



Laurea Magistrale in informatica
Università di Salerno

Master Test Plan

Team member

Saverio Napolitano - 0522501400

Gerardo Festa - 0522501452

Alessandra Parziale - 0522501422

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

Sommario

| | |
|---------------------------------------|---|
| Obiettivi e Contesto:..... | 3 |
| Strategie di Test:..... | 3 |
| Attività di Test:..... | 3 |
| Classi Testate:..... | 4 |
| Classi/Package Escluse dai Test:..... | 4 |
| Coverage:..... | 4 |

Obiettivi e Contesto:

- **Nome del Progetto:** InfoZilla
- **Descrizione del Prodotto:** Una libreria e tool che si occupa dell'estrazione di dati strutturati da fonti di dati non strutturati come e-mail, discussioni e segnalazioni di bug.
- **Obiettivi del Test:** Garantire la qualità del prodotto e verificare la corretta funzionalità.

Strategie di Test:

- **Testing di Unità:** Il testing sarà condotto sulla base di:
 - Coverage, dunque in maniera white-box. La coverage servirà a guidare l'attività di testing; cercheremo di coprire i branch scoperti, selezionando input appropriati per raggiungerli.
 - Regular Expression/Patterns: i filtri utilizzano pattern con i quali fanno il match delle stringhe in input. Cercheremo di fornire quanti più input che possano mettere in difficoltà il sistema in tal senso, utilizzando delle stringhe che siano realistiche, ma che possano essere male interpretate dalle regex.

Ad ogni esecuzione di un caso di test, verifichiamo che il contenuto restituito dall'esecuzione dei filtri sia uguale a quello atteso.

- **Testing di Integrazione:** valuteremo le combinazioni di interazione tra i vari filtri (2-3-tutti), in quanto l'output di un filtro è l'input del successivo.
- **Testing di Sistema:** La strategia presa in considerazione è il **category partitioning**; pertanto, suddividiamo il campo dei dati in ingresso in classi di casi di test, identifichiamo i parametri relativi al file di input, ovvero le informazioni che lo strumento è in grado di estrarre, e successivamente le opzioni possibili per i parametri del file di input.

Attività di Test:

- Creazione dei Casi di Test
- Esecuzione dei Test
- Report dei Risultati
- Gestione dei Difetti

Classi Testate:

- La classe *FilterPatches* sarà sottoposta a test di Unità.
- La classe *PatchParser* sarà sottoposta a test di Unità.
- La classe *FilteSourceCodeJava* sarà sottoposta a test di Unità.
- La classe *FilterEnumeration* sarà sottoposta a test di Unità.
- La classe *FilterStackTraceJAVA* sarà sottoposta a test di Unità.
- La classe *FilterChainEclipse* sarà sottoposta a test di Unità, che permetterà di sottoporre le classi elencate precedentemente a test di Integrazione.

Classi/Package Escluse dai Test:

- La classe *FilterChainEclipsePS* sarà esclusa dai test poiché è considerata esterna al contesto.
- La classe *FilterChainMozilla* sarà esclusa dai test poiché è considerata esterna al contesto.
- La classe *FilterTalkBack* sarà esclusa dai test poiché non è utilizzata.
- Il package *bugreports* sarà escluso dai test poiché non utilizzato.
- Il package *datasources* sarà escluso dai test poiché non utilizzato.

Coverage:

La percentuale di copertura della branch per ciascun filtro al momento è pari allo 0% ci poniamo l'obiettivo di raggiungere una percentuale \geq al 70%.