

---

# Change Requests Document

Versione 1.0

CodeSmile

Team Members:

Matteo Ercolino — 0522501462

Simone Silvestri — 0522501419

Repository: [GitHub link](#)

Anno Accademico 2024/2025

# Indice

<b>1</b>	<b>Introduzione</b>	<b>3</b>
1.1	Panoramica del Sistema . . . . .	3
1.1.1	Analisi Statica del Codice . . . . .	3
1.1.2	Web App Interattiva . . . . .	3
1.1.3	CLI (Command Line Interface) . . . . .	3
1.1.4	GUI (Graphic User Interface) . . . . .	4
1.2	Scopo del Documento . . . . .	4
<b>2</b>	<b>Change Requests</b>	<b>5</b>
2.1	Change Request #1: Dashboard interattiva nella Web App . . . . .	5
2.1.1	Descrizione . . . . .	5
2.1.2	Dettagli . . . . .	5
2.1.3	Piano di Implementazione . . . . .	5
2.2	Change Request #2: Integrazione con strumenti di CI/CD (GitHub Actions) . .	5
2.2.1	Descrizione . . . . .	5
2.2.2	Dettagli . . . . .	5
2.2.3	Piano di Implementazione . . . . .	5
2.3	Change Request #3: Aggiunta di una Modalità “Quick Scan” per Analisi Veloci	6
2.3.1	Descrizione . . . . .	6
2.3.2	Dettagli . . . . .	6
2.3.3	Piano di Implementazione . . . . .	6
<b>3</b>	<b>Tracciamento delle Modifiche</b>	<b>7</b>
3.1	Modifiche Approvate . . . . .	7
3.2	Modifiche in Attesa . . . . .	7

# 1 Introduzione

Il presente documento illustra le principali richieste di modifica (*Change Requests*) per il sistema **CodeSmile**. Tale sistema è concepito per individuare e analizzare *code smells* in progetti software, con particolare attenzione a quelli legati al *Machine Learning* e all'Intelligenza Artificiale.

## 1.1 Panoramica del Sistema

CodeSmile è uno strumento avanzato progettato per il rilevamento e l'analisi dei *code smells* nei progetti software, con un'attenzione particolare ai problemi specifici del *Machine Learning*. Il sistema si compone di diversi moduli che lavorano insieme per garantire un'analisi efficace e dettagliata del codice.

### 1.1.1 Analisi Statica del Codice

L'analisi statica del codice è uno dei pilastri fondamentali di CodeSmile. Essa si basa sull'uso di *Abstract Syntax Trees* (AST) per esaminare la struttura del codice senza doverlo eseguire. CodeSmile è in grado di individuare specifici *ML-specific Code Smells*, tra cui ad esempio:

- Uso errato delle API *in-place* di Pandas;
- Mancata liberazione della memoria durante i cicli di addestramento;
- Uso improprio dei metodi `forward()` nei modelli PyTorch;
- Errori nelle chiamate alle funzioni di librerie di Machine Learning come TensorFlow, PyTorch e NumPy.

### 1.1.2 Web App Interattiva

Per facilitare l'utilizzo del sistema, CodeSmile include una Web App sviluppata con *React* e *Node.js*. Tramite questa interfaccia grafica, gli utenti possono:

- Caricare file o interi progetti da analizzare;
- Visualizzare in tempo reale i risultati dell'analisi;
- Consultare una dashboard con metriche avanzate e suggerimenti di correzione;
- Esportare report in PDF.

### 1.1.3 CLI (Command Line Interface)

Per gli utenti più avanzati, CodeSmile offre una interfaccia a riga di comando (CLI) che consente di:

- Eseguire analisi su singoli file o interi progetti;
- Specificare directory di input e output per i risultati;
- Configurare opzioni avanzate come l'esecuzione parallela;

### 1.1.4 GUI (Graphic User Interface)

Oltre alla Web App e alla CLI, CodeSmile fornisce anche una GUI basata su *Tkinter*, che rappresenta un'interfaccia più semplice per l'esecuzione delle analisi. Tramite la GUI, gli utenti possono:

- Selezionare file o cartelle da analizzare;
- Configurare le opzioni di esecuzione tramite menu intuitivi;

## 1.2 Scopo del Documento

Lo scopo di questo documento è raccogliere, descrivere e tracciare le *Change Requests* relative a CodeSmile. Le *Change Requests* riguardano sia migliorie di tipo **perfettivo** (focalizzate sul miglioramento delle funzionalità esistenti), sia modifiche di tipo **adattivo** (orientate all'integrazione con nuovi strumenti o ambienti). Ogni richiesta viene presentata con una descrizione, una tabella di dettagli chiave e un piano di implementazione indicativo. L'obiettivo finale è di fornire una visione chiara e completa del processo di evoluzione del sistema.

## 2 Change Requests

### 2.1 Change Request #1: Dashboard interattiva nella Web App

#### 2.1.1 Descrizione

Questa richiesta si propone di aggiungere una *dashboard interattiva* nella Web App di CodeSmile, in modo che gli utenti possano visualizzare:

- Metriche avanzate sui code smells rilevati;
- Grafici avanzati.

L'obiettivo è migliorare la **visibilità** e la **comprensione** dei problemi, aiutando gli sviluppatori a intervenire rapidamente e in modo più efficace.

#### 2.1.2 Dettagli

Tipo	Perfettiva
Priorità	Alta
Data di Presentazione	1 maggio 2025
Stato	Approvata

#### 2.1.3 Piano di Implementazione

1. Progettare il layout della dashboard (metriche e grafici).
2. Integrare i dati provenienti dal motore di analisi (API Gateway).
3. Implementare visualizzazioni interattive (ad esempio, usando librerie di grafica).
4. Validare il corretto funzionamento con repository di test.

### 2.2 Change Request #2: Integrazione con strumenti di CI/CD (GitHub Actions)

#### 2.2.1 Descrizione

Lo scopo è permettere l'esecuzione automatica di CodeSmile direttamente nelle pipeline di integrazione continua (CI) e di delivery continuo (CD). Questo consentirà un **feedback rapido** sui code smells ogni volta che viene effettuato un push o aperta una pull request.

#### 2.2.2 Dettagli

Tipo	Adattiva
Priorità	Alta
Data di Presentazione	1 maggio 2025
Stato	Approvata

#### 2.2.3 Piano di Implementazione

1. Aggiornare la documentazione con le istruzioni di setup per CI/CD.

2. Preparare script di configurazione per GitHub Actions.
3. Aggiungere l'esecuzione di CodeSmile nelle pipeline per ogni commit.
4. Automatizzare la generazione dei report con link diretti ai log degli errori.
5. Validare su repository di prova.

## 2.3 Change Request #3: Aggiunta di una Modalità “Quick Scan” per Analisi Veloci

### 2.3.1 Descrizione

Questa richiesta introduce una modalità di analisi rapida che consenta agli sviluppatori di controllare solamente i file modificati di recente (ad esempio, ultimo commit Git), senza dover eseguire un'analisi completa dell'intero progetto.

### 2.3.2 Dettagli

<b>Tipo</b>	Perfettiva
<b>Priorità</b>	Alta
<b>Data di Presentazione</b>	1 maggio 2025
<b>Stato</b>	Approvata

### 2.3.3 Piano di Implementazione

1. Implementare una funzione per rilevare i file modificati nell'ultimo commit (es. tramite comandi Git).
2. Integrare la nuova funzione di *Quick Scan* nel motore di analisi, in modo da analizzare solo i file effettivamente cambiati.
3. Aggiornare il CLI per consentire l'opzione di *analisi rapida* tramite parametro specifico.
4. Validare il funzionamento con repository di esempio, confrontando i risultati con l'analisi completa.

## 3 Tracciamento delle Modifiche

### 3.1 Modifiche Approvate

ID	Titolo	Stato
CR-01	Dashboard Interattiva nella Web App	Approvata
CR-02	Integrazione con strumenti di CI/CD	Approvata
CR-03	Aggiunta di una Modalità “Quick Scan” per Analisi Veloci	Approvata

### 3.2 Modifiche in Attesa

ID	Titolo	Stato	Data Stimata
-	-	-	-