



Laurea Magistrale in informatica
Università di Salerno

Report Finale

Team member

Saverio Napolitano - 0522501400

Gerardo Festa - 0522501452

Alessandra Parziale - 0522501422

<https://github.com/GerardoFesta/infozilla>

Sommario

1. Introduzione.....	3
2. Esito delle modifiche.....	3
3. Deviazioni da quanto pianificato.....	4
4. Lessons Learned.....	4

1.Introduzione

- **Contesto:** Questo report presenta i risultati del test di regressione effettuato per valutare le prestazioni del nostro sistema.
- **Descrizione del Sistema:** **InfoZilla** è una libreria e un tool che si occupa dell'estrazione di dati strutturati da fonti di dati non strutturati come e-mail, discussioni e segnalazioni di bug.

InfoZilla è in grado di identificare i seguenti elementi:

- **Java Source Code Regions:**
Codice sorgente Java, di piccole o medie dimensione, che è utilizzato per illustrare un problema, descriverne il contesto di programmazione in cui si è verificato o rappresentare un possibile esempio di soluzione.
- **Patches:**
Le patch rappresentano una piccola porzione di software progettata per aggiornare problemi in un programma. Il formato maggiormente impiegato per le patch è il diff unificato (utilizzato per illustrare le modifiche apportate ad un file rispetto alla sua versione precedente).
- **Enumerations and Itemizations:**
Vengono utilizzate per elencare elementi o presentare una sequenza di azioni che lo sviluppatore deve intraprendere al fine di risolvere un problema. Esse danno una struttura alla descrizione di un problema e agevolano la comprensione.
- **Java Stacktraces:**
Elencano i frame dello stack attivi nella pila delle chiamate durante l'esecuzione di un programma. Vengono utilizzati per assistere la correzione di errori fornendo delle indicazioni sull'origine di un problema.
- **Talkback Traces:**
Dettagliano i contesti di crash e le informazioni sull'ambiente quando viene rilevato un problema.

2.Esito delle modifiche

Dopo le modifiche apportate, il tool InfoZilla ha subito significativi miglioramenti rispetto alla sua versione iniziale.

Due di questi miglioramenti si concentrano principalmente sul riconoscimento delle *Patch* e delle *Enumeration*.

Le modifiche relative alla **CR_1** hanno permesso di risolvere i problemi legati al filtro delle patch e hanno portato a diverse migliorie. Queste includono una maggiore precisione nell'identificazione delle patch all'interno del testo di input, con conseguente eliminazione delle patch dal file "Cleaned" quando necessario. Inoltre, è stata implementata una gestione più efficiente delle patch multiple, garantendo che tutte vengano riconosciute e applicate correttamente, e un riconoscimento accurato degli "hunk", evitando la rimozione erronea di elementi validi.

Le modifiche legate alla **CR_3** hanno permesso di risolvere i problemi associati al filtro Enumeration. Questo ha portato a un conteggio preciso delle enumerazioni della stessa tipologia, a una corretta formattazione delle enumerazioni senza l'inclusione di tutte le successive e all'eliminazione corretta dell'ultimo elemento.

Inoltre, grazie alle modifiche apportate nella **CR_2**, ora il tool è in grado di riconoscere in modo accurato le patch in formato Unified Diff, evitando così errori nel conteggio degli *"hunk"* causati dalle patch non riconosciute.

Un'altra modifica di rilievo è stata implementata nella **CR_4**, che ha reso il tool più flessibile per gli utenti e compatibile con diverse applicazioni e strumenti di gestione dati. Questo è stato ottenuto attraverso la possibilità di esportare dati in formati più comuni.

Infine, l'implementazione della **CR_5** ha reso il tool più comodo per gli utenti, consentendo loro di automatizzare l'estrazione di informazioni da repository con molte issue.

3.Deviazioni da quanto pianificato

Durante l'esecuzione del progetto, si sono verificate alcune variazioni rispetto alla pianificazione iniziale, che hanno richiesto l'adozione di un approccio differente. Queste variazioni sono emerse dopo aver eseguito un test preliminare.

Il problema principale che abbiamo riscontrato riguardava la presenza di vari bug nei processi di parsing, il che ci ha portato a modificare la change request originale. Nella versione modificata infatti, abbiamo mantenuto gli obiettivi originali considerati di maggiore importanza (**CR_2**, **CR_4**, **CR_5**) e abbiamo aggiunto due ulteriori richieste (**CR_1**, **CR_3**) per affrontare i bug individuati, in quanto la risoluzione di questi bug è stata considerata fondamentale per il progresso del progetto.

4.Lessons Learned

Durante il processo di sviluppo di questo progetto, abbiamo imparato molte lezioni preziose, tra cui:

- **Riconoscere l'importanza dei test:** Dato che inizialmente, il nostro sistema non include alcun tipo di test, abbiamo appreso quanto essi siano fondamentali per individuare gli errori.
- **Migliorare la gestione e utilizzo di GitHub:** Abbiamo affinato le nostre competenze nell'utilizzo di GitHub, sfruttando aspetti come la gestione dei branch che ci ha permesso di gestire complessivamente il progetto in modo più efficiente, pianificare le attività e distribuire il lavoro in modo più adeguato.