



# Ingegneria dei Requisiti

## Cosa? - difficoltà, tecniche e processi

Andrea Polini

Ingegneria del Software  
Corso di Laurea in Informatica

# Sommario

- 1 Generalità
- 2 Attività dell'ingegneria dei requisiti
  - Studio di fattibilità
  - Elicitazione
  - Validazione
  - Gestione
- 3 Documenti dei Requisiti

# Sommario

- 1 Generalità
- 2 Attività dell'ingegneria dei requisiti
  - Studio di fattibilità
  - Elicitazione
  - Validazione
  - Gestione
- 3 Documenti dei Requisiti

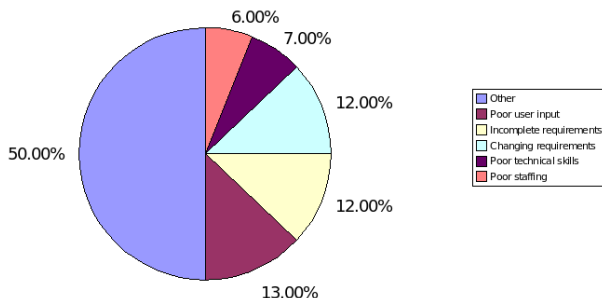
# Ingegneria dei Requisiti

Disciplina che si occupa di definire **cosa un sistema debba fare**, le sue **proprietà essenziali** ed i vincoli a cui deve rispondere. Scoprire, analizzare, documentare e validare i requisiti sono attività investigate dalla disciplina dell'**ingegneria dei requisiti**

Attività che presentano forte **interazione e comunicazione con il cliente**. Dunque non soltanto attività dagli aspetti tecnici ma forti **implicazioni socio-antropologiche** (ci si riferisce a questa attività come “comunicazione”)

# Rilevanza dei Requisiti

- La fase di gestione dei requisiti è probabilmente la più critica
- Problemi inseriti in questa fase dello sviluppo sono i più costosi da rimuovere.
- Studi rivelano che circa il 37% dei problemi, nello sviluppo di sistemi software “challenging”, sono relativi alla fase dei requisiti



# Tipici problemi

- La visione e lo scopo del progetto non sono state chiaramente definiti
- I clienti erano troppo impegnati per lavorare con gli analisti e gli sviluppatori
- Il team di sviluppo non ha potuto mai interagire con gli utenti
- I clienti hanno dichiarato tutti i requisiti come critici, nessuna prioritizzazione
- Gli sviluppatori hanno incontrato ambiguità durante lo sviluppo
- La comunicazione si è focalizzata su come le interfacce dovessero essere rappresentate e non sulle funzionalità
- Il cliente non ha mai approvato i requisiti
- L'ambito del progetto è continuamente aumentato mentre lo sviluppo progrediva
- Richieste di modifiche ai requisiti sono state ignorate
- ...

# Tipici problemi

- La visione e lo scopo del progetto non sono state chiaramente definiti
- I clienti erano troppo impegnati per lavorare con gli analisti e gli sviluppatori
- Il team di sviluppo non ha potuto mai interagire con gli utenti
- I clienti hanno dichiarato tutti i requisiti come critici, nessuna prioritizzazione
- Gli sviluppatori hanno incontrato ambiguità durante lo sviluppo
- La comunicazione si è focalizzata su come le interfacce dovessero essere rappresentate e non sulle funzionalità
- Il cliente non ha mai approvato i requisiti
- L'ambito del progetto è continuamente aumentato mentre lo sviluppo progrediva
- Richieste di modifiche ai requisiti sono state ignorate
- ...

# Tipici problemi

- La visione e lo scopo del progetto non sono state chiaramente definiti
- I clienti erano troppo impegnati per lavorare con gli analisti e gli sviluppatori
- Il team di sviluppo non ha potuto mai interagire con gli utenti
- I clienti hanno dichiarato tutti i requisiti come critici, nessuna prioritizzazione
- Gli sviluppatori hanno incontrato ambiguità durante lo sviluppo
- La comunicazione si è focalizzata su come le interfacce dovessero essere rappresentate e non sulle funzionalità
- Il cliente non ha mai approvato i requisiti
- L'ambito del progetto è continuamente aumentato mentre lo sviluppo progrediva
- Richieste di modifiche ai requisiti sono state ignorate
- ...



# Tipici problemi

- La visione e lo scopo del progetto non sono state chiaramente definiti
- I clienti erano troppo impegnati per lavorare con gli analisti e gli sviluppatori
- Il team di sviluppo non ha potuto mai interagire con gli utenti
- I clienti hanno dichiarato tutti i requisiti come critici, nessuna prioritizzazione
- Gli sviluppatori hanno incontrato ambiguità durante lo sviluppo
- La comunicazione si è focalizzata su come le interfacce dovessero essere rappresentate e non sulle funzionalità
- Il cliente non ha mai approvato i requisiti
- L'ambito del progetto è continuamente aumentato mentre lo sviluppo progrediva
- Richieste di modifiche ai requisiti sono state ignorate
- ...

# Tipici problemi

- La visione e lo scopo del progetto non sono state chiaramente definiti
- I clienti erano troppo impegnati per lavorare con gli analisti e gli sviluppatori
- Il team di sviluppo non ha potuto mai interagire con gli utenti
- I clienti hanno dichiarato tutti i requisiti come critici, nessuna prioritizzazione
- Gli sviluppatori hanno incontrato ambiguità durante lo sviluppo
- La comunicazione si è focalizzata su come le interfacce dovessero essere rappresentate e non sulle funzionalità
- Il cliente non ha mai approvato i requisiti
- L'ambito del progetto è continuamente aumentato mentre lo sviluppo progrediva
- Richieste di modifiche ai requisiti sono state ignorate
- ...

# Tipici problemi

- La visione e lo scopo del progetto non sono state chiaramente definiti
- I clienti erano troppo impegnati per lavorare con gli analisti e gli sviluppatori
- Il team di sviluppo non ha potuto mai interagire con gli utenti
- I clienti hanno dichiarato tutti i requisiti come critici, nessuna prioritizzazione
- Gli sviluppatori hanno incontrato ambiguità durante lo sviluppo
- La comunicazione si è focalizzata su come le interfacce dovessero essere rappresentate e non sulle funzionalità
- Il cliente non ha mai approvato i requisiti
- L'ambito del progetto è continuamente aumentato mentre lo sviluppo progrediva
- Richieste di modifiche ai requisiti sono state ignorate
- ...

# Tipici problemi

- La visione e lo scopo del progetto non sono state chiaramente definiti
- I clienti erano troppo impegnati per lavorare con gli analisti e gli sviluppatori
- Il team di sviluppo non ha potuto mai interagire con gli utenti
- I clienti hanno dichiarato tutti i requisiti come critici, nessuna prioritizzazione
- Gli sviluppatori hanno incontrato ambiguità durante lo sviluppo
- La comunicazione si è focalizzata su come le interfacce dovessero essere rappresentate e non sulle funzionalità
- Il cliente non ha mai approvato i requisiti
- L'ambito del progetto è continuamente aumentato mentre lo sviluppo progrediva
- Richieste di modifiche ai requisiti sono state ignorate
- ...

# Tipici problemi

- La visione e lo scopo del progetto non sono state chiaramente definiti
- I clienti erano troppo impegnati per lavorare con gli analisti e gli sviluppatori
- Il team di sviluppo non ha potuto mai interagire con gli utenti
- I clienti hanno dichiarato tutti i requisiti come critici, nessuna prioritizzazione
- Gli sviluppatori hanno incontrato ambiguità durante lo sviluppo
- La comunicazione si è focalizzata su come le interfacce dovessero essere rappresentate e non sulle funzionalità
- Il cliente non ha mai approvato i requisiti
- L'ambito del progetto è continuamente aumentato mentre lo sviluppo progrediva
- Richieste di modifiche ai requisiti sono state ignorate
- ...

# Tipici problemi

- La visione e lo scopo del progetto non sono state chiaramente definiti
- I clienti erano troppo impegnati per lavorare con gli analisti e gli sviluppatori
- Il team di sviluppo non ha potuto mai interagire con gli utenti
- I clienti hanno dichiarato tutti i requisiti come critici, nessuna prioritizzazione
- Gli sviluppatori hanno incontrato ambiguità durante lo sviluppo
- La comunicazione si è focalizzata su come le interfacce dovessero essere rappresentate e non sulle funzionalità
- Il cliente non ha mai approvato i requisiti
- L'ambito del progetto è continuamente aumentato mentre lo sviluppo progrediva
- Richieste di modifiche ai requisiti sono state ignorate
- ...

# Processo di Ingegneria dei requisiti

Non esiste processo definitivo, attività tipicamente parte di un processo di ingegneria dei requisiti:

- Elicitazione ed analisi dei requisiti
  - Scoperta dei requisiti
  - Classificazione ed organizzazione dei requisiti
  - Prioritizzazione dei requisiti e negoziazione
  - Documentazione dei Requisiti
- Validazione
- Gestione

Anche in questo caso le varie attività possono essere organizzate in diverse maniere. E.g. Iterativo - il peso delle varie attività dunque varierà nelle varie fasi.

# Come specificare i requisiti

Differenti tecniche possibili ovviamente non esclusive (requisiti possono essere definiti e poi raffinati con diversi stili):

- **Informali**: usano tipicamente linguaggi naturali
- **Semi formali**: usano notazioni grafiche per cui la semantica non è sempre precisamente definita
- **Formali**: attraverso modelli matematici

**Esempio**: si consideri la specifica di un sistema di controllo apertura sbarra passaggio a livello



# Requisiti Utente vs. Requisiti di Sistema

## Requisiti Utente

- si rivolgono principalmente all'utente
- alto livello di astrazione
- usano linguaggio naturale e diagrammi

## Requisiti di Sistema

- si rivolgono principalmente ai progettisti/sviluppatori
- alto livello di dettaglio e precisione

# Esempio: il sistema di gestione della biblioteca

## Definizione di requisito utente:

- *Il sistema deve tener traccia di tutti i dati richiesti dalla normativa sul copyright*

## Requisiti di Sistema

- *Il sistema deve prevedere un meccanismo di autenticazione per gli utenti*
- *il sistema deve memorizzare ogni richiesta per almeno 5 anni*
- *Il sistema deve mantenere traccia di tutte le richieste fatte*
- *Il sistema deve memorizzare informazioni sugli utenti sul materiale oggetto della richiesta, e sul personale che ha gestito la richiesta (??)*
- *Il sistema deve produrre resoconti mensili da inviare alle agenzie di controllo.*

# Scelte tecnologiche

Si considerino i seguenti requisiti:

- Il sistema deve permettere autenticazione degli utenti che accedono al prestito
- Il sistema deve permettere autenticazione degli utenti tramite login/password per accesso al prestito
- il sistema deve autenticare l'accesso dell'utente al prestito tramite un meccanismo di login e password

Quali sono le conseguenze di ognuno di questi requisiti? Come possono essere interpretati? Livello di astrazione e scelte tecnologiche?

# Ambiguità



“Figure” ambigue sono molto facili da creare con il linguaggio naturale

# Ambiguità nel linguaggio naturale

Esistono diverse possibili sorgenti di ambiguità nell'uso del linguaggio naturale che possono impattare la specifica di requisiti:

- **Lessicale:** I termini usati sono polisemici, ovvero possono avere più significati.
  - Al di fuori di un cane, un libro è il miglior amico dell'uomo; da dentro è troppo difficile leggere (Groucho Marx)
- **Sintattica:** la frase ha più di un albero sintattico
  - Chiara ha visto Luca in giardino con il canocchiale
  - Una vecchia legge la regola
- **Semantica:** l'equivalente della frase nella logica dei predicati ha più di un'interpretazione
  - Tutti i linguisti preferiscono una teoria
- **Pragmatica:** l'interpretazione dipende dal contesto - "l'entrata è sulla destra"

# Requisiti utente

Specificano il comportamento del sistema in modo comprensibile al cliente. Si occupano del **comportamento osservabile** del sistema (input, output, eccezioni) per l'utente e non dovrebbero contenere specifiche di design.

Tipici problemi:

- Mancanza di chiarezza - **verbosità vs. precisione**
- Confusione - Le diverse tipologie di **requisiti sono mischiati** tra loro
- Accorpamento - Molti requisiti vengono specificati come un singolo requisito

# Requisiti utente

esempio

**Griglia di supporto:** nell'assistere l'utente nel posizionamento delle entità in un diagramma, l'utente può attivare una griglia, che fornisca sia i centimetri che i pollici, attraverso un'opzione nel pannello di controllo. Inizialmente la griglia è disattivata. La griglia può essere attivata/disattivata in qualsiasi momento. Un'opzione griglia verrà fornita nella vista adatta-a-dimensioni ma il numero di linee mostrate sarà ridotto per evitare di riempire diagrammi più piccoli con linee di griglia.

Problemi?

# Requisiti utente

esempio

Molti differenti tipi di requisiti sono mischiati nella prima frase:

- funzionali: la griglia
- non funzionali: centimetri / pollici
- non funzionale: dove il meccanismo si trova

Nota: Troppi dettagli tecnici limitano il raggio di azione degli sviluppatori che invece potrebbero fornire soluzioni innovative.



# Requisiti utente

esempio

L'editor deve fornire una funzionalità griglia dove una **matrice di linee orizzontali e verticali** vengano visualizzate come **background** della vista nella finestra dell'editor. La griglia dovrebbe essere **passiva** e l'allineamento alla griglia deve essere fatto su iniziativa dell'utente.

*Motivazione: una griglia aiuta l'utente nella creazione di un diagramma più pulito con entità ben spaziate. Una griglia attiva può essere utile ma può creare effetti indesiderati come posizionamenti imprecisi. L'utente è la persona più appropriata a decidere il posizionamento*

Sorgente: Micky Mouse

# Requisiti utente

## esercizio

Il sistema di gestione della biblioteca intende fornire un supporto alla gestione dei “conti” che in particolare riporti tutti i pagamenti fatti dagli utenti del sistema. I gestori del sistema devono poter configurare il sistema in modo da poter accordare sconti ad utenti regolari.

Problemi?

# Requisiti utente

## reccomandazioni

- Definite un **formato standard** per la definizione dei requisiti
- Utilizzate linguaggio consistentemente - **attenzione alle parole** “deve”, “dovrebbe”
- **The MoSCoW principle**
- Utilizzate meccanismi di **evidenziazione del testo**
- Non usare, per quanto possibile, **gergo tecnico del dominio informatico**.

# Requisiti di sistema

Aggiungono dettagli per capire come gli obiettivi specificati dai requisiti utente possono essere effettivamente raggiunti dal sistema. Anche questi si dovrebbero **limitare al comportamento osservabile e non contenere scelte che dovrebbero competere alle attività di design**. Ma

...

- Potreste aver bisogno di definire un'architettura iniziale per strutturare i requisiti.
- In molti casi il sistema interagirà con sistemi pre-esistenti che dunque in un certo qual modo forzano scelte progettuali  
*e.g. avete bisogno di certificare il sistema rispetto a norme di safety*  
*il sistema interagisce con un sistema che utilizza formati XML*

# Requisiti di sistema

Aggiungono dettagli per capire come gli obiettivi specificati dai requisiti utente possono essere effettivamente raggiunti dal sistema. Anche questi si dovrebbero **limitare al comportamento osservabile e non contenere scelte che dovrebbero competere alle attività di design**. Ma

...

- Potreste aver bisogno di definire un'architettura iniziale per strutturare i requisiti.
- In molti casi il sistema interagirà con sistemi pre-esistenti che dunque in un certo qual modo forzano scelte progettuali  
*e.g. avete bisogno di certificare il sistema rispetto a norme di safety*  
*il sistema interagisce con un sistema che utilizza formati XML*

# Requisiti di sistema

Problemi nell'uso di linguaggio naturale:

- si basa sulla comune comprensione dei concetti nel sistema
- troppo flessibile
- difficile modularizzare requisiti scritti con linguaggio naturale

Uso di notazioni semi-formali o formali

- Linguaggio Naturale Strutturato
- Linguaggi di Descrizione Progettuale
- Notazioni grafiche
- Specifiche Matematiche

# Formato definizione requisiti

Informazioni minime da riportare in un formato strutturato per la specifica di requisiti:

**ID:** “Identificativo unico - scegliete formato utile agli scopi”

**Nome:** “Nome Mnemonico tipicamente azione nome”

Il “Sistema o parte di esso” Deve/Dovrebbe/Può/Potrebbe “descrizione funzionalità”

**Descrizione:** fornisce ulteriori indicazioni che servono a migliorare comprensione del requisito

**Sorgente:** chi o cosa ha originato il requisito?

<b>ID</b>	<i>Identificativo unico</i>
<b>Tipo</b>	<i>Tipologia del requisito</i>
<b>Evento/CU correlato</b>	<i>A quale CU o evento si riferisce</i>
<b>Descrizione</b>	<i>Una frase che caratterizzi il sistema</i>
<b>Motivazione</b>	<i>Breve descrizione a contestualizzare il requisito</i>
<b>Sorgente</b>	<i>Chi ha richiesto inserimento requisito</i>
<b>Criterio di Valutazione</b>	<i>Come valutare il soddisfacimento del requisito</i>
<b>Soddisfazione Cliente</b>	<i>Grado di soddisf. se requisito sarà implementato</i>
<b>Insoddisfazione Cliente</b>	<i>Grado di insoddisf. se requisito non sarà implementato</i>
<b>Conflitti</b>	<i>conflitti con altri requisiti</i>
<b>Priorità</b>	<i>importanza per il cliente</i>
<b>Materiale di supporto</b>	<i>documentazione che può migliorare comprensione</i>
<b>Storia</b>	<i>Creazione, modifiche</i>



# Specifica dei punti di interazione

Praticamente tutti i sistemi software si trovano ad interagire con altri sistemi software. Le interfacce di interazione devono essere definite formalmente:

- Application Programming Interface (API)
- Strutture dati
- Rappresentazione dei dati

# Requisiti funzionali, non-funzionali e di dominio

## Requisiti funzionali:

- cosa deve fare il sistema. Come deve reagire agli stimoli esterni. Anche cosa il sistema non deve fare.

## Requisiti non-funzionali:

- proprietà del sistema che devono essere soddisfatte. Influenza il sistema nel suo complesso.

## Requisiti di dominio:

- Categoria trasversale - riguarda quei requisiti che derivano direttamente dallo specifico dominio applicativo.

Attenzione distinzione non sempre netta!!

# Requisiti funzionali

Vengono descritte le funzionalità in dettaglio - input, output, eccezioni  
Completezza e consistenza dei requisiti

# Requisiti non funzionali

## classificazione

Si riferiscono a proprietà del sistema

Sono per certi versi **critici tanto quanto i requisiti funzionali**

Classificati in:

- **Requisiti di Prodotto**
- **Requisiti Organizzativi**
- **Requisiti Esterni** - (interazione con altri sistemi, legislazione, ...)

# Requisiti non-funzionali

## esempi

### Product:

- Il sistema deve essere capace di gestire 3000 richieste al minuto

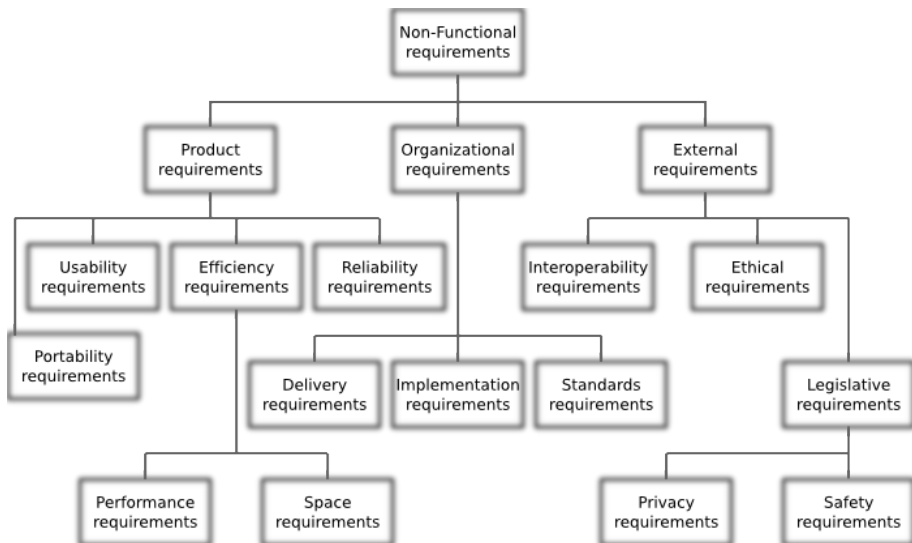
### Organizzativi

- La documentazione ed il processo di sviluppo devono essere conformi a quanto specificato nello standard XYZStand-2007 usato presso la nostra azienda

### Esterno

- Il sistema deve essere conforme agli standard di accessibilità

# Requisiti non-funzionali



# Requisiti non-funzionali

## problemi

I requisiti non-funzionali sono spesso difficili e costosi da verificare!!

Si consideri:

*Il sistema deve risultare usabile ad un utenza esperta*

Come e quando possiamo dire che il requisito è stato soddisfatto?

Spesso non è comunque facile definire **metriche** per le caratteristiche non funzionali

# Requisiti non funzionali

esempi di metriche

Velocità/Performance

Dimensioni

Facilità d'uso

Affidabilità

Portabilità

Tempo per transazione, tempi di risposta

K bytes, Numero di chip nella RAM

Tempi di training, numero di frame di aiuto

MTBF, probabilità di indisponibilità ...

Percentuale degli statement dipendenti dalla  
piattaforma, numero di sistemi target

...

...



# Sommario

- 1 Generalità
- 2 Attività dell'ingegneria dei requisiti**
  - Studio di fattibilità
  - Elicitazione
  - Validazione
  - Gestione
- 3 Documenti dei Requisiti

# Sommario

- 1 Generalità
- 2 Attività dell'ingegneria dei requisiti**
  - **Studio di fattibilità**
  - Elicitazione
  - Validazione
  - Gestione
- 3 Documenti dei Requisiti

# Studio di fattibilità

**Studio preliminare** sulle implicazioni che il sistema avrà una volta costruito e sulla sua convenienza. Risultato di questa fase sarà una **raccomandazione sul continuare o meno lo sviluppo**.

Le domande a cui tipicamente uno studio di fattibilità dovrà rispondere sono:

- Il sistema contribuisce al **raggiungimento degli obiettivi** dell'organizzazione a cui è rivolto? Qual'è il suo impatto? (**Chiaramente questione di business**)
- Può il sistema essere implementato con le **tecnologie correnti** e con costi e tempi "prevedibili"?
- Può il sistema essere integrato con sistemi pre-esistenti?

# Studio di fattibilità

Nella raccolta delle informazioni sarà necessario interagire con il “cliente”. Alcune domande a cui dovrete cercare risposta sono:

- Come l'organizzazione risolverebbe il problema se non fosse possibile implementare il sistema?
- Quali sono i problemi con i processi attuali e come il sistema potrà risolverli?
- Quale contributo il sistema apporterà al raggiungimento degli obiettivi?
- Le informazioni possono essere **trasferite verso o da altre organizzazioni?**
- Il sistema richiederà l'introduzione di **nuove tecnologie?**
- Quali attività il sistema dovrà supportare e **cosa potrà essere lasciato fuori?**

# Sommario

## 1 Generalità

## 2 Attività dell'ingegneria dei requisiti

- Studio di fattibilità
- **Elicitazione**
- Validazione
- Gestione

## 3 Documenti dei Requisiti

# Elicitazione ed analisi dei requisiti

## Difficoltà?

Primo passo è l'individuazione degli “*stakeholders*” (attori).

L'elicitazione dei requisiti è resa difficile da alcuni problemi “inevitabili”:

- attori non hanno piena **coscienza di ciò di cui hanno bisogno**. Hanno difficoltà ad identificare i limiti del sistema di cui hanno bisogno. Possono fornire dettagli che confondono e rendono difficile la focalizzazione sull'obiettivo principale. Possono trovare difficile esprimersi o possono richiedere sistemi inattuabili (dati anche corrispondenti costi)
- Uso di **linguaggio tecnico** del dominio applicativo
- **Stesso requisito** può essere espresso diversamente da differenti persone
- Requisiti aggiunti al fine di poter raggiungere obiettivi personali
- L'ambiente è tipicamente dinamico e le **condizioni possono mutare** anche repentinamente

# Elicitazione ed analisi dei requisiti

## sub-attività

- Scoperta dei requisiti
- Classificazione ed organizzazione dei requisiti
- Prioritizzazione e negoziazione
- Documentazione

# Elicitazione ed analisi dei requisiti

Difficoltà?

Elicitazione influenzata dalle caratteristiche dei processi cognitivi umani. In particolare attività di elicitazione devono considerare i processi mentali di:

- rimozione
- distorsione
- generalizzazione

Particolare attenzione ai termini: tutto, ogni, sempre, mai, nessuno, niente



# Scoperta dei requisiti

## punti di vista

Punti di vista permettono di **classificare gli attori**. Questo permette di avere un'idea della copertura ottenuta sui possibili requisiti. *Meglio intervistare 3 attori da gruppi differenti piuttosto che 10 da uno stesso gruppo.*

Tipicamente si distingue tra:

- **Punto di vista diretto**: chi interagisce direttamente con il sistema
- **Punto di vista indiretto**: chi non interagisce con il sistema ma è interessato al suo comportamento
- **Punto di vista di dominio**

# Scoperta dei requisiti: tecniche

- Interviste
- Workshops
  - Scelta dei partecipanti ed invito
  - Scelta delle location
  - Definizione degli obiettivi
  - identificazione di un moderatore e di un redattore delle minute
- Focus groups
- Observation
- ...

# Tecniche accessorie

- Brainstorming
- KJ Method
- Prototyping
- Mind mapping

# Scoperta dei requisiti

## l'intervista

Meeting nel quale si ha interazione con i vari attori. Obiettivo è mettere l'attore in una condizione di **massimo agio** in modo che possa esprimersi nel modo che più sente naturale rispetto ai requisiti del sistema, senza remore.

- a domande chiuse
- interviste aperte

Interviste sono un buon strumento per raggiungere una **comprensione generale** su cosa il sistema debba fare, ma forniscono scarsa comprensione del dominio applicativo e dettagli specifici.

# Scoperta dei requisiti

## l'intervista

Il risultato dell'intervista chiaramente dipende dall'intervistatore:

- ottime capacità di relazione
- ascoltare
- no preconcetti

# Scoperta dei requisiti

## Descrizione di scenari

Attori trovano più semplice dire **come intendono utilizzare il sistema o come credono debba essere utilizzato**. È più semplice criticare l'uso del sistema che un singolo requisito.

Elicitazione di requisiti tramite **descrizione di scenari d'uso**

Nella forma più generale uno scenario comprende:

- Cosa ci si aspetta quando lo scenario parte
- La descrizione del flusso normale dello scenario
- Descrizione di cosa può andar male nell'esecuzione del flusso normale
- Informazione su attività che potrebbero svolgersi in parallelo
- Una descrizione dello stato del sistema alla fine

# Esempio di scenario

il sistema elettronico di biblioteca

**Assunzioni iniziali:** L'utente si è autenticato ed ha localizzato il link al documento che vuole scaricare

**Flusso Normale:** l'utente seleziona il documento. Il sistema richiede di fornire dettagli di pagamento. Pagamento può essere fatto con CC o con numero di conto da addebitare.

Viene richiesto all'utente di riempire un form di copyright che viene sottoposto al sistema.

Se transazione approvata il PDF del documento viene reso disponibile e l'utente viene informato. Nel caso di documento "print-only" si chiede di scegliere una stampante.

# Esempio di scenario

il sistema elettronico di biblioteca

**Cosa può andar storto:** Copyright form riempito scorrettamente. Si informa il cliente e si chiede di riempire nuovamente il form. Nel caso di errore la transazione viene rifiutata.

Il pagamento non va a buon fine. La transazione viene rifiutata. La stampa può fallire. Nel caso di documento “print-only” la transazione viene abortita ed il cliente viene riaccreditato dell'ammontare corrispondente.

**Attività in parallelo:** molti utenti possono essere connessi al sistema e potrebbero richiedere il download. Lo stesso utente potrebbe tener aperte più sessioni.

**Stato del sistema alla fine:** L'utente viene riportato ad una pagina di benvenuto, l'articolo è stato stampato e nel caso di “print-only” è stato eliminato da eventuali aree disco temporanee.



# Limitazioni dello strumento degli Scenari

Molto efficaci per raccogliere requisiti da punti di vista diretti

Non adatto a rappresentare requisiti derivanti da punti di vista **indiretti** o di **dominio** e a definire requisiti **extra-funzionali** (caratteristiche globali)

# Sommario

## 1 Generalità

## 2 Attività dell'ingegneria dei requisiti

- Studio di fattibilità
- Elicitazione
- **Validazione**
- Gestione

## 3 Documenti dei Requisiti

# Validazione dei requisiti

La fase di validazione dei requisiti cerca di rimuovere possibili problemi nella specifica dei requisiti. Possibili verifiche sono:

- **Controllo di validità:** verificare che ciò che è stato specificato coincide effettivamente con quanto necessario all'utente
- **Controllo di consistenza:** i requisiti non devono essere contraddittori
- **Controllo di completezza:** i requisiti dovrebbero specificare tutte le possibili funzionalità
- **Controllo di concretezza:** verificare che il requisito richieda qualcosa che effettivamente possa essere implementato date anche le tecnologie adottate, i costi e le scadenze imposte
- **Verificabilità:** requisiti devono essere scritti in modo da poter verificare la loro soddisfazione.

# Validazione dei requisiti

## tecniche

Tecniche che si sono rivelate utili nella validazione dei requisiti sono:

- **Revisione dei requisiti:** processo **manuale** che coinvolge team misti cliente/contractor. Può essere **formale** o **informale**. Verifiche che potrebbero essere fatte includono:
  - Verificabilità
  - Comprensibilità
  - Tracciabilità
  - Adattabilità
- **Prototipizzazione**
- **Generazione di casi di test**

# Sommario

## 1 Generalità

## 2 Attività dell'ingegneria dei requisiti

- Studio di fattibilità
- Elicitazione
- Validazione
- **Gestione**

## 3 Documenti dei Requisiti

# Gestione dei requisiti

Requisiti sono costantemente sottoposti a **spinte di cambiamento**. Il requisito una volta definito non è fissato per sempre, anche considerando non solo le fasi di post-rilascio. Molte motivazioni per questo:

- Comunità estesa di utenti con richieste differenti ed anche conflittuali
- Acquirenti ed utenti diretti spesso non sono la stessa entità.
- L'ambiente di esecuzione cambia velocemente.

L'attività di gestione dei requisiti si occupa di far **emergere, permettere e gestire** modifiche ai requisiti

# Gestione dei requisiti

...di cosa c'è bisogno?

- Identificazione dei requisiti tramite ad esempio definizione di ID
- Definire un processo di modifica dei requisiti: tutte le modifiche sono trattate egualmente e consistentemente
- Definire meccanismi di tracciabilità
- uso e supporto da parte di CASE tool/environment (database, fogli elettronici etc. possono essere sufficienti)

Requisiti stabili vs. requisiti volatili

# Tracciabilità dei requisiti

Differenti tipi di tracciabilità tipicamente si immagazzinano le informazione in una matrice:

- **Sorgente**: Attore x Requisito
- **Relazioni con altri requisiti**: Requisito x Requisito
- **Design**: Sottosistema x Requisito

Matrici possono diventare particolarmente estese e poco gestibili. Uso di CASE (database) di supporto alle varie fasi:

- Immagazzinamento
- Gestione delle modifiche
- Gestione della tracciabilità



# Attività di gestione delle modifiche

L'attività di gestione della modifica dei requisiti sarà strutturata su sotto-attività quali:

- Analisi del problema e specifica della modifica
- analisi del cambiamento e valutazione del costo
- implementazione della modifica

# Sommario

- 1 Generalità
- 2 Attività dell'ingegneria dei requisiti
  - Studio di fattibilità
  - Elicitazione
  - Validazione
  - Gestione
- 3 Documenti dei Requisiti

# Documento dei requisiti software

Il documento dei requisiti software è **ciò che deve essere implementato dagli sviluppatori**. Contiene generalmente **sia requisiti utente che di sistema**.

Differenti utenti . . . differenti “requisiti” . . . differenti formati

Formato dipendente anche da **processo adottato!**

Metodologie agili in molti casi suggeriscono di non derivare un documento dei requisiti ma di annotarli su apposite cards che verranno poi priorizzate.

# IEEE/ANSI 830-1998

Suggerisce la seguente struttura:

- Introduction
  - 1 Scopo del documento dei requisiti
  - 2 Scopo del prodotto
  - 3 Definizione, acronimi ed abbreviazioni
  - 4 Riferimenti
  - 5 Overview dell'intero documento
- Descrizione generale
  - 1 Prospettive sul prodotto
  - 2 Funzioni del prodotto
  - 3 Caratteristiche degli utenti
  - 4 vincoli generali
  - 5 Assunzioni e dipendenze
- Requisiti specifici
- Appendici
- Indici

# Contenuto generale

- 1 Prefazione
- 2 Introduzione
- 3 Glossary
- 4 Definizione dei requisiti utente
- 5 Architettura del sistema
- 6 Definizione dei requisiti di sistema
- 7 Modelli del sistema
- 8 Evoluzione del sistema
- 9 Appendici
- 10 Indici