

Università Degli Studi Di Salerno
Corso di Laurea Triennale in Informatica

Ingegneria Del Software

Componenti del gruppo:

Antonio Gambale n.m 05121
05346 Raissa Francesca Colicino n.m
05121 05341

Full-Beer

1.Introduzione

Il testing di unità rappresenta una delle fasi di testing più importanti, in quanto consiste della verifica e del collaudo delle singole unità software di un sistema. Essendo così importante va affrontata con particolare cautela, per evitare imprevisti e un conseguente slittamento dei tempi.

Il documento UNIT TEST PLAN descrive l'approccio al testing ed il framework generale che guiderà i test.

Ha l'obiettivo di rimuovere eventuali difetti critici prima dell'inizio dei prossimi livelli di test.

Il documento introduce quindi:

- Approccio di Unit Testing, ovvero le regole alla base del test e la descrizione del processo di impostazione di un test valido
- Pass/Fail criteri per la gestione dei test: processo per gestire la logistica del test e tutti gli eventi che si verificano durante l'esecuzione

2.Relazione con altri documenti

Dalla correlazione con altri documenti derivano diversi criteri di accettazione del test.

I documenti di Test Case Specification infatti devono essere disponibili prima dell'inizio della fase di progettazione del test.

Per individuare correttamente i test case si deve tener conto dei documenti prodotti precedentemente. Infatti, la fase di testing è strettamente legata alle fasi precedenti che costituiranno un punto di partenza indispensabile per poter effettuare un testing corretto ed adeguato.

2.1 Relazione con RAD

La relazione tra Unit Test Plan e RAD interessa i requisiti funzionali e non funzionali del sistema, in quanto i test che saranno eseguiti terranno conto delle specifiche espresse nel documento.

2.2 Relazione con SDD

Nel System Design Document si è suddiviso il sistema in sottosistemi e l'architettura in tre livelli:

Presentation Layer, Application Layer e Data Layer

Il test dovrà rimanere il più possibile fedele alle suddivisioni.

2.3 Relazione con ODD

Il test d'integrazione dovrà attenersi quanto più possibile alle interfacce delle classi e i package nell'ODD.

3.Dettagli di Unit Testing

3.1Approccio di Unit Testing

Il testing si compone di tre fasi.

Inizialmente verranno eseguiti i testing di unità dei singoli componenti in modo da testare la correttezza di ciascuna unità constatando il corretto funzionamento delle singole unità di codice. Questa fase verrà effettuata al completamento di ogni unità realizzata per poter individuare tempestivamente gli errori presenti nel codice.

3.2Testing di unità

Per effettuare il testing di unità verrà utilizzata la tecnica "Black-Box testing" attraverso il framework JUnit.

In questa fase saranno analizzate le funzionalità dell'applicazione ed il comportamento delle singole componenti senza tener conto della loro struttura interna ma focalizzandosi sui comportamenti di I/O. I risultati del testing verranno analizzati e usati per correggere gli errori che hanno causato dei fallimenti del sistema. Se si dovessero verificare errori con dei risultati inattesi si interverrà tempestivamente sulla componente in modo da renderla correttamente funzionante e procedere con le fasi di testing successive.

4.Pass/Fail criteria

Lo scopo del testing è quello di dimostrare la presenza di errori all'interno del sistema. Le attività di testing infatti, saranno mirate all'identificazione degli errori e alla loro successiva correzione.

Il testing avrà successo se l'output osservato risulterà diverso dall'output atteso, quindi si parla di successo per il test se esso rileva un errore. L'errore verrà poi analizzato e, se legato ad un fault, si procederà alla sua correzione. Sarà infine iterata la fase di testing per verificare che la modifica apportata non abbia avuto alcun impatto su altre componenti del sistema.

Il test fallirà quando non individuerà nessun errore.

5.Dettagli Test Results

5.1Testing di Unità

Di seguito sono riportati tutti i test effettuati per le classi "Model" del sistema FullBeer tramite testing Junit