



Specifica dei test

Versione 1.0

Stato del documento:

*Formale ed
esterno*

Sommario :

Il documento intende fornire dettagli sullo svolgimento dei test sul software.

Redazione

Nominativo	Ruolo	Data
Roberto Pordon	Verificatore	20 febbraio 2007
Lucia Meneghello	Verificatore	20 febbraio 2007

Lista di Distribuzione

Nominativo	Ruolo
Tullio Vardanega	Committente
Renato Conte	Committente
Lucia Meneghello	Verificatore
Margherita Collicelli	Progettista
Eric Miotto	Progettista/Amministratore
Stefano Gazzola	Progettista
Roberto Pordon	Verificatore
Lorenzo Daniele	Responsabile
Alberto Meneghello	Amministratore

Approvato da:

Versione	Nominativo	Data
1.0	Lorenzo Daniele	23 febbraio 2007

Registro delle Modifiche:

Versione	Autore	Data
1.0	Lorenzo Daniele	23 febbraio 2007
Approvazione per la Revisione di Progetto Definitivo.		
0.3	Eric Miotto	23 febbraio 2007
Verifica del documento.		
0.2	Roberto Pordon	21 febbraio 2007
Inseriti dettagli per test di unità		
0.1	Roberto Pordon	20 febbraio 2007
	Lucia Meneghello	
Prima bozza del documento.		

Indice

1	Introduzione.....	5
1.1	Scopo documento.....	5
1.2	Riferimenti.....	5
2	Test di unità.....	6
2.1	Strumenti.....	6
2.2	Test funzionale.....	6
2.2.1	<i>Verifica informale del programmatore.....</i>	<i>6</i>
2.2.2	<i>Verifica formale.....</i>	<i>6</i>
2.3	Test strutturale.....	6
3	Test di integrazione.....	6
4	Test di sistema.....	7
4.1	Compatibilità.....	7
4.1.1	<i>Parametri di verifica.....</i>	<i>7</i>

1 Introduzione

1.1 Scopo documento

Il documento intende illustrare gli strumenti e le modalità per la conduzione dei test sul prodotto software.

1.2 Riferimenti

- [IS] Lucidi del corso di Ingegneria del software, proff. Tullio Vardanega e Renato Conte, Università di Padova, a.a. 2006/2007
<http://www.math.unipd.it/~tullio/IS-1/2006/calendario.html>
<http://www.math.unipd.it/~conte>

2 Test di unità

2.1 Strumenti

Per i test di funzionali si utilizzerà JUnit (junit.org), un framework opensource integrato anche nell'ambiente NetBeans, che garantisce la ripetibilità delle prove.

Per i test strutturali, il software Cobertura (cobertura.sourceforge.net) fornirà dati sul livello di branch e statement coverage raggiunti dai test.

2.2 Test funzionale

2.2.1 Verifica informale del programmatore

Il programmatore può effettuare dei test sulle unità che sviluppa mano a mano che procede nel lavoro. Queste verifiche sono ammesse soltanto se costituiscono un reale ausilio allo sviluppo e non ne causano un eccessivo rallentamento. In ogni caso queste verifiche non hanno alcun valore formale, di conseguenza non vanno documentate.

2.2.2 Verifica formale

Al termine dello sviluppo di un modulo i verificatori procedono al test dello stesso. Il test con JUnit deve accertare il corretto comportamento di tutte le funzionalità del modulo. I risultati delle verifiche formali devono essere verbalizzati.

2.3 Test strutturale

Per il test strutturale saranno definiti dei casi di prova che coprano la maggior parte possibile del codice. I dati percentuali di copertura del codice saranno forniti dal software Cobertura. Tutti i casi di test dovranno essere documentati e dovrà essere fornita la percentuale del codice coperto.

3 Test di integrazione

I test di integrazione potranno essere attuati solo successivamente al corretto completamento dei test di unità.

L'integrazione darà la precedenza alle parti con meno dipendenze, per poi passare a quelle più complesse, che sono vincolate a quelle precedenti.

4 Test di sistema

4.1 Compatibilità

Per assicurare una verifica oggettiva, l'esecuzione avverrà su più macchine aventi architetture hardware e software diverse in massima parte.

4.1.1 Parametri di verifica

Verranno considerati di rilevanza i seguenti parametri specifici

- utilizzo della memoria principale (RAM);
- efficienza, risorse macchina impiegate e tempistica;
- layout grafico;
- conseguenze di input (azioni, dati...) imprevisti.