UNIVERSITÀ DEGLI STUDI DI SALERNO
DIPARTIMENTO DI INFORMATICA

Esame di Ingegneria, Gestione ed Evoluzione del Software

PROGETTO CODESMILE

Test Incident Report (Pre-Modifications)

TEAM MEMBERS

Dario Mazza - 0522501553

Nicolò Delogu - 0522501556

REPOSITORY

`https://github.com/xDaryamo/smell_ai`

VERSIONE

0.1

Anno Accademico 2024-2025

1 Introduzione

Nel presente documento saranno illustrate le problematiche emerse durante l'esecuzione dei casi di test prima dell'applicazione delle modifiche a CodeSmile. L'obiettivo principale di questo report è fornire un'analisi dettagliata dei fault individuati, cioè delle incongruenze tra il comportamento atteso (oracolo) e i risultati effettivamente ottenuti. Attraverso un'analisi approfondita di ciascuna problematica, verranno proposte soluzioni che puntano a migliorare il sistema e ottimizzare il processo di sviluppo, al fine di garantire una qualità superiore del codice analizzato.

2 Incident Report

2.1 STI1 - System Testing Incident 1

2.1.1 Informazioni di Base

- **Progetto:** Code Smile
- **Ambiente:** Ambiente di Test
- **Nome Incident:** STI1
- **Test Case:**Tutti i test case (TC_1 - TC_18)
- **Priorità:** Alta
- **Gravità:** Alta

2.1.2 Descrizione dell'Incidente

- **Descrizione:** Tutti i test case falliscono quando si esegue il tool CodeSmile dalla root del progetto poiché viene cercato un file di log in un percorso errato. L'esecuzione corretta richiede che il comando sia lanciato dalla directory `controller`.
- **Passi per la riproduzione:** Posizionarsi nella root del progetto. Eseguire il tool tramite CLI o GUI (`python -m controller.analyzer` o `python -m controller.GUI`.

  Verificare l'errore relativo al percorso del file di log.
- **Comportamento attuale**: Il tool non trova il file di log e restituisce un errore, impedendo la corretta esecuzione dei test.

2.1.3 Risoluzione e Pianificazione

- **Stato:** Aperto
- **Cause identificate:**L'errore si verifica perché il tool cerca il file di log in un percorso errato se eseguito dalla root del progetto. Non c'è un meccanismo che permetta di adattare il percorso in modo dinamico.

- **Componente coinvolta:** Modulo controller per gestione dei percorsi di log.
- **Pianificazione:** Implementare una verifica dinamica del percorso del file di log. Aggiungere indicazioni nella documentazione per eseguire il tool dalla directory corretta.
- **Soluzione:** Modificare il modulo controller per supportare esecuzioni sia dalla root che dalla directory specifica. Aggiornare il manuale d'uso per indicare chiaramente il percorso di esecuzione corretto.

2.2 STI2 - System Testing Incident 2

2.2.1 Informazioni di Base

- **Progetto:** Code Smile
- **Ambiente:** Ambiente di Test
- **Nome Incident:** STI2
- **Test Case:** TC4
- **Priorità:** Alta
- **Gravità:** Alta

2.2.2 Descrizione dell'Incidente

- **Descrizione:** Il code smell di tipo generico **Deterministic Algorithm Option Not Used** non viene rilevato e non è incluso nell'output del tool.
- **Passi per la riproduzione:** Eseguire il tool da GUI, fornire in input il percorso di TC4, fornire in output il percorso della cartella Output, abilitare l'esecuzione parallela, specificare un numero di *workers* uguale a 5 e premere "run".
- **Comportamento attuale**: Il tool non rileva il code smell di tipo generico **Deterministic Algorithm Option Not Used**.

2.2.3 Risoluzione e Pianificazione

- **Stato:** Aperto
- **Cause identificate:** Riga 33-34 del codice della componente coinvolta, manca un controllo completo e ben strutturato su gli usi della funzione **use_deterministic_algorithms** del modulo **torch**. Attualmente la verifica del relativo nodo AST prevede solo la verifica del sotto-attributo *id* dell'attributo *func* del nodo.
- **Componente coinvolta:** Modulo *cs_detector*, sotto-modulo *detection_rules*, file Generic.py
- **Pianificazione:** aggiungere un controllo completo e ben strutturato alla visita del nodo dell'AST .

- **Soluzione:** modificare il codice sorgente della funzione **deterministic_algorithm_option_not_used** per includere un doppio controllo sul nodo dell'AST visitato: *isinstance(node.func, ast.Attribute)* e *isinstance(node.func, ast.Name)* in modo da riconoscere entrambi gli usi di **use_deterministic_algorithms**.

2.3 STI3 - System Testing Incident 3

2.3.1 Informazioni di Base

- **Progetto:** Code Smile
- **Ambiente:** Ambiente di Test
- **Nome Incident:** STI3
- **Test Case:** TC3, TC6, TC8, TC13, TC15, TC18, TC21
- **Priorità:** Media
- **Gravità:** Media

2.3.2 Descrizione dell'Incidente

- **Descrizione:** Nonostante il tool analizzi dei progetti vuoti (o che contengono file non Python) o privi di smells, vengono creati dei file *to_save.csv* vuoti (sia se eseguito da GUI sia da CLI). Se il tool viene eseguito da CLI, la console stampa un errore non specificato: **Error.**.
- **Passi per la riproduzione:** Eseguire il tool tramite GUI o CLI, specificare i percorsi di input (progetti vuoti o pribi di smells) e di output e gli altri relativi parametri. Premere "run".
- **Comportamento attuale**: il tool crea un file *to_save.csv* ogni volta che analizza un progetto vuoto (o che contiene file non Python) o privo di smells. Inoltre restituisce un errore non chiaro nella console (se eseguito tramite CLI).

2.3.3 Risoluzione e Pianificazione

- **Stato:** Aperto
- **Cause identificate:** Righe 29 e 93 del codice sorgente
- **Componente coinvolta:** Modulo controller, file analyzerṗy
- **Pianificazione:** Ispezionare il codice della componente coinvolta per individuare il motivo del comportamento inaspettato del tool.

- **Soluzione:** Aggiungere un controllo alla riga 93 per verificare se la variabile *to_save* sia vuoto o meno. In caso di assenza di smells, evitare di restituire la variabile *to_save*. Nella riga 29, cambiare il messaggio che viene stampato dalla funzione **merge_results** per evitare fraintendimenti sul significato dello stesso.