

UNIVERSITÀ DEGLI STUDI DI SALERNO

Dipartimento di Informatica



Corso di Laurea Magistrale in Informatica

Change request per il progetto

csDetector

Professore
Andrea De Lucia

Tutor
Gianmario Voria

Repository
<https://github.com/gianwario/csDetector>

Studente
Riccardo Imparato 0522501613

Forked repository
<https://github.com/r4004/csDetector>

ANNO ACCADEMICO 2023/2024

Sommario

1 Contesto del progetto.....	3
2 Analisi di csDetector	3
3 Obiettivi del progetto.....	3
4 Reverse engineering	4
5 Proposte di modifica.....	4
5.1 Refactoring generale.....	4
5.2 Evoluzione del codice.....	4
6 Analisi dell'impatto	4
7 Implementazione delle modifiche.....	5
7.1 Refactoring generale.....	5
7.2 Evoluzione del codice.....	5
7.3 Nuova matrice delle dipendenze	6
8 Conclusioni	6

1 Contesto del progetto

In recenti studi di ingegneria del software, la comunità ha cercato di indentificare schemi comportamentali, introducendo quindi il concetto di “community smell”.

I community smell sono degli schemi organizzativi e socio-tecnici subottimali. Molti dei project manager non sono a conoscenza di questo concetto, ed inoltre non esistono strumenti automatici per rilevarli.

2 Analisi di csDetector

Il csDetector è uno strumento per rilevare automaticamente i community smell, che utilizza il machine learning.

Poiché è uno strumento che è stato creato per la ricerca, possiede diversi problemi, come un’elevata difficoltà di installazione, la scarsa affidabilità e la scarsa manutenibilità.

3 Obiettivi del progetto

Per rendere il progetto più avvicinabile, si vuole eseguire principalmente un lavoro di refactoring che mira ad aumentare la manutenibilità, per semplificare successivi sviluppi e migliorare la qualità del codice.

4 Reverse engineering

Prima di fare delle proposte di modifica, si esegue una semplice fase di reverse engineering in cui si crea una matrice delle dipendenze.

Module	Id	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
aliasWorker	0			1			1													1					
authorAliasExtractor	1			1			2									1				1					
cadocsLogger	2																								
centralityAnalysis	3			1			1											1		1					
commitAnalysis	4			1			1											1		1					
configuration	5																								
csDetector	6			1									1												
csDetectorAdapter	7							1																	
csDetectorWebService	8								1																
custmException	9																								
devAnalysis	10			1			1																		
devNetwork	11	1		1	1	1	1				1	1			1	1	1		1						
perspectiveAnalysis	12						1																		
politenessAnalysis	13						1				1							1							
repoLoader	14			1			1																		
smelDetection	15			1			1																		
statsAnalysis	16																								
tagAnalysis	17			1			1												1						
utils	18																								
graphqlAnalysis.	19																								
graphqlAnalysisHelper																									
graphqlAnalysis.	20			1	1		1				1			1				1							
issueAnalysis																									
graphqlAnalysis.	21			1	1		1							1				1							
prAnalysis																									
graphqlAnalysis.	22			1			1											1							
releaseAnalysis																									
webService.	23								1																
csDetectorWebService																									

5 Proposte di modifica

5.1 Refactoring generale

Come già anticipato, sarà svolto un refactoring generale per aumentare la manutenibilità, correggere qualche errore, e prevenire possibili problemi e difficoltà in sviluppi futuri.

5.2 Evoluzione del codice

Sarà implementata una nuova funzionalità che permette di selezionare differenti branch e verrà aggiunta un'interfaccia basilare per il web. Inoltre, gli sviluppatori avranno la possibilità di usare costanti segrete.

6 Analisi dell'impatto

Dopo aver condotto uno studio sulle modifiche proposte, non ci si aspetta nessun impatto sulle dipendenze, nonostante la pianificazione dell'eliminazione di una classe (da cui le altre classi non dipendono).

7 Implementazione delle modifiche

Come già previsto, le modifiche non hanno provocato cambi nelle dipendenze. Per trasparenza, verrà comunque mostrata la nuova matrice delle dipendenze alla fine di questo capitolo, riportando anche la classe eliminata.

7.1 Refactoring generale

Il refactoring generale comprende sia cambi che non hanno conseguenze sul programma, sia vere e proprie correzioni del codice che altrimenti avrebbero bloccato l'esecuzione. Le modifiche implementate sono:

- Refactoring per la leggibilità.
- Correzione di molti bug riguardanti la creazione di cartelle e gli errori “too many values to unpack”.
- Correzione dei namespace conflict.
- Correzione di codice deprecato.
- Eliminazione di una classe duplicata.

7.2 Evoluzione del codice

L'evoluzione del codice comprende nuove funzionalità e l'aggiunta dei test. Le modifiche implementate sono:

- Interfaccia grafica di base ottenuta attraverso il rendering di un template HTML.
- Possibilità ed obbligo di fornire il branch desiderato.
- Possibilità per gli sviluppatori di poter utilizzare delle costanti segrete.
- Propagazione degli errori allo scopo di rendere i test possibili.
- Aggiornamento di tutti i test poiché si riferivano a versioni molto vecchie del programma.
- Nuovi test.

7.3 Nuova matrice delle dipendenze

Module	Id	0	1	2	3	4	5	6	7	-	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
aliasWorker	0			1			1													1					
authorAliasExtractor	1			1			2									1				1					
cadocsLogger	2																								
centralityAnalysis	3			1			1											1		1					
commitAnalysis	4			1			1											1		1					
configuration	5																								
csDetector	6			1									1												
csDetectorAdapter	7							1																	
csDetectorWebService	-								-																
custmException	9																								
devAnalysis	10			1			1																		
devNetwork	11	1		1	1	1	1				1	1			1	1	1		1						
perspectiveAnalysis	12						1																		
politenessAnalysis	13						1			1								1							
repoLoader	14			1			1																		
smelDetection	15			1			1																		
statsAnalysis	16																								
tagAnalysis	17			1			1											1							
utils	18																								
graphqlAnalysis.	19																								
graphqlAnalysisHelper																									
graphqlAnalysis.	20			1	1		1			1			1					1							
issueAnalysis																									
graphqlAnalysis.	21			1	1		1						1					1							
prAnalysis																									
graphqlAnalysis.	22			1			1											1							
releaseAnalysis																									
webService.	23								1																
csDetectorWebService																									

8 Conclusioni

Allo scopo di migliorare la qualità e manutenibilità del codice, si è svolto un lavoro di revisione, refactoring ed evoluzione.

Le modifiche implementate hanno aumentato la leggibilità, risolto bug critici, migliorato la sicurezza e la gestione degli errori, aggiunto un'interfaccia grafica web, aggiunto la funzionalità di selezione del branch e reintrodotta i test.

In conclusione, c'è stato un notevole contributo, soprattutto nei confronti dei futuri sviluppatori che continueranno il lavoro su questo progetto.