



REGIONE SICILIANA
PRESIDENZA



PRESIDENZA
DEL CONSIGLIO DEI MINISTRI
DIPARTIMENTO DELLA PROTEZIONE CIVILE



Commissario Delegato per l'Emergenza Bonifiche
e la Tutela delle Acque in Sicilia

PIANO DI TUTELA DELLE ACQUE DELLA SICILIA


(di cui all'art. 121 del Decreto Legislativo 3 aprile 2006, n° 152)



Sistema WEB-GIS per la fruizione delle informazioni del
Piano di Tutela delle Acque in Sicilia

SIPTAS

Piano di Test

COORDINAMENTO GENERALE A CURA DI	DOCUMENTO	REDATTO DA	DATA	APPROVATO
 SOCIETÀ GESTIONE IMPIANTI IDRICI Unità Operativa di Palermo	ALL.21.2	SOGESID S.p.A. Consulenza informatica: INTEA S.r.l.	DICEMBRE 2007	

INDICE

1	Introduzione	1
2	Processo dei Test	1
3	Test Funzionali	1
3.1	Ambiente operativo	1
3.1.1	Ambiente Hardware	2
3.1.2	Ambiente software	2
3.2	Progettazione dei test	2
3.2.1	Matrice Requisiti di Business	2
3.2.2	Requisiti e Script di test.....	3
4	Test non Funzionali.....	3
4.1	Test Prestazionali	4
4.2	Test di Sicurezza	4
4.3	Test di non Regressione	5
4.4	Test di Sistema – Integrazione	5
5	Documenti di riferimento	5

1 Introduzione

Il presente documento descrive la metodologia adottata per l'esecuzione dei test del software prodotto per i sistemi WEB-GIS per il Catasto degli Scarichi (CATSCA), per il Piano di Tutela delle Acque della Sicilia (SIPTAS) e per il Piano Regolatore Generale degli Acquedotti della Sicilia (SIPRA).

Il processo di testing ha costituito una fase fondamentale per l'ottenimento della qualità del software prodotto. Essa ha avuto anche l'obiettivo di rilevare ed intervenire su eventuali problematiche prima del collaudo dei singoli prodotti.

2 Processo dei Test

Il processo di test ha previsto una serie di attività dirette a sollecitare le varie aree delle applicazioni in modo da verificare che i comportamenti di quest'ultime risultassero conformi alle aspettative.

Le attività per il processo di test sono state:

- **Progettazione.** In questa fase si è proceduto all'identificazione delle funzionalità, dei requisiti e delle condizioni per giudicare l'oggetto di test adeguato alle aspettative.
- **Definizione e popolamento degli ambienti.** Fase in cui è stato definito l'ambiente operativo e la base dati in cui sono stati eseguiti i test.
- **Esecuzione.** Fase in cui sono state effettuate le prove, atte a verificare la conformità dei risultati a quanto definito nei documenti di progettazione.
- **Collezione dei risultati.** Fase in cui è avvenuta la registrazione dell'esito delle esecuzioni e dello stato delle anomalie.
- **Analisi delle anomalie.** Fase in cui le anomalie sono state classificate secondo i livelli di gravità, per procedere successivamente alla loro correzione.

3 Test Funzionali

In questa fase è stato condotto il processo di testing per valutare i requisiti funzionali dei Sistemi. I test non funzionali, di sistema e di non regressione hanno seguito lo stesso iter procedurale.

3.1 Ambiente operativo

Di seguito è descritto l'ambiente operativo (hardware e software) in cui sono stati effettuati i test. Lo stesso ambiente può essere riprodotto per la fase di collaudo.

3.1.1 Ambiente Hardware

Server

- Processore: Pentium IV – 2.4 GHz
- Memoria centrale: 2 GB
- Hard Disk: Ultra SCSI – RAID 5
- Scheda di rete: Ethernet 100TX

Client

- Processore: Pentium IV – 2.4 GHz
- Memoria centrale: 1 GB
- Hard Disk: SATA
- Scheda video: GeForce 4200
- Scheda di rete: Ethernet 100TX
- Monitor: LCD 17”

3.1.2 Ambiente software

Server

- Sistema operativo: Microsoft Windows 2003 Server
- Database: Microsoft SQL2000
- Internet Map Server: ESRI ArcIMS 9.1

Client

- Sistema operativo: Microsoft Windows XP SP2
- Browser HTML: Internet Explorer 6.1, Netscape Navigator 7.2, Mozilla Firefox 2.0

3.2 Progettazione dei test

Nella fase di “progettazione” sono state identificate le funzionalità, i requisiti e le condizioni da esercitare per valutare i Sistemi oggetto di test.

3.2.1 Matrice Requisiti di Business

Sono state individuate le funzionalità da sottoporre a test. Per ciascuna di essa è stato valutato se si trattava di funzionalità di tipo elementare. In caso contrario, si è provveduto a decomporre in funzionalità elementari.

3.2.2 Requisiti e Script di test

Per ciascuna funzionalità elementare è stato identificato un requisito di test e quindi sviluppato uno script di test. Uno script test consiste in una sequenza di azioni da svolgere, nell'identificazione dei dati da fornire in input e nell'individuazione dei risultati attesi.

Le anomalie sono state classificate secondo quattro livelli di gravità. La tabella che segue ne illustra i criteri di valutazione che sono stati previsti.

Livelli Gravità Anomalie	
Livello Gravità	Anomalie
Alta	L'anomalia risulta bloccante per funzioni dell'applicazione, oppure provoca contaminazioni sulle informazioni archiviate, oppure provoca rallentamenti evidenti per servizi vitali o critici
Media	L'anomalia provoca rallentamenti evidenti per servizi non vitali né critici, oppure presenta gravi carenze di usabilità, oppure provoca il blocco della funzione se vengono immessi dati o compiute operazioni scorrette
Bassa	L'anomalia provoca lievi problemi di usabilità, oppure provoca comportamenti non previsti dell'applicazione se vengono immessi dati o compiute operazioni scorrette ma non ne provoca il blocco né gravi difficoltà funzionali
Warning	L'anomalia non provoca problemi, ma vengono evidenziati ambiti di miglioramento

4 Test non Funzionali

I test non funzionali sono stati finalizzati a verificare funzionalità non applicative e relative ad aspetti e/o requisiti di sicurezza, di prestazione, di usabilità.

Sono state previste le categorie di seguito elencate:

- Test delle prestazioni, atti a verificare che il sistema soddisfi i requisiti di prestazioni richiesti.
- Test di sicurezza, atti a verificare che il sistema realizzato garantisca i requisiti di sicurezza richiesti.
- Test di compatibilità, atti ad assicurare il corretto funzionamento del sistema realizzato su differenti piattaforme hardware, di sistema operativo, di browser, ecc.

- Test di usabilità, atti a verificare che il sistema realizzato soddisfi tutti i requisiti di usabilità richiesti, come la facilità di comprensione dell'interfaccia o l'adeguatezza dei contenuti delle schermate. Questa tipologia di test ha permesso di identificare criticità e colli di bottiglia dell'interfaccia (corretti in fase di design), e di prevedere il comportamento dell'utente per scoprire eventuali difficoltà di cui tener conto in fase di progettazione.
- Test di stress o di carico del sistema, atti a garantire la capacità del nuovo sistema di funzionare correttamente anche quando si raggiungono i suoi limiti in termini di quantità di dati trattati o di utenti connessi.

4.1 Test Prestazionali

Lo scopo del “performance testing” non è stato quello di rilevare errori nell'applicazione, ma di eliminare potenziali colli di bottiglia e stabilire una “baseline” prestazionale per i futuri test di regressione.

Condurre test di questo tipo è equivalente all'esecuzione di un processo controllato di misurazione e analisi. Un insieme chiaramente definito di “cosa ci si aspetta” è essenziale per un test significativo.

Per un'applicazione WEB è necessario conoscere almeno:

- Carico atteso in termini di utenti concorrenti o connessioni HTTP
- Tempi di risposta considerati accettabili

Sono stati definiti i requisiti prestazionali per ogni funzione elementare, in particolare le misure sono state effettuate nei termini di:

- Carico massimo previsto
- Tempo di risposta massimo accettabile

I test sono stati effettuati creando situazioni operative e simulando accessi tali da stimolare il sistema in scenari operativi reali.

4.2 Test di Sicurezza

Non esistono metodi standard per collaudare la sicurezza di un'applicazione. Inoltre, sono disponibili pochi strumenti per collaudare in modo approfondito tutti gli aspetti della sicurezza. La tecnica principale dei test della sicurezza consiste nel tentare di violare i controlli di protezione predefiniti. Questa tecnica assicura che i meccanismi di protezione impediscano l'accesso indesiderato al sistema.

Esistono due tipi di attacchi: interni ed esterni. Per attacco interno si intende un attacco messo in atto da un dipendente ed alcune delle minacce più pericolose per l'infrastruttura di un'organizzazione sono costituite dagli attacchi provenienti dalla rete interna. Gli

attacchi interni che potrebbero causare potenzialmente i danni maggiori sono quelli risultanti dalle attività di persone a cui sono assegnati i privilegi più alti, ad esempio gli amministratori di rete. Un attacco esterno, invece, proviene dall'esterno dell'organizzazione.

Possibili categorie di attacco sono:

- Tentativi di compromissione delle credenziali
- Sfruttamento dei punti di vulnerabilità
- Accesso a computer non autorizzato

I test eseguiti hanno simulato le condizioni che seguono:

- Attacco con dizionario
- Invio di un elevato numero di richieste continue al sistema
- Creazione volontaria di errori di sistema per tentare di accedervi durante il ripristino delle normali condizioni operative

4.3 Test di non Regressione

Con tale tipo di test si è inteso accertamento il regolare funzionamento dei programmi anche dopo aver apportato aggiornamenti al software.

Anche per questo test la metodologia adottata è stata analoga a quella usata per i test funzionali.

4.4 Test di Sistema – Integrazione

Al termine delle singole sessioni di test, è stato eseguito un “Test di Sistema”, come verifica conclusiva del corretto funzionamento del sistema in rilascio, considerando integrate tutte le sue diverse componenti. Può accadere, infatti, che anche se i singoli moduli sono corretti, il sistema ottenuto integrandoli potrebbe non esserlo.

Nel test di sistema è stato incluso, oltre all'esecuzione del piano completo dei test funzionali, anche l'esecuzione dello “stress testing” e del “performance test”.

5 Documenti di riferimento

L'architettura del software ed il modello dei dati dei sistemi WEB-GIS CATSCA, SIPTAS e SIPRA sono descritti rispettivamente nei documenti di seguito elencati:

- a) CATSCA – Progettazione del software - Novembre 2006
- b) CATSCA – Modello dati - Novembre 2006

- c) SIPTAS – Progettazione del software – Aprile 2006
- d) SIPTAS – Modello dati – Aprile 2006
- e) SIPRA – Progettazione del software – Aprile 2006
- f) SIPRA – Modello dati – Aprile 2006