

UNIVERSITÀ DEGLI STUDI DI SALERNO

DIPARTIMENTO DI INFORMATICA



Ingegneria del Software: Tecniche Avanzate

Progetto ISTA – Evoluzione di CodeSmile

Pre-Modification Test Incident Report

Professore:

Ch.mo Prof.

Andrea De Lucia

Tutor di riferimento:

Giammaria Giordano

Gilberto Recupito

Studenti:

Antonio D'Auria

Mat. NF22500063

Emanuele D'Auria

Mat. 0522501918

Luca Casillo

Mat. NF22500037

ANNO ACCADEMICO 2025/2026

INDICE

Indice

Elenco delle tabelle

1	Introduzione	1
2	Struttura dei Livelli di Test	2
3	Incident Report	3
3.1	Static Analysis Tool	3
3.1.1	Unit Test Incidents	3
3.1.2	Integration Test Incidents	3
3.1.3	System Test Incidents	3
3.2	Web Application	4
3.2.1	Unit Test Incidents	4
3.2.2	Integration Test Incidents	4
3.2.3	End-to-End Test Incidents	5
4	Incident Resolution	7
4.1	Static Analysis Tool	7
4.1.1	Unit Test Fixes	7
4.2	Web Application	7
4.2.1	Integration Test Fixes	7
4.2.2	End-to-End Test Fixes	8

ELENCO DELLE TABELLE

3.1 Static Analysis Tool - Unit Test Outcomes	3
3.2 Static Analysis Tool - System Test Outcomes	4
3.3 Web Application - Integration Test Outcomes	4
3.4 Web Application - End-to-End Test Outcomes	5

1 Introduzione

Il presente documento riporta il *Pre-Modification Test Incident Report* relativo alla versione corrente del sistema, antecedente all'introduzione delle Change Request. Il suo obiettivo è documentare gli incidenti emersi durante l'esecuzione delle attività di testing pre-modifica definite nel documento Pre-Modification System Testing.

Il contenuto di questo report costituisce la baseline di riferimento per:

- distinguere eventuali difetti pre-esistenti da quelli potenzialmente introdotti dalle modifiche;
- motivare eventuali correzioni o adeguamenti della test suite.

2 Struttura dei Livelli di Test

In coerenza con quanto riportato nel piano di test pre-modifica, l'esecuzione dei test è articolata sui seguenti livelli:

- **Static Analysis Tool** (directory test/):
 - *Unit Testing* (test/unit_testing): verifica dei package e moduli in isolamento (es. cli, code_extractor, detection_rules, report);
 - *Integration Testing* (test/integration_testing): verifica della collaborazione tra moduli cooperanti del core di analisi;
 - *System Testing* (test/system_testing): validazione del comportamento complessivo tramite casi di test strutturati basati su progetti e directory di input.
- **Web Application** (directory webapp/):
 - *Unit Testing* (webapp/_tests_): test dei componenti UI mediante Jest;
 - *Integration Testing* (webapp/integration_tests): test dell'integrazione tra gateway e servizi backend;
 - *End-to-End Testing* (webapp/cypress): test E2E dei principali flussi utente tramite Cypress.

Prima di procedere all'esecuzione dei test, sono state installate tutte le dipendenze definite nel file `requirements.txt` presente nella root del progetto. Per la webapp, inoltre, sono state installate anche le dipendenze definite nel file `requirements.txt` presente nel percorso `webapp/gateway/requirements.txt`, nonché le dipendenze npm necessarie, installate nella directory webapp.

Si osserva che, nel seguito del documento, i riferimenti ai file di test sono espressi in forma abbreviata, in quanto la collocazione dei diversi livelli di testing all'interno della struttura del repository è già stata esplicitata nella presente sezione.

3 Incident Report

Di seguito sono riportati gli incidenti emersi durante le attività di test. Gli incidenti sono organizzati per componente e tipologia di testing.

3.1 Static Analysis Tool

Questa sottosezione riporta gli incidenti rilevati durante l'esecuzione dei test relativi allo Static Analysis Tool.

3.1.1 *Unit Test Incidents.* Questa tabella riassume gli incidenti rilevati negli unit test.

Tabella 3.1. Static Analysis Tool - Unit Test Outcomes

ID	Test File	Funzione di Test
INC-SAU-01	components/test_project_analyzer.py	test_analyze_project_empty_directory
INC-SAU-02	gui/test_code_smell_detector_gui.py	test_run_program_missing_paths

INC-SAU-01. Durante l'esecuzione del test `test_analyze_project_empty_directory`, l'analisi di una directory di input priva di file Python provoca il sollevamento di un'eccezione `ValueError`.

Il comportamento causa il fallimento del test che verifica la gestione di progetti vuoti.

INC-SAU-02. Durante l'esecuzione del test `test_run_program_missing_paths` si manifesta un errore di tipo `_tkinter.TclError`, dovuto a una inizializzazione non deterministica del sottosistema Tkinter all'interno della suite di test. Il comportamento intermittente dell'errore qualifica il test come flaky.

3.1.2 *Integration Test Incidents.* Sono stati eseguiti i test di integrazione presenti nella directory `test/integration_testing`. L'esecuzione ha prodotto esito positivo per tutti i casi di test previsti (6/6), e non sono stati rilevati incidenti a questo livello di testing.

3.1.3 *System Test Incidents.* I test di sistema dello Static Analysis Tool sono stati eseguiti mediante CLI sui casi di test presenti in `test/system_testing/` (TC1-TC21), automatizzando l'esecuzione tramite uno script di supporto che invoca `cli.cli_runner` e produce un file di riepilogo in formato CSV.

L'esecuzione ha evidenziato due categorie di esiti non riconducibili a fault di detection:

- **SKIP_NO_PY:** casi di test privi di file Python (`.py`) e quindi non analizzabili;
- **PASS_NO_OUTPUT:** casi eseguiti correttamente con 0 code smell rilevati, per i quali non viene generato `overview.csv` (assenza di risultati da consolidare).

Tabella 3.2. Static Analysis Tool - System Test Outcomes

Esito	Test	Osservazione
SKIP_NO_PY	TC1, TC5, TC15, TC21	Directory di input prive di file .py; esecuzione non applicabile.
PASS_NO_OUTP	TC6, TC8, TC18	Esecuzione completata con 0 smell rilevati; non viene prodotto overview.csv in assenza di risultati da consolidare.

Per tutti gli altri casi di test di sistema (TC2–TC4, TC7, TC9–TC14, TC16–TC17, TC19–TC20) l’esecuzione ha prodotto esito positivo e i report attesi sono stati generati.

Gli esiti riportati confermano la correttezza del comportamento del sistema nei casi di test applicabili, senza evidenziare fault funzionali a livello di system testing.

3.2 Web Application

Questa sottosezione riporta gli incidenti rilevati durante l’esecuzione dei test relativi alla Web Application.

3.2.1 *Unit Test Incidents.* Sono stati eseguiti i test unitari del frontend mediante jest (npm test nella directory webapp/).

L’esecuzione ha prodotto esito positivo per tutte le suite e i casi di test previsti (6/6 test suite, 26/26 test), pertanto non sono stati rilevati incidenti a questo livello di testing.

3.2.2 *Integration Test Incidents.* Sono stati eseguiti i test di integrazione presenti nella directory webapp/integration_tests tramite PyTest (python -m pytest -q webapp/integration_tests nella root directory). L’esecuzione ha evidenziato fallimenti dovuti a disallineamenti tra le asserzioni dei test e il formato effettivo delle risposte restituite dal gateway (contract mismatch), e non a eccezioni o crash durante l’esecuzione delle API.

Tabella 3.3. Web Application - Integration Test Outcomes

ID	Test File	Funzione di Test
INC-WAU-01	integration_tests/test_gateway_to_inspector.py	test_gateway_to_static_analysis_no_smell
INC-WAU-02	integration_tests/test_gateway_to_inspector.py	test_gateway_to_static_analysis_with_smell
INC-WAU-03	integration_tests/test_gateway_to_generate_report.py	test_generate_report_valid_data
INC-WAU-04	integration_tests/test_gateway_to_model.py	test_gateway_to_ai_analysis_no_smell
INC-WAU-05	integration_tests/test_gateway_to_model.py	test_gateway_to_ai_analysis_with_smell

INC-WAU-01. Il test fallisce a causa di un disallineamento nel contratto della risposta JSON dell'endpoint `/api/detect_smell_static`. In particolare, la risposta effettiva del gateway include un campo addizionale `success` che non è previsto nell'asserzione del test, il quale utilizza un confronto di uguaglianza stretta sull'intero oggetto JSON restituito.

INC-WAU-02. Analogamente a quanto osservato in *INC-WAU-01*, il test fallisce per un disallineamento nel formato della risposta JSON dell'endpoint `/api/detect_smell_static`. La presenza del campo addizionale `success`, restituito dal gateway insieme alla lista `smells`, provoca un `mismatch` rispetto alla struttura attesa dal test.

INC-WAU-03. Il test fallisce a causa di un disallineamento di tipo nel campo `filename` del report combinato generato dal gateway. Nella risposta effettiva, il valore di `filename` è rappresentato come stringa (ad esempio `"1"`), mentre il test si aspetta un valore numerico (ad esempio `1`), determinando il fallimento dell'asserzione.

INC-WAU-04. Il test fallisce a causa dell'indisponibilità del servizio di analisi basato su modello AI. L'endpoint `/api/detect_smell_ai` restituisce una risposta di errore contenente il campo `detail` con valore `Error during AI analysis`, anziché la struttura di risposta attesa dal test. Il fallimento è imputabile alla dipendenza del test dalla corretta configurazione e disponibilità del servizio esterno basato su Ollama.

INC-WAU-05. Analogamente a quanto osservato in *INC-WAU-04*, il test fallisce a causa dell'impossibilità di completare correttamente l'analisi AI tramite l'endpoint `/api/detect_smell_ai`. In assenza di una risposta valida dal modello, il gateway restituisce un messaggio di errore, determinando il fallimento dell'asserzione del test. Anche in questo caso, l'incidente è riconducibile a una dipendenza infrastrutturale esterna e non a un errore logico o di contratto nell'implementazione del gateway.

3.2.3 End-to-End Test Incidents. I test end-to-end della Web Application sono stati eseguiti in modalità headless tramite Cypress (`npm run cypress` nella root directory), utilizzando Electron. L'esecuzione ha prodotto esito positivo per 2 specifiche su 4 (`homepage.spec.ts`, `report.spec.ts`), mentre sono stati rilevati incidenti in due flussi utente.

Tabella 3.4. Web Application - End-to-End Test Outcomes

ID	Spec	Test
INC-WAE-01	<code>cypress/e2e/upload-project.spec.ts</code>	should handle API failure gracefully
INC-WAE-02	<code>cypress/e2e/upload-python.spec.ts</code>	should display progress and results in AI mode

INC-WAE-01. Il test fallisce in modo non deterministico con un'eccezione (`TypeError: Cannot read properties of null (reading 'get')`). L'errore emerge durante la validazione del comportamento in caso di fallimento dell'API di analisi statica nella pagina *Upload Project* e interrompe l'esecuzione delle asserzioni previste.

Il comportamento osservato è riconducibile a una condizione di race tra l'intercettazione della richiesta HTTP simulata e il rendering condizionale dei messaggi di errore nell'interfaccia, causando flakiness del test.

INC-WAE-02. Il test fallisce per timeout dell'asserzione: Cypress non rileva entro il tempo massimo la presenza dell'elemento atteso (Smell #1) nella modalità *AI-Based* della pagina *Upload Python*. L'incidente è compatibile con un disallineamento tra risposta attesa e comportamento effettivo in modalità AI.

4 Incident Resolution

In questa sezione sono documentate le correzioni applicate agli incidenti emersi durante le attività di test, organizzate per componente e livello di testing.

4.1 Static Analysis Tool

4.1.1 Unit Test Fixes.

INC-SAU-01. Il test `test_analyze_project_empty_directory` è stato modificato per riflettere correttamente il comportamento del metodo `analyze_project` in presenza di una directory priva di file Python.

In particolare, l'asserzione è stata aggiornata utilizzando `pytest.raises(ValueError)`, verificando esplicitamente il sollevamento dell'eccezione attesa e allineando il test al flusso di controllo effettivo dell'implementazione.

INC-SAU-02. L'errore `_tkinter.TclError` è stato risolto rendendo deterministica l'inizializzazione di Tkinter all'interno dei test GUI.

È stata introdotta un'unica istanza di `Tk()` condivisa a livello di sessione di test, riutilizzata da tutti i test della GUI, limitando la creazione e distruzione ai soli widget. Questa modifica ha eliminato la flakiness del test e stabilizzato l'esecuzione della suite.

4.2 Web Application

4.2.1 Integration Test Fixes.

INC-WAU-01. Il test `test_gateway_to_static_analysis_no_smell` è stato corretto aggiornando l'asserzione sulla risposta JSON per riflettere il contratto effettivo dell'endpoint `/api/detect_smell_static`.

In particolare, la risposta attesa è stata modificata includendo il campo `success: True`, restituito dal gateway insieme al messaggio `smells`. La logica applicativa non è stata modificata.

INC-WAU-02. Analogamente a quanto effettuato per *INC-WAU-01*, il test `test_gateway_to_static_analysis_with_smell` è stato aggiornato per includere il campo `success: True` nella risposta attesa.

La correzione allinea il test al contratto reale dell'endpoint `/api/detect_smell_static`, mantenendo invariata l'asserzione sul contenuto della lista `smells`.

INC-WAU-03. Il test `test_generate_report_valid_data` è stato corretto allineando il tipo del campo `filename` nella risposta attesa.

In particolare, il valore atteso è stato aggiornato da tipo numerico a stringa ("1"), coerentemente con il formato restituito dal gateway e dal servizio di report. Anche in questo caso, la correzione ha interessato esclusivamente il test di integrazione.

INC-WAU-04. L'incidente non è stato risolto in quanto dipendente dalla disponibilità e configurazione del servizio di analisi AI basato su modello esterno. Il fallimento del test è attribuibile a una dipendenza infrastrutturale e non a un errore di implementazione del gateway o dei test.

INC-WAU-05. Analogamente a *INC-WAU-04*, l'incidente non è stato oggetto di intervento. Il fallimento del test è causato dall'impossibilità di ottenere una risposta valida dal servizio AI esterno e non riflette un fault logico nell'applicazione o nel contratto dell'API.

4.2.2 *End-to-End Test Fixes.*

INC-WAE-01. L'incidente è stato risolto correggendo la struttura del test `should handle API failure gracefully` per allinearlo a un modello di esecuzione deterministico.

In particolare, sono stati rimossi costrutti asincroni JavaScript (`async/await`, `Promise.all`). Il caricamento dei file è stato reso sequenziale tramite `cy.fixture().then(...)` e le asserzioni sono state eseguite solo dopo la conferma dell'avvenuta intercettazione della richiesta HTTP e della verifica del relativo `statusCode`.

La modifica ha eliminato le condizioni di race responsabili della flakiness del test, rendendo il comportamento stabile e ripetibile.

INC-WAE-02. L'incidente non è stato risolto in quanto relativo alla modalità di analisi AI-Based. Il fallimento del test dipende dalla risposta del servizio di analisi AI esterno e dal suo comportamento non deterministico.

Analogamente a quanto già osservato nei test di integrazione della Web Application, l'incidente è attribuibile a una dipendenza infrastrutturale esterna e non a un errore logico nell'applicazione o nei test end-to-end.