



UNIVERSITÀ DEGLI STUDI DI SALERNO
DIPARTIMENTO DI INFORMATICA

Esame di Ingegneria, Gestione ed Evoluzione del Software

PROGETTO CODESMILE

Modifications & Regression Testing

TEAM MEMBERS

Dario Mazza - 0522501553

Nicolò Delogu - 0522501556

REPOSITORY

https://github.com/xDaryamo/smell_ai

VERSIONE 0.1

-

Anno Accademico 2024-2025

1 Introduzione

Il presente documento descrive le attività di testing condotte per verificare l'affidabilità e la correttezza del sistema a seguito delle modifiche introdotte dalle Change Requests (CR). L'obiettivo principale è garantire che le nuove funzionalità rispettino i requisiti specificati e che il sistema esistente non venga compromesso dalle modifiche. A tal fine, sono stati eseguiti due tipi principali di testing:

- **Testing Post-Modifiche:** Finalizzato a verificare il corretto funzionamento delle nuove funzionalità introdotte dalle CR.
- **Testing di Regressione:** Condotta per assicurare che le modifiche non abbiano introdotto anomalie o comportamenti inattesi nel sistema esistente.

2 Testing Post-Modifiche

2.1 Risultati per la CR1

Per testare il sistema post-ristrutturazione orientata agli oggetti, abbiamo definito delle test suite di unità e di integrazione per le classi che hanno sostituito il vecchio codice procedurale. Questi test sono stati progettati per verificare che le nuove classi implementassero correttamente le funzionalità precedenti e che le interazioni tra i vari componenti fossero gestite senza introdurre anomalie. In particolare, sono stati eseguiti:

- **Test di unità per le singole classi:** Ogni classe è stata testata singolarmente per garantire che implementasse correttamente i metodi e le funzionalità previsti, in linea con i requisiti.
- **Test di integrazione tra le classi:** Sono stati eseguiti test per verificare che l'interazione tra le classi fosse fluida e senza errori, simulando scenari d'uso completi che coinvolgevano più classi.

Esito:

- I test di unità hanno confermato che le classi implementano correttamente le funzionalità previste.
- I test di integrazione hanno verificato che l'interazione tra le classi non ha introdotto regressioni e ha mantenuto il comportamento atteso del sistema.

2.2 Risultati per la CR2

La CR2 ha introdotto una nuova web application per l'analisi dei code smells, sviluppata come modulo indipendente e senza modificare i componenti esistenti del sistema. Nonostante l'assenza di modifiche dirette al codice preesistente, sono stati eseguiti diversi livelli di testing per garantire che il nuovo modulo non introducesse anomalie o comportamenti inattesi.

Approccio:

- **Testing unitario per le componenti del frontend:** Le componenti della web application sono state testate singolarmente per verificare il corretto comportamento dei singoli elementi dell'interfaccia e delle relative logiche.
- **Testing di integrazione per i servizi:** Sono stati effettuati test mirati per verificare che i servizi funzionassero in modo corretto
- **Testing end-to-end con Cypress:** È stato utilizzato Cypress per eseguire test end-to-end, simulando flussi utente per validare il funzionamento del modulo nella sua integrità

Esito:

- I test end-to-end hanno confermato il corretto funzionamento della nuova funzionalità in tutti i flussi principali.
- I test di unità e di integrazione sono stati eseguiti senza riscontrare errori o anomalie.

Conclusioni: L'introduzione della web application non ha causato regressioni o alterazioni nelle funzionalità preesistenti. Grazie al testing unitario, di integrazione e end-to-end, il comportamento complessivo del sistema è stato confermato come stabile e invariato.

3 Testing di Regressione

3.1 Risultati per la CR1

Durante l'integrazione del branch relativo alla ristrutturazione del sistema da un'architettura procedurale a un'architettura orientata agli oggetti, è stato condotto un test di regressione facendo uso della test suite (di sistema) sviluppata in maniera preliminare rispetto alla CR1. I test sono stati eseguiti sia sulla versione precedente del sistema (architettura procedurale) sia sulla nuova versione ristrutturata (architettura OO), per verificare che il comportamento delle componenti restasse invariato al fronte della cambio architetturale.

Esito dei test di regressione:

- L'esecuzione della test suite pre-ristrutturazione ha identificato alcune problematiche che sono state analizzate e documentate nel *Test Incident Report*.
- L'esecuzione della test suite post-ristrutturazione è stata eseguita senza riscontrare anomalie e le problematiche presenti nel *Test Incident Report* sono state risolte con successo.

Conclusioni: I test hanno confermato che il comportamento del sistema è invariato rispetto alla versione precedente ed inoltre esso non presenta le problematiche discusse nella sezione 3.1

3.2 Risultati per la CR2

La CR2 ha introdotto una nuova web application per l'analisi dei code smells, sviluppata come modulo indipendente e senza modificare i componenti esistenti del sistema. Nonostante l'assenza di modifiche dirette al codice preesistente, è stato eseguito un test di regressione per garantire che l'integrazione della nuova funzionalità non introducesse comportamenti inattesi nel sistema esistente.

Approccio: La test suite definita nella CR precedente è stata rieseguita per verificare che non ci fossero anomalie nel sistema.

Esito:

- Non sono state rilevate anomalie durante l'esecuzione della test suite.
- Non sono stati riscontrati problemi di compatibilità con l'ambiente o con altri moduli del sistema.

Conclusioni: L'introduzione della web application non ha causato regressioni o alterazioni nelle funzionalità preesistenti. Il comportamento del sistema è stato confermato come stabile e invariato.

3.3 Risultati per la CR3

Durante l'implementazione della CR3, è stato introdotto un metodo automatizzato per la generazione di dataset con code smells utilizzando Modelli di Linguaggio di Grandi Dimensioni (LLM). L'obiettivo era consentire la creazione di varianti di codice con specifici difetti, utili per l'addestramento di modelli di rilevamento.

Approccio:

- È stato utilizzato Qwen2.5Coder14B per iniettare smells nei codebase, convalidando le modifiche attraverso un'analisi manuale preliminare.
- La test suite definita nella CR1 è stata rieseguita per verificare l'integrità del sistema post integrazione.

Esito:

- Non sono state rilevate anomalie nel comportamento del sistema esistente.
- I dataset generati sono risultati conformi ai requisiti di R8, consentendo la creazione di codice con difetti specifici.

Conclusioni: L'uso degli LLM ha permesso di automatizzare con successo la generazione di dataset con code smells, senza introdurre regressioni o alterazioni nel sistema preesistente.

3.4 Risultati per la CR4

La CR4 si è focalizzata sull'addestramento di un modello di intelligenza artificiale per il rilevamento automatico di code smells, sfruttando i dataset generati dagli LLM in CR3.

Approccio:

- È stato progettato e addestrato un modello utilizzando altre varianti di Qwen2.5Coder sui dataset creati nella CR3.
- Sono state definite metriche di valutazione (accuratezza, precisione, richiamo e F1 score) per analizzare le prestazioni del modello.
- È stata rieseguita la test suite (unità, integrazione e sistema) per garantire che l'aggiunta del nuovo modulo non introducesse problemi.

Esito:

- Il modello ha raggiunto prestazioni soddisfacenti.
- Non sono state rilevate anomalie nel sistema durante i test di regressione.

Conclusioni: L'integrazione è stata completata senza alterare la stabilità del sistema.