



# Specifica dei test per C04/PQS

Versione 2.0

Stato del documento:

*Formale ed*

*esterno*

## Sommario :

Il documento intende fornire dettagli sullo svolgimento dei test sul software relativamente al capitolato C04: "Sistema *software* per l'informatizzazione della gestione di qualità a norma ISO 9000:2000 nelle scuole".

## 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	Responsabile
Eric Miotto	Amministratore/Programmatore
Stefano Gazzola	Programmatore
Roberto Pordon	Verificatore
Lorenzo Daniele	Programmatore
Alberto Meneghello	Programmatore

## Approvato da:

Versione	Nominativo	Data
2.0	Margherita Collicelli	16 marzo 2007

## **Registro delle Modifiche:**

<b>Versione</b>	<b>Autore</b>	<b>Data</b>
2.0	Margherita Collicelli	16 marzo 2007
Approvazione per la Revisione di qualifica.		
1.1	Roberto Pordon	13 marzo 2007
Abbozzata strategia test per le componenti.		
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

<b>1</b>	<b>Introduzione.....</b>	<b>5</b>
1.1	Scopo documento.....	5
1.2	Riferimenti.....	5
<b>2</b>	<b>Test di unità.....</b>	<b>6</b>
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>
2.2.3	<i>Test di WSDidattica.....</i>	<i>6</i>
2.2.4	<i>Test di APPDidattica.....</i>	<i>7</i>
2.3	Test strutturale.....	7
2.4	Test di regressione.....	7
<b>3</b>	<b>Test di integrazione.....</b>	<b>7</b>
<b>4</b>	<b>Test di sistema.....</b>	<b>7</b>
4.1	Compatibilità.....	7
4.1.1	<i>Parametri di verifica.....</i>	<i>8</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>
- risultati dei test disponibili nelle cartelle di ogni progetto

## 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.

I risultati dei test saranno stampati in formato pdf attraverso l'utilizzo di JunitPdfReport; dunque sarà attraverso la lettura di tali documenti che si procederà al controllo degli esiti ottenuti.

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.

#### 2.2.3 Test di WSDidattica

Per il test di WSDidattica abbiamo creato:

- uno stub di WEB SERVICE-API (chiamato API\_stub) che fornisca l'interfaccia a WSDidattica. API\_stub dispone di un Web Service usato dal driver (descritto successivamente) per imporre lo stato necessario per i test;
- un driver (WSDidattica\_driver) che consumi i web service di WSDidattica e che contiene i test da effettuare.

Una prima verifica (già effettuata con successo) consiste nel richiedere i WSDidattica tramite il driver e verificare che API\_stub ricevesse correttamente i comandi tramite stampe su console.

Una verifica successiva consentirà di mantenere la persistenza dei dati in API\_stub e

quindi di programmare dei test automatizzati che verifichino tutti i possibili comportamenti di WSDidattica.

#### 2.2.4 Test di APPDidattica

Il test dei JavaBean sarà supportato da uno stub di WSDidattica che avrà un'interfaccia per pilotare il suo stato (e quindi per automatizzare i test).

Stiamo valutando il modo migliore di effettuare i test sulla GUI.

### 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.

### 2.4 Test di regressione

Nel caso in cui, a seguito di test, si debba effettuare delle modifiche alle entità testate, sarà necessario ritestare sia le classi modificate che tutte quelle ad esse correlate.

Entrando più nel dettaglio, nel caso si effettuino cambiamenti trasparenti, sarà possibile replicare i test esclusivamente nelle entità modificate.

Inoltre si farà attenzione a che gli aggiornamenti, nel risolvere delle anomalie, non ne introducano di nuove (“Principio di non regressione”).

## 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.

L'ordine d'integrazione sarà il seguente:

- API\_Stub (o se possibile la API di Swell Systems);
- WSDidattica;
- APPDidattica.

## 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.