

UNIVERSITÀ DEGLI STUDI DI TRIESTE

FACOLTÀ DI INGEGNERIA

Corso di Laurea Triennale in Ingegneria Gestionale

---

Tirocinio in

RICERCA OPERATIVA

**DOCUMENTO DI REQUISITI E SPECIFICHE  
PER L'INTERFACCIA DI UN SISTEMA  
INFORMATIVO DI TIMETABLING**

**Cristina CALUCCI**

---

Trieste, 22 aprile 2005

# Indice

<b>1</b>	<b>Introduzione</b>	<b>3</b>
<b>2</b>	<b>Requisiti utente</b>	<b>4</b>
2.1	Contenuto . . . . .	4
2.2	Requisiti funzionali . . . . .	5
2.2.1	Attori . . . . .	6
2.2.2	Casi d'uso . . . . .	6
2.3	Requisiti non funzionali . . . . .	8
<b>3</b>	<b>Piano di qualità</b>	<b>13</b>
3.1	Contenuto . . . . .	13
3.2	Controllo di qualità . . . . .	15
3.3	Standard di prodotto . . . . .	16
3.4	Processo di sviluppo . . . . .	16
3.5	Gestione dei rischi . . . . .	17
<b>4</b>	<b>Specifiche software</b>	<b>18</b>
4.1	Contenuto . . . . .	18
4.2	Disegno architettuale . . . . .	18
4.3	Tecnologia ed infrastruttura . . . . .	19
4.4	Dettaglio dei casi d'uso . . . . .	19
<b>5</b>	<b>Pianificazione di rilascio</b>	<b>28</b>
5.1	Contenuto . . . . .	28
5.2	Organizzazione del progetto . . . . .	28
5.3	Risorse hardware e software . . . . .	28
5.4	Attività . . . . .	29
5.5	Analisi dei rischi . . . . .	29
5.6	Stima dei costi . . . . .	30
5.6.1	Stima dei costi . . . . .	30
	<b>Bibliografia</b>	<b>31</b>
	<b>Elenco delle figure</b>	<b>32</b>

# Capitolo 1

## Introduzione

Questo documento fa parte della fase preliminare allo sviluppo di un sistema informativo di Timetabling per la facoltà di Ingegneria dell'Università degli Studi di Trieste. Tale fase preliminare consiste nello stabilire quali sono i requisiti e le specifiche per la realizzazione di un'Interfaccia Utente idonea al sistema. L'obiettivo che ci si pone in questo documento è dunque quello di analizzare come dovrà presentarsi l'Interfaccia in modo da garantire il suo utilizzo in modo pratico, semplice e duraturo.

Il focus fondamentale di questo documento è fare in modo che l'uso dell'Interfaccia risulti gradevole all'utenza. Questo significa che l'utente dovrà essere soddisfatto di interagire con l'Interfaccia, e implicitamente con l'intero sistema. Solo in questo modo si può giungere alla conclusione che il vecchio sistema di compilazione degli orari è stato completamente rimpiazzato nel modo più efficiente ed efficace possibile.

Il lavoro è articolato in quattro parti per renderlo più scorrevole e facile da consultare:

1. REQUISITI UTENTE, focus sull'analisi di quali sono i requisiti funzionali (cioè a *cosa serve* il sistema, cosa dovrà fare per essere di aiuto all'utente) e i requisiti non funzionali (vincoli ai quali deve sottostare un software per essere conforme alle norme vigenti);
2. PIANO DI QUALITA', focus su quelle che sono le variabili che stabiliscono la qualità di un software;
3. SPECIFICHE SOFTWARE, focus su quello che l'Interfaccia dovrà fare concretamente;
4. PIANIFICAZIONE DI RILASCIO, focus sull'ottimizzazione dello sviluppo (organizzazione, risorse necessarie, analisi dei rischi, ecc).

## Capitolo 2

# Requisiti utente

### 2.1 Contenuto

Questo capitolo descrive i requisiti utente per la realizzazione dell'Interfaccia di un sistema informativo di Timetabling per la Facoltà di Ingegneria dell'Università degli Studi di Trieste. Tali requisiti sono il risultato del colloquio con la Responsabile degli orari della segreteria di Facoltà. L'Interfaccia e il software di Timetabling hanno il primo obiettivo di automatizzare l'attuale sistema in uso per la compilazione delle tabelle di orari, il quale momentaneamente è in prevalenza manuale. Il nuovo sistema andrà dunque a velocizzare le attività per quanto concerne la realizzazione degli orari di studenti e docenti dell'Università. L'Interfaccia dovrà permettere:

- buona comunicazione con l'utenza;
- facilità di scelta delle metodologie di compilazione;
- facilità di ricerca e modifica dell'archivio orari.

Essa deve quindi svolgere attività interattive di consultazione, archiviazione e deve garantire inoltre la possibilità di scegliere la modalità di sviluppo degli orari impiegando hardware standard (che verrà illustrato nel capitolo 4 e nel capitolo 5 ).

Non sono emerse specifiche necessità di interfacciamento ma è bene che il sistema sia conforme con le correnti tecnologie LAN, WAN e Internet in particolare. E' ovviamente ritenuta indispensabile la conformità alle specifiche della rete elettrica e telefonica italiana e alle norme relative all'emissione di onde elettromagnetiche sia per quanto riguarda l'interferenza che gli effetti sulla salute. Gli utenti devono poter usufruire dell'Interfaccia mediante periferiche di input e output di uso comune come mouse, tastiera, monitor e stampante.

Il sistema di calcolo degli orari verrà utilizzato essenzialmente dalla Segreteria e dalla Presidenza della Facoltà di Ingegneria, quindi si può supporre

che non sia strettamente necessaria una formazione adeguata del personale in quanto certamente avrà già a che fare con altri sistemi software. In ogni caso la semplicità dell'Interfaccia renderà l'utilizzo piacevole e confortevole all'utenza. La parte relativa alla consultazione e alla ricerca potrà inoltre essere utilizzata da docenti e studenti interessati. A questo scopo si rende utile un'Interfaccia Utente uniforme. Il sistema deve implementare le funzioni descritte in conformità con le norme relative alla qualità del software e rispettare gli standard tecnologici indicati.

Viene inoltre individuata una futura linea di evoluzione del sistema: l'Interfaccia Utente e il software potrebbero venir estesi a tutto l'Ateneo. Ciò comporterebbe l'aumentare del numero di corsi e di docenti e dunque l'aumentare delle variabili in input al sistema.

## 2.2 Requisiti funzionali

I requisiti funzionali descrivono una funzionalità del software: sono una definizione formale di ciò che farà il software e in particolare le sue parti viste come moduli. Per "modulo" infatti si intende proprio "funzione", "funzionalità". Nel caso più semplice vengono presi in considerazione solo i rapporti tra gli input e gli output per ogni modulo.

I requisiti funzionali qui raccolti sono incompleti in quanto verranno in seguito dettagliati nella fase di specifica dei requisiti di sistema (specifiche software) e illustrati con dovizia di particolari nel capitolo 4.

Lo scopo di definire i requisiti funzionali è quello di individuare per prima cosa gli "attori", cioè gli utilizzatori primari del sistema. Una volta individuati gli attori bisogna andare a definire i "casi d'uso", cioè come gli attori interagiscono con il sistema. Tali casi d'uso rappresentano proprio ciò che il sistema dovrà fare, le funzioni che dovrà svolgere, sono dunque la soluzione ai quesiti che l'utente ci ha posto a fronte di un problema che gli si è presentato. Nel nostro caso l'utente ha richiesto un software che faciliti la compilazione degli orari, quindi dobbiamo cercare di rendere chiare le idee su cosa tale sistema dovrà fare per portare un beneficio all'utenza.

Per descrivere i requisiti funzionali è stato scelto di usare il linguaggio UML (Unified Modeling Language), in particolare la descrizione di Use Case. L'UML è uno strumento per progettisti di sistemi informativi che consente di modellare e documentare sistemi software utilizzando forme grafiche. Lo scopo del linguaggio UML è dunque quello di garantire un approccio visuale alla progettazione. I Casi d'Uso (Use Case) sono dei particolari tipi di diagrammi UML, sono particolarmente utili per schematizzare una vista esterna del sistema con il fine di rendere più chiaro il dialogo tra utilizzatore finale e sistema.

Andiamo dunque a illustrare quali saranno i futuri utenti (attori) e le future funzionalità (casi d'uso) del sistema. Ciò è utile per individuare i

confini del sistema, dettagliarne i compiti ed eventualmente progettarne i test.

### 2.2.1 Attori

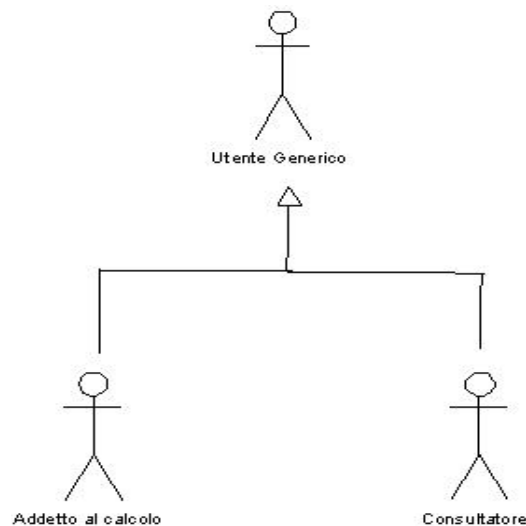


Figura 2.1: Attori

- **Utente generico:** questo attore è un utente generico da cui tutti gli altri attori sono derivati.
- **Addetto al calcolo:** qualsiasi persona che può accedere all'area di calcolo degli orari; a differenza del consultatore, può avere accesso alla parte di software relativa al calcolo e alla modifica, si suppone quindi l'inserimento di un nome utente e di una password per garantire la sicurezza dei dati.
- **Consultatore:** qualsiasi persona, studenti, docenti o addirittura esterni che sono interessati a consultare gli orari; essi possono accedere solo ed esclusivamente alla parte di software relativa alla ricerca di un orario nell'archivio.

### 2.2.2 Casi d'uso

- **Obiettivi dell'utente generico: ACCESSO AL SISTEMA**

Con questa operazione l'Interfaccia Utente mette a disposizione dell'utente una serie di scelte opzionali per la navigazione all'interno del sistema: RICERCA, CALCOLO, MODIFICA.

- **Obiettivi dell'addetto al calcolo: AUTENTICAZIONE**

Con questa operazione l'addetto al calcolo dichiara la propria identità. Il sistema mette a disposizione dell'utente autenticato la possibilità di calcolare gli orari o modificare orari preesistenti.

- **Obiettivi dell'addetto al calcolo: CALCOLO DEGLI ORARI**

Questa operazione prevede l'inserimento di una serie di input:

1. il semestre di cui si intende calcolare gli orari; questa distinzione permette di snellire gli input seguenti;
2. tutti i vincoli e i parametri necessari ad elaborare l'intero orario.

Il sistema dovrebbe evidenziare se ci sono vincoli in conflitto tra loro, in tal caso è necessario modificare gli input. Una volta calcolato, l'orario va salvato in un archivio dal quale può essere estratto ogni qual volta ne venga fatta richiesta esplicita.

- **Obiettivi dell'addetto al calcolo: MODIFICA DI UN ORARIO PREESISTENTE**

Questa operazione prevede la modifica di uno o più orari presenti in archivio a causa di variazioni delle condizioni iniziali. Attraverso l'immissione degli input come nel Caso d'Uso precedente individuo il docente o l'insegnamento di cui bisogna modificare l'orario. Si prosegue con un nuovo calcolo alla fine del quale il nuovo orario viene salvato nuovamente nell'archivio al posto di quello modificato.

- **Obiettivi dell'addetto al calcolo: ASSEGNAZIONE MANUALE DI UN ORARIO**

Questa operazione prevede l'inserimento manuale di una tabella di orari. Attraverso l'immissione dei parametri di ricerca, il sistema dà in output una tabella (già compilata nel caso in cui l'orario esista, vuota in caso contrario) in cui l'addetto può assegnare manualmente le ore di lezione. Il sistema elabora le informazioni e verifica che esse siano compatibili con orari esistenti, poi procede con la memorizzazione se la compatibilità è verificata.

- **Obiettivi del consultatore: RICERCA DI UN ORARIO SPECIFICO**

Questa operazione prevede la ricerca di un orario già presente nell'archivio a seguito della specifica di uno di questi criteri di ricerca:

1. per Corso di Laurea;
2. per docente;
3. per aula.

## 2.3 Requisiti non funzionali

I requisiti non funzionali rappresentano i vincoli ai quali deve sottostare il software. L'Interfaccia Utente e il sistema devono soddisfare gli attributi di qualità definiti dallo standard ISO 9126.

La Norma ISO 9126 ("Information Technology - Software product evaluation - Quality characteristics and guidelines for their use"), pubblicata nella sua prima versione nel 1991, definisce il modello dei requisiti qualitativi del Prodotto Software.

Secondo il modello proposto dalla norma, i requisiti sono raggruppabili in 6 "caratteristiche" e in 21 "sottocaratteristiche", distinte fra caratteristiche esterne (orientate all'utente) e caratteristiche interne (orientate allo sviluppo e manutenzione).

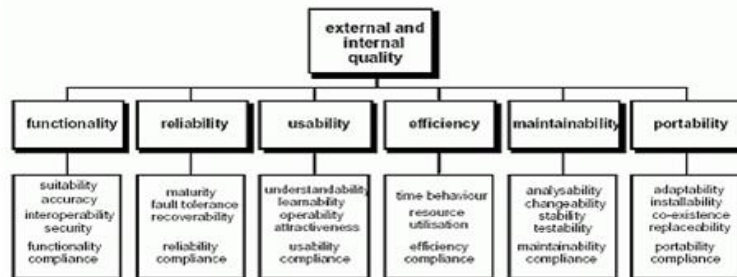


Figura 2.2: Caratteristiche e sottocaratteristiche

La presenza dei requisiti richiesti può essere verificata tramite riesami e test. Ogni caratteristica deve essere valutata tramite questionari e tabelle, da compilare assieme al cliente.



Figura 2.3: ISO 9126

Vengono ora illustrate le caratteristiche e sottocaratteristiche analizzate per il progetto in questione.

1. **FUNZIONALITA'**, cioè capacità di fornire funzioni tali da soddi-



sfare i requisiti funzionali espliciti o impliciti (il software fa ciò per cui è stato sviluppato)

- (a) *ADEGUATEZZA*, cioè si parla di funzioni appropriate per compiti specifici; l'indicatore da usare è:

$$\text{copertura funzionale} = \frac{\text{numero di funzioni presenti nel prodotto}}{\text{numero di funzioni specificate nel progetto esecutivo}}$$

Si misura con visite ispettive sul prodotto. Soglia minima: 100

- (b) *ACCURATEZZA*, cioè capacità di fornire risultati ed effetti in accordo con i requisiti; l'indicatore da usare è:

$$\text{accuratezza funzionale} = \frac{\text{numero di funzioni esattamente descritte nei requisiti}}{\text{numero totale di funzioni}}$$

Si misura con verifiche nella documentazione. Soglia minima: 80

- (c) *INTEROPERABILITA'*, cioè capacità di interagire con altri sistemi; gli indicatori da usare sono:

- accessibilità agli archivi con tecnologie standard = 
$$\frac{\text{numero entità accessibili}}{\text{numero entità specificato nel progetto esecutivo}}$$
 Si misura con verifiche sul DBMS (Data Base Management System, è un sistema software in grado di gestire una collezione di dati). Soglia minima: 100

- capacità del sistema di essere usato con periferiche di uso corrente = 
$$\frac{\text{numero di funzioni accessibili}}{\text{numero di funzioni specificato nel progetto esecutivo}}$$
 Si misura con verifiche ispettive sul prodotto. Soglia minima: 100

- (d) *SICUREZZA*, cioè capacità di evitare accessi non autorizzati a programma e dati; l'indicatore da usare è:

$$\text{sicurezza delle informazioni} = \frac{\text{num informazioni con accesso limitato}}{\text{num informaz con accesso limitato specificato nel progetto esec}}$$

Si misura con verifiche ispettive sul prodotto. Soglia minima: 100

- (e) *ADERENZA*, cioè rispetto di standard, convenzioni e leggi relative alle funzioni svolte dall'applicazione; l'indicatore da usare è:

$$\text{possibilità di accedere alle funzioni tramite Web} = \frac{\text{num funzioni pubbliche accessibili}}{\text{num funzioni pubbliche accessibili specificato nel progetto esec}}$$

Si misura con verifiche ispettive sul prodotto con Browsers diversi. Soglia minima: 80

2. *AFFIDABILITA'*, cioè capacità di mantenere le prestazioni stabilite nelle condizioni e per periodi di tempo prefissati

- (a) **ROBUSTEZZA**, cioè capacità di limitare malfunzionamenti a seguito di difetti nel software; l'indicatore da usare è:  
tempo medio tra il manifestarsi di malfunzionamenti = 
$$\frac{\text{ore di attività del sistema trascorse}}{\text{numero di malfunzionamenti manifestati}}$$
Si misura con verifiche nell'archivio di segnalazione d'errore. Soglia minima: 1000 ore
- (b) **TOLLERANZA AGLI ERRORI**, cioè capacità di mantenere determinati livelli di prestazione in caso di difetti nel software; l'indicatore da usare è:  
tasso di errori che hanno provocato un fermo del sistema = 
$$\frac{\text{numero degli errori applicativi che hanno provocato fermi}}{\text{numero totale degli errori applicativi}}$$
Si misura con verifiche nell'archivio di segnalazione d'errore. Soglia minima: 1
- (c) **RECUPERABILITA'**, cioè capacità di ripristinare i livelli di prestazione prestabiliti e di recuperare dati coinvolti in seguito a un malfunzionamento; l'indicatore da usare è:  
tempo medio di rimozione malfunzionamenti = 
$$\frac{\text{tempo totale di attesa per rimozione dei malfunzionamenti}}{\text{numero totale di malfunzionamenti}}$$
Si misura con verifiche nell'archivio di segnalazione d'errore. 50 ore
3. **USABILITA'**, cioè capacità del software di essere usato (quindi compreso e appreso) dagli utenti
- (a) **COMPENSIBILITA'**, cioè impegno necessario agli utenti per comprendere la logica di funzionamento; l'indicatore da usare è:  
tempo necessario ad un operatore medio per eseguire una funzione (elaborazione esclusa)  
Si verifica mediante l'osservazione dell'utenza. Soglia massima: 10 minuti
- (b) **APPRENDIBILITA'**, cioè impegno richiesto per utilizzare il software; l'indicatore da usare è:  
disponibilità di help in linea = 
$$\frac{\text{numero totale di help in linea}}{\text{numero totale di funzioni}}$$
Si misura con verifiche sul prodotto. Soglia minima: 50
- (c) **OPERABILITA'**, cioè impegno richiesto per utilizzare il software; gli indicatori da usare sono:
- tempo medio tra un errore umano e l'altro = 
$$\frac{\text{tempo totale di osservazione}}{\text{numero totale di errori umani osservati}}$$
Si misura con osservazioni sugli utenti. Soglia massima: 20 minuti

- uniformità dei comandi = 
$$\frac{\text{numero dei comandi con formato standard}}{\text{numero totale dei comandi}}$$
 Si misura con verifiche sul prodotto. Soglia minima: 100
  - disponibilità di valori predefiniti = 
$$\frac{\text{num valori inseribili con valorizz predefinita}}{\text{num tot valori inserib con valorizz predef secondo prog eseg}}$$
 Si misura con verifiche sul prodotto. Soglia minima: 90
  - disponibilità di liste di selezione dei possibili valori in input = 
$$\frac{\text{num liste di selezione disponibili}}{\text{num tot campi con liste di selezione secondo il prog eseg}}$$
 Si misura con verifiche sul prodotto. Soglia minima: 90
4. **EFFICIENZA**, cioè rapporto tra prestazioni e quantità di risorse utilizzate, in condizioni prestabilite di funzionamento (il software usa bene le risorse disponibili)
- (a) *COMPORAMENTO RISPETTO AL TEMPO*, cioè tempi di risposta e tempi di elaborazione; gli indicatori da usare sono:
- tempo medio di risposta alla richiesta di esecuzione di una funzione = 
$$\frac{\text{totale dei tempi di risposta osservati}}{\text{numero totale di funzioni richieste al sistema}}$$
 Si misura con simulazioni. Soglia massima: 5 secondi
  - tempo medio di elaborazione = 
$$\frac{\text{totale dei tempi di elaborazione osservati}}{\text{numero totale di funzioni richieste al sistema}}$$
 Si misura con simulazioni. Soglia massima: 15 secondi
- (b) *IMPIEGO DI RISORSE*, cioè quantità e durata di impiego eseguendo le funzioni attese; l'indicatore da usare è:
- costo del Server = 
$$\frac{\text{costo dell'hardware}}{\text{numero di operatori serviti contemporaneamente}}$$
 Si misura con simulazioni. Soglia massima: 500 Euro
5. **PORTABILITA'**, cioè facilità con cui il software può essere trasferito da un sistema operativo ad un altro (il software risulta essere libero da dipendenze vincolanti)
- (a) *ADATTABILITA'*, cioè capacità del software di essere adattato a diversi sistemi operativi applicando le sole azioni previste per questa operazione; l'indicatore da usare è:
- tasso di esportabilità dei dati = 
$$\frac{\text{numero di entità esportabili}}{\text{numero di entità presenti in archivio}}$$
 Si misura con verifiche ispettive sul prodotto. Soglia minima: 100

- (b) *INSTALLABILITA'*, cioè impegno richiesto per installare il software su un sistema operativo; l'indicatore da usare è il tempo impiegato.  
Si misura mediante l'osservazione di un addetto medio. Soglia massima: 4 ore
- (c) *COESISTENZA*, cioè capacità del software di coesistere con altri software condividendo risorse. Se si verificano problemi vanno considerati come malfunzionamenti.
- (d) *SOSTITUIBILITA'*, cioè capacità del software di essere utilizzato al posto di un altro software per svolgere gli stessi compiti. E' una caratteristica che in questo caso non è richiesta.
- (e) *ADERENZA*, rispetto di standard e norme relative alla portabilità. E' una caratteristica che in questo caso non è richiesta.

## Capitolo 3

# Piano di qualità

### 3.1 Contenuto

Questo capitolo intende definire quali sono gli obiettivi di qualità del progetto e definire gli step per raggiungerli. Gli elementi analizzati sono: attributi di qualità, processo di sviluppo adatto, modalità di controllo della qualità, rischi che possono affliggere la qualità. Tali elementi forniscono un aiuto per organizzare le attività del progetto, in particolare per quanto riguarda la qualità.

Il progetto di automazione dell'attività di Timetabling della Facoltà di Ingegneria dell'Università degli Studi di Trieste è destinato a sostituire l'attuale compilazione prevalentemente manuale delle tabelle orari. Il nuovo software dunque mira a snellire la procedura in uso diminuendo tempi, costi e sforzi. Il progetto in analisi verrà sviluppato su misura per le particolari esigenze della Segreteria e Presidenza dell'Università degli Studi di Trieste. Seguono eventuali richieste di evoluzione dell'utilizzo per tutto l'Ateneo.

Non ci sono reali vincoli temporali per il completamento del software, quindi il tempo non è un elemento critico ai fini della riuscita del progetto.

Gli obiettivi di qualità emersi dall'intervista con l'utenza sono espressi in modo dettagliato nel capitolo 2 seguendo gli standard del modello di qualità ISO 9126.

Gli obiettivi di qualità di interesse del produttore riguardano la manutenibilità e verranno qui precisati secondo lo standard ISO 9126.

La **MANUTENIBILITA'** è la capacità del software di essere modificato con impegno contenuto a causa di correzioni, miglioramenti o adattamenti dovuti a cambiamenti nell'ambiente di impiego o nei requisiti o nelle specifiche funzionali (il software segue l'evoluzione dell'organizzazione)

1. **ANALIZZABILITA'**, cioè impegno per identificare malfunzionamenti e quindi le parti da modificare; gli indicatori da usare sono:

- percentuale di classi pubbliche i cui servizi sono documentati e modellati = 
$$\frac{\text{num classi pubbliche i cui servizi sono documentati e modellati}}{\text{numero totale di classi pubbliche}}$$
 Si misura con verifiche ispettive sul codice. Soglia minima: 75
- percentuale di classi pubbliche correlate con i requisiti utente = 
$$\frac{\text{numero di funzioni utente con Use Case}}{\text{numero totale di funzioni utente}}$$
 Si misura con verifiche ispettive sul codice. Soglia minima: 70

2. **MODIFICABILITA'**, cioè impegno richiesto per effettuare modifiche; gli indicatori da usare sono:

- profondità di nidificazione delle istruzioni condizionali, serve a migliorare la leggibilità del codice. Si misura con analisi automatiche del codice.
- complessità di controllo di flusso del programma. Si misura con analisi automatiche del codice.
- percentuale di metodi in overriding = 
$$\frac{\text{numero dei metodi in overriding}}{\text{numero totale di metodi}}$$
 Riduce la dipendenza strutturale del codice. Si misura con analisi automatiche del codice. Soglia massima: 20

3. **STABILITA'**, cioè ridotto rischio di effetti inaspettati a seguito di modifiche; gli indicatori da usare sono:

- profondità massima dell'albero di ereditarietà Si misura con analisi automatiche del codice. Soglia massima: 5
- numero medio di chiamate da e per un metodo di un'altra classe = 
$$\frac{\text{numero totale chiamate da e per un metodo di un'altra classe}}{\text{numero totale di metodi}}$$
 Si misura con analisi automatiche del codice. Soglia massima: 5

4. **PROVABILITA'**, cioè impegno richiesto per validare le modifiche apportate al software; l'indicatore da usare è:

$$\text{percentuale dei requisiti che dispongono di test regressivi} = \frac{\text{numero dei requisiti che dispongono di test regressivi}}{\text{numero totale dei requisiti}}$$

Si misura con ispezioni del codice. Soglia minima: 80

### 3.2 Controllo di qualità

Gli obiettivi di qualità sono stati espressi attraverso attributi di qualità stimati con indicatori misurabili. Effettuare tali misure comporta dei costi che risultano necessari per:

1. ridurre i malfunzionamenti che portano il software a non essere più utile allo scopo;
2. ridurre i costi che l'utente sostiene in termini di perdita di informazioni o interruzione esercizio;
3. ridurre i costi di evoluzione che aumentano con la complessità del sistema.

Il controllo di qualità si concretizza con la misurazione degli indicatori descritti. Tale misurazione verrà svolta a campione su alcune parti del prodotto che saranno scelte con il fine di individuare il campione più rappresentativo:

1. parti che presentano maggiore incertezza o difficoltà;
2. parti che hanno maggiore influenza sul resto;
3. parti ritenute critiche dall'utenza.

In base all'esito delle misurazioni si prosegue come segue:

- se i valori sono conformi alle aspettative non sono necessarie azioni correttive;
- se i valori sono conformi alle aspettative ma molto prossimi ai limiti è necessario effettuare altri controlli;
- se i valori sono minori alla soglia minima accettabile dovranno essere attuate opportune azioni correttive con un controllo più attento.

Gli ultimi due punti causano un consistente aumento dei costi. E' per questo che è bene adottare opportuni standard di prodotto e di processo di sviluppo.

### 3.3 Standard di prodotto

Gli standard di prodotto permettono una buona economia di scala in quanto danno la possibilità di avere a che fare con classi di prodotti omogenee da cui prendere spunto. Lo stesso vale per i processi di sviluppo.

Gli standard di prodotto riguardano:

1. ***Documentazione e modellazione***
2. ***Selezione delle tecnologie:***
  - (a) Sistema Operativo (Linux, Windows 98, ecc);
  - (b) Supporto Run-Time (.NET Framework);
  - (c) Web Server (MS IIS);
  - (d) Ambiente di sviluppo (Visual Studio .NET);
  - (e) Linguaggio (C++, C#, ecc);
  - (f) DBMS (Oracle, ecc).
3. ***Selezione delle opzioni tecnologiche e infrastrutturali:***
  - (a) Distributed Object Broker;
  - (b) Data Services (ADO.NET);
  - (c) Opzioni del Web Browser (HTML 4.0, DHTML, ecc);
  - (d) Estensioni del modello relazionale (XML).
4. ***Selezione dell'architettura applicativa in base alla distribuzione fisica di Data Services, Business Services e User Services***
5. ***Selezione dell'architettura software:***
  - (a) Struttura del sistema (Repository, Multi-Tier);
  - (b) Modello di controllo.
6. ***Regole di codifica (convenzioni sui nomi, ecc)***

### 3.4 Processo di sviluppo

Gli elementi che caratterizzano il progetto dell'Interfaccia Utente sono i seguenti:

- dimensione del prodotto piccola;
- gruppo di sviluppo di dimensioni ridotte ( è sufficiente un programmatore);



- vincolo di tempo trascurabile;
- obiettivi di qualità di media complessità.

Da queste condizioni si evince che il modello di sviluppo che appare adeguato potrebbe essere il Modello a Cascata, magari semplificato o adattato a seconda delle esigenze, cui affiancare una fase di Prototipizzazione per incrementare la partecipazione e l'interesse dell'utente. Il Modello a Cascata è uno dei diversi modelli per lo sviluppo di un prodotto software. Prevede la realizzazione del prodotto seguendo una serie di fasi consecutive, "a cascata" per l'appunto. Tali fasi possono essere schematizzate come segue:

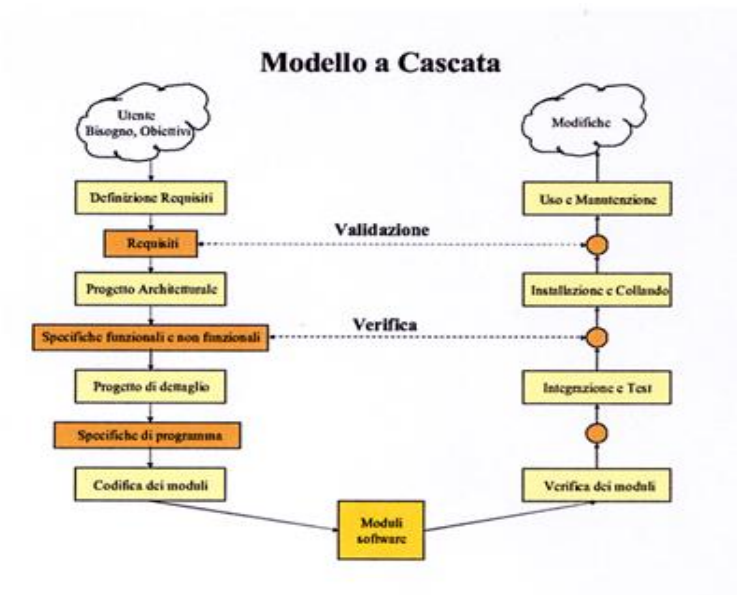


Figura 3.1: Modello a cascata

La Prototipizzazione consiste nel costruire un sistema in grado di mostrare tutte le funzionalità del prodotto finale, ma che non rispetta per forza tutti i requisiti non funzionali.

### 3.5 Gestione dei rischi

Rischio: malfunzionamento del software

Descrizione: instabilità o inaffidabilità

Probabilità: moderata

Conseguenze: serie

Piano per abbassare la probabilità di rischio: effettuare test preventivi e garantire assistenza.

Ulteriori rischi verranno elencati nel capitolo 5.

## Capitolo 4

# Specifiche software

### 4.1 Contenuto

Questo capitolo descrive le specifiche software dell'Interfaccia Utente per il sistema informativo di Timetabling della Facoltà di Ingegneria dell'Università degli Studi di Trieste. Le specifiche comprendono:

- definizione dei dettagli generali dell'architettura;
- definizione delle tecnologie impiegate per il software di base e l'infrastruttura;
- definizione in dettaglio dei Casi d'Uso.

Questo documento parte dal DOCUMENTO DEI REQUISITI UTENTE e dal PIANO DI QUALITA' ed è alla base della successiva attività di pianificazione.

### 4.2 Disegno architetturale

Il sistema di Timetabling deve avere una struttura a livelli in modo da avere un sistema distribuito:

1. il primo livello è il Repository, cioè un DataBase che fornisce i dati;
2. il secondo livello è il Business Services, cioè la logica, l'algoritmo in grado di calcolare gli orari;
3. il terzo livello è lo User Services, dove è necessaria l'Interfaccia Utente.

Un sistema distribuito garantisce l'indipendenza e la snellezza dei suoi componenti: è infatti possibile modificare ciascuno dei tre livelli senza dover

per forza intaccare gli altri. Ciò risulta utile in caso di obsolescenza dell'algoritmo (modifico il secondo livello senza intaccare i dati) oppure nel caso in cui si desidera cambiare l'Interfaccia (non serve modificare la logica).

Il terzo livello richiede competenze di grafica, sicurezza e programmazione in quanto fornisce all'utente finale servizi di presentazione, interazione, autenticazione, autorizzazione. Esso riguarda le finestre dell'Interfaccia Utente. L'interazione si appoggia al livello della logica ed eventualmente al DataBase per la ricerca. L'autenticazione e l'autorizzazione riguardano espressamente il personale della Segreteria e della Presidenza e si verificano mediante l'inserimento di nome utente e password; si rende dunque necessario l'uso di un DataBase che contenga i dati di tutti gli utenti che possono avere accesso alla parte del sistema che effettua il calcolo degli orari.

### 4.3 Tecnologia ed infrastruttura

Le scelte tecnologiche operate per il progetto di Interfaccia Utente quindi dal livello User Services sono:

- Sistema Operativo: Windows 98 e successivi;
- Web Server: MS IIS;
- Supporto Run-Time: .NET Framework;
- Ambiente di sviluppo: Visual Studio .NET;
- Linguaggio: C#;
- Distributed Object Broker;
- Web Browser standard (Internet Explorer 3.0 e successivi).

### 4.4 Dettaglio dei casi d'uso

#### 1. ACCESSO AL SISTEMA

- *Descrizione*  
Questa operazione prevede l'avvio del sistema da parte di un utente generico
- *Prerequisiti*  
Installazione di hardware e software necessari
- *Avvio del Caso d'Uso*  
L'utente generico vuole accedere al sistema perché è interessato a uno dei servizi offerti



Figura 4.1: Accesso al sistema

- *Conclusioni del Caso d'Uso*  
L'utente generico sceglie una delle opzioni disponibili
- *Errori*  
Possono essere applicativi (errori formali) o di malfunzionamento (operazione interrotta)
- *Flusso di esecuzione*
  - (a) L'utente avvia il sistema
  - (b) Accede tramite un click all'area interessata

## 2. AUTENTICAZIONE AL SISTEMA

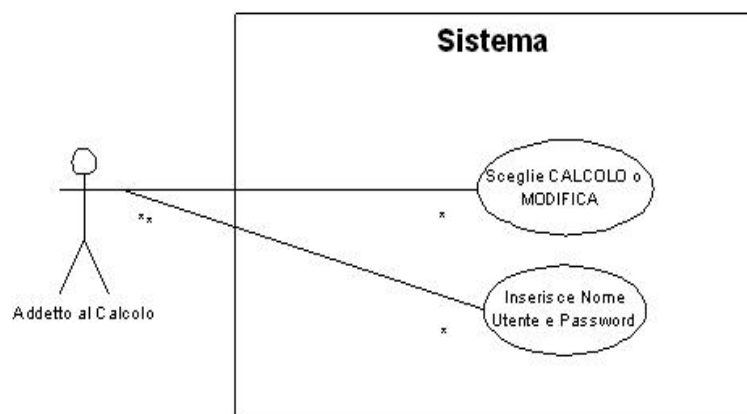


Figura 4.2: Autenticazione al sistema

- *Descrizione*  
Questa operazione prevede l'inserimento di nome utente e pas-

sword da parte del personale autorizzato al CALCOLO e alla MODIFICA degli orari

- *Prerequisiti*  
L'addetto deve essere riuscito ad accedere al sistema
- *Avvio del caso d'Uso*  
Tra le possibili opzioni l'utente ha scelto il CALCOLO o la MODIFICA. Verrà visualizzata una finestra con delle textbox dove inserire nome utente e password.
- *Conclusione del Caso d'Uso*  
L'utente può accedere senza problemi, oppure non essere inserito nella lista senza poter completare l'operazione
- *Errori*  
Possono essere applicativi (errori formali) o di malfunzionamento (operazione interrotta)
- *Postcondizioni*  
Risultati possibili:
  - Si verifica un errore
  - L'autenticazione viene effettuata con successo
  - L'autenticazione non viene effettuata con successo, si rende quindi necessaria da parte dell'utente un'altra scelta
- *Flusso di esecuzione*
  - (a) L'utente effettua una delle scelte che prevede l'autenticazione
  - (b) L'utente digita nome utente e password
  - (c) Il sistema verifica se l'utente fa parte del personale autorizzato:
    - Se fa parte del personale autorizzato allora l'autenticazione è completata e il Caso d'Uso termina con successo
    - Se non fa parte del personale autorizzato il Caso d'Uso termina e l'utente deve effettuare un'altra scelta che non prevede l'autenticazione

### 3. RICERCA DI UN ORARIO

- *Descrizione*  
Questa operazione prevede la ricerca di un orario già compilato da parte di un utente generico
- *Prerequisiti*  
L'utente deve aver completato l'accesso al sistema
- *Avvio del Caso d'Uso*  
L'utente sceglie l'opzione RICERCA e sceglie il semestre cui è interessato. A questo punto il sistema gli offre 3 modalità di ricerca tra cui scegliere:

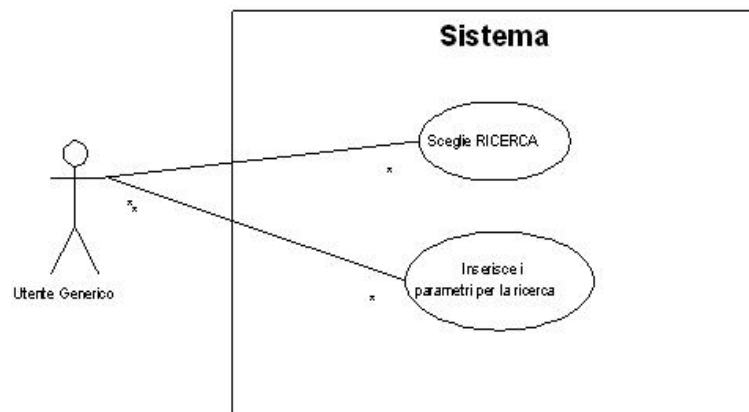


Figura 4.3: Ricerca di un orario

- Per Corso di Laurea
- Per docente
- Per aula
- *Conclusioni del Caso d'Uso*  
Una volta scelta la modalità di ricerca il sistema visualizza una tabella con l'orario corrispondente ai criteri in input
- *Errori*  
Possono essere applicativi (errori formali) o di malfunzionamento (operazione interrotta)
- *Postcondizioni*  
Risultati possibili:
  - La ricerca viene completata con successo
  - L'orario corrispondente ai criteri in input ancora non è disponibile
  - Si verifica un errore
- *Flusso di esecuzione*
  - (a) L'utente sceglie l'opzione RICERCA
  - (b) L'utente sceglie il semestre
  - (c) L'utente sceglie la modalità
  - (d) L'utente cerca in una combobox il Corso di Laurea / docente / aula che gli interessa a seconda della modalità che ha scelto nel punto precedente
  - (e) Il sistema elabora le informazioni. A questo punto può dare in output:
    - Una tabella con gli orari corrispondenti

- Una messagebox con l’avviso che l’orario cercato non esiste ancora; il Caso d’Uso termina senza aver avuto successo
- (f) L’utente può stampare l’orario o effettuare una nuova ricerca; il Caso d’Uso termina con successo

#### 4. CALCOLO DELL’ORARIO

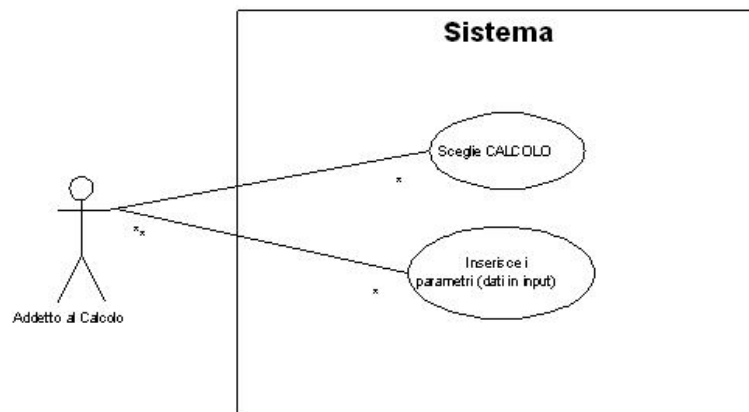


Figura 4.4: Calcolo dell’orario

- *Descrizione*  
Questa operazione prevede il calcolo dell’orario con le caratteristiche immesse dall’operatore
- *Prerequisiti*  
L’utente deve aver avuto accesso e deve essere stato autenticato dal sistema
- *Avvio del Caso d’Uso*  
L’utente sceglie CALCOLO DEGLI ORARI e risulta autorizzato dal sistema. Deve ora scegliere il semestre che gli interessa. A questo punto deve immettere tutte le caratteristiche, i vincoli e i parametri necessari per la realizzazione degli orari. Tali vincoli riguardano tutti i docenti, i Corsi di Laurea ed eventualmente tutte le aule contemporaneamente al fine di fornire al sistema una visione globale della situazione reale
- *Conclusione del Caso d’Uso*  
L’orario viene calcolato, il sistema visualizza un messaggio di operazione effettuata con successo, le informazioni vengono salvate nell’archivio e l’utente può direttamente passare alla RICERCA per verificare se i parametri sono stati verificati

- *Errori*  
Possono essere applicativi (errori formali), di malfunzionamento (operazione interrotta) oppure possono essere causati da conflitti tra i vincoli espressi negli input
- *Postcondizioni*  
Risultati possibili:
  - Il calcolo e il salvataggio delle informazioni vengono completati con successo
  - Si verifica un errore
  - L'orario non può essere calcolato a causa di conflitti negli input
- *Flusso di esecuzione*
  - (a) L'utente sceglie CALCOLO e viene autenticato
  - (b) L'utente sceglie il semestre
  - (c) L'utente immette come input una serie di vincoli e parametri riguardanti Corsi di Laurea e/o docenti espressi mediante una tabella
  - (d) Il sistema elabora le informazioni. A questo punto può dare in output:
    - Una messagebox dove segnala che l'operazione richiesta è andata a buon fine;
    - Una messagebox dove segnala che alcuni parametri in input sono in conflitto e il processo per il calcolo non può concludersi correttamente; il Caso d'Uso termina.
  - (e) L'utente può ora accedere all'area RICERCA per visualizzare l'output dato dal sistema

## 5. MODIFICA DI UN ORARIO ESISTENTE

- *Descrizione*  
Questa operazione prevede la modifica di un orario esistente
- *Prerequisiti*  
L'utente deve aver avuto accesso e deve essere stato autenticato dal sistema
- *Avvio del Caso d'Uso*  
L'utente sceglie MODIFICA e risulta autorizzato dal sistema. A questo punto deve scegliere il semestre e la modalità di ricerca dell'orario da modificare:
  - Per Corso di Laurea
  - Per docente



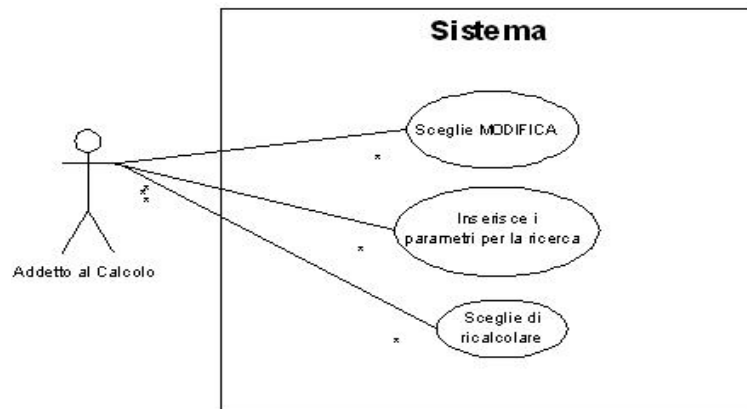


Figura 4.5: Modifica di un orario esistente

- *Conclusioni del Caso d'Uso*  
L'orario corrispondente ai criteri di ricerca viene cancellato, ne viene calcolato uno nuovo che verrà memorizzato
- *Errori*  
Possono essere applicativi (errori formali) o di malfunzionamento (operazione interrotta)
- *Postcondizioni*  
Risultati possibili:
  - Il nuovo orario viene calcolato e salvato nell'archivio con successo
  - Si verifica un errore
  - L'orario con i criteri definiti in precedenza ancora non esiste, in questo caso non verrà effettuata alcuna cancellazione ma solo il calcolo e la memorizzazione
- *Flusso di esecuzione*
  - (a) L'utente sceglie MODIFICA e viene autenticato
  - (b) L'utente sceglie il semestre
  - (c) L'utente sceglie la modalità
  - (d) L'utente cerca in una combobox il Corso di Laurea / docente che gli interessa
  - (e) Il sistema elabora le informazioni e visualizza una tabella con il risultato della ricerca
  - (f) Se l'utente conferma la cancellazione viene avviato un nuovo calcolo altrimenti il Caso d'Uso termina
  - (g) Il sistema elabora il nuovo orario che viene visualizzato e memorizzato

- (h) L'utente può stampare l'orario visualizzato oppure modificarne un altro

## 6. ASSEGNAZIONE MANUALE DI UN ORARIO

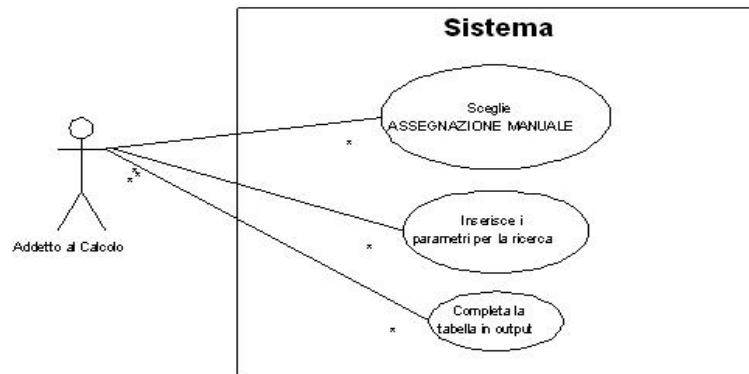


Figura 4.6: Assegnazione manuale

- *Descrizione*  
Questa operazione prevede l'inserimento manuale di un orario da parte di un addetto
- *Prerequisiti*  
L'utente deve aver completato l'accesso al sistema e deve essere stato autenticato
- *Avvio del Caso d'Uso*  
L'utente sceglie l'opzione ASSEGNAZIONE MANUALE e risulta autorizzato dal sistema. A questo punto deve scegliere il semestre e la modalità di ricerca dell'orario da inserire:
  - Per Corso di Laurea
  - Per docente
  - Per aula
- *Conclusione del Caso d'Uso*  
L'orario corrispondente ai criteri di ricerca viene visualizzato in una tabella direttamente modificabile dall'operatore manualmente (se l'orario non è ancora stato compilato tale tabella è vuota). L'orario inserito verrà memorizzato
- *Errori*  
Possono essere applicativi (errori formali) o di malfunzionamento (operazione interrotta)
- *Postcondizioni*  
Risultati possibili:

- Il nuovo orario viene inserito nell'archivio con successo
- Si verifica un errore
- *Flusso di esecuzione*
  - (a) L'utente sceglie ASSEGNAZIONE MANUALE e viene autenticato dal sistema
  - (b) L'utente sceglie il semestre
  - (c) L'utente sceglie la modalità
  - (d) L'utente cerca in una combobox il Corso di Laurea / docente / aula che gli interessa a seconda della modalità che ha scelto nel punto precedente
  - (e) Il sistema elabora le informazioni e visualizza in output una tabella (eventualmente vuota nel caso in cui l'orario corrispondente non esista)
  - (f) L'utente effettua le assegnazioni che desidera
  - (g) Il sistema verifica che le informazioni inserite siano compatibili con gli orari già memorizzati. Se non lo sono dà un messaggio di errore, altrimenti memorizza le assegnazioni appena inserite
  - (h) L'utente può stampare l'orario oppure effettuare una nuova assegnazione; il Caso d'Uso termina con successo

## Capitolo 5

# Pianificazione di rilascio

### 5.1 Contenuto

Questo capitolo descrive la pianificazione per la realizzazione dell'Interfaccia Utente per il sistema informativo di Timetabling della Facoltà di Ingegneria dell'Università degli Studi di Trieste e si basa sulle informazioni raccolte nei capitoli precedenti.

Verranno qui indicati obiettivi, vincoli, gestione dei rischi, risorse necessarie e dunque una stima dei costi in termini di tempo. Gli obiettivi ed i vincoli generali sono indicati nel capitolo 2. Gli obiettivi correlati al livello di qualità attesa sono indicati nel capitolo 3. I vincoli relativi ai tempi di realizzazione sono indicati nel capitolo 3. Infine i vincoli architetturali e tecnologici prescelti sono indicati nel capitolo 4.

### 5.2 Organizzazione del progetto

Per il progetto dell'Interfaccia Utente è sufficiente uno sviluppatore responsabile di progettazione di interfaccia e dello sviluppo del livello di User Services. Dovrà eventualmente occuparsi se necessario della documentazione per l'utente. Per quanto riguarda la gestione dei rischi egli dovrà effettuare dei test mirati oltre che test di unità, delle componenti e dell'eventuale riutilizzo. Dovrà rendere la grafica gradevole e usabile agli utenti testandola con un Prototipo.

### 5.3 Risorse hardware e software

L'hardware necessario comprende:

- 1 GB di RAM;
- RAID SCSI con un Disk Array RAID5;

- Sistema multiprocessore simmetrico.

L'hardware così configurato è sovradimensionato per permettere lo sviluppo delle esigenze future. Il software necessario è indicato in modo dettagliato nel capitolo 4.

## 5.4 Attività

L'ordine con cui i requisiti verranno implementati dipende dalla loro importanza. I Casi d'Uso sono messi in quest'ordine di priorità:

1. CALCOLO DELL'ORARIO;
2. ASSEGNAZIONE MANUALE DI UN ORARIO;
3. MODIFICA DI UN ORARIO ESISTENTE;
4. RICERCA DI UN PARTICOLARE ORARIO;
5. ACCESSO AL SISTEMA;
6. AUTENTICAZIONE DELL'UTENTE.

## 5.5 Analisi dei rischi

Oltre ai rischi legati agli obiettivi di qualità discussi nel PIANO DI QUALITÀ', ne sono stati individuati degli altri:

- Rischio: sottostima delle dimensioni del progetto  
Descrizione: eventuali requisiti non considerati emergono in fase di sviluppo  
Probabilità: moderata  
Conseguenze: serie  
Piano per abbassare la probabilità di rischio: implementare prima i requisiti con maggiore incertezza sulla loro definizione o comprensione e fare partecipare l'utente
- Rischio: cambiamento dei requisiti  
Descrizione: l'utente cambia i requisiti durante la fase di realizzazione  
Probabilità: moderata  
Conseguenze: serie  
Piano per abbassare la probabilità di rischio: supportare l'utente nella documentazione con opportuno livello di dettaglio; stimare il tempo richiesto dal nuovo requisito

## 5.6 Stima dei costi

Fattori che concorrono alla definizione dei costi in termini di tempo:

- Volatilità dei requisiti: il dominio applicativo è stabile dunque la volatilità è bassa.
- Termini contrattuali: il vincolo temporale non è stringente; più stringenti sono i parametri di qualità esplicitati nel capitolo 2 e nel capitolo 3 il cui rispetto è necessario.
- Incertezze: riguardano la stima iniziale dei costi in quanto problemi inattesi potrebbero emergere solo in fase di realizzazione.

### 5.6.1 Stima dei costi

1. Preparazione hardware e software: hardware e software precedentemente indicati vanno installati e configurati. Su richiesta va fornita assistenza. COSTO: 3 giornate uomo.
2. Implementazione classi: prevede il disegno, la codifica, i test, il controllo di qualità ed eventualmente la documentazione. COSTO: 10 giornate uomo.
3. Implementazione dei Casi d'Uso: prevede la specifica dettagliata dei Casi d'Uso, il disegno, la codifica, i test, il controllo di qualità ed eventualmente la documentazione. COSTO: 10 giornate uomo.

La stima finale è di 23 giornate uomo.

# Bibliografia

- [1] [http://www.tecnoteca.it/tesi/certificazione/qualita\\_software/standart\\_normative](http://www.tecnoteca.it/tesi/certificazione/qualita_software/standart_normative)
- [2] [http://www.artea.it/newentry/iso9001\\_software.htm](http://www.artea.it/newentry/iso9001_software.htm)
- [3] Dispense dei corsi di Sistemi Informativi delle Università di Trieste, Udine, Pisa e dispensa del corso di Impianti Speciali per la documentazione sulle normative ISO.

# Elenco delle figure

2.1	Attori . . . . .	6
2.2	Caratteristiche e sottocaratteristiche . . . . .	8
2.3	ISO 9126 . . . . .	8
3.1	Modello a cascata . . . . .	17
4.1	Accesso al sistema . . . . .	20
4.2	Autenticazione al sistema . . . . .	20
4.3	Ricerca di un orario . . . . .	22
4.4	Calcolo dell'orario . . . . .	23
4.5	Modifica di un orario esistente . . . . .	25
4.6	Assegnazione manuale . . . . .	26