

**Analisi, modellazione e  
specifica dei requisiti**

# Indice

8. Analisi, modellazione e specifica dei requisiti .....	3
8.1 Determinazione dei requisiti .....	4
8.1.1 Raccolta dei requisiti .....	5
8.1.2 Identificazione dei requisiti .....	5
8.1.3 Classificazione dei requisiti .....	6
8.1.4 Requisiti non funzionali .....	6
8.1.5 Requisiti funzionali .....	6
8.2 Analisi e specifica di requisiti funzionali .....	7
8.2.1 Modelli di casi d'uso .....	7
8.2.2 Diagrammi di casi d'uso .....	8
8.2.3 Diagrammi di package di casi d'uso .....	8
8.2.4 Descrizioni di casi d'uso .....	9
8.2.5 Scenario principale .....	10
8.2.6 Estensioni .....	10
8.3 Negoziazione e validazione di requisiti .....	10
8.3.1 Analisi di consistenza di requisiti .....	11
8.4 Esercizi: specifica di requisiti .....	11
8.4.1 Premessa .....	11
8.4.2 Esercizio 1: requisiti non funzionali .....	11
8.4.3 Esercizio 2: requisiti di accessibilità .....	11
8.4.4 Esercizio 3: requisiti funzionali .....	11
8.5 Note .....	12
8.6 Bibliografia .....	12

## 8. Analisi, modellazione e specifica dei requisiti

*Everything should be made as simple as possible, but not one bit simpler.*  
Albert Einstein (attribuita) <sup>1</sup>

### Scopo della lezione



- **traduzione di obiettivi di qualità in requisiti di sistema, verificabili dall'utente, rivolti al loro conseguimento**
- **struttura del Documento di Specifica dei Requisiti (DSR)**
- **introduzione a tecniche standard di specifica dei requisiti funzionali (modelli di casi d'uso UML)**

Una delle principali difficoltà che frequentemente si incontra nella stesura di un documento dei requisiti di sistema è causata dalla inerente duplicità del contesto a cui esso è rilevante.

Il contesto d'uso, da un canto, esige la massima semplicità e chiarezza della descrizione dei requisiti, con astrazione da dettagli tecnici e aspetti propri della realizzazione che non siano pertinenti all'uso del sistema.

Il contesto produttivo, d'altro canto, prescrive la verificabilità di ogni requisito attraverso procedure di collaudo, il che spesso richiede precisione formale e dettagli tecnici non appropriati al contesto d'uso.

Per superare questo *gap* fra livelli di astrazione nella descrizione, sono state proposte varie tecniche di specifica di requisiti, il cui grado di formalità varia notevolmente, dalla informalità di di descrizioni narrative in linguaggio naturale, alla semi-formalità di descrizioni che usano notazioni diagrammatiche convenzionali e standardizzate, fino alla rigorosa formalità di specifiche matematicamente ben fondate e verificabili.

Una panoramica delle tecniche di specifica di requisiti, anche solo di quelle più rilevanti ai vari gradi di formalità, esula dai confini della presente trattazione. In questa lezione ci si limita ad introdurre le principali problematiche connesse alla specifica di requisiti, e ad illustrare alcune semplici tecniche di soluzione attraverso degli esempi. Per ulteriori approfondimenti si rinvia ai riferimenti citati nelle sezioni che seguono.

Nella progettazione di sistemi complessi, la determinazione di nuovi requisiti relativi ad una parte del sistema piuttosto che al sistema nel suo complesso può avvenire a livelli di astrazione più bassi di quello iniziale, e non necessariamente coinvolge gli utenti del sistema, in quanto coinvolge gli utenti della parte in questione, che può ben essere del tutto interna al sistema ed inaccessibile agli utenti finali.

Il *raffinemento* di requisiti mira ad assicurare la consistenza della specifica di requisiti attraverso vari livelli di astrazione, e può dunque essere iterativamente adoperato come tecnica di sviluppo dalla (macro)progettazione del sistema alla sua *microprogettazione*, cioè alla progettazione delle sue parti, fino al livello di dettaglio sufficiente per la loro implementazione. Questa tecnica verrà ulteriormente illustrata nella prossima lezione.

In questa lezione:

- **Determinazione dei requisiti (Sez. 8.1)**

è il processo in cui si realizza la traduzione di nostro interesse. In esso si presta attenzione ai seguenti aspetti particolarmente rilevanti.

- **Raccolta dei requisiti**

è il primo passo del processo di traduzione di nostro interesse. La sua realizzazione può richiedere consultazione con utenti e/o committenti, analisi dei rischi, indagini di mercato, analisi dei vincoli derivanti dal contesto produttivo, etc.

- **Identificazione univoca dei requisiti**

per garantirne la *tracciabilità* nelle successive attività di analisi, sviluppo e collaudo, nonché attraverso cambiamenti che i requisiti stessi possano subire nel corso di dette attività o in iterazioni successive del processo produttivo.

- **Classificazione dei requisiti**

in base a caratteristiche generali di qualità a cui possono essere riferiti.

- **Distinzione fra requisiti funzionali e non funzionali**

per queste due categorie principali di requisiti sono state elaborate tecniche descrittive e di specifica notevolmente diverse.

- **Tecniche di specifica dei requisiti funzionali (Sez. 8.2)**

La determinazione dei requisiti, di solito svolta in collaborazione con gli utenti, mira a descrizioni caratterizzate da semplicità, oggettività ed immediata intellegibilità. La descrizione dei requisiti funzionali, tuttavia, richiede di entrare nel merito delle funzionalità previste per il sistema, sia pur astraendo dal *come* esse siano da realizzare. In Sez. 8.2 si introduce una semplice tecnica di *analisi di requisiti funzionali*, rivolta alla loro formulazione per l'inclusione in un *documento di specifica dei requisiti* (DSR) adeguato a detti scopi.

- **Negoziazione e validazione dei requisiti (Sez. 8.3)**

Una specifica di requisiti costituisce un *contratto* fra produttori e utenti del sistema, o di una sua parte, e come tale è naturalmente risultato di una *negoziazione*, riferimento (anche legale) per il riconoscimento di *validità* delle sue clausole nel prodotto a cui si applica, e oggetto di possibile *cambiamento*, previa rinegoziazione ed accordo fra i contraenti.

- **Proposte di lavoro (Sez. 8.4)**

La serie di proposte di lavoro avanzate nelle tre lezioni precedenti si conclude con ulteriori proposte, che completano le precedenti rispetto agli argomenti di questa lezione.

## 8.1 Determinazione dei requisiti

### Documento di Specifica dei Requisiti (DSR)



**parti interessate (stakeholders):**

- utenti, committenti
- progettisti, sviluppatori
- collaudatori
- manager, addetti alle vendite

## 8.1.1 Raccolta dei requisiti

- **Modalità di raccolta**

Dati degli obiettivi di qualità, le modalità attraverso cui si giunge alla determinazione dei requisiti dipendono molto dalla natura del prodotto e dalle caratteristiche degli utenti. Per prodotti destinati al mercato aperto, spesso non si ha un committente e gli utenti sono caratterizzati in modo astratto; in tal caso la raccolta dei requisiti è effettuata da un gruppo interamente interno alla struttura produttiva, e può avvalersi di indagini di mercato, sondaggi, esperienza passata, etc.. Altrimenti, il gruppo preposto alla definizione dei requisiti consisterà di almeno un rappresentante degli utenti o del committente e di almeno un esponente del gruppo di produzione. La determinazione dei requisiti porta ad un contratto fra questi due gruppi, ed è importante che la comprensione che ne hanno entrambi sia chiara e scevra da equivoci. A questo scopo, è rilevante quanto segue.

- **Tecniche di descrizione**

La specifica dei requisiti in un linguaggio formale è altamente raccomandabile, quando è fattibile. Come notato in Sez. 1.2.1, l'adozione di metodi formali di specifica richiede però un certo grado di familiarità con i fondamenti matematici degli stessi, condizione che non è sempre agevole soddisfare. Inoltre, in sede di prima definizione dei requisiti, può sussistere un elevato grado di incertezza rispetto ai dettagli che la formalizzazione solitamente richiede. Queste ragioni giustificano l'uso, almeno in prima approssimazione, di tecniche informali o semi-formali di descrizione.

- **Consistenza e completezza dei requisiti**

Va inoltre notato che, anche quando i requisiti siano definiti con rigore e precisione matematica, risulta problematico *dimostrare* formalmente la *consistenza* e *completezza* dei requisiti rispetto agli obiettivi di qualità, per la evidente difficoltà di formalizzare questi ultimi, almeno quelli di natura generale. Questi due principi agiscono tuttavia come guida di orientamento in sede di raccolta, definizione ed organizzazione testuale dei requisiti.

- **Organizzazione del DSR**

Anche se in una stesura informale, un documento di definizione di requisiti sarà meglio adeguato alla sua funzione contrattuale, alla possibilità di mutamenti, ed alla sua stretta relazione con gli obiettivi di qualità, se rispetterà alcuni canoni elementari di organizzazione, di cui due sono particolarmente importanti: li esaminiamo qui appresso.

## 8.1.2 Identificazione dei requisiti

- **Univocità dell'identificazione**

Il termine "identificazione" va qui inteso nel senso, strettamente sintattico, di "etichettatura", cioè assegnazione di un unico *identificatore* a ciascun requisito nel documento, al fine di potervi fare riferimento.

Tale etichetta può semplicemente essere un numero di capitolo, sezione, sottosezione, paragrafo, etc., o possono essere escogitati altri meccanismi di identificazione, ma l'esistenza ed unicità dell'identificatore per ogni requisito sono essenziali. Esso infatti consente riferimenti semplici e univoci ai requisiti anche dall'esterno del documento (si pensi ai requisiti come a clausole di un contratto).

- **Usi dell'identificazione dei requisiti**

Detti riferimenti risultano utili a molti scopi, ad esempio: (ri-)negoiazione, collaudo,

validazione, risoluzione di controversie fra committente e produttore, etc..

Inoltre, la possibilità di *evoluzione* dei requisiti richiede ovviamente che ciascun requisito sia identificabile indipendentemente dalla forma in cui è espresso, che può cambiare.

### 8.1.3 Classificazione dei requisiti

- **Caratteristiche di qualità come categorie di classificazione**

Per agevolare la corrispondenza dei requisiti ad obiettivi di qualità, è utile usare questi ultimi come categorie di classificazione dei primi. In generale, risulta conveniente una classificazione in cui caratteristiche standard di qualità, ad esempio quelle della norma ISO 9126 (in Tab. 3.1), classifichino corrispondenti gruppi di requisiti e strutturino il documento di definizione dei requisiti in rispettive sezioni e sottosezioni.

- **Distinzione fondamentale: requisiti funzionali e non funzionali**

Sulla base di quanto detto, vanno innanzitutto distinti i requisiti funzionali, corrispondenti alla caratteristica generale di funzionalità, dai requisiti non funzionali, corrispondenti alle altre cinque caratteristiche generali definite dalla norma suddetta, in quanto queste due classi di requisiti si differenziano notevolmente nelle loro tipiche tecniche di descrizione.

### 8.1.4 Requisiti non funzionali

- **Oggettività e collaudabilità**

I *requisiti non funzionali*, nelle cinque categorie suddette, sono asserzioni relative al comportamento del sistema, la cui validità deve essere oggettivamente sperimentabile dall'utente.

Così, ad esempio, un enunciato quale "l'interfaccia di utente deve essere intuitiva", pur essendo consistente con la caratteristica generale di usabilità, e in particolare con quella di apprendibilità (v. Tab. 3.1), manca di oggettività. Questa è invece evidente in un requisito quale "l'interfaccia di utente presenta menù a tendina nel formato standard per questa linea di prodotti".

- **Vincoli quantitativi su fattori di qualità**

Talvolta la formulazione di requisiti non funzionali coincide con l'imposizione di vincoli quantitativi su determinati fattori di qualità. Ad esempio, nella categoria generale di efficienza rispetto al tempo, non si richiederà che il tempo di risposta sia "molto basso" bensì che non superi un certo valore.

### 8.1.5 Requisiti funzionali

- **Analisi del problema**

I *requisiti funzionali*, a differenza dei precedenti, possono essere determinati solo entrando nel merito delle funzionalità del prodotto, e dunque richiedono un, sia pur minimo, lavoro preliminare di analisi del problema, eventualmente in collaborazione con gli utenti al fine di meglio coglierne le esigenze.

- **Notazione standard per la specifica dei requisiti funzionali**

Per la descrizione dei requisiti funzionali si può adoperare una notazione, che in parte è diagrammatica ed in parte fa ricorso al linguaggio naturale, standardizzata dall'*Object Management Group* (OMG) all'interno della definizione di una più generale notazione per la

modellazione orientata agli oggetti, denominata *Unified Modeling Language* (UML).

- **Modellazione dei casi d'uso**

La programmazione orientata agli oggetti costituisce oggi il paradigma dominante nell'industria del software. Va però detto che la tecnica descrittiva che stiamo per introdurre, pur essendo parte di UML, non contiene nulla di specifico al paradigma degli oggetti, e può dunque essere impiegata qualunque sia l'approccio concettuale alla definizione dei requisiti. La tecnica in questione prende il nome di *modellazione dei casi d'uso*, ed è stata propugnata per primo da Ivar Jacobson (Jacobson et al. 1992). Ne illustriamo i tratti salienti.

## 8.2 Analisi e specifica di requisiti funzionali

### Casi d'uso



**Analisi: decomposizione della funzionalità del sistema in casi d'uso dello stesso**

Il tentativo di rendere quanto più semplice possibile la descrizione di requisiti che costituiscono una complessa specifica di funzionalità può, in accordo con i principi di separazione enunciati in Sez. 1.2, puntare a decomporre l'uso del sistema in *casi* distinti, ciascuno dei quali sia o relativamente semplice da descrivere o, altrimenti, decomponibile secondo lo stesso principio.

Si conviene che un caso d'uso sia abbastanza semplice da descrivere direttamente quando il comportamento stimolo-risposta del sistema nel caso d'uso in questione possa essere espresso in modo accettabilmente conciso da uno o più *scenari* del caso d'uso, cioè sequenze di interazioni elementari fra il sistema e le entità ad esso esterne, dette *attori* del caso d'uso, fra cui in primo luogo gli utenti.<sup>2</sup>

### 8.2.1 Modelli di casi d'uso

Un *modello di casi d'uso* è una descrizione di requisiti funzionali che consta di:

- Uno o più **diagrammi di casi d'uso**.

In ciascuno di questi diagrammi si evidenziano casi d'uso (rappresentati da ovali), eventuali relazioni fra essi (rappresentate da frecce etichettate, dove l'etichetta designa il tipo di relazione, di solito estensione o inclusione), e le interazioni (rappresentate da frecce) di ciascun caso d'uso con gli attori che vi intervengono, simboleggiati da un grafismo antropomorfo (anche quando l'attore non rappresenta un utente umano).

- Per ciascun caso d'uso, uno o più **scenari** delle interazioni tra il sistema e gli attori che vi intervengono.

Solitamente, gli scenari sono delle descrizioni narrative delle sequenze di interazioni dette.

- Zero o più **diagrammi di package di casi d'uso**: con questi diagrammi si descrive la composizione dell'insieme di requisiti funzionali in gruppi di casi d'uso.

Un *package* di casi d'uso, rappresentato da un rettangolo con un rettangolino attaccato, può contenere un diagramma di casi d'uso o un diagramma di package di casi d'uso; un diagramma di package di casi d'uso può mostrare *relazioni di dipendenza* fra package, rappresentate da frecce tratteggiate.

## 8.2.2 Diagrammi di casi d'uso

A mo' di esempio, in Fig. 8.1 è mostrato un diagramma di casi d'uso per un servizio di catalogazione e consultazione via Web di un archivio di documenti. Si suppone che l'archivio propriamente detto sia esterno al sistema che realizzerà il servizio. Ciò potrebbe essere stabilito da un vincolo posto nello studio di fattibilità (Sez. 6.1.3), ad esempio un archivio iniziale già esiste, o da una scelta operativa effettuata in fase di pianificazione (Sez. 6.2.2). Il fatto che l'archivio sia esterno al sistema di cui si tracciano i casi d'uso giustifica la sua rappresentazione come attore, denominato *Archivio* nel diagramma.

Si suppone ancora che la consultazione sia accessibile a tutti gli utenti, mentre l'accesso alla catalogazione sia riservato solo ad alcuni di essi. Anche questa differenziazione potrebbe essere frutto di un vincolo predeterminato. Tale distinzione nei diritti di accesso si riflette nella presenza di due attori diversi nel diagramma, rispettivamente denominati *Utente* e *Gestore*.

Notiamo infine che il caso d'uso *Consultare documento* estende il caso d'uso *Consultare catalogo*: questo significa che il secondo può essere eseguito come parte del primo, a determinate condizioni specificate negli scenari.

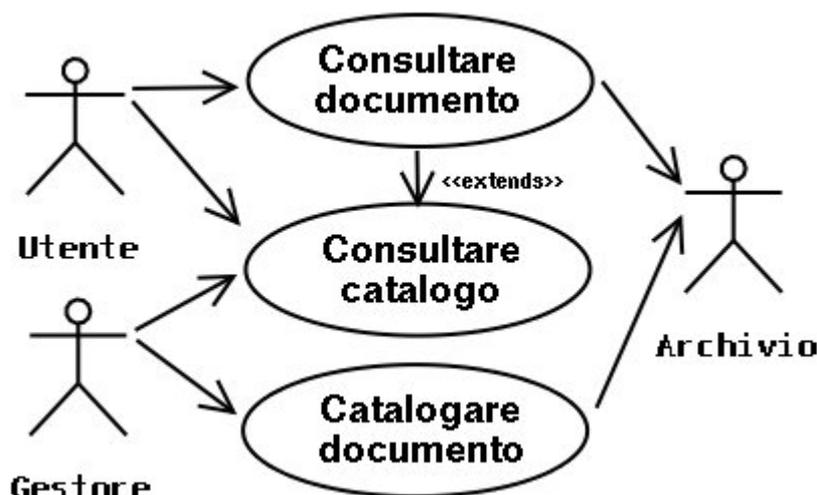


Figura 8.1: Un diagramma di casi d'uso

## 8.2.3 Diagrammi di package di casi d'uso

Il diagramma in Fig. 8.1 potrebbe essere il contenuto del package *Servizi on-line* di un diagramma di package che potrebbe contenere altri package, come ad esempio in Fig. 8.2, dove si è ipotizzato che alla funzionalità del package *Servizi on-line* siano da aggiungere quelle di gestione degli utenti, ad esempio per la creazione e rimozione delle rispettive registrazioni, e di autenticazione degli stessi, quale ad esempio quella tipica che prevede l'immissione di un nome e di una *password* da parte dell'utente per accedere ai servizi on-line.

Le dipendenze indicate nel diagramma prevedono che almeno un caso d'uso in *Autenticazione* dipenda da qualche caso d'uso in *Gestione utenti*, e che almeno uno in *Servizi on-line* (ad es., il caso d'uso *Consultare catalogo*) dipenda da qualche caso d'uso in *Autenticazione*. Intuitivamente, ciò significa: il buon esito dell'autenticazione è possibile solo se è previamente avvenuta una registrazione dell'utente, e questi ha accesso ai servizi on-line solo

se la sua precedente autenticazione ha buon esito.

Che siano *queste* le previste dipendenze non è però esprimibile nel diagramma di package, né può essere espresso nei diagrammi dei casi d'uso: occorre entrare nel dettaglio delle interazioni di cui si compongono i casi d'uso, ossia nella descrizione degli scenari.

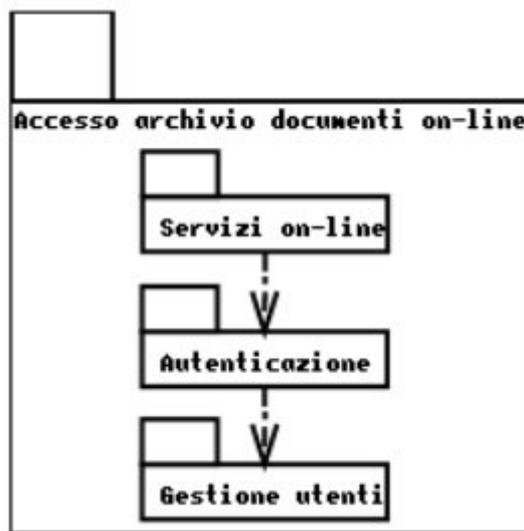


Figura 8.2: Un diagramma di package di casi d'uso

### 8.2.4 Descrizioni di casi d'uso

**Tecnica di descrizione frequentemente adottata: scenari in linguaggio naturale**

Mentre i diagrammi di casi d'uso, e gli eventuali diagrammi di package, danno una visione globale delle funzionalità presenti in un modello di casi d'uso, le *descrizioni di casi d'uso*, o *scenari*, specificano i dettagli, relativi a ciascun singolo caso d'uso, necessari alla precisione richiesta per il documento di specifica dei requisiti funzionali.

Uno stile diffuso di stesura degli scenari (v. ad esempio (Cheesman et al. 2001)) ne prevede una descrizione narrativa nella forma di una lista articolata di descrizioni di interazioni fra attori e sistema. È anche possibile utilizzare altri diagrammi UML, quali i diagrammi di sequenza, come ad esempio mostrato in (Conallen 2003), per ottenere sostanzialmente lo stesso risultato, ma in questa sede ci limitiamo ad illustrare lo stile narrativo di stesura.

Una caratteristica comune dei due approcci è quella di prevedere uno *scenario principale*, che costituisce il corso normale o il più frequente delle interazioni, a cui si aggiungono delle *estensioni*, che costituiscono diramazioni alternative da punti ben identificati dello scenario principale.

Per identificare agevolmente i punti di diramazione, detti anche *punti di estensione* dello scenario principale, tutti i punti di quest'ultimo sono numerati progressivamente. Anche quelli delle estensioni lo sono, giacché anch'esse possono a loro volta avere estensioni in dati punti di estensione.

Così, con riferimento all'esempio precedente, uno scenario per il caso d'uso Consultare catalogo in Fig. 8.1, che preveda il caso d'uso Consultare documento come estensione, potrebbe essere descritto come segue, dove il primo punto nello scenario principale è un riferimento ad un caso d'uso che ipotizziamo essere definito nel package Autenticazione.

## 8.2.5 Scenario principale

Identificatore: Consultare catalogo

Scenario principale:

1. Autenticazione:: Accedere ai servizi online  
{L'Utente accede alla URL del servizio e si identifica.}
2. L'Utente richiede di consultare il catalogo.
3. Il sistema presenta la pagina di ricerca del catalogo.
4. L'Utente immette un criterio di selezione di documenti.
5. Il sistema presenta la lista di elementi del catalogo, con rispettivi codici di collocazione a cui sono associati collegamenti ipertestuali, che soddisfano il criterio immesso al punto 4.
6. L'Utente reitera il punto 4.

## 8.2.6 Estensioni

Estensione:

Identificatore: Consultare documento

- 4. L'Utente attiva il collegamento ipertestuale associato ad un codice di collocazione, così richiedendo la consultazione del corrispondente documento in `Archivio`.
- 4.1 Il sistema estrae dall'`Archivio` il documento richiesto al punto 4.
- 4.2 Il sistema presenta all'Utente il documento estratto dall'`Archivio` al punto 4.1.
- 4.3 L'Utente reitera il punto 4.

Estensione:

Identificatore: Terminare la sessione di consultazione

- 4. L'Utente abbandona la pagina di ricerca del catalogo.

## 8.3 Negoziazione e validazione di requisiti

- **Contesto tipico: ciclo di vita evolutivo**

Il notevole livello di dettaglio tipico dei requisiti di sistema getta una nuova luce sulla negoziazione e validazione dei requisiti, specie nelle metodologie di progettazione partecipativa secondo modelli evolutivi, quali ad esempio i modelli a spirale ([Fig. 5.2](#)).

- **Evoluzione dei requisiti**

Da un canto, i requisiti sono praticamente soggetti ad un cambiamento "permanente" (per quanto ciclicamente controllato), dunque la loro stabilità è limitata ad una iterazione del processo di sviluppo, e ciò non tanto per mutamenti ambientali quanto per l'accumularsi di esperienza realizzativa che suggerisce nuove soluzioni e propone cambiamenti ai requisiti avendone sperimentato fattibilità e costi.

- **Partecipazione degli utenti alla revisione dei requisiti**

D'altro canto, il numero e la complessità dei requisiti di sistema pongono nei fatti una sfida alla loro effettiva negoziabilità e convalidabilità da parte degli utenti. Per garantire queste ultime occorre innanzitutto una ben disciplinata organizzazione dei requisiti, come suggerito in [Sez. 8.1.1](#), ma di questa va anche fatto uso per agevolare la partecipazione degli utenti al processo permanente di revisione dei requisiti, ad esempio con modalità analoghe alle ispezioni formali a cui si è accennato in [Sez. 4.2.3](#).

### 8.3.1 Analisi di consistenza di requisiti

- **Matrice di dipendenza dei requisiti**

Per l'efficacia della validazione occorre poi tener presente che essa acquista una decisa valenza tecnica, in quanto non si riduce a mera constatazione di accettabilità da parte degli utenti (specie se non adeguatamente informati), ma esige una attenta *analisi di consistenza* dei requisiti stessi. All'analisi di consistenza può essere utile la costruzione di una *matrice di dipendenza* dei requisiti (Sommerville et al. 1997), (Kotonya et al. 1998), che, analogamente a quanto discusso in [Sez. 2.3.1](#), prevede che requisiti distinti possano essere indipendenti, o in sovrapposizione, o in conflitto.

- **Risoluzione di conflitti fra requisiti**

La negoziazione dei requisiti richiede dunque tecniche di risoluzione di conflitti fra gli stessi. A tale scopo torna utile l'attribuzione di *priorità* ai requisiti, preferibilmente quale risultato di un'attenta analisi dei rischi, di cui si è detto nella [lezione precedente](#).

## 8.4 Esercizi: specifica di requisiti

### 8.4.1 Premessa



ipotesi di lavoro: v. [Sez. 5.3.1](#)

### 8.4.2 Esercizio 1: requisiti non funzionali

Dare un esempio di requisito non funzionale, di validità oggettivamente sperimentabile nel sito web di cui all'ipotesi di lavoro, per ciascuna delle cinque caratteristiche generali di qualità (oltre a "Funzionalità") definite dalla norma ISO 9126 (v. [Tab. 3.1](#)).

### 8.4.3 Esercizio 2: requisiti di accessibilità

Dare esempi di requisiti per il sito web di cui all'ipotesi di lavoro, che siano di validità oggettivamente sperimentabile, e classificabili nella categoria dei requisiti di accessibilità.

### 8.4.4 Esercizio 3: requisiti funzionali

Specificare un modello di casi d'uso, per i requisiti funzionali del sito web di cui all'ipotesi di lavoro, che consti di almeno un diagramma di casi d'uso, e stilare la descrizione di almeno uno dei casi d'uso del modello.

## 8.5 Note

1. Tutto dovrebbe essere reso il più semplice possibile, ma neanche un poco più semplice.
2. Ma non *esclusivamente* gli utenti, come si illustra nell'esempio che segue, e contrariamente a quanto sostenuto in (Conallen 2003), che su questo semplifica un po' troppo.

## 8.6 Bibliografia

- **Cheesman, J., Daniels, J.**, 2001. *UML Components - Prima edizione italiana*. Addison-Wesley, Pearson Education Italia.  
Web: <http://hpe.pearsoned.it>
- **Conallen, J.**, 2003. *Applicazioni web con UML - II Ed. Italiana*. Addison-Wesley, Pearson Education Italia.  
Web: <http://hpe.pearsoned.it>
- **Jacobson, I., Christerson, M., Jonsson, P., & Övergaard, G.**, 1992. *Object-Oriented Software Engineering: A Use Case Driven Approach*. Addison-Wesley.
- **Kotonya, G. & Sommerville, I.**, 1998. *Requirements Engineering: Processes and Techniques*. John Wiley & Sons.  
Web: <http://eu.wiley.com>
- **Sommerville, I. & Sawyer, P.**, 1997. *Requirements Engineering: A Good Practice Guide*. John Wiley & Sons.  
Web: <http://eu.wiley.com>
- **OMG**. *Object Management Group*. Web: <http://www.omg.org>.
- **UML**. *Unified Modeling Language*. Web: <http://www.omg.org/uml>.