy License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-sa/2.5/it/> or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.

Ingegneria dei requisiti

- Stabilire che cosa richiedono gli utenti da un sistema software

Ingegneria dei requisiti

- E' il processo di stabilire i servizi che i clienti richiedono dal sistema, e i vincoli sotto cui il sistema viene sviluppato e opererà
- Due tipi di requisiti:
 - Funzionali: Descrivono le funzionalità e i servizi offerti dal sistema
 - Non-funzionali: Descrivono vincoli che il sistema o il processo di sviluppo devono soddisfare

Cos'è un requisito?

- Può variare da una descrizione informale dei servizi che il sistema deve fornire, o dei vincoli che deve soddisfare, a una specifica funzionale dettagliata e formalmente definita
- La ragione è che i requisiti servono molteplici scopi
 - Come base per una offerta di contratto, quindi aperti alle interpretazioni
 - Come base del contratto stesso, quindi definiti in dettaglio

L'Ingegneria dei requisiti

- Porta alla specifica delle caratteristiche operative (funzionalità, dati e comportamenti) del software
- Indica una interfaccia del software con gli altri elementi del sistema
- Stabiliscono i vincoli cui il sistema deve sottostare

Definizione/Specifica dei requisiti

- Definizione dei requisiti
 - Delle frasi in linguaggio naturale, più eventuali diagrammi dei servizi che il sistema deve offrire e dei suoi vincoli operativi. E' rivolta al cliente
- Specifica dei requisiti
 - Documento strutturato che fissa una descrizione dettagliata dei servizi che il sistema deve offrire. Viene scritta come contratto tra cliente e fornitore del software
- Specifica del software
 - Una descrizione dettagliata del software che serva come base per il disegno dell'architettura o per l'implementazione. E' rivolta agli sviluppatori.

Definizioni e specifiche

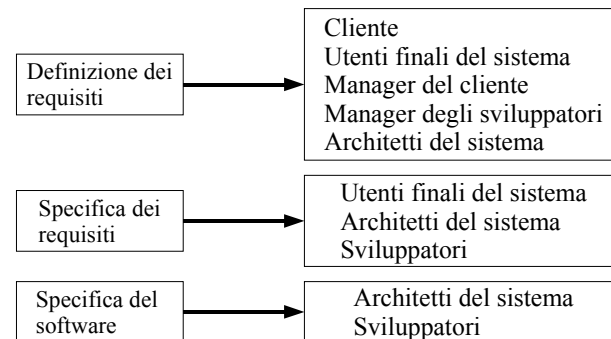
User requirement definition

1. The software must provide a means of representing and accessing external files created by other tools.

System requirements specification

- 1.1 The user should be provided with facilities to define the type of external files.
- 1.2 Each external file type may have an associated tool which may be applied to the file.
- 1.3 Each external file type may be represented as a specific icon on the user's display.
- 1.4 Facilities should be provided for the icon representing an external file type to be defined by the user.
- 1.5 When a user selects an icon representing an external file, the effect of that selection is to apply the tool associated with the type of the external file to the file represented by the selected icon.

A chi sono rivolti i requisiti



Il sistema LIBSYS

- Consideriamo un ipotetico sistema che fornisca una interfaccia unica ad un insieme di database di articoli e documenti.
- Gli utenti possono effettuare ricerche, acquistare, scaricare e stampare gli articoli per uso e studio personale.

Esempi di requisiti funzionali

- “L'utente deve essere in grado di effettuare ricerche su tutti i database disponibili o su un sottoinsieme di essi.”
- “Il sistema deve fornire dei *viewers* appropriati per consentire agli utenti di leggere i documenti disponibili.”
- “Per ogni ordine di acquisto deve esistere un identificatore unico (ORDER_ID) che l'utente deve poter salvare in modo permanente.”

Imprecisione dei requisiti

- Considerare i termini “*viewers* appropriati”
 - Nelle intenzioni degli utenti—programmi special-purpose in grado di leggere tipi specifici di documenti;
 - Interpretazione degli sviluppatori—un unico visualizzatore di testo che mostri il contenuto di qualsiasi tipo di documento.

Ragioni per le inconsistenze

- Utenti diversi hanno diversi requisiti e priorità. I requisiti sono frutto di un compromesso
- Utenti finali e committenti (coloro che pagano per far sviluppare il sistema) possono avere requisiti diversi
- La prototipazione è spesso molto utile per chiarire i requisiti

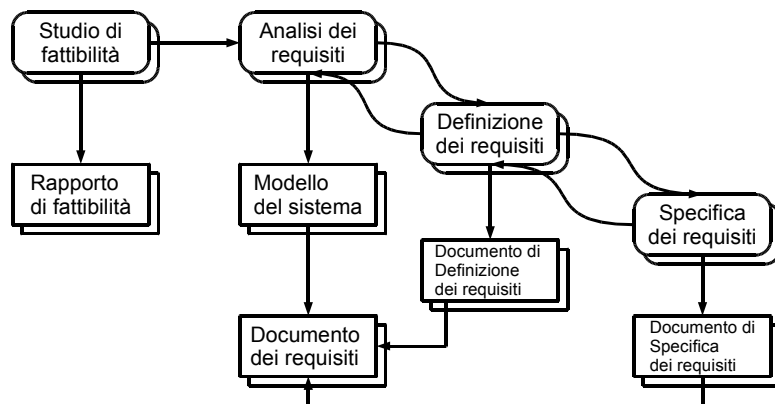
Requisiti non funzionali

- Definiscono proprietà e vincoli sul sistema (es., affidabilità, tempo di risposta, occupazione di memoria).
- Possono essere dati anche vincoli di processo, che impongono l'uso di un particolare sistema CASE, linguaggio di programmazione e/o metodo di sviluppo.
- I requisiti non funzionali possono essere maggiormente critici rispetto ai requisiti funzionali.
 - Se i requisiti funzionali non sono soddisfatti, il sistema può risultare inutilizzabile.

Il processo di Ingegneria dei Requisiti

- Studio di fattibilità
 - Le esigenze del cliente possono essere soddisfatte considerando il livello tecnologico corrente e il budget a disposizione?
- Analisi dei requisiti
 - Individuare cosa viene richiesto dal sistema
- Definizione dei requisiti
 - Descrivere i requisiti in forma comprensibile per il cliente
- Specifica dei requisiti
 - Definire i requisiti in dettaglio

Le fasi del processo di Ingegneria dei Requisiti



Il Documento dei Requisiti

- Il documento dei requisiti definisce ufficialmente ciò che viene richiesto dal sistema
 - Cioè cosa è richiesto che gli sviluppatori facciano
- Include sia la specifica che la definizione dei requisiti
- Non è un documento di design
 - Dovrebbe dire COSA il sistema deve fare, piuttosto di COME lo deve fare

Avvio del processo / 1

- La tecnica più usata per individuare i requisiti consiste nel condurre riunioni o interviste
- Di solito si parte con domande acontestuali, cioè domande che conducano a fissare in termini generali il problema
 - Da dove parte la richiesta di questo lavoro?
 - Chi utilizzerà la soluzione
 - Quali saranno i benefici economici di una soluzione ottimale?
 - La soluzione si può trovare altrove?

Avvio del processo / 2

- Successivamente l'analista pone domande per comprendere in maniera più approfondita il problema, e per dare al cliente l'opportunità di esprimere la soluzione percepita
 - Quali sono, secondo voi, le caratteristiche di un output "soddisfacente" prodotto dalla soluzione?
 - Quali problemi deve affrontare la soluzione?
 - Potete mostrare (o descrivere) l'ambiente in cui la soluzione dovrà operare?
 - Esistono questioni specifiche relative alle prestazioni o vincoli che incidono sulla soluzione?

Avvio del processo / 3

- L'ultimo gruppo di domande concerne l'efficacia della riunione
 - Siete voi la persona più indicata per rispondere alle mie domande? Le vostre risposte sono "ufficiali"?
 - Le mie domande sono rilevanti rispetto al problema?
 - Sto facendo troppe domande?
 - Ci sono persone in grado di darmi informazioni supplementari?
 - C'è qualcos'altro che dovrei chiedere?

Validazione dei requisiti

- Dimostrare che i requisiti definiti sul sistema coincidono esattamente con ciò che vuole il cliente
- Errori nei requisiti causano costi elevati, quindi la validazione è importante
 - Correggere un errore nei requisiti è molto più costoso che correggere un errore di implementazione
- La prototipazione è un mezzo efficace per validare i requisiti

Controllo dei requisiti / 1

- Validità
 - Il sistema fornisce le funzionalità che meglio soddisfano le esigenze del cliente?
- Consistenza
 - Ci sono requisiti in conflitto fra di loro?
- Completezza
 - Tutti i requisiti del cliente sono stati individuati?
- Realismo
 - I requisiti possono essere implementati data la tecnologia e il budget a disposizione?

Controllo dei requisiti / 2

- Verificabilità
 - Il requisito può essere realisticamente testabile?
- Comprensibilità
 - Il requisito è stato adeguatamente compreso?
- Tracciabilità
 - L'origine del requisito è chiaramente indicata?
- Adattabilità
 - Può il requisito essere cambiato senza grossi impatti sugli altri requisiti?

Revisione dei requisiti

- E' opportuno effettuare revisioni periodiche durante la formulazione dei requisiti
 - Revisioni formali
 - Revisioni informali
- Le revisioni dovrebbero coinvolgere sia il cliente che l'azienda che implementa il sistema
- Una adeguata comunicazione tra cliente, sviluppatori e utenti finali può aiutare a individuare e risolvere problemi appena si manifestano

FAST

(Facilitated Application Specification Technique)

- E' una tecnica per la specifica di applicazioni
- Promuove la formazione di un team congiunto di clienti e sviluppatori che collaborano a:
 - Individuare il problema
 - Proporre elementi della soluzione
 - Negoziare strategie diverse
 - Specificare una famiglia preliminare di requisiti

Fasi di FAST / 1

- Si svolge una riunione a cui partecipano ingegneri del software e clienti
- Si stabiliscono le regole per la preparazione e la partecipazione
- Si propone un'agenda
 - Sufficientemente formale da abbracciare tutti i punti importanti
 - Abbastanza informale da incoraggiare l'apporto di nuove idee

Fasi di FAST / 2

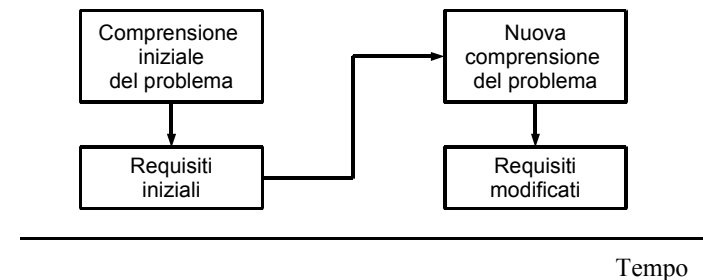
- Un “moderatore” modera la riunione
 - Può essere un cliente, uno sviluppatore, una persona esterna
- Si applica un “meccanismo di definizione”
 - Una lavagna, fogli appesi ad una parete o altro
- Lo scopo è
 - Individuare il problema
 - Proporre eventuali soluzioni
 - Confrontare strategie diverse
 - Specificare un insieme preliminare di requisiti

Pianificazione collaborativa

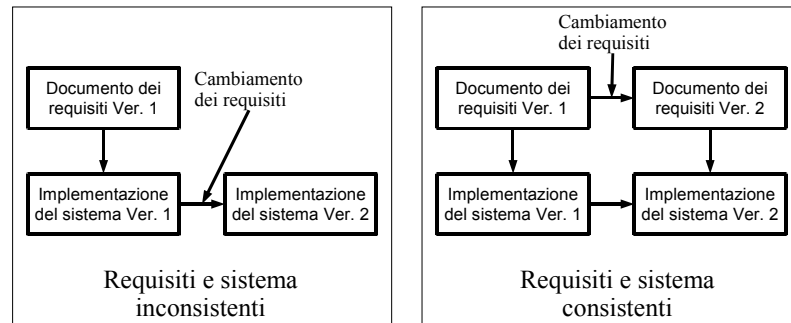


Evoluzione dei requisiti

- I requisiti evolvono man mano che le esigenze del cliente sono meglio comprese e gli obiettivi dell'organizzazione cambiano
- E' necessario tener conto di questa mutabilità dei requisiti mentre il sistema viene implementato e viene utilizzato



Evoluzione controllata



Classi di requisiti

- **Requisiti duraturi**
 - Requisiti stabili che derivano dalla natura stessa dell'attività del cliente. Ad esempio, un ospedale avrà sempre a che fare con pazienti, medici, infermieri...
- **Requisiti volatili**
 - Requisiti che possono cambiare durante lo sviluppo del sistema, o quando questo è in uso. Ad esempio, in un ospedale, i requisiti derivanti dalle politiche di cura dei pazienti, o dalla politica sanitaria nazionale

Casi d'uso / 1

- Man mano che i requisiti vengono raccolti, l'analista può creare un insieme di scenari che identificano una tendenza d'uso per il sistema
- Gli *Scenari* (detti anche *Casi d'Uso*) descrivono i modi in cui il sistema verrà usato
 - **Attori**
Utenti o altre entità che interagiscono con il sistema. Formalmente, un attore è qualunque cosa che comunica con il sistema ma è esterno ad esso
 - **Casi d'uso**
Descrivono il modo in cui un attore interagisce con il sistema

Casi d'uso / 2

- **Domande a cui trovare risposte**
 - Quali compiti o funzionalità vengono svolte dall'attore?
 - Quali informazioni di sistema vengono acquisite, prodotte o modificate dall'attore?
 - L'attore dovrà informare il sistema sui cambiamenti nell'ambiente esterno?
 - Quali informazioni l'attore desidera dal sistema?
 - L'attore vuole essere informato di ogni cambiamento inatteso?