



IT INTEGRATION TEST PLAN



Sommario

1. Introduzione
2. Test di integrazione
 - 2.1 Approccio Testing di integration
3. Struttura delle componenti da testare



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. Questo documento ha il compito di identificare la strategia di testing di integrazione per il sistema Tutto-Elettronica.

Il **Test di Integrazione** rileva bug che non sono stati determinati durante il Test di Unità, focalizzando l'attenzione su un insieme di componenti che vengono integrate. Due o più componenti vengono integrate e analizzate, e quando dei bug sono rilevati, possono essere aggiunte nuove componenti per correggerli. Siamo, quindi, nella situazione in cui l'intero sistema è visto come una collezione di sottosistemi determinati durante il system e l'object design. L'ordine in cui i sottosistemi vengono selezionati per il testing e per l'integrazione determina la **strategia di testing**.

2. Test di integrazione

2.1 Approccio Testing di Integration

La strategia adottata per il testing di integrazione è quella di tipo "Bottom-up".

I sottosistemi al livello più basso della gerarchia sono testati individualmente. I successivi sottosistemi ad essere testati sono quelli che chiamano i sottosistemi testati in precedenza. Si ripete quest'ultimo passo finché tutti i sottosistemi non sono stati testati. Per il testing vengono utilizzati i Test Driver (che simulano le componenti dei livelli più alti, che non sono state ancora integrate).

Vantaggi:

- Utile per integrare sistemi object-oriented, real-time, con rigide richieste di performance.
- Non richiede stub.
- Il processo di test ed integrazione delle unità è intuitivamente più semplice da realizzare

Svantaggi:

- Questo approccio non è buono per sistemi decomposti funzionalmente, perché testa i sottosistemi più importanti solo alla fine, quindi si percepisce la funzionalità del sistema solo alla fine.
- Può richiedere driver anche complessi.
- Non individua errori di progettazione di alto livello se non molto avanti nel processo.
- Più tardi viene scoperto un errore, più è costoso eliminarlo, in quanto richiede la sua correzione, solitamente con un modulo foglia e la ripetizione parziale del test fino ai moduli nei quali è stato trovato l'errore.

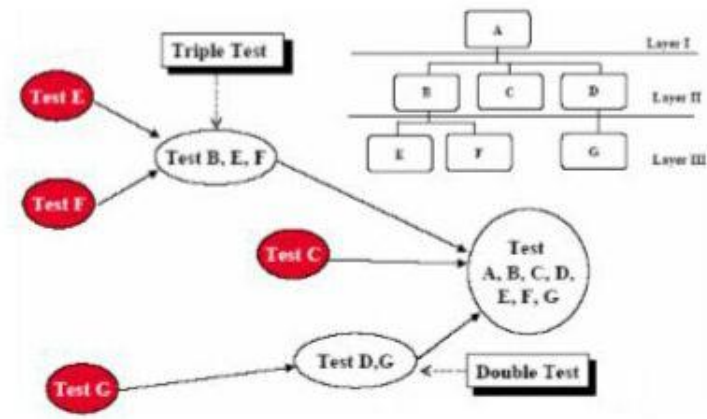


Per preparare il testing di integrazione sono stati seguiti questi passi:

- Delineare l'architettura del sistema
- Identificare i moduli
- Capire la funzionalità di ogni modulo
- Capire come i dati interagiscono con gli altri moduli
- Capire input e output del sistema
- Creare layer (insieme di moduli) da testare
- Identificare e specificare delle condizioni di test
- Prendere una condizione alla volta e formulare un test case

3. Struttura delle componenti da testare

- Si sviluppano prima i moduli di più basso livello, e si testano con i driver necessari
- Poi si procede verso l'alto, introducendo a ogni livello i driver necessari





Con la strategia bottom-up vengono prima collaudati i moduli di più basso livello nella gerarchia prodotta dalla progettazione, si tratta cioè delle unità più piccole ed elementari che compongono il programma. Quando sono valutati e corretti si passa al livello superiore che viene testato utilizzando le funzionalità del precedente livello, e si risale quindi così fino al sistema intero.

