

9. Analisi dei requisiti (II)

Prof. Paola Barra
a.a. 2022/2023

Requisiti funzionali
VS
Requisiti non funzionali

Tipologie di requisiti

- Requisiti funzionali

- Descrivono le interazioni tra il sistema ed il suo ambiente indipendentemente dalla implementazione

- Requisiti non funzionali

- Aspetti non direttamente legati ad un comportamento funzionale
 - «il tempo di risposta deve essere inferiore al secondo»

- Vincoli

- Imposti dal cliente o dall'ambiente
 - «Il linguaggio per l'implementazione deve essere Java»
 - Chiamato anche «pseudo requisiti»

Requisiti funzionali VS requisiti non funzionali

Requisiti funzionali

Descrivono compiti utente che il sistema deve supportare

Espressi come azioni

- «Pubblicizza una nuova lega»
- «Programma un calendario»
- «Notifica un nuovo gruppo di interesse»

Requisiti non funzionali

Descrivono le proprietà del sistema o del dominio

Espressi come vincoli o affermazioni negative

- «tutti gli input dell'utente devono fornire la conferma di ricezione entro un secondo»
- «Un crash del sistema non dovrebbe comportare perdita di dati»

REQUISITI FUNZIONALI

- Descrivono le funzionalità che il sistema deve realizzare
 - In termini di:
 - azioni che il sistema deve compiere
 - come il sistema software reagisce a specifici tipi di input
 - come si comporta in situazioni particolari.
- Tradizionalmente i requisiti a cui è dato maggior valore
- Esempi
 1. Il sistema software deve fornire un appropriato visualizzatore per i documenti memorizzati
 2. L'utente deve essere in grado di effettuare ricerche sia sull'intero insieme di basi di dati che su un loro sottoinsieme
 3. Ad ogni nuovo ordine deve essere associato un identificatore unico (Order_ID)

REQUISITI NON FUNZIONALI

- Aka requisiti di qualità
- Descrivono le proprietà del sistema software in relazione a determinati servizi o funzioni e possono anche essere relativi al processo:
 - caratteristiche di qualità quali:
 - efficienza, affidabilità, safety, usabilità, interfaccia, security, robustezza ecc.
 - caratteristiche del processo di sviluppo:
 - standard di processo, uso di ambienti CASE, linguaggi di programmazione, metodi di sviluppo, ecc.
 - caratteristiche esterne:
 - interoperabilità con sistemi di altre organizzazioni, vincoli legislativi, ecc.
 - requisiti fisici (hw)

Requisiti funzionali per SatWatch

SatWatch è un orologio da polso che visualizza l'ora in base alla posizione corrente

- SatWatch usa il GPS per determinare la propria posizione e struttura dati interne per convertire questa ubicazione in un fuso orario

Le informazioni memorizzate in SatWatch e la sua precisione sono tali per cui il proprietario non ha mai la necessità di resettare l'ora

- SatWatch regola l'ora e la data visualizzata ogniqualvolta il proprietario attraversa fusi orari o confini politici. Per questa ragione, SatWatch non ha pulsanti o controlli per l'utente

SatWatch determina la posizione usando un satellite GPS e, come tale, soffre delle stesse limitazioni di tutti i dispositivi GPS (ad esempio, incapacità di determinare la posizione in certi orari del giorno in regioni montagnose).

- Durante i momenti di black-out, SatWatch assume di non attraversare fusi orari o confini politici. SatWatch corregge il proprio fuso orario non appena termina il momento di black-out

Requisiti funzionali per SatWatch (II)

SatWatch ha un display a due linee che mostra, sulla linea in alto, l'ora (ore, minuti, secondi, fuso orario) e sulla linee in basso, la data (giorno, mese, anno)

- La tecnologia del display usata è tale che il proprietario dell'orologio può guardare l'ora e la data anche al buio

Quando cambiano i confini politici, il proprietario dell'orologio può aggiornare il software dell'orologio usando il dispositivo WebifyWatch (fornito con l'orologio) e un pc connesso ad Internet

Acquisire e descrivere i requisiti funzionali : Casi d'uso

- I casi d'uso sono un altro modo per acquisire i requisiti
- Il caso d'uso è un modo in cui un utente può usare il prodotto
- Si prospetta al committente insieme al risultato atteso e si aspettano i commenti
- I casi d'uso devono includere non solo la sequenza di eventi corretta ma anche comportamenti inattesi: **le eccezioni.**

Tipi di requisiti non funzionali

Requisiti di qualità

- Usabilità
- Affidabilità
 - Robustezza
 - Sicurezza
- Prestazioni
 - Tempo di risposta
 - Scalabilità
 - Throughput
 - Disponibilità
- Supportabilità
 - Adattabilità
 - Manutenibilità

Implementazione

Interfaccia

Funzionamento

Packaging

Legale

- Licenza (GPL, LGPL)
- Certificazione
- Regolamentazione

Vincoli o pseudo requisiti

Esempi di requisiti non funzionali

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

Requisiti funzionali VS requisiti non funzionali

- E' bene tenerli separati.
- Esempio:
 - Il sistema deve validare il pin inserito dal cliente entro 3 secondi



- Il sistema deve validare il pin inserito dal cliente
- La validazione del pin deve essere completata entro 3 secondi

Esercizio Bancomat

Si consideri un sistema per il prelievo automatizzato di denaro contante (Bancomat). Per poter utilizzare il sistema l'utente deve possedere una tessera magnetica. Il sistema deve mettere a disposizione le funzioni di prelievo, saldo, estratto conto. Il sistema deve essere disponibile a persone non vedenti, deve garantire un tempo di risposta inferiore al minuto, e deve essere sviluppato su architettura X86. Le operazioni di prelievo devono richiedere autenticazione tramite un codice segreto memorizzato sulla carta. Il sistema deve essere facilmente espandibile, e adattabile alle future esigenze bancarie.

Curiosità

- Si calcola che il **95% delle operazioni di un bancomat** venga eseguita tramite software scritto in **Cobol** e che in totale ci siano 200 miliardi di righe di codice che gestiscono ogni giorno 30 miliardi di transazioni.

[Antonio Dini su Wired 10-1-22]

Qualche definizione di requisiti di qualità

- **Usabilità:** Facilità con cui gli attori possono eseguire una funzione in un sistema
 - Usabilità è uno dei termini più spesso usato in modo improprio («il sistema è facile da usare»)
 - L'**usabilità** deve essere misurabile, altrimenti è solo marketing
 - Esempio: specifica del numero di passi – la misura! – per eseguire un acquisto su internet con un browser
- **Robustezza:** La capacità di un sistema di garantire una funzione
 - Anche se l'utente immette un input errato
 - Anche se ci sono cambiamenti nell'ambiente
 - Esempio: il sistema può tollerare temperature fino a 90°C
- **Disponibilità:** Il rapporto tra il tempo atteso di funzionamento e la somma dei tempi attesi di funzionamento e non funzionamento
 - Esempio: il sistema è *down* non più di 5 minuti a settimana

Requisiti di qualità per Sat Watch

- Un utente che sa come leggere un orologio digitale e capire le abbreviazioni del fuso orario internazionale dovrebbe essere in grado di usare *SatWatch* senza il manuale utente [[Usabilità](#)]
- Poiché *SatWatch* non ha pulsanti, non dovrebbero verificarsi errori software che richiedono il reset dell'orologio [[Affidabilità](#)]
- *SatWatch* dovrebbe visualizzare il fuso orario corretto dopo al più 5 minuti dal termine del periodo di black-out del GPS [[Prestazioni](#)]
- *SatWatch* dovrebbe visualizzare correttamente l'ora in tutti i 24 fusi [[Prestazioni](#)]
- *SatWatch* dovrebbe accettare aggiornamenti via interfaccia seriale Webify Watch [[Supportabilità](#)]

Vincoli per SatWatch

- Tutti i software associati con *SatWatch*, incluso il software proprietario, saranno scritti usando Java, per conformarsi con la corrente politica della compagnia [[Implementazione](#)]
- *SatWatch* è conforme con le interfacce fisiche, elettriche e software definite dalle API Webify Watch 2.0 [[Interfaccia](#)]

Requisiti non funzionali: altri esempi

- «gli spettatori devono essere in grado di guardare un incontro senza doversi prima registrare e senza avere informazioni preliminari sull'incontro»
 - Requisito di usabilità
- «il sistema deve supportare 10 turni in parallelo»
 - Requisito di prestazione
- «L'operatore deve essere in grado di aggiungere nuove partite senza modificare il sistema esistente»
 - Requisito di supportabilità

Cosa non dovrebbe andare nei requisiti?

- Struttura del sistema, tecnologia di implementazione
- Metodologia di sviluppo
- Ambiente di sviluppo
- Linguaggio di implementazione
- Riutilizzabilità
- E' desiderabile che nessuno di questi siano vincolati dal cliente

Validazione dei requisiti

- La validazione dei requisiti è un passo per assicurare la qualità, di solito eseguito dopo la scoperta dei requisiti o dopo l'analisi
- Correttezza
 - I requisiti rappresentano la visione dell'utente
- Completezza
 - Sono descritti tutti i possibili scenari in cui il sistema può essere usato
- Consistenza
 - Non ci sono requisiti in contraddizione reciproca

Validazione dei requisiti (2)

- Chiarezza

- I requisiti possono essere interpretati in un unico modo

- Realismo

- I requisiti possono essere implementati e consegnati

- Tracciabilità

- Ogni componente e comportamento del sistema può essere imputato ad un insieme di requisiti funzionali

- Problemi con la validazione dei requisiti

- I requisiti cambiano velocemente durante la scoperta
- Le inconsistenze si aggiungono facilmente con le modifiche
- Necessari tool di supporto

Tool per la gestione dei requisiti

- DOORS (IBM/Rational)
 - Gestore di requisiti multiplatforma per team che lavorano nella stessa locazione geografica
- RequisitePro (IBM/Rational)
 - Integrazione con MS-Word
 - Confronto tra progetti
- Unicae (<http://unicase.org>)
 - Tool per lo sviluppo collaborativo dei modelli di sistema
 - I partecipanti possono essere distribuiti geograficamente

Differenti tipi di scoperta dei requisiti

- **Greenfield Engineering**

- Lo sviluppo parte da zero, non c'è alcun sistema precedente, i requisiti provengono dagli utenti e dai clienti
- Innescata dalle necessità dell'utente

- **Re-engineering**

- Riprogettazione e/o re-implementazione di un sistema esistente usando una tecnologia più moderna
- Innescata dalla nuova tecnologia

- **Interface engineering**

- Fornitura di servizi esistenti in un nuovo ambiente
- Innescata dalla nuova tecnologia e da nuove necessità del mercato

Assegnare priorità ai requisiti

- Alta priorità

- Affrontato durante l'analisi, la progettazione e l'implementazione
- Un caratteristica ad alta priorità deve essere dimostrata

- Media priorità

- Affrontato durante l'analisi e la progettazione
- Solitamente dimostrato nella seconda iterazione

- Bassa priorità

- Affrontato solo durante l'analisi
- Illustra come il sistema sarà usato in futuro con una tecnologia non ancora disponibile

Modello del documento di analisi dei requisiti (RAD)

1. Introduzione
2. Sistema corrente
3. Sistema proposto
 - 3.1 Panoramica
 - 3.2 Requisiti funzionali
 - 3.3 **Requisiti non funzionali**
 - 3.4 Vincoli (“Pseudo requisiti”)
 - 3.5 Modelli del sistema
 - 3.5.1 Scenari
 - 3.5.2 Modello casi d’uso
 - 3.5.3 Modello ad oggetti
 - 3.5.3.1 Dizionario dei dati
 - 3.5.3.2 Diagrammi delle classi
 - 3.5.4 Modelli dinamici
 - 3.5.5 Interfaccia utente
4. Glossario

Sezione 3.3 : Requisiti non funzionali

3.3.1 Interfaccia utente e fattori umani

3.3.2 Documentazione

3.3.3 Considerazioni hardware

3.3.4 Caratteristiche delle prestazioni

3.3.5 Gestione errori e condizioni limite

3.3.6 Interfacciamento del sistema

3.3.7 Problemi di qualità

3.3.8 Modifiche del sistema

3.3.9 Ambiente fisico

3.3.10 Problemi di sicurezza

3.3.11 Problemi di risorse e gestione

Requisiti non funzionali

(domande per superare il blocco dello scrittore)

- Interfaccia utente e fattori umani
 - Che tipo di utenti useranno il sistema?
 - Il sistema sarà usato da più di una tipologia di utenti?
 - Che addestramento sarà richiesto per ogni tipologia di utenti?
 - E' importante che il sistema si facile da imparare?
 - Gli utenti dovrebbero essere protetti dal fare errori?
 - Che tipo di periferiche di input/output sono disponibili?
- Documentazione
 - Che tipo di documentazione è richiesta?
 - Che uditorio dovrebbe essere affrontato da ogni documento?

Requisiti non funzionali (2)

- Considerazioni hardware
 - Su quale hardware deve essere usato il sistema proposto?
 - Quali sono le caratteristiche dell'hardware, incluso dimensione della memoria e spazio di memorizzazione ausiliario?
- Caratteristiche delle prestazioni
 - Sul sistema ci sono vincoli di velocità, throughput, tempi di risposta?
 - Ci sono vincoli di dimensione o capacità sui dati da elaborare?
- Gestione errori e condizioni limite
 - Il sistema come dovrebbe rispondere agli errori di input?
 - Il sistema come dovrebbe rispondere a condizioni limite?

Requisiti non funzionali (3)

- Interfacciamento del sistema
 - Ci sono input provenienti da sistemi esterni al sistema proposto?
 - L'output è diretto a sistemi esterni al sistema proposto?
 - Ci sono restrizioni sul formato o sul mezzo che deve essere usato per l'input o l'output?
- Problemi di qualità
 - Quali sono i requisiti per l'affidabilità?
 - Il sistema deve intercettare i guasti?
 - Qual è il tempo per riavviare il sistema dopo un fallimento?
 - C'è un downtime accettabile su 24h?
 - E' importante che il sistema sia portatile?

Requisiti non funzionali (4)

- Modifiche al sistema
 - Quali parti del sistema è probabile modificare?
 - Che tipi di modifiche ci si attende?
- Ambiente fisico
 - Dove funzionerà il sistema?
 - E' distribuito in diverse posizioni?
 - Le condizioni ambientali saranno normali?
- Problemi di sicurezza
 - L'accesso ai dati o al sistema deve essere controllato?
 - La sicurezza fisica è un problema?

Requisiti non funzionali (5)

- Problemi di risorse e gestionali
 - Quanto spesso il sistema sarà sottoposto a backup?
 - Chi sarà il responsabile del backup?
 - Chi sarà responsabile della installazione del sistema?
 - Chi sarà responsabile per la manutenzione del sistema?