



Università degli Studi di Napoli
“Federico II”

Ingegneria del Software

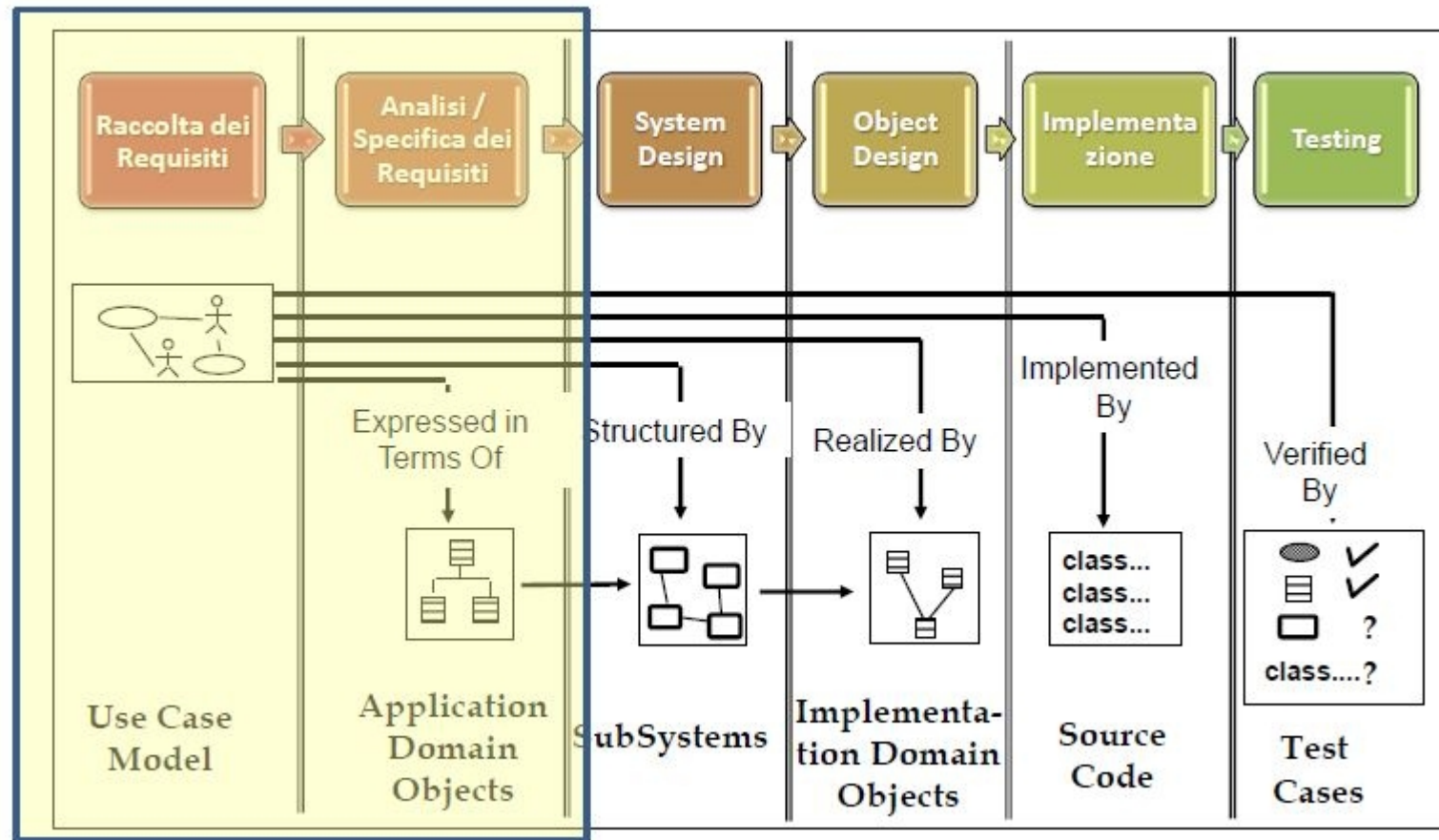
a.a. 2013/14

I Requisiti Software

Contenuti della lezione

- Comprendere il concetto di Ingegneria dei Requisiti
- Comprendere le diverse Tipologie di Requisiti Software
 - Requisiti Utente vs. Requisiti di Sistema
 - Requisiti Funzionali vs. Requisiti Non Funzionali

Ciclo di vita Software: Attività



Requisiti

Def. (IEEE Glossary) requirement.

(1) A condition or capability needed by a user to solve a problem or achieve an objective.

(2) A condition or capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification, or other formally imposed documents.

(3) A documented representation of a condition or capability as in (1)(2).

- Ingegneria dei requisiti: Definire i requisiti del sistema di interesse.
 - **Raccolta dei requisiti**
 - **Analisi dei requisiti**

Esempio

- Cosa deve fare il sistema + vincoli operativi
 - Il sistema dovrà archiviare i dati di una biblioteca, in particolare descrittori bibliografici e relativi a materiale librario e periodici secondo formati compatibili con gli standard accreditati.
 - Il sistema dovrà gestire il patrimonio librario e di periodici posseduto identificando i possedimenti con un codice a barra.
 - Il sistema dovrà gestire sistemi di prestiti.
 - Il sistema dovrà permettere la consultazione via Web del catalogo usando qualsiasi campo del descrittore bibliografico.
 - L'interfaccia al sistema dovrà essere realizzata utilizzando un World-Wide-Web browser.
 - Il sistema dovrà gestire almeno 20 transazioni per secondo
 - Il sistema dovrà riconoscere l'utente attraverso una smart-card

Ingegneria dei requisiti

- **L'ingegneria dei requisiti** comprende tutte le attività connesse alla gestione dei requisiti:
 - Studio di fattibilità
 - Raccolta dei requisiti
 - Analisi dei requisiti
 - Specifica / Documentazione dei requisiti
 - Verifica e Validazione dei requisiti

Tipologie di Requisiti

- I Requisiti SW possono essere classificati secondo due diversi punti di vista:
- **Livello di dettaglio**
 - Requisiti Utente
 - Requisiti di Sistema
- **Tipo di requisito rappresentato**
 - Requisiti Funzionali
 - Requisiti Non Funzionali
 - Requisiti di Dominio

Livello di dettaglio

- Diversi livelli di granularità e diversa astrazione nei formalismi impiegati:
- **Requisiti Utente**
 - Descrivono i servizi richiesti al sistema (comportamento osservabile dall'esterno) e i vincoli operativi del sistema.
 - Scritti in linguaggio naturale o debolmente strutturato (non tecnico).
 - Il punto di vista è quello dell'utente (a volte Committente).
- **Requisiti di Sistema**
 - Formulazione dettagliata, strutturata, di servizi e vincoli.
 - Scritti in linguaggio naturale, notazioni semi-formali, linguaggi formali
 - Il punto di vista è quello del Contraente, che li può usare anche per il contratto con il Committente
- **Specifica Software**
 - Una specifica astratta del sistema software che può essere usata per progettazione o implementazione.

Un requisito utente, molti requisiti di sistema

Requirements definition (Un requisito Utente)

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

Requirements specification (diventa PIU' requisiti di Sistema)

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

Specifica (IEEE Glossary)

- **Specification.** A document that specifies, in a complete, precise, verifiable manner, the requirements, design, behavior, or other characteristics of a system or component, and, often, the procedures for determining whether these provisions have been satisfied.

Una specifica è scritta con un linguaggio di specifica

- **Specification language.** A language, often a machine-processible combination of natural and formal language, used to express the requirements, design, behavior, or other characteristics of a system or component.
- **Formal specification.**
 - (1) A specification written and approved in accordance with established standards.
 - (2) A specification written in a formal notation, often for use in proof of correctness.

Tipi di requisiti

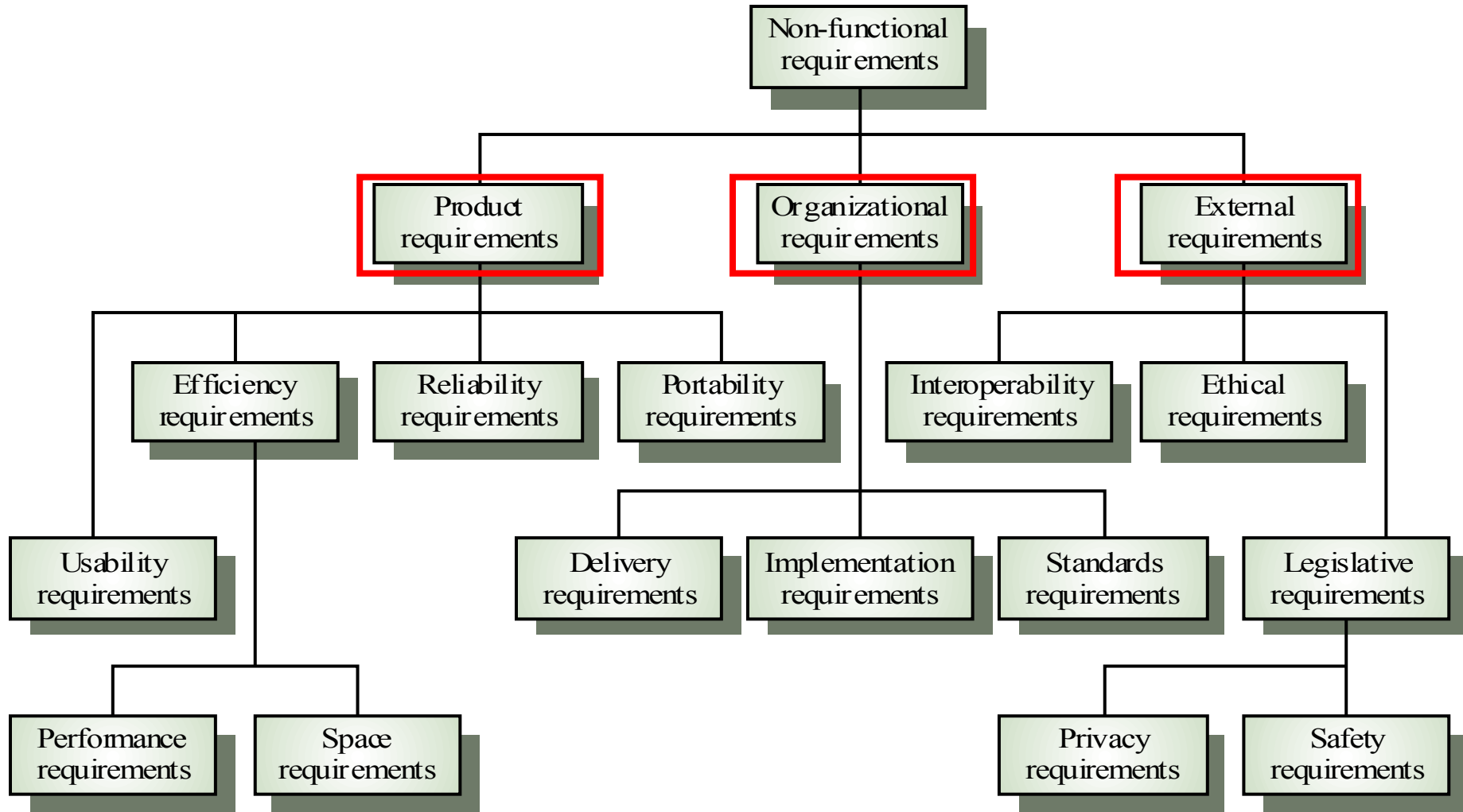
- I **Requisiti Funzionali** descrivono i servizi, o funzioni, offerti dal sistema (normalmente attivati da user-input)
 - “Quando l’utente richiede la visualizzazione dell’estratto conto... allora il sistema deve ...”
- I **Requisiti Non-Funzionali** descrivono vincoli sui servizi offerti dal sistema, e sullo stesso processo di sviluppo
 - “La visualizzazione dell’estratto conto deve avvenire entro 4 secondi dalla sua richiesta”
- I **Requisiti di Dominio** (funzionali e non-funzionali) riflettono caratteristiche generali del dominio applicativo
 - “L’accesso alla cassa continua da parte dell’addetto bancario al rifornimento deve avvenire secondo le consuete procedure di sicurezza a doppia-chiave”

Requisiti Funzionali

functional requirement. (IEEE Glossary) A requirement that specifies a function that a system or system component must be able to perform.

- Esempio
 - Il sistema POS NextGen è un'applicazione informatica che deve permettere di registrare le vendite ed in pagamenti in negozi e supermercati.
 - Il sistema dovrà permettere ad un utente che vuole pagare con PagoBancomat di inserire il PIN su un'apposita tastiera, dopo aver visualizzato su un display l'elenco degli articoli acquistati.
 - Il Display deve mostrare l'elenco degli articolo acquistati, con il relativo prezzo, un eventuale sconto applicato, ed il costo complessivo dell'intera spesa.

Requisiti non-funzionali (Sommerville)



Il modello FURPS

- FURPS è un acronimo per la definizione dei requisiti
- FURPS classifica i Requisiti in:
 - **Functionality**
 - **Usability = Usabilità**
 - **Reliability = Affidabilità**
 - **Performance**
 - **Supportability = Supportabilità**

Requisiti funzionali



Requisiti Non
Funzionali

Requisiti Non-Funzionali

- **Functional requirement. (IEEE Glossary)** A requirement that specifies a function that a system or system component must be able to perform.
- I requisiti **Non Funzionali** descrivono aspetti del sistema non direttamente collegati al comportamento funzionale del sistema.
- Diverse classificazioni di requisiti non funzionali:
 - Modello FURPS+
 - Classificazione Sommerville
 -

Requisiti non funzionali

- Requisiti del sistema che non sono materialmente legati alle funzionalità e ai servizi del sistema:
 - ad es. aspetti di efficienza, affidabilità, uso di risorse, parametri qualitativi etc
 - Varie classificazioni (Sommerville, FURPS+, ...)
- Esempio (POS NextGEN).
 - Risposta all'autorizzazione entro 30 secondi, nel 90% dei casi
 - Il sistema deve essere in grado di gestire diversi tipi di interfacce utente, quali un display multiriga, un web-browser, etc...
 - Qualora ci siano problemi di connettività, il sistema deve comunque registrare in locale il pagamento, in modo da permettere il normale svolgimento degli acquisti nel negozio.

Modello FURPS+

- **FURPS**: Acronimo per **F**unctionality, **U**sability, **R**eliability, **P**erformance, **S**upportability
- + indica una estensione a categorie di requisiti aggiuntivi spesso incluse tra i requisiti non funzionali:

- **USABILITY (IEEE Glossary)**

The ease with which a user can learn to operate, prepare inputs for, and interpret outputs of a system or component.

- Ad esempio
 - Convenzioni usate per le interfacce utenti
 - Copertura degli help online
 - Livello della documentazione utente
 -

Modello FURPS+ (2)

- **RELIABILITY (IEEE Glossary)**

The ability of a system or component to perform its required functions under stated conditions for a specified period of time.

- Ad esempio

- Il tempo medio accettabile per un fallimento

- Capacità di riconoscere certi fault specificati

- Capacità di reggere a specifici attacchi di sicurezza

- A volte inclusa nella categoria **Dependability (da cui si può dipendere)**

- Capacità del sistema di mostrarsi “affidabile” nei confronti degli utilizzatori

-

Modello FURPS+ (2b)

- **(Dependability)**
 - **Robustness** (comportamento corretto in presenza di input non valido o condizioni di stress ambientale)
 - **Robustness. (IEEE Glossary)** The degree to which a system or component can function correctly in the presence of invalid inputs or stressful environmental conditions. ***See also: error tolerance; fault tolerance.***
 - **Safety** Capacità di non arrecare conseguenze catastrofiche per l'ambiente, cose e persone
 - **Security** Capacità di fornire confidenzialità (accessi autorizzati) e integrità dello stato del sistema e dei dati.

Modello FURPS+ (3)

- **PERFORMANCE requirement (IEEE Glossary)**

A requirement that imposes conditions on a functional requirement; for example a requirement that specifies the speed, accuracy, or memory usage with which a given function must be performed.

- Alcune sottocategorie della performance:

- **Response Time:** velocità di reazione agli input

- **Throughput:** quantità di lavoro in uno specifico lasso di tempo

- **Throughput IEEE Glossary** The amount of work that can be performed by a computer system or component in a given period of time; for example, number of jobs per day.

Modello FURPS+ (3b)

- **PERFORMANCE requirement (IEEE Glossary)**
- Alcune sottocategorie della performance:
 - **Availability:** grado di disponibilità e accessibilità del servizio alla richiesta
 - availability. **IEEE Glossary**
 - The degree to which a system or component is operational and accessible when required for use. Often expressed as a probability. See also: error tolerance; fault tolerance; robustness.
 - **Accuracy:** accuratezza nello svolgimento del servizio.
 - **accuracy. IEEE Glossary**
 - (1) A qualitative assessment of correctness, or freedom from error.
(2) A quantitative measure of the magnitude of error. *Contrast with:* precision.

Modello FURPS+ (4)

- **SUPPORT (IEEE Glossary)**

The set of activities necessary to ensure that an operational system or component fulfills its original requirements and any subsequent modifications to those requirements. For example, software or hardware maintenance, user training.

- Facilità di gestione dei cambiamenti dopo la messa in funzione del sistema:

- **Adaptability:** capacità di adattare il sistema a richieste aggiuntive

adaptability. IEEE Glossary The ease with which a system or component can be modified for use in applications or environments other than those for which it was specifically designed.

Modello FURPS+ (4b)

- **Support (continua)**

Maintainability. IEEE Glossary

(1) The ease with which a software system or component can be modified to correct faults, improve performance or other attributes, or adapt to a changed environment. *See also: extendability; flexibility.*

(2) The ease with which a hardware system or component can be retained in, or restored to, a state in which it can perform its required functions.

- **Portability:** la facilità con cui il sistema può essere trasferito da un ambiente HW/SW ad un altro.

Modello FURPS+ (5)

- **Categorie aggiuntive (+ in FURPS+)**
 - **Implementation requirements:** vincolo sull'uso di piattaforme HW, tool specifici, linguaggi di programmazione.
 - **Interface requirements:** vincoli imposti da sistemi esterni (preesistenze o legacy), formati di scambio, interfacce.
 - **Operation requirements:** vincoli sull'amministrazione e gestione del sistema quando operativo.
 - **Packaging requirements:** vincoli sulle modalità di consegna del software (ad es. supporto per l'installazione etc)
 - **Legal requirements:** riguardano aspetti delle licenze, regolamenti, certificazioni rispetto a standard di qualità, etc.

Requisiti di qualità per POS NextGEN

- Il testo deve essere visibile da un metro di distanza (**Usabilità**)
- Qualora ci siano problemi di connettività, il sistema deve comunque registrare in locale il pagamento, in modo da permettere il normale svolgimento degli acquisti nel negozio. (**Affidabilità**)
- Risposta all'autorizzazione entro 30 secondi, nel 90% dei casi (**Performance**)
- Deve essere possibile cambiare la lingua del testo visualizzato (**Supportabilità / Usabilità**)

Requisiti di dominio

- Si riferiscono a caratteristiche del sistema che derivano da caratteristiche generali del dominio applicativo (piuttosto che dalla istanza specifica del problema affrontato dal sistema).
- Problemi:
 - **Spesso sono requisiti impliciti:** vengono dati per scontati dal committente e non espressi esplicitamente.
 - **A volte sono requisiti espliciti ma oscuri:** Il Committente descrive requisiti utilizzando termini e concetti con i quali il Contraente/Sviluppatore non ha dimestichezza (Glossario)

Requisiti di dominio – Esempi

- Gestione biblioteca
 - ‘Ci deve essere una interfaccia utente a tutti i database bibliografici adeguata allo standard Z39.50’
 - Lo Standard è noto al personale della biblioteca, non agli sviluppatori
- POS NextGEN
 - Il codice identificativo dell’articolo, letto mediante codice a barre, può essere basato sui seguenti schemi di codifica:
 - UPC
 - EAN
 - JAN
 - SKU

Caratteristiche dei requisiti Funzionali

- **Completezza:** devono comprendere tutti i servizi di interesse per gli utenti
- **Coerenza:** i requisiti non devono avere definizioni conflittuali e/o contraddittorie
- **Correttezza:** il sistema descritto dai requisiti deve essere quello inteso dal Committente.
- **Non ambiguità:** il sistema risulta chiaramente e univocamente definito.
- Per sistemi consistenti è difficile ottenere requisiti completi e coerenti
 - Gli utenti interessati hanno esigenze diverse, spesso in conflitto.

Caratteristiche dei requisiti funzionali

- **Realismo:** Il sistema può essere implementato nei limiti imposti dai vincoli.
- **Verificabilità:** E' possibile disegnare dei test per il sistema sviluppato per dimostrare che i requisiti sono soddisfatti.
- **Tracciabilità:**
 - Ogni requisito puo' essere propagato nello sviluppo del software alla funzione che lo realizza.
 - Ogni funzione sviluppata nel sistema può essere riferita all'insieme di requisiti che realizza..

Conflitti fra requisiti non-funzionali

- Esempio: Sistema di rilevamento pattume spaziale sullo Shuttle
 - Req.1 - System should fit into 4Mbytes of memory
 - Req.2 - System should be written in ADA
- Può risultare impossibile compilare un programma ADA con le funzionalità richieste, e che occupi solo 4Mbytes: uno dei requisiti va escluso

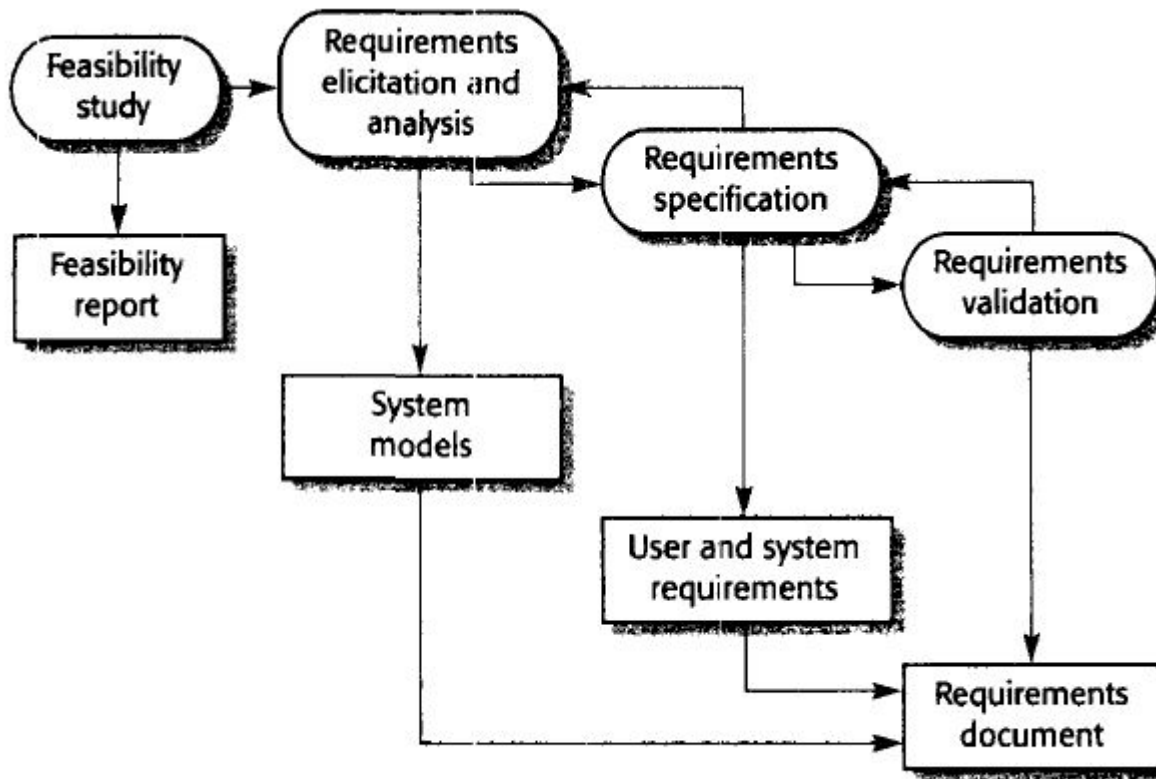
Verificabilità di requisiti non funzionali

- I Requisiti non funzionali devono sempre essere **Verificabili**
 - Dovrebbero sempre esprimere vincoli quantitativi e non puramente qualitativi
- Es. Requisito funzionale **non verificabile**
 - ‘Il sistema deve essere facile da usare per controllori esperti, e deve essere tale da minimizzare gli errori degli utenti’
- Es. Requisito non-funzionale **verificabile**
 - ‘controllori esperti devono poter imparare a usare tutte le funzioni del sistema in max. 2 ore di apprendimento.
 - Dopo l’apprendimento, il controllore deve essere in grado di operare senza commettere piu’ di 2 errori al giorno.’

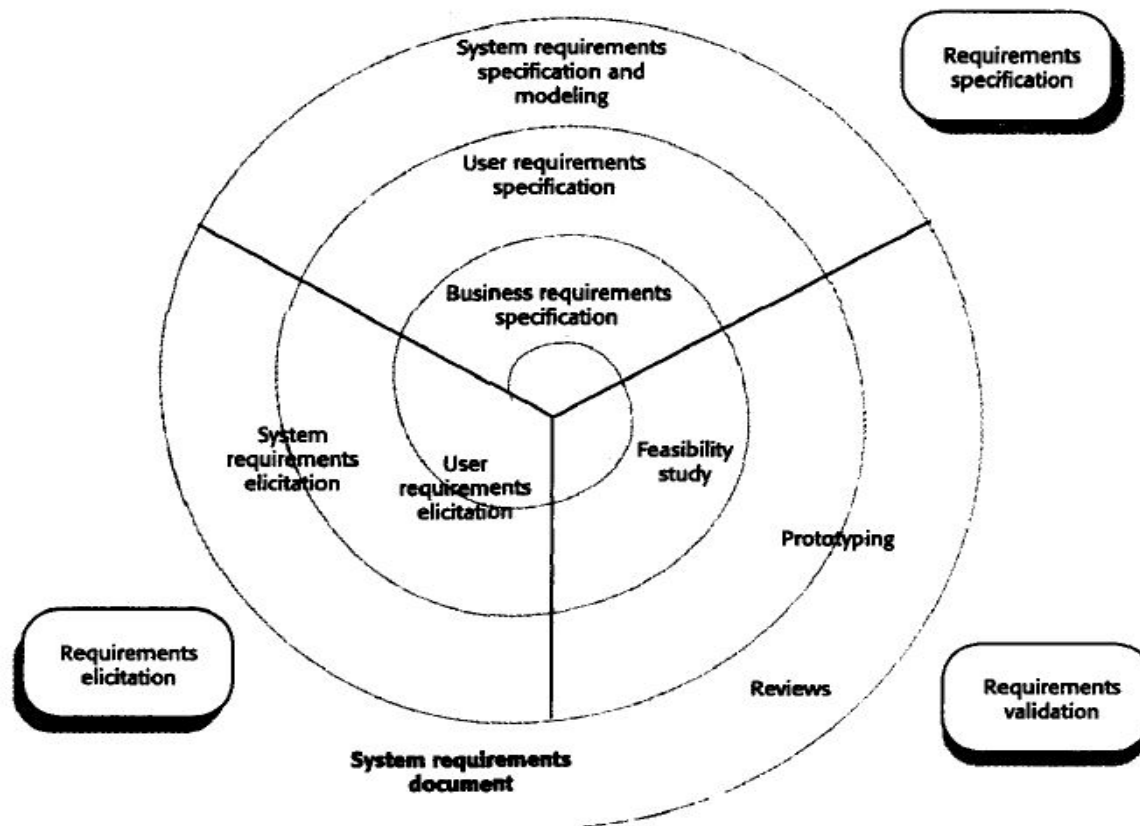
Quantificazione di requisiti non-funzionali

Property	Measure
Speed	Processed transactions/second User/Event response time Screen refresh time
Size	K Bytes Number of RAM chips
Ease of use	Training time Number of help frames
Reliability	Mean time to failure Probability of unavailability Rate of failure occurrence Availability
Robustness	Time to restart after failure Percentage of events causing failure Probability of data corruption on failure
Portability	Percentage of target dependent statements Number of target systems

Ingegnerizzazione dei requisiti



Ingegnerizzazione dei requisiti: Processo



Tipi di raccolta dei requisiti

- Classificati sulla base della sorgente dei requisiti:
 - **Greenfield Engineering**
 - Lo sviluppo parte da zero, nessun sistema esiste in precedenza, così i requisiti sono “estratti” dagli utenti e dal cliente
 - È guidata dalla necessità dell’utente o dalle esigenze del mercato
 - **Re-engineering**
 - Un sistema esistente viene riprogettato a causa della disponibilità di nuove tecnologie o per estendere funzionalità del sistema
 - **Interface Engineering**
 - Per fornire i servizi di un sistema esistente in un nuovo ambiente
 - Vengono riprogettate le interfacce
 - Sistemi legacy lasciati inalterati, eccetto che per le interfacce
 - È guidata dall’introduzione di nuove tecnologie, e da nuove necessità del mercato

Raccolta dei requisiti

- E' spesso considerata l'attività più difficile, perché richiede la collaborazione tra più gruppi di partecipanti con differenti background.
 - **Stakeholders:** “insieme dei soggetti che hanno un interesse nei confronti di un'organizzazione e che con il loro comportamento possono influenzarne l'attività”.

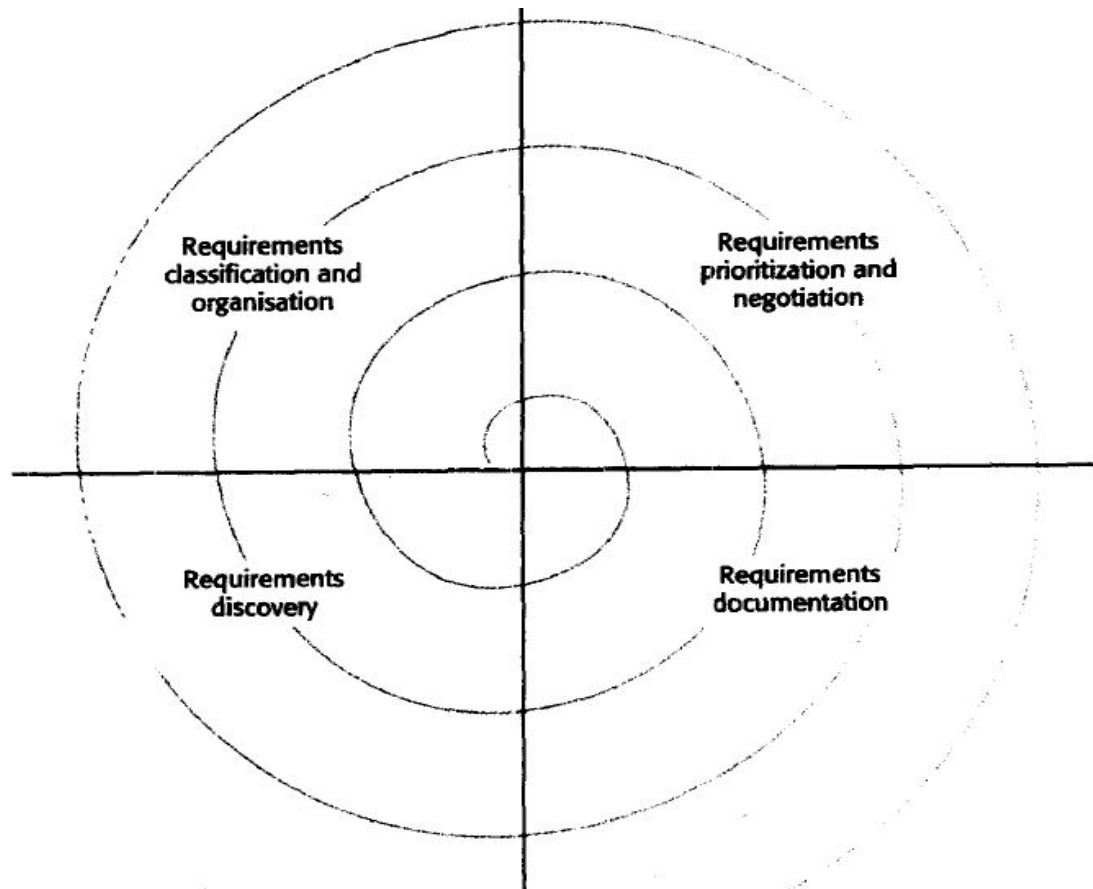
Tutte le persone in qualche modo interessate alla messa in opera del sistema

- **Il cliente e gli utenti finali** sono esperti nel loro dominio e hanno una idea generale di cosa il sistema debba fare, e poca (o nulla) esperienza nello sviluppo del software
- **Gli sviluppatori** hanno esperienza nel produrre sistemi software, ma hanno una conoscenza limitata del dominio di applicazione (ambiente degli utenti finali)

Raccolta dei requisiti

- Tutti gli stakeholders comunicano tra di loro per definire il sistema da realizzare.
 - Fallimenti nella comunicazione (tra i diversi domini) portano a un sistema difficile da usare o che non supporta le funzionalità richieste
- Gli errori introdotti in questa fase sono difficili (e costosi) da risolvere perché sono scoperti nelle ultime fasi del processo di sviluppo del software
- Rischi possibili:
 - una funzionalità che il sistema dovrebbe supportare non è specificata;
 - funzionalità incorrette o obsolete;
 - interfacce utenti poco intuitive e difficile da usare.

Raccolta e analisi dei requisiti: Processo



Validazione dei requisiti (2)

Realismo:

- La specifica dei requisiti è realistica se può essere implementata tenendo conto dei vincoli

• Verificabilità:

- La specifica dei requisiti è verificabile se, una volta che il sistema è stato costruito, test ripetuti possono essere delineati per dimostrare che il sistema soddisfa i requisiti

• Tracciabilità:

- Ogni funzione del sistema può essere individuata e ricondotta al corrispondente requisito funzionale.
- Include anche l'abilità di tracciare le dipendenze tra i requisiti, le funzioni del sistema, gli artefatti, incluso componenti, classi, metodi e attributi di oggetti
- **È cruciale per lo sviluppo di test e per valutare i cambiamenti**

Validazione dei requisiti

- E' un passo critico nel processo di sviluppo
- I requisiti sono continuamente validati da cliente e utenti
- La validazione dei requisiti richiede di controllare:
 - **Correttezza:** una specifica è corretta se rappresenta accuratamente il sistema che il cliente richiede e che gli sviluppatori intendono sviluppare;
 - **Completezza:** una specifica è completa se tutti i possibili scenari per il sistema sono descritti, incluso i comportamenti eccezionali
 - **Coerenza:** se i requisiti non si contraddicono tra di loro
 - **Chiarezza:** una specifica è chiara se non è possibile interpretare la specifica in due modi diversi

Tecniche di validazione dei requisiti

- **Reviews dei requisiti:** revisioni sistematiche di team di revisori.
- **Prototipazione:** viene prodotto un modello eseguibile del sistema e sottoposto all'approvazione degli utenti
- **Generazione dei casi di test:** la generazione dei casi di test come parte del processo di validazione permette di evidenziare problemi
- **Analisi di consistenza automatica:** se i requisiti sono espressi in modello che utilizza un linguaggio formale possono essere sottoposti a un controllo di consistenza automatico.

Tracciabilità dei requisiti

- **Identificabilità:**

- I requisiti devono avere un **identificativo** per poter essere referenziati.
- Occorre scegliere il giusto livello di **granularità** dei requisiti per permettere adeguati riferimenti.

- **CASE Tool:**

- La creazione e gestione delle tracce deve essere supportata da CASE tool specializzati e opportuni sistemi di memorizzazione (database)

Tracciabilità dei requisiti

- **Tipologie di informazione di tracciabilità**
- **Tracciabilità della sorgente:**
 - Legame tra il requisito e lo stackholder proponente insieme al rationale per il requisito
 - Lo stackholder può essere consultato in caso di cambiamento.
- **Tracciabilità tra requisiti:**
 - Legame che esplicita la dipendenza dei requisiti
 - Stabilisce l'impatto che un cambiamento ha sul complesso
- **Tracciabilità di progettazione:**
 - Legame tra requisiti e moduli di progettazione
 - Stabilisce l'impatto che un cambiamento di requisiti ha sulla progettazione del sistema.