
Ingegneria, Gestione ed Evoluzione del Software - Change Request Infozilla

Componenti Gruppo:

Gerardo Festa - 0522501452 - g.festa22@studenti.unisa.it

Alessandra Parziale - 0522501422 - a.parziale8@studenti.unisa.it

Saverio Napolitano - 0522501400 - s.napolitano44@studenti.unisa.it

Proposte change request

1. Risoluzioni errori estrazione Patch
2. Aggiunta formato Unified Diff
3. Risoluzioni errori estrazione Enumeration
4. Export in diversi formati
5. Lancio dell'analisi su URL di una issue fornito in input
6. Lancio dell'analisi su tutte le issue di una repository Github con URL fornito in input

1 Risoluzione errori estrazione Patch (Problematica riscontrata durante il testing)

Situazione attuale:

1. Attualmente, il processo di identificazione si basa sulla ricerca dell'header presente nella patch all'interno del testo dato in input. Tuttavia, questa metodologia di identificazione si è dimostrata non corretta a causa della differenza di formattazione tra l'header della patch e il testo circostante.
 2. Attualmente, il sistema esegue il match considerando la prima patch, ignorando le patch successive con lo stesso header. Questo comporta problemi di riconoscimento e applicazione delle patch multiple.
 3. Un ulteriore problema risiede nella formattazione del testo degli hunk, in cui la presenza di "\n\n" viene erroneamente interpretata come la fine di uno hunk valido. Tuttavia, questa interpretazione può portare all'eliminazione di elementi validi all'interno dell'hunk.
-

Modifica: La modifica punta a risolvere le problematiche riscontrate per ristabilire il corretto funzionamento di questa componente.

Impatto: Per fare ciò, è necessario modificare PatchParser e FilterPatches con impatto che potrebbe estendersi anche ad altri filtri.

2 Aggiunta formato Unified Diff (Problematica riscontrata durante il testing)

Situazione attuale: Attualmente Infozilla non è in grado di riconoscere il formato Unified Diff, tra i più diffusi in quanto è quello fornito da Git di default, e funziona solo con Unified Diff With Index Line.

Modifica: La modifica punta a rendere il sistema compatibile con questo nuovo formato.

Impatto: Per fare ciò, è necessario modificare la classe PatchParser con impatto che potrebbe estendersi anche ai filtri successivi (FilterStackTraces FilterSourceCodeJAVA, FilterEnumeration)

3 Risoluzione errori estrazione Enumeration (Problematica riscontrata durante il testing)

Situazione attuale:

1. Conteggio Errato delle Enumerazioni della Stessa Tipologia: L'applicazione non riconosce correttamente le enumerazioni della stessa tipologia, in quanto il contatore viene erroneamente azzerato, ma il primo elemento dell'enumerazione successiva è già stato letto. Ciò comporta che se dopo la prima enumerazione è presente un'altra enumerazione di soli due elementi allora quest'ultima non verrà riconosciuta.
2. Enumerazione Contiene Tutte le Enumerazioni Successive: L'applicazione riconosce le diverse enumerazioni, ma la prima enumerazione contiene tutte le successive, la seconda contiene le successive a quella, e così via.
3. Eliminazione dell'Ultimo Elemento dell'Enumerazione: L'applicazione non elimina l'ultimo elemento dell'enumerazione.

Modifica: La modifica punta a risolvere le problematiche riscontrate per ristabilire il corretto funzionamento di questa componente.

Impatto: La seguente modifica andrà ad impattare la classe Main contenuta nel package cli. Ci aspettiamo che questa sia l'unica classe impattata dalla modifica: l'automatismo infatti, prevede di utilizzare il tool su tutte le issue, ma con esecuzione sequenziale (una issue per volta).

4 Export in diversi formati

Situazione attuale: Attualmente Infozilla è in grado di esportare solo in formato xml

Modifica: La modifica punta ad estendere questa funzionalità aggiungendo i seguenti formati: Json, csv e xls

Impatto: Per fare ciò, è necessario aggiungere un'opzione per permettere la scelta tra i vari formati.

La seguente modifica andrà ad impattare la classe Main contenuta nel package "cli": La procedura di output, infatti, è contenuta in questa classe, quindi ci aspettiamo che questa sia l'unica impattata dalla modifica.

A seguito della modifica, l'utente potrà decidere di esportare in diversi formati, pur garantendo il funzionamento attuale, ovvero, laddove non venga indicato nulla, il tool esporterà in xml.

5 Lancio dell'analisi su un URL fornito in input

Situazione attuale: Attualmente Infozilla è in grado di identificare ed estrarre snippet di codice Java, Stacktrace in Java, patches, enumerazioni e Talkback traces solo attraverso un file di testo in input.

Modifica: La modifica punta ad estendere questa funzionalità fornendo in input un url di una issue Github.

Impatto: Per fare ciò, è necessario aggiungere un'opzione alla command line interface per specificare la modalità di utilizzo (file locale o url) desiderata.

La command line interface è contenuta nella classe Main nel package cli, quindi ci aspettiamo che questa sia l'unica classe impattata dalla modifica; il tool, infatti, dovrebbe funzionare allo stesso modo, con l'aggiunta della funzionalità che dall'url porti il testo della issue in locale.

6 Lancio dell'analisi su tutte le issue di una repository Github con URL fornito in input

Situazione attuale: Similmente a quanto descritto nella change request numero 5, Infozilla funziona su un singolo file testuale locale

Modifica: La modifica punta ad automatizzare l'esecuzione del tool su tutte le issue di una repository github.

Inoltre sarà possibile parametrizzare per ottenere le issue solo in un determinato range di tempo (di apertura e di chiusura), per assegnatario, per labels e per stato (aperto/chiuso).

Impatto: La seguente modifica andrà ad impattare la classe Main contenuta nel package cli. Ci aspettiamo che questa sia l'unica classe impattata dalla modifica: l'automatismo infatti, prevede di utilizzare il tool su tutte le issue, ma con esecuzione sequenziale (una issue per volta).
