



Università degli Studi di Salerno

Laurea in Informatica



Progetto Ingegneria del Software

GESTIONE SERVIZI WEB



Gruppo di lavoro:

Giardino Pasquale

Prota Antonio

Nocerino Dario

D'Ambrosio Giuseppe

Farisano Gino

Anno Accademico:2011/2012



Test Planning

Indice Test Planning

- 1. Introduzione**
- 2. Documenti correlati**
- 3. Relazioni con il documento di analisi dei requisiti (RAD)**
- 4. Relazioni con il documento di System Design Description (SDD)**
- 5. Overview del sistema**
 - 5.1 Sottosistema amministratore
 - 5.2 Sottosistema utente
- 6. Funzionalità testate e non testate**
 - 6.1 Funzionalità testate
 - 6.2 Funzionalità non testate
- 7. Criteri Pass/Fail**
- 8. Approccio**
 - 8.1 Black Box Testing
 - 8.2 White Box Testing
- 9. Sospensione e Ripristino**
 - 9.1 Criteri di Sospensione
 - 9.2 Criteri di Ripristino
- 10. Materiale per il Testing (Requisiti Hardware/Software)**
- 11. Casi di Test (Test Cases)**
 - 11.1 White Box Test Case
 - 11.2 Black Box Test Case
- 12. Pianificazione del Testing**
- 13. Glossario**



1.Introduzione

Il testing è una tecnica che ha lo scopo di rompere il sistema, cioè quello di identificare errori con l'obiettivo di evitare che si presentino durante l'utilizzo del software da parte dell'utente, e non quello di garantire che il codice sia corretto. In altre parole si cerca di creare quanti più fallimenti o stati erronei in modo pianificato in modo da consentire agli sviluppatori di correggerli prima che il prodotto sia consegnato al cliente. Lo scopo di questo documento è quello quindi di descrivere e pianificare l'attività di testing su determinate funzionalità del sito GSW (Gestione Servizi Web) con l'obiettivo di individuare errori mediante i casi di test. Il testing andrà a testare soltanto una singola componente del sistema, quella che si occupa della gestione degli utenti e delle applicazioni da parte dell'amministratore.

2.Documenti correlati

Questo documento è correlato a molti artefatti nella progettazione del sito. Innanzitutto non si è potuta avviare all'attività di testing prima che gran parte dell'implementazione fosse stata completata. Poi vi sono strette correlazioni tra i documenti precedentemente realizzati e quest'ultimo specificate qui di seguito.

3.Relazioni con il documento di analisi dei requisiti (RAD)

Questo documento è correlato a molti artefatti nella progettazione del sito. Innanzitutto non si è potuta avviare all'attività di testing prima che gran parte dell'implementazione fosse stata completata. Poi vi sono strette correlazioni tra i documenti precedentemente realizzati e quest'ultimo specificate qui di seguito.



4.Relazioni con il System Design Document (SDD)

Il testing di unità e il System Design sono correlate, perchè le unità candidate per il test sono scelte dalla decomposizione in sottoinsiemi. Inoltre vi è una relazione tra i design goal dell'SDD e il performance testing.

5.Overview del Sistema

Come specificato nel SDD il sistema verrà diviso in sottosistema amministratore e sottosistema utente. Ognuno di essi costituito da diverse componenti in base all'area considerata. Di seguito sono riportate le diverse funzioni sia del sottosistema amministratore sia del sottosistema utente.

5.1Sottosistema amministratore

login()

logout()

aggiungi_applicazione()

visualizza_applicazione()

visualizza_cliente()

visualizza_azienda()

elimina_account()

5.2 Sottosistema utente

login()

logout()



crea_account()

modifica_account()

effettua_acquisto()

6.Funzionalità testate e non testate

6.1 Funzionalità testate

Le funzionalità che verranno testate tra quelle implementate sono:

aggiungi_applicazione()

6.2 Funzionalità non testate

Le funzionalità che non verranno testate tra quelle implementate sono:

login()

logout()

aggiungi_cliente()

aggiungi_azienza()

visualizza_cliente()

visualizza_azienda()

visualizza_applicazione()

modifica_account()

elimina_account()

aggiungi_licenza()

7.Criteri Pass/Fail

Gli input per i vari casi di test saranno suddivisi in classi di equivalenza con lo scopo di favorire il processo di Blackboxtesting ed evitare un eccessivo costo per questa



fase. I criteri di successo saranno raggiunti nel momento in cui l'output verificato coinciderà con quello atteso. Quest'ultimo sarà definito da un oracolo umano (sviluppatore), che conosce preventivamente quale sarà il risultato dell'esecuzione del test case. Viceversa, un test case fallirà nel momento in cui il risultato atteso e quello ottenuto non coincideranno.

Motivazioni: si è scelto di adoperare un oracolo umano dopo aver preso in considerazione la taglia del sottosistema implementato da testare. Essendo quest'ultimo limitato, è stato deciso che un oracolo umano potesse adempiere facilmente a questo scopo, senza la necessità di costruirne uno meccanico.

8.Approccio

Dopo una meticolosa valutazione da parte del Team Manager, si è pensato di applicare come approccio il Test di Unità al fine di valutare il comportamento di ogni singola componente. Il test di unità infatti verrà effettuato su componenti isolate con lo scopo di valutarne la correttezza in termini di input/output e di struttura. In particolare si adopereranno due tecniche, la Black Box Testing e la White Box Testing.

8.1 Black Box Testing

Verrà effettuato con particolare attenzione su ogni componente isolata, specificando opportunamente le classi di equivalenza ed utilizzando il WeakEquivalenceClass Testing. Ogni errore riscontrato durante questa fase sarà tempestivamente corretto.

8.2 White Box Testing

Il test sulla struttura interna delle componenti sarà trattato solo in parte, visto che il codice non presenta particolari path complessi. Per questa ragione, il WhiteBoxTesting si concentrerà solo sul Branch Test relativo all'Inserimento di una nuova applicazione.

9.Criteri di sospensione e ripristino



9.1 Criteri di sospensione

Dopo un'attenta valutazione del tempo stimato rimasto e della taglia del sottosistema da testare, il Team Manager ha deciso che il Test potrà essere svolto in maniera corposa e abbastanza completa, venendo sospeso solo nel caso in cui venga sfiorato il monte di ore previsto o dopo aver testato il sottosistema nella sua interezza.

9.2 Criteri di ripristino

Nel momento in cui si individuino errori durante la fase di test, si apporteranno le dovute

condizioni per poi ripristinare il test per verificare nuovamente la componente.

10. Materiale per il Testing

L'hardware necessario per la fase di testing è il seguente:

un computer portatile per ogni membro del Test Team

Il software necessario per la fase di testing è il seguente:

DBMS MySql ;

Browser: Google Chrome e MozillaFirefox ;

Server Apache;

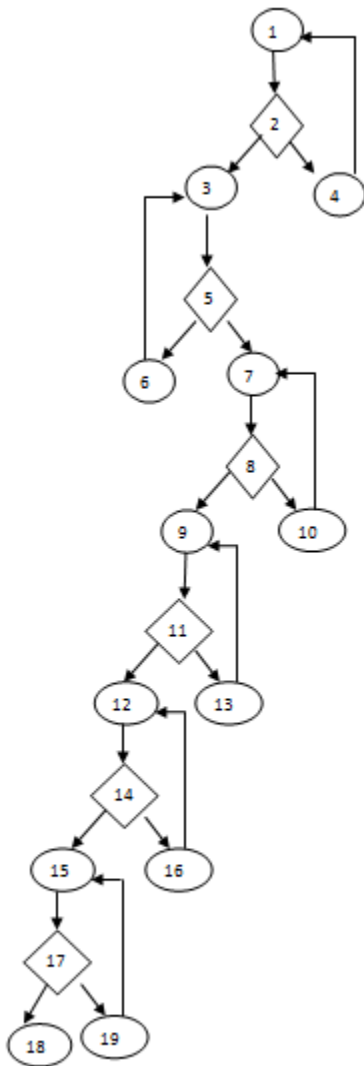


11.Casi di test (Test Cases)

11.1 WhiteBox Test Case

Inserimento applicazione

Flow graph:





Legenda:

1. Inserimento id applicazione
2. Branch correttezza id (if 2 is true go to 3 else go to 4)
3. Inserimento Nome applicazione
4. Branch correttezza Nome applicazione (if 5 is true go to 6 else go to 7)
5. Inserimento Tipo applicazione
6. Branch correttezzaTipo applicazione (if 8 is true go to 9 else go to 10)
7. Inserimento Autori applicazione
8. Branch correttezza Autori applicazione (if 11 is true go to 12 else go to 13)
9. Inserimento prezzo applicazione
- 10.Branch correttezza prezzo applicazione (if 14 is true go to 15 else go to 16)
- 11.Inserimento link applicazione
- 12.Branch correttezza link applicazione (if 17 is true go to 18 else go to 19)

Classi

Input		Risultato Branch ID
Classi valide	CL1	True
Classi non valide	CL2	False

Input		Risultato Branch Nome
Classi valide	CL3	True
Classi non valide	CL4	False

Input		Risultato Tipo
Classi valide	CL5	True
Classi non valide	CL6	False

Input		Risultato Branch Autori
-------	--	-------------------------



Classi valide	CL7	True
Classi non valide	CL8	False

Input		Risultato Branch Prezzo
Classi valide	CL9	True
Classi non valide	CL10	False

Input		Risultato Branch Link
Classi valide	CL11	True
Classi non valide	CL12	False

Identificazione Test Case

Inserimento Applicazione_1

Input	Classe	Membro
Risultato Branch Id	CL1	True
Risultato Branch Nome	CL3	True
Risultato Branch Tipo	CL6	False
Risultato Branch autori	CL7	True
Risultato Branch prezzo	CL10	False
Risultato Branch link	CL11	True

Inserimento Applicazione_2

Input	Classe	Membro
Risultato Branch Id	CL2	False
Risultato Branch Nome	CL3	True
Risultato Branch Tipo	CL5	True



Risultato Branch autori	CL7	True
Risultato Branch prezzo	CL10	False
Risultato Branch link	CL12	False

Inserimento Applicazione_3

Input	Classe	Membro
Risultato Branch Id	CL1	True
Risultato Branch Nome	CL3	True
Risultato Branch Tipo	CL6	False
Risultato Branch autori	CL7	True
Risultato Branch prezzo	CL9	True
Risultato Branch link	CL11	True

11.2 BlackBox Test Case

Inserimento Applicazione

Classi

Gli input sono: id, NomeApplicazione, tipo, autori, prezzo, link;

Input		ID
Classi valide	CL13	Stringa numerica 01
Classi valide	CL14	Stringa numerica 02
Classi non valide	CL15	Stringa nulla
Classi non valide	CL16	Qualsiasi altra stringa alfanumerica

Input		Nome
Classi valide	CL17	Stringa alfanumerica di lunghezza maggiore di 4



		e minore di 100 caratteri
Classi non valide	CL18	Stringa alfanumerica minore o uguale di 4 caratteri.
Classi non valide	CL19	Stringa alfanumerica maggiore di 100 caratteri.
Classi non valide	CL20	Stringa nulla.

Input		Tipo
Classi valide	CL21	Stringa alfanumerica di lunghezza maggiore di 4 e minore di 100 caratteri
Classi non valide	CL22	Stringa alfanumerica minore o uguale di 4 caratteri.
Classi non valide	CL23	Stringa alfanumerica maggiore di 100 caratteri.
Classi non valide	CL24	Stringa nulla.

Input		Autori
Classi valide	CL25	Stringa alfanumerica di lunghezza maggiore di 4 e minore di 100 caratteri
Classi non valide	CL26	Stringa alfanumerica minore o uguale di 4 caratteri.
Classi non valide	CL27	Stringa alfanumerica maggiore di 100 caratteri.
Classi non valide	CL28	Stringa nulla.

Input		Prezzo
Classi valide	CL29	Intero positivo



Classi non valide	CL30	Numero negativo
Classi non valide	CL31	Numero nullo.

Input		Link
Classi valide	CL32	Stringa alfanumerica di formato www.xx.xx www.xxx.xxx Dove xxx rappresenta una stringa alfanumerica
Classi non valide	CL33	Stringa alfanumerica che non comincia con www
Classi non valide	CL34	Stringa nulla.

Test Case:

Inserimento Applicazione_4

Input	Classe	Membro
Id	CL13	01
Nome	CL17	"Programma e-commerce"
Tipo	CL21	"Software"
Autori	CL25	"Giardino Pascale"
Prezzo	CL29	125
Link	CL32	www.compraonline.com

Inserimento Applicazione_5

Inserimento Applicazione_4



Input	Classe	Membro
Id	CL15	""
Nome	CL17	"Programma gestione clienti"
Tipo	CL22	"Sof"
Autori	CL25	"Giardino Pascale"
Prezzo	CL30	-3000
Link	CL33	gestione.com

12. Pianificazione del Testing

Il test sarà effettuato nell'arco di massimo 4 giorni (escluso il tempo necessario per la pianificazione). Il compito di testare il sottosistema implementato è stato assegnato dal Team Manager ai componenti del Team che non hanno partecipato all'implementazione, fornendo loro tutto il materiale necessario. Tuttavia, a causa dell'esiguo numero di membri del team, il Project Manager incaricherà anche gli implementatori di collaborare alla fase di test, lasciando loro però un ruolo più marginale. Questi ultimi inoltre, avranno il compito di effettuare le correzioni necessarie dopo la rivelazione di eventuali failure e far ripristinare il test della componente malfunzionante. Il risultato della fase di test sarà suddiviso in 2 documenti:

- TestPlanning
- TestCaseSpecification

13. Glossario

ODD: Documento di Object Design.

SDD: Documento di System Design.

RAD: Documento di Raccolta e Analisi dei Requisiti.

TestPlan: Documento in cui si pianifica l'attività di testing.



TestCaseSpecification: Documento in cui si specificano i Test Case.

Scheduling: Pianificazione dei tempi e delle precedenze nell'impiego di risorse materiali e umane per un buon svolgimento del processo produttivo; è una delle fasi del processo di pianificazione e si inserisce tra il routing (configurazione del ciclo) e il dispatching (assegnazione agli addetti degli ordini da eseguire). La schedulazione della produzione fissa i tempi e l'ubicazione dei componenti affinché uomini e mezzi siano disponibili al posto giusto e nel momento giusto. Si serve delle tecniche reticolari (PERT) e del diagramma di GANTT.

TestCase: Insieme di informazioni e variabili che, testate, possono determinare se un'applicazione funziona correttamente o no.

BlackBox Testing: Tecnica di test d'unità che si focalizza sull'I/O.

WhiteBox Testing: Tecnica di test d'unità che si focalizza sulla struttura interna