TP:
Test Plan

ScheduFIRE

Riferimento	
Versione	0.1
Data	05/12/2019
Destinatario	Top Management
Presentato da	Giordano Domenico, Sarno Orazio, Sansone Michela
Approvato da	

Revision History

Data	Versione	Cambiamenti	Autori
29/11/19	0.1	Criteri Pass/Fail e Criteri di sospensione e ripresa. Revisone	Labanca Nicola, Bruno Biagio, Cipolletta Ciro
30/11/19	0.2	Features da testare e Features da non testare. Revisione	Giuliano Alfredo, Annunziata Giusy, Bombardelli Emanuele
01/12/19	0.3	Introduzione. Revisione	Bombardelli Emanuele, Perillo Francesca.
02/12/19	0.4	Approccio, Testing di unità e testing di integrazione. Revisione	Annunziata Giusy, Bruno Biagio, Sottile Eugenio.
04/12/19	0.5	Test Caregory Partition e Testing di Sistema. Revisione	Bruno Biagio, Sottile Eugenio, Giuliano Alfredo, Annunziata Giusy

Sommario

1. Introduzione

Lo scopo di questo documento è quello di analizzare e gestire lo sviluppo e le attività di Testing riguardanti il software ScheduFIRE. Questa sessione di lavoro deve verificare il corretto funzionamento di ScheduFIRE in diversi casi, studiati appositamente per mettere alla prova ogni singola funzionalità e caratteristica del sistema, al fine di verificare se esistono incongruenze tra il comportamento atteso e il comportamento osservato. Andremo quindi a rilevare gli eventuali errori prodotti all'interno del codice, per evitare che essi si presentino nel momento in cui il sistema verrà utilizzato dall'utente finale, e per assicurare a quest'ultimo un corretto output del software per la maggior parte del tempo. Tutte le gestioni precedentemente individuate saranno sottoposte a testing.

Si tenga presente che nelle prime versioni di questo documento verranno sviluppati test di unità di tipo "*Black-Box*", e verranno aggiunte in itinere altre metodologie di testing atte a ridurre al minimo il rischio di incorrere in fallimenti dovuti all'utilizzo malevolo, scorretto e/o comunque non pianificato in fase di implementazione.

2. Riferimenti ai documenti correlati

Il Test Plan è in stretta relazione con il RAD (Requirement Analysis Document) prodotto fino a questo momento. Tale documento infatti, contiene la descrizione dei requisiti funzionali (paragrafo 3.2) e dei requisiti non funzionali (paragrafo 3.3) che il sistema ScheduFIRE dovrà implementare. Nella fase di testing si definiranno le caratteristiche che i dati di input al sistema dovrà avere per un corretto utilizzo.
La relazione fra il test plan e il RAD riguarda in particolare i requisiti funzionali e non funzionali del sistema, poiché i test verranno eseguiti su alcune delle funzionalità espresse nel Requirement Analysis Document.

3. Dettagli per il Testing

3.1 Features da testare

I sottosistemi che verranno sottoposti alla fase di testing black box sono i seguenti:

- Package "Gestione Ferie": verrà effettuato il test riguardante il requisito "Aggiungere periodo di ferie"
- Package "Gestione malattia": verrà effettuato il test riguardante il requisito "Aggiungere periodo di malattia"
- Package "Gestione personale": verrà effettuato il test riguardante i requisiti "Aggiungere VF" e "Modificare VF"
- Package "Sessione utente": verrà effettuato il test riguardante i requisiti "Login"

3.2 Features da non testare

Non verranno testati con black box i requisiti per cui non è richiesto nessun inserimento di input da parte dell'utente, quali:

3.3 - Criteri Pass/Fail

I dati di input saranno individuati dalle specifiche, suddivisi e raggruppati in insiemi che presentano caratteristiche comuni, in modo da identificare gli effettivi valori di input dai valori diversi. In questo modo, sarà possibile effettuare test case su combinazioni di unici elementi rappresentati dei diversi insiemi, mediante il **Category Partition Testing.**
I diversi test case saranno considerati superati (**Pass**) se, grazie alla loro esecuzione, si verificheranno failure in presenza di input corretto e se quindi, ci sarà discordanza con il risultato atteso.

Per failure si intende l'incapacità di svolgere una determinata funzione, anche se in presenza di corretto input (output inatteso).
I diversi test case ed il test saranno considerati falliti (**Fail**) quando non saranno riscontrati errori mediante l'esecuzione dei test case.

3.4 - Criteri di sospensione e ripristino

I criteri di sospensione comprendono tutti quei casi critici dove gli errori hanno un dannoso impatto sul progresso dell'attività di testing. Questi possono essere:

- Crash del database
- Crash del server
- Crash della connessione
- Problemi relativi all'ambiente di sviluppo relativo al testing

Inoltre, l'attività di testing verrà sospesa quando saranno ottenuti i risultati attesi e precedentemente previsti, in accordo con il tempo di sviluppo pianificato.

Per quanto riguarda il ripristino, l'attività di testing potrà riprendere in seguito a modifiche attuate per risolvere ciò che precedentemente ha generato errori o fallimenti.
In questo caso, verrà effettuato testing di regressione in modo da assicurarsi che le modifiche apportate non abbiano introdotto ulteriori errori.
Verranno quindi effettuati nuovamente i test case somministrati precedentemente al sistema, in modo da assicurarsi dell'effettiva risoluzione del problema.

4. Approccio

L'approccio nella fase di testing avviene mediante il Black-Box testing per testare i comportamenti di I/O. Esso consiste in 3 fasi:

- Testing di unità, che verifica i singoli bean;
- Testing di integrazione, che verifica la corretta integrazione delle classi attraverso il testing dei manager;
- Testing di sistema, che testa il sistema funzionante in base alle sue funzionalità;

4.1 Testing di unità

In questa prima fase si effettuerà il controllo delle varie classi e metodi del sistema, ricercando le condizioni di fallimento andando ad evidenziare gli errori. Per realizzare il testing di ogni singola componente verrà utilizzata la tecnica "Black-Box testing". Così facendo andremo ad esaminare le funzionalità dell'applicazione ed il comportamento input/output delle singole componenti senza tener conto della loro struttura interna. Verranno realizzate delle classi di equivalenza, poiché risulta impossibile testare tutti i casi possibili di input, scegliendo per ognuna un test case per ridurre la ridondanza e rendere il test più efficiente.

4.2 Testing di integrazione

Il testing di integrazione consente di individuare i problemi che si verificano quando due unità si combinano. Dopo aver sottoposto ogni componente al testing di unità, ed aver corretto gli eventuali errori scaturiti dal test, essi verranno integrati in sottosistemi più grandi per sottoporli ad un test di integrazione. Verrà utilizzato un approccio di tipo "bottom-up", che prevede il solo utilizzo dei driver, senza dover ricorrere all'utilizzo degli stub.

4.3 Testing di sistema

Prima di essere pronto all'uso, il sistema affronterà l'ultima fase di testing, quello di sistema, che testeremo con un apposito tool. Con il testing di sistema verrà effettuato un controllo della conformità dell'intero sistema con i requisiti dell'utente finale. Lo scopo di questa fase è testare le funzionalità più importanti, usate maggiormente e con maggior probabilità di fallimento.

4.3.1Category Partition

I casi di test sono stati progettati utilizzando il metodo del Category Partition. Questo è un tipo di test combinatorio in cui si identificano i diversi attributi, valori rilevanti e possibili combinazioni separando l'identificazione dei valori che caratterizzano l'input dalla combinazione di valori diversi nei diversi casi di test completi, permettendo di fornire una stima dei casi di test molto presto.

Con il Category Partition si possono generalizzare casi di test funzionali, seguendo tre passi:

• **Decomposizione delle specifiche in caratteristiche indipendentemente verificabili**: si identificano le feauture che devono essere testate in modo separato, identificando i parametri ed elementi nell'ambiente di esecuzione da cui dipende come, ad esempio, un Database. Per ciascun parametro ed elemento dell'ambiente si identificano le caratteristiche del parametro che sono dette categorie;

• **Identificare i valori di riferimento**: questo passo prevede l'identificazione di un insieme di valori rappresentativi per ciascuna caratteristica di ogni parametro definito nella fase precedente;

• **Generazione delle specifiche dei casi di test**: Per ogni test case realizzato, saranno prodotti i relativi Test Case Specifications contenenti gli esempi di tutte le casistiche individuate per ogni funzionalità e i relativi risultati.