Università degli Studi di Salerno

Corso di Ingegneria, Gestione ed Evoluzione Software

TEMEVI

Test Metrics Visualisation

TEST PLAN

Versione 1.0

TOP MANAGER:
Prof. Andrea De Lucia

PARTECIPANTI:
Giosuè Sulipano
Gianluca Di Lillo

Revision History

Data	Versione	Descrizione	Autore
18/02/2020	1.0	Stesura del documento	Giosuè Sulipano Gianluca di Lillo

Indice

1. Scopo del documento

Lo scopo di questo documento è quello di illustrare l'approccio, le risorse e lo scheduling delle attività di testing del plugin **TEMEVI**. La fase di testing servirà a verificare il funzionamento effettivo del plugin, controllando che il comportamento atteso specificato nei requisiti di sistema combaci con il comportamento osservato.

2. Panoramica del sistema

Il sistema è stato suddiviso fondamentalmente in due layer:

1) **GUI Layer**, caratterizzato dai sottosistemi AnalysisResults e ConfigUI.
2) **Application Layer**, caratterizzato dai seguenti sottosistemi:
    a) **Actions**;
    b) **StaticProcessor**;
    c) **DynamicProcessor**;
    d) **ConfigurationManager**;
    e) **ReportManager**.

I servizi offerti da ciascun sottosistema saranno oggetto della fase di testing. In particolare, saranno elencate di seguito quali sono le specifiche funzionalità da testare.

3. Funzionalità da testare

Le funzionalità da esercitare durante la fase di testing di sistema sono le seguenti:

1) Avvio dell'analisi dei fattori legati ai test.
2) Visualizzazione di Line e Branch Coverage.
3) Visualizzazione di Mutation Coverage.
4) Visualizzazione di Flaky Tests.
5) Filtro della lista delle classi di test per Test Smell.
6) Filtro dei grafici relativi alle metriche dei Test Smell in base a mese ed anno dei reports.
7) Modifica soglia di rilevamento e di guardia di un Test Smell.

4. Criteri Pass/Fail

I casi di input verranno suddivisi in classi di equivalenza e raggruppati in insiemi in base alle caratteristiche comuni, per i quali sarà sufficiente testare solo l'elemento rappresentativo di tale insieme.
Si considera un *successo* quando il comportamento del test item esercitato con specifici dati di input combacia con quello definito nell'oracolo associato.
Nel caso in cui si presenti una percentuale del 60 % di fallimento dei test cases, si dovrà sospendere il testing finchè non saranno stati risolti i problemi che hanno portato al fallimento dei suddetti.

5. Approccio

L'attività di testing si compone principalmente di tre fasi:

1. Test di Unità,
2. Test di Integrazione,
3. Test di Sistema.

L'obiettivo principale è quello di individuare quanti più errori possibili da correggere nel codice sorgente. L'approccio utilizzato è di tipo black box; il testing, dunque, sarà effettuato ponendo il focus sulle funzionalità del sistema.

5.1 Testing di Unità

È prevista una classe di test per ogni classe del codice sorgente da testare coinvolta nell'esecuzione del plugin e, attraverso l'utilizzo del framework JUnit, verranno testati i singoli metodi. L'obiettivo di questa fase è quello di ottenere una line coverage di almeno 80%.
Verranno utilizzati eventuali mock object per isolare completamente l'unità da esercitare dalle sue dipendenze.

5.2 Testing di Integrazione

In seguito al Testing di Unità sarà effettuato il Testing di Integrazione, il quale prevede anch'esso l'utilizzo di JUnit per la scrittura e l'esecuzione delle classi di test. In questa fase, la strategia utilizzata sarà bottom-up, per verificare che le componenti comunichino con successo.

5.3 Testing funzionale di Sistema

In quest'ultima fase si valuterà l'esecuzione effettiva del plugin, verificando che il comportamento del sistema combaci con il comportamento atteso o previsto.
I casi di test consistono nell'esecuzione dell'interfaccia grafica del plugin e nell'analisi di ciò che viene effettivamente visualizzato in base ai diversi input.

6. Risorse per il testing

Di seguito sono presentate le risorse necessarie per effettuare il testing:

- **Computer**: necessaria una macchina con sistema operativo Windows 10, non necessariamente connesso ad Internet. Inoltre deve essere correttamente stato installato java 8.
- **Test Tools**: verranno utilizzati JUnit e PowerMock per il test di unità. Il test di integrazione sarà effettuato anche esso con JUnit.

7. Test Cases di Sistema

Funzionalità: **Avvio dell'analisi dei fattori legati ai test**.

Parametro: Progetto da analizzare	
Categorie	**Scelte**
Struttura delle cartelle del progetto [SC]	1. Il progetto rispetta la struttura delle cartelle attesa dal plugin. [property SC1] 2. Il progetto non rispetta la struttura delle cartelle attesa dal plugin. [error]
Presenza di codice sorgente di test [TSRC]	1. Il progetto contiene del codice sorgente di test nella folder attesa. [if SC1] [property TSRC1] 2. Il progetto non contiene codice sorgente di test. [if SC1] [error]
Esistenza di classi di test compilate [TCOMP]	1. Il progetto contiene classi di test compilate nella folder attesa. [if SC1 && TSRC1 && AD1] [property TCOMP1] 2. Il progetto non contiene classi di test compilate. [if SC1 && TSRC1 && AD1] [error]

Parametro: Checkboxes analisi da effettuare	
Categorie	**Scelte**
Esecuzione di un tipo di analisi dinamica [AD]	1. L'utente seleziona un tipo di analisi dinamica [if SC1 && TSRC1] [property AD] 2. L'utente non seleziona un tipo di analisi dinamica [if SC1 && TSRC1] [property AD2]

Parametro: JavaLocation (String)	
Categorie	**Scelte**
Formato della stringa JavaLocation [JRE]	1. JRE 8 è installato nel sistema, il plugin ha individuato la sua locazione e l'ha salvata nella stringa JavaLocation [if SC1 && TSRC1 && AD1 && TCOMP1] [property JRE1] 2. JRE 8 non è installato nel sistema e/o il plugin non ha individuato la sua locazione. [if SC1 && TSRC1 && AD1 && TCOMP1] [error]

Codice	**Combinazione**	**Esito**
TC_1.0	SC1, TSRC1, AD1, TCOMP1, JRE1	Corretto
TC_1.1	SC1, TSRC1, AD2	Corretto
TC_1.2	SC1, TSRC1, AD1, TCOMP1, JRE2	Errore
TC_1.3	SC1, TSRC2	Errore
TC_1.4	SC1, TSRC1, AD1, TCOMP2	Errore
TC_1.5	SC2	Errore

Funzionalità: **Estrazione e visualizzazione di Line e Branch Coverage**.

Parametro: Previous Coverage Report	
Categorie	**Scelte**
Esistenza del file di report dell'esecuzione precedente della classe [CR]	1. Il report dell'esecuzione precedente della classe esiste. [property CR1] 2. Non è stata effettuata alcuna analisi in precedenza per la classe. [property CR2]

Parametro: Test Class	
Categorie	**Scelte**
Valore delle coverage precedenti [VC]	1. Le coverage precedenti sono più basse delle correnti. [if CR1] [property VC1] 2. Le coverage precedenti sono più alte delle correnti. [if CR1] [property VC2] 3. Le coverage precedenti sono uguali alle correnti. [if CR1] [property VC3]

Codice	**Combinazione**	**Esito**
TC_2.0	CR1	Corretto
TC_2.1	CR1, VC1	Corretto
TC_2.2	CR1, VC2	Corretto
TC_2.3	CR1, VC3	Corretto

Funzionalità: **Estrazione e visualizzazione di Mutation Coverage**.

Parametro: Mutation Coverage Timeout	
Categorie	**Scelte**
Valore Timeout [VT]	1. Il timeout consente la totale esecuzione dell'analisi della classe [property VT1] 2. Il timeout è troppo breve e non consente il completamento dell'esecuzione dell'analisi della classe [error]

Parametro: Mutation Coverage Report	
Categorie	**Scelte**
Esistenza del file di report [CR]	1. L'analisi del tool è andata a buon fine e il report è stato generato. [property CR] 2. L'analisi del tool non è andata a buon file e/o il report non è stato generato. [error]

Codice	**Combinazione**	**Esito**
TC_3.0	VT2, CR2	Errore
TC_3.1	VT2, CR1	Corretto
TC_3.2	VT1, CR1	Corretto

Funzionalità: **Estrazione e visualizzazione di Flaky Tests**.

Parametro: Test Class	
Categorie	**Scelte**
Numero di flaky tests nella test class [FT]	1. La classe di test non ha alcun flaky test. [property FT1] 2. La classe di test ha almeno un flaky test. [property FT2]

Codice	**Combinazione**	**Esito**
TC_4.0	FT1	Corretto
TC_4.1	FT2	Corretto

Funzionalità: **Filtro della lista delle classi di test per Test Smell**.

Parametro: Test Smell selezionato (Dropdown Menu)	
Categorie	**Scelte**
Filtro abilitato [FILTER]	1. L'utente abilita il filtro selezionando uno smell. [property FILTER1] 2. L'utente non abilita il filtro selezionando l'opzione "ALL" [property FILTER2]
Tipo di Test Smell [TS]	1. L'utente seleziona lo smell "Assertion Roulette". [if FILTER1] [property TS1] 2. L'utente seleziona lo smell "Eager Test". [if FILTER1] [property TS2] 3. L'utente seleziona lo smell "General Fixture". [if FILTER1] [property TS3] 4. L'utente seleziona lo smell "Indirect Testing". [if FILTER1] [property TS4] 5. L'utente seleziona lo smell "Sensitive Equality". [if FILTER1] [property TS5] 6. L'utente seleziona lo smell "Resource Optimism". [if FILTER1] [property TS6]

	7. L'utente seleziona lo smell "Mystery Guest". [if FILTER1] [property TS7]

Codice	**Combinazione**	**Esito**
TC_5.0	FILTER2	Corretto
TC_5.1	FILTER1, TS1	Corretto
TC_5.2	FILTER1, TS2	Corretto
TC_5.3	FILTER1, TS3	Corretto
TC_5.4	FILTER1, TS4	Corretto
TC_5.5	FILTER1, TS5	Corretto
TC_5.6	FILTER1, TS6	Corretto
TC_5.7	FILTER1, TS7	Corretto

Funzionalità: **Visualizzazione e filtro dei grafici relativi alle metriche dei Test Smell in base a mese ed anno dei reports**.

Parametro: File di report	
Categorie	**Scelte**
Esistenza di files di report [RE]	1. Sono presenti files di report storici per la classe selezionata [property RE1] 2. Non esistono files di report storici per la classe selezionata [error]

Codice	**Combinazione**	**Esito**
TC_6.0	RE1	Corretto
TC_6.1	RE2	Errore

Funzionalità: **Modifica soglia di guardia e rilevamento di un Test Smell**.

Parametro: Soglia di guardia / rilevamento	
Categorie	**Scelte**
Formato delle soglie date in input [FS]	1. Il formato delle soglia rispetta il tipo double. [property FS1] 2. Il formato della soglia non rispetta il tipo double. [error]
Valore della soglia data in input [VS]	1. Il valore della soglia è compreso nel range accettato dal plugin. [if FS1] [property VS1] 2. Il valore della soglia supera il valore massimo. [if FS1] [error] 3. Il valore della soglia è al di sotto del valore minimo. [if FS1] [error]
Relazione tra le soglie [RS]	1. La soglia di guardia è maggiore di quella di rilevamento. [if FS1 && VS1] [property RS1] 2. La soglia di guardia è minore di quella di rilevamento. [if FS1 && VS1] [error] 3. La soglia di guardia è uguale a quella di rilevamento. [if FS1 && VS1] [error]

Codice	**Combinazione**	**Esito**
TC_7.0	FS2	Errore
TC_7.1	FS1, VS2	Errore
TC_7.2	FS1, VS3	Errore
TC_7.3	FS1, VS1, RS2	Errore
TC_7.4	FS1, VS1, RS3	Errore
TC_7.5	FS1, VS1, RS1	Corretto