Ingegneria dei Requisiti

A. Gianoli

Univ. Ferrara - Corso di laurea in Informatica

Dove?

- Pressmann, cap 9 in parte
- Sommerville, cap 6

Cosa significa?

- le attività con cui si stabilisce ciò che il committente chiede al sistema software e i vincoli che il sistema deve soddisfare in fase di sviluppo e in fase operativa
-

Analisi dei requisiti

- Studio della realtà applicativa
- Identificazione dei confini tra l'applicazione da sviluppare e "resto del mondo"
- Comprensione degli attributi di qualità richiesti

- attività esplorativa e incrementale
- In questa fase non ci chiediamo come realizzeremo l'applicativo, ma ci concentriamo nell'elencare cosa deve fare

Requisiti?

"Requirements": ci servono per specificare cosa costruire

- ci dicono "cosa", non "come"
- ci dicono il problema, non la soluzione
- riflettono (sono pertinenti) al system design, non al software design

“Cosa vs. Come”: è relativo...

- One person’s *what* is another person’s *how*...
  - “One person’s constant is another person’s variable” [Mythical Man Month]

- Input file processing è il cosa, parsing è il come
- Parsing è il cosa, uno stack è il come
- Uno stack è il cosa, un array o una linked list è il come
- Una linked list è il cosa, una doubly linked list è il come

Perché l'analisi dei requisiti?

- Alcuni delle ragioni per cui l'analisi dei requisiti è necessaria
  - comprendere precisamente che cosa si richiede dal sistema
  - comunicare questa comprensione in modo preciso a tutti gli sviluppatori coinvolti
  - controllare la produzione per assicurarsi che il sistema soddisfi le specifiche (comprese le eventuali variazioni in corso d'opera)

Ruoli nell'analisi dei requisiti

- customers: spiegano cosa deve essere consegnato (base contrattuale)
- managers: controllano scheduling e avanzamento lavori
- designers: producono le specifiche del design
- programmatori: preparano lista di implementazioni accettabili/output
- QA/testers: pensare a un insieme di test di base, per la validazione, verifica, ...

Definizione vs Specifica dei requisiti

- **Definizione dei requisiti**
  - delle frasi in linguaggio naturale, con eventuale aggiunta di 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; di solito viene scritta come contratto tra il cliente e il 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

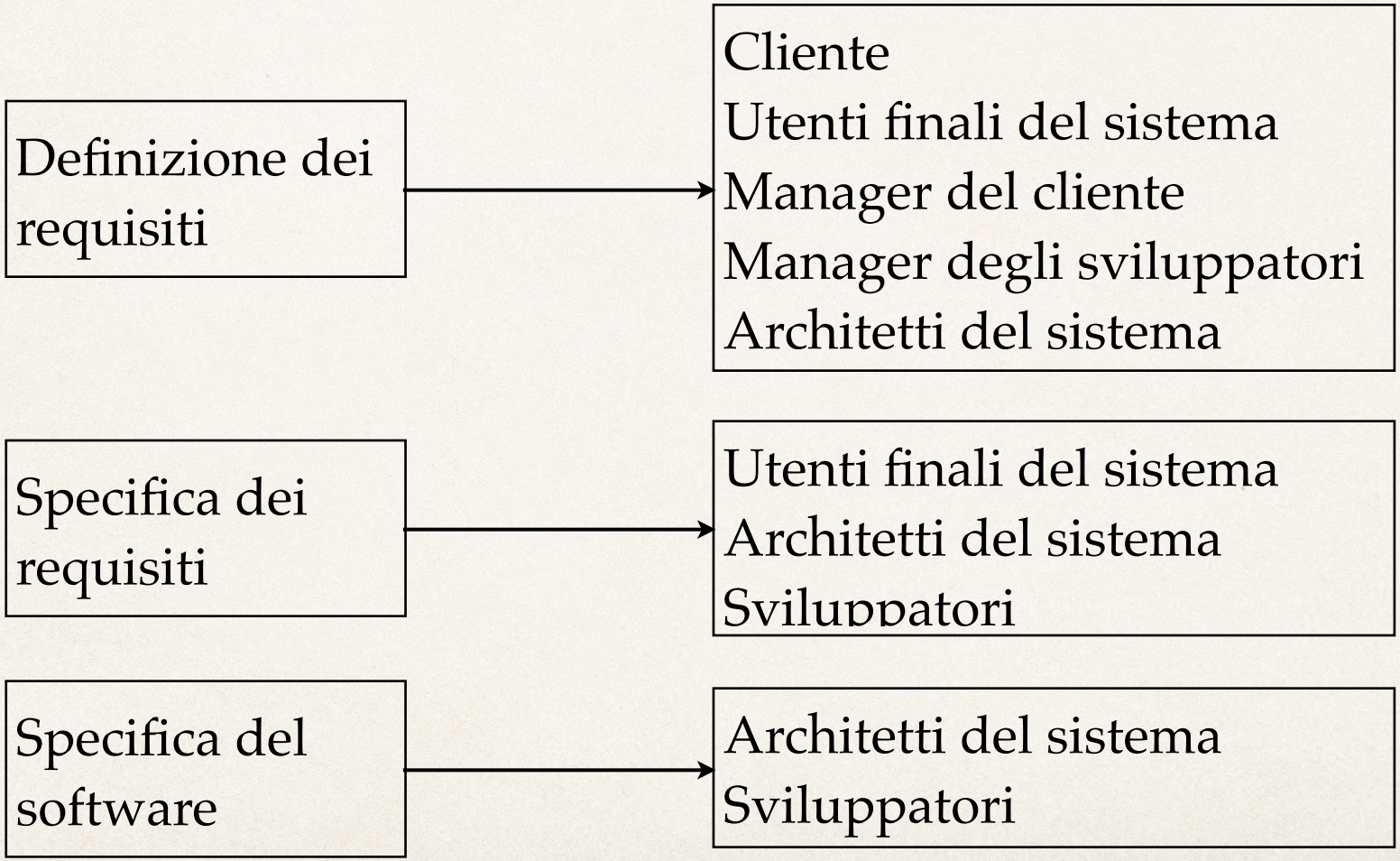
User requirement definition

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?



Descrizione dei requisiti

- Metodo classico per classificare i requisiti: due tipi di requisiti
  - *funzionali*: descrivono i servizi offerti dal sistema o le sue funzionalità
  - *non funzionali*: vincoli sul processo di sviluppo o sul sistema in generale

Requisiti di dominio?

- derivano dal dominio di applicazione del sistema, non dagli utenti
- terminologia propria del dominio di applicazione o si riferiscono a suoi concetti
- possono essere nuovi requisiti funz. o porre vincoli a altri req.

decel $D_{treno} = D_{controllo} + D_{gradiente}$ con $D_{gradiente}$=9.81 ms^2 * grad compensazione/alpha terreno

Cosa intendiamo con “requisito”?

- Grande variabilità, 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
- Il motivo di ciò è che i requisiti servono a molteplici scopi
  - può essere la base di un’offerta per un lavoro: aperta a interpretazioni e modifiche
  - può essere parte fondamentale del contratto stesso: massimo livello di dettaglio, niente interpretazioni

Requisiti: come ottenerli?

- Cosa fare:
  - parlare con gli utenti, o lavorare per un po' con loro, per capire in che modo lavorano
  - fate domande nelle varie fasi per "scavare a fondo", non date le cose per scontate
  - pensate al perché un utente fa qualcosa, non soltanto a cosa fa
  - aspettatevi fin dall'inizio che i requisiti possano evolvere e cambiare
- Cosa non fare:
  - non siate troppo specifici o dettagliati (non a questo livello)
  - non provate a pensare a tutto in anticipo (fallireste)
  - non aggiungete features non richieste: non sono necessarie

Esempio: LIBSYS

- Prendiamo come esempio un ipotetico sistema che fornisce una interfaccia unica ad un insieme di database di 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 fare una ricerca su tutti i database dispnibili 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 nei requisiti

- Cosa intendiamo con “viewers appropriati”?
- nelle intenzioni del committente, un programma specifico per ogni diverso tipo di documento
- nell’interpretazione dello sviluppatore, un unico visualizzatore di testo che mostri il contenuto di qualsiasi tipo di documento

Imprecisioni: quali le ragioni?

- I problemi nascono quando i requisiti non sono enunciati con precisione
- Requisiti ambigui possono essere interpretati in modo diverso da sviluppatori e utenti
- In principio i requisiti dovrebbero essere sia completi che consistenti
  - completi: dovrebbero includere la descrizione di tutto ciò che è richiesto
  - consistenti: non ci dovrebbero essere conflitti o contraddizioni nelle descrizioni delle funzionalità richieste
- In pratica è impossibile produrre un documento dei requisiti completo e consistente: sotto questo aspetto la prototipazione può essere molto utile per chiarire le cose

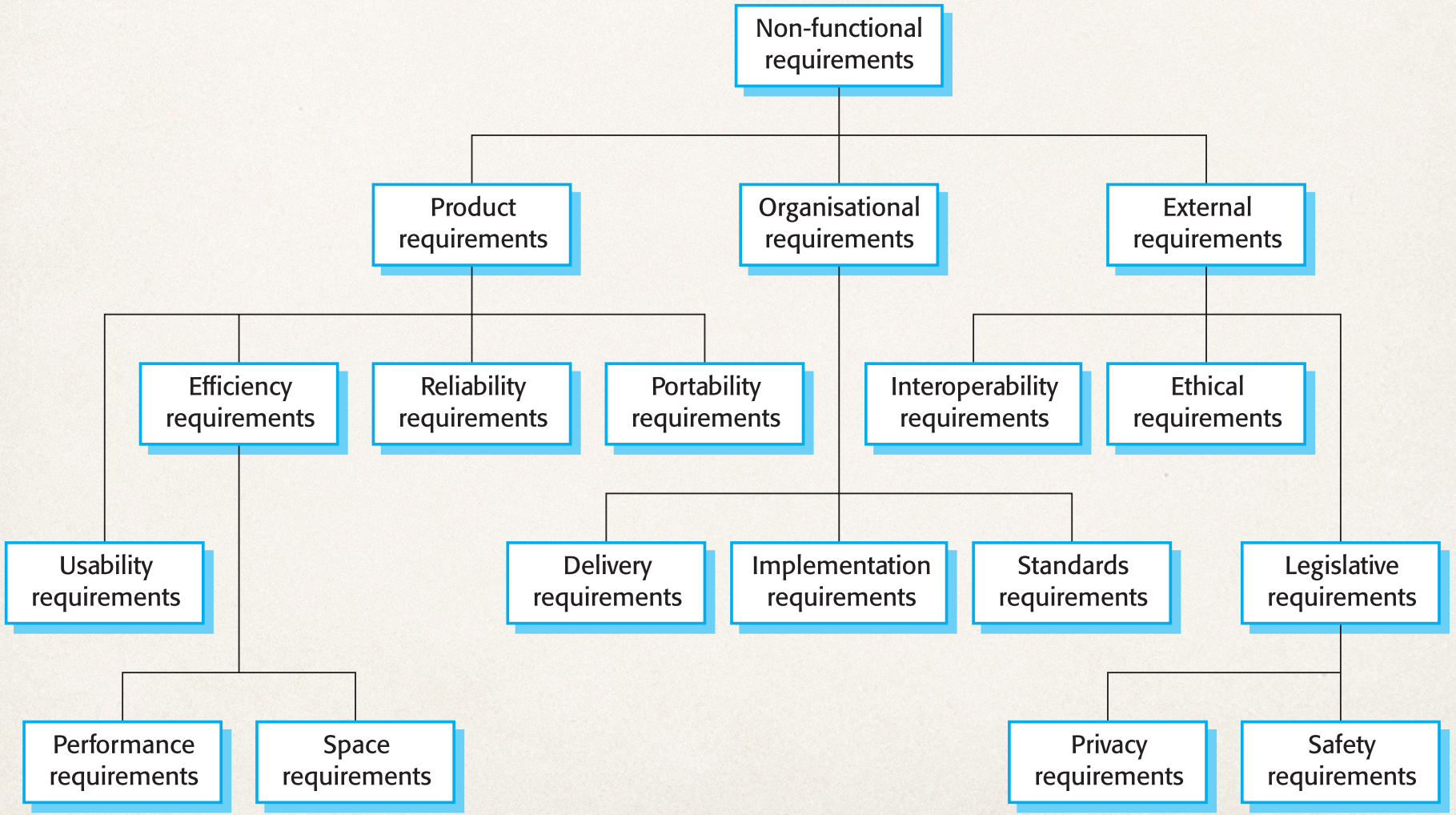
Requisiti non funzionali

- Definiscono proprietà e vincoli sul sistema (p.e. affidabilità, tempo di risposta, occupazione, ...)
- Possono anche essere vincoli di processo che impongono l'uso di un particolare sistema di sviluppo, linguaggio di programmazione o libreria software
- I requisiti non funzionali possono anche essere più critici di quelli funzionali: se non soddisfatti il sistema può essere inutilizzabile

Requisiti non funzionali

- Product requirements
  - vincoli che specificano il comportamento del prodotto finale, velocità, reliability, ...
- Requisiti organizzativi
  - sono conseguenza di policy organizzative del cliente, standard di processo da usare, requisiti implementativi, ...
- Requisiti esterni
  - sono esterni al sistema e al suo processo di sviluppo, p.e. requisiti di interoperabilità, requisiti legislativi, ...

Tipi di requisiti non funzionali



Problema dei Requisiti di dominio

- Come abbiamo detto in precedenza, derivano dal dominio dell'applicazione
- Comprensibilità
  - questi requisiti sono formulati nel linguaggio del dominio applicativo
  - non è affatto detto che gli sviluppatori "parlino" questo linguaggio
- Implicità
  - lo specialista di dominio può conoscere la sua area di lavoro così bene da non pensare di dover esplicitamente richiedere alcune cose: per lui sono implicite

Ingegneria dei requisiti

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

Uso dei requisiti

1. definizione dei bisogni del committente
   - spesso non chiari neppure al committente
   - committente e realizzatore non si capiscono

   esigenza di *comprensibilità*

   necessità *di convalida*

Uso dei requisiti

2. Definizione dei requisiti dell'implementazione
   - documento di riferimento per i progettisti
   - esigenza di *chiarezza* e *precisione*

Uso dei requisiti

3. Documento di riferimento per la manutenzione
   - distinguere chiaramente tra manutenzione correttiva e adattiva / perfettiva
   - tutte le precedenti qualità

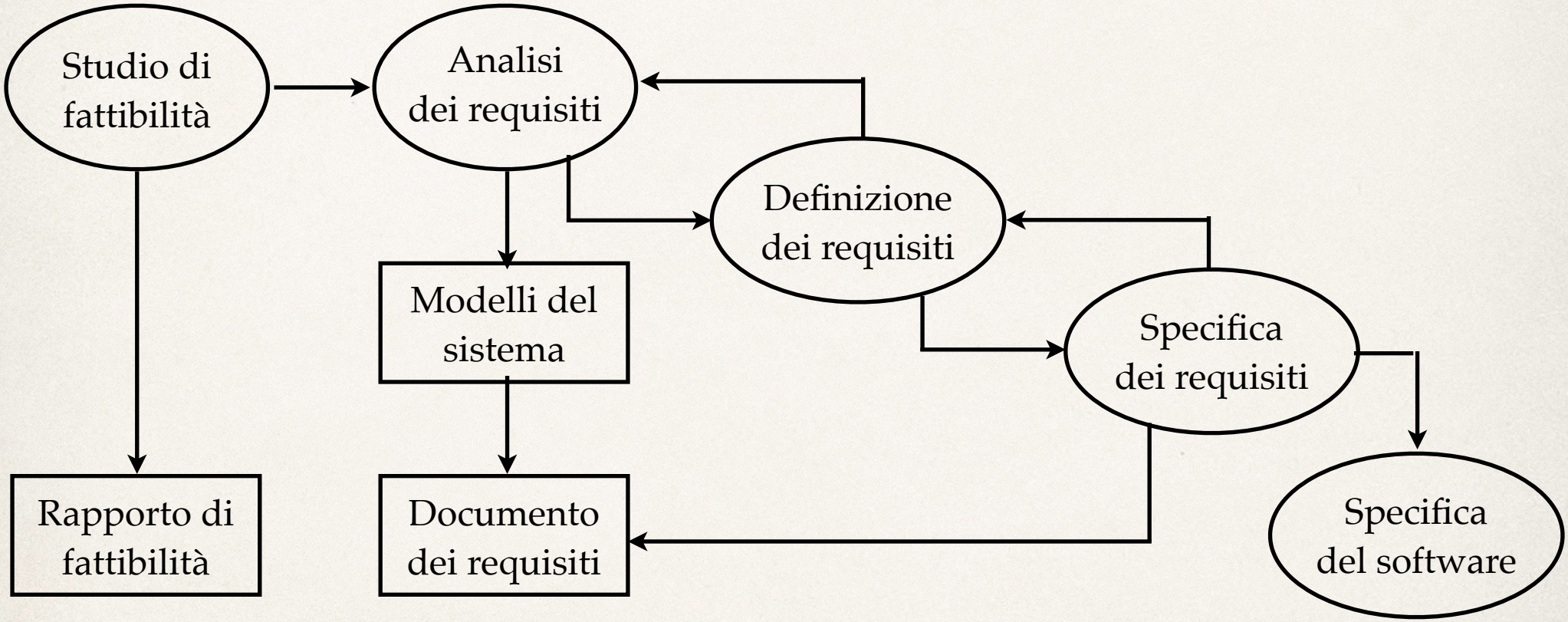
Le fasi del processo di ingegneria dei requisiti

1. Studio fattibilità
   - quali sono le necessità che possono essere soddisfatte con la tecnologia e le risorse disponibili?
2. Analisi dei requisiti
   - quali servizi il sistema deve offrire, qual'è il dominio di applicazione, performances richieste, vincoli hw, ...
3. Definizione dei requisiti
   - frasi in linguaggio naturale più eventuali diagrammi dei servizi che il sistema deve offrire e dei suoi vincoli operativi. E' rivolta al cliente

Le fasi del processo di ingegneria dei requisiti

4. Specifica dei requisiti
   - documento strutturato che fissa una descrizione dettagliata dei servizi che il sistema deve offrire. E' alla base del contratto
5. Specifica del software
   - descrizione dettagliata del software che serva come base per il disegno dell'architettura o per l'implementazione. E' rivolta agli sviluppatori

Le fasi del processo di ingegneria dei requisiti



Il documento dei requisiti

- Il documento dei requisiti definisce ufficialmente ciò che viene richiesto al sistema
  - cioè cosa è richiesto che gli sviluppatori producano
- Include sia la specifica che la definizione dei requisiti
- Non è un documento di design
  - dovrebbe limitarsi a dire "cosa" il sistema deve fare, ma non il "come" lo deve fare

Avvio del processo

- La tecnica più usata per individuare i requisiti consiste nel condurre riunioni o interviste col committente e gli utilizzatori finali
- Solitamente si parte da domande che conducano a fissare in termini generali il problema
  - chi c'è dietro la richiesta di questo lavoro?
  - chi utilizzerà la soluzione?
  - quali saranno i vantaggi economici di una soluzione di successo?
  - esiste un'altra fonte per la soluzione di cui si ha bisogno?
- Serve per identificare gli stakeholder, i benefici misurabili e le alternative

Avvio del processo

- secondo insieme di domande: consente all'analista di acquisire maggiore comprensione del problema, e al committente di comunicare le proprie percezioni riguardo una soluzione
  - come caratterizzare un "buon" output generato da una soluzione di successo?
  - quali problemi deve risolvere questa soluzione?
  - potete mostrare/descrivere l'ambiente in cui la soluzione dovrà operare?
  - esistono particolari richieste di prestazioni o vincoli generali che incidono sulla soluzione?

Avvio del processo

- l'ultimo gruppo di domande riguarda l'attività di comunicazione (meta-domande)
  - siete voi la persona corretta per rispondere a queste domande? le vostre risposte sono "ufficiali"?
  - le mie domande sono rilevanti rispetto al problema?
  - sto facendo troppe domande?
  - ci sono altre persone in grado di fornirmi informazioni supplementari?
  - c'è qualcos'altro che dovrei chiedere?

Raccolta

L'avvio può sembrare ingannevolmente semplice. Ci sono una serie di problemi che aiutano a comprendere perché la raccolta dei requisiti è difficile

- **problemi di scope**: limiti del sistema mal definiti, oppure committente specifica dettagli tecnici non necessari che possono confondere invece di chiarire gli obiettivi generali
- **problemi di comprensione:** committente non è completamente sicuro di ciò che vuole, ha un'idea vaga delle limitazioni e delle possibilità, non conoscono bene il dominio del problema, omettono informazioni che ritengono "ovvie", hanno problemi a comunicare i bisogni all'analista, ...
- **problemi di volatilità:** cambiano i requisiti nel corso del tempo

Validazione dei requisiti

- Dimostrare che i requisiti definiti riguardo il sistema coincidono esattamente con i desideri del committente
- errori nei requisiti sono in genere molto costosi, quindi la valutazione è importante
- la prototipazione spesso è un modo efficace per validare i requisiti

Controllo dei requisiti

- Validità
  - il sistema fornisce le funzionalità che meglio soddisfano le richieste del committente?
- Consistenza
  - ci sono requirements in conflitto tra loro?
- Completezza
  - tutti i requisiti del cliente sono stati individuati?
- Realismo
  - è fattibile soddisfare tutti i requisiti con la tecnologia e i budget a disposizione?

Controllo dei requisiti

- Verificabilità
  - il requisito è realisticamente testabile?
- Comprensibilità
  - il requisito è stato adeguatamente compreso?
- Tracciabilità
  - l'origine del requisito è chiaramente indicata?
- Adattabilità
  - è possibile cambiare il requisito senza grossi impatti su altri requisiti?

Revisione dei requisiti

- E' necessario svolgere delle revisioni periodiche durante la formulazione dei requisiti
  - possono essere sia formali che informali
- Le revisioni devono coinvolgere sia il cliente che gli sviluppatori
- Lo scopo è sempre avere una adeguata comunicazione tra committente, sviluppatori e utenti finali per individuare e risolvere i problemi appena si manifestano

FAST
Facilitated Application Specification Technique

- Tecnica per la specifica delle applicazioni
- Consiste nella formazione di un team misto di sviluppatori e clienti che collabori a
  - individuare e specificare il problema
  - proporre elementi della soluzione
  - negoziare strategie diverse per valutarle
  - specificare un insieme preliminare di requisiti

FAST
Facilitated Application Specification Technique

- viene indetta una riunione a cui partecipano sia gli ingegneri del software che i clienti
- si stabiliscono delle regole per la preparazione e la partecipazione
- viene proposta una agenda
  - abbastanza formale da coprire tutti i punti importante
  - abbastanza informale da incoraggiare l'apporto di nuove idee/ punti di vista

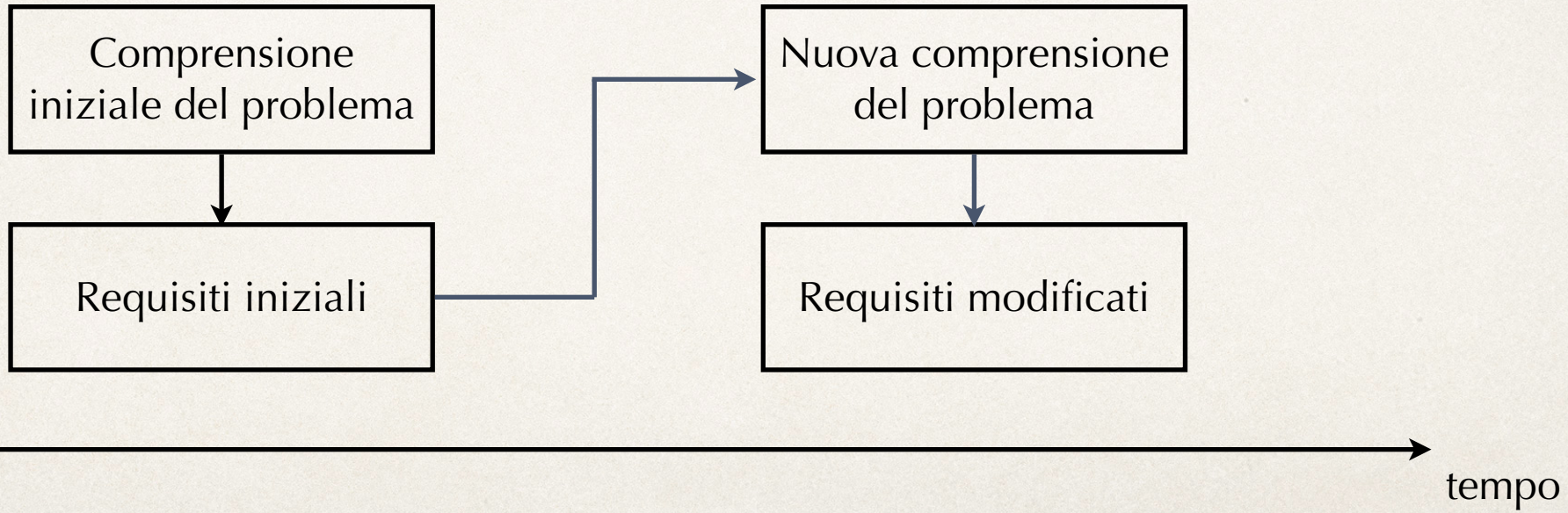
FAST
Facilitated Application Specification Technique

- viene stabilito un “moderatore” per gestire la riunione
- può essere chiunque: cliente, sviluppatore, persona esterna
- si applica un “meccanismo di definizione”
- lavagna, fogli appesi ad una parete, altro...
- quello che si vuole ottenere è:
  - specificare il problema
  - proporre eventuali soluzioni
  - confrontare varie strategie
  - arrivare ad un insieme preliminare di requisiti

Collaborative planning

Evoluzione dei requisiti

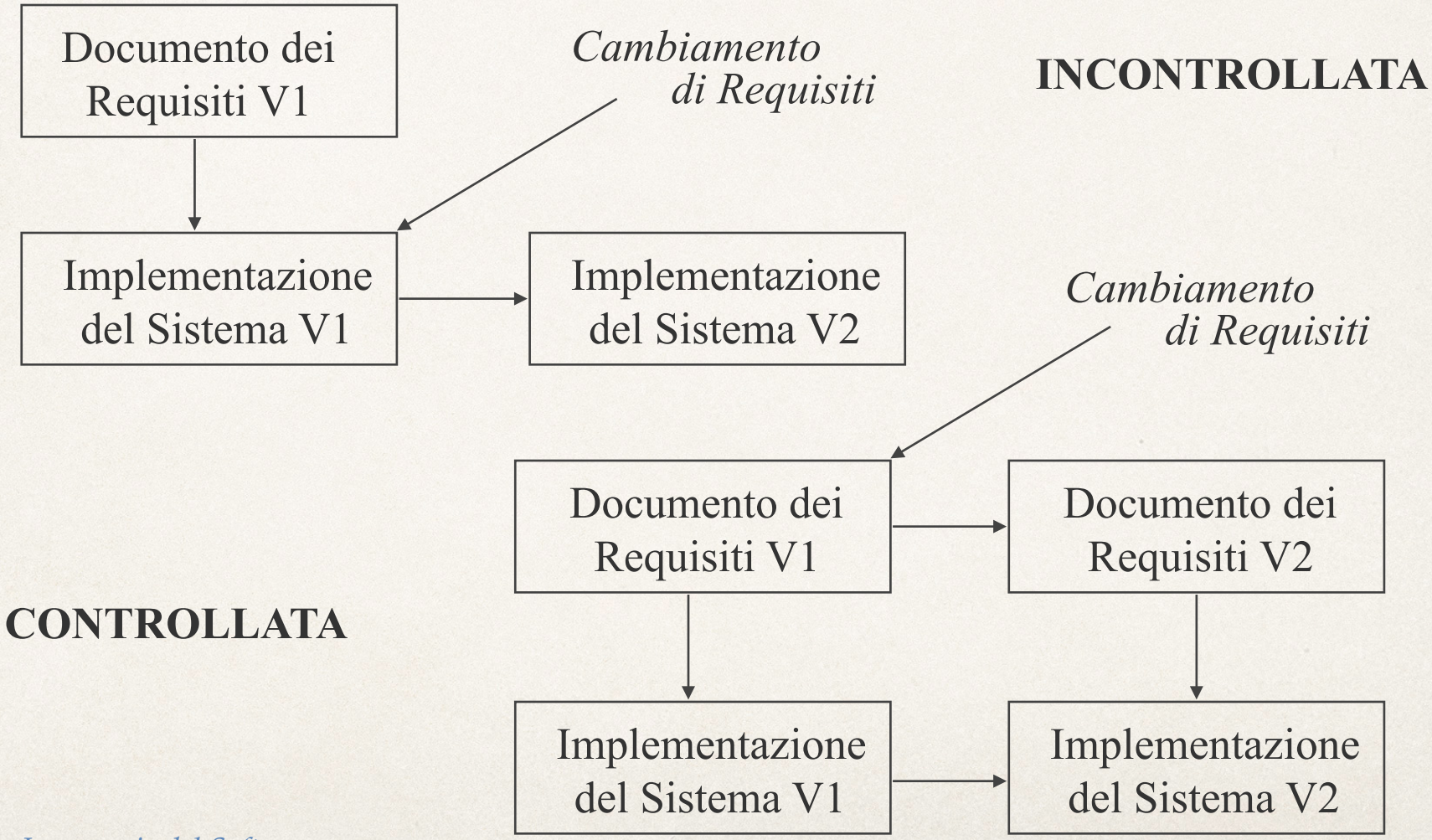
- Non importa quanto stiate attenti: i requisiti evolvono; man mano che le vere esigenze del cliente sono meglio capite o perché gli obiettivi stessi dell'organizzazione cambiano
- Occorre che il sistema sia aperto a modifiche dei requisiti sia durante lo sviluppo che durante l'utilizzo



Classi di requisiti

- Requisiti durevoli
  - derivano dalla natura stessa dell'attività del committente; possono essere ottenuti dalla modellizzazione del domino (rappresentazione astratta entità e relazioni che caratterizzano un dominio di applicazione)
  - Es: un ospedale avrà sempre a che fare con medici, infermieri...
- Requisiti volatili
  - possono cambiare durante lo sviluppo o durante l'uso del sistema
  - Es: in un ospedale i requisiti di una politica sanitaria possono cambiare

Evoluzione controllata



Casi d'uso

- Man mano che i requisiti vengono raccolti l'analista può creare un insieme di scenari che identificano un possibile "modo" di usare il sistema
- Gli scenari (Use Cases) descrivono una funzionalità dal punto di vista di chi la utilizza
  - Attori
    utenti o altra entità che interagisce col sistema; formalmente è una qualunque cosa che comunica col sistema ma ne è esterno
  - Casi d'uso
    descrivono l'interazione tra attori e sistema, ma non la "logica interna" del sistema

Casi d'uso

- A quali domande dobbiamo rispondere?
  per esempio:
  - quali compiti o funzionalità vengono svolte dagli attori?
  - quali informazioni di sistema vengono acquisite, prodotte o modificate dall'attore?
  - l'attore dovrà informare il sistema sui cambiamenti esterni?
  - l'attore vuole essere informato di ogni cambiamento inatteso?