

System Test Plan

Revision History

<i>Data</i>	<i>Versione</i>	<i>Descrizione</i>	<i>Autori</i>
15/12/2016	1.0	Creazione documento	Paolo Zirpoli

Sommario

Revision History	1
1. Introduzione	3
2.Riferimenti	3
3.Test di integrazione	3
3.1 Approccio di Integration Testing	3
3.2 Componenti da testare.....	3
4.Test Criteria.....	4
4.1 Suspension Criteria	4
4.2 Exit Criteria.....	4
4.3 Pass/Fail Criteria.....	4
5.Test Enviroment	4

1. Introduzione

Il testing di integrazione rappresenta una delle fasi di testing più importanti, in quanto consiste nella verifica delle interazioni tra due o più componenti. L'obiettivo del testing consiste nella verifica della corretta interazione tra le componenti e il rispetto delle interfacce.

Il presente documento espone la strategia di testing attuata per l'implementazione della piattaforma BeVoyager.

2. Riferimenti

Per verificare la corretta integrazione dei sottosistemi della piattaforma, sono stati predisposti dei test case basati sulla suddivisione proposta in fase di System Design.

Le specifiche dei test sono reperibili nel documento Test Case Specification

3. Test di integrazione

3.1 Approccio di Integration Testing

Il Sistema da noi proposto è sviluppato seguendo un'implementazione test driven. Cioè, viene implementato un sistema iniziale, che è poi sottoposto ai test descritti nella Test Case Specification. Il sistema è considerato affidabile quando passa una specifica soglia percentuale di test.

I test in questione saranno test automatizzati, per permettere di avere un riscontro sullo stato dell'implementazione in maniera veloce ed affidabile, eliminando la variabile umana.

Per l'implementazione della piattaforma seguiremo l'approccio non incrementale, o big-bang testing, che prevede l'implementazione completa del sistema in una fase, che presenterà già tutti i servizi offerti, che sarà poi testata e via via raffinata fino al raggiungimento di un risultato soddisfacente. Modifiche al codice, quindi, devono essere fatte in caso di fallimento di un test o di non raggiungimento di requisiti qualitativi.

3.2 Componenti da testare

Le componenti da testare sono tutte quelle individuate in fase di design ed implementate successivamente.

4. Test Criteria

4.1 Suspension Criteria

Se si rileva che più del 20% dei test da eseguire non vengono superati dall'implementazione corrente, l'attività di testing verrà sospesa per dare l'opportunità agli sviluppatori di migliorare il codice, con l'obiettivo di passare i test precedentemente non superati.

4.2 Exit Criteria

Specifica i criteri che denotano il completamento positivo di una fase di test :

- La Run rate deve essere obbligatoriamente 100%, cioè tutti i test devono essere stati eseguiti almeno una volta
- Il Pass rate è del 90%, soglia obbligatoria.

4.3 Pass/Fail Criteria

Il testing ha successo se l'output osservato è diverso dall'output atteso: ciò significa che parliamo di successo se il test individua una failure.

In caso di test eseguito con successo, bisognerà analizzare la failure individuata e correggerla al più presto.

Si procederà quindi ad eseguire il regression testing, cioè verranno rieseguiti tutti i test per accertarsi che le nuove modifiche non abbiano introdotto nuove failures.

Si parla di fallimento del test se questo non individua errori.

5. Test Environment

Il test delle funzionalità sarà eseguito con Selenium sul browser Firefox.