

# UNIVERSITÀ DEGLI STUDI DI SALERNO



## PROGETTO DI INGEGNERIA DEL SOFTWARE II REPOMINER EVOLUTION

---

### Test Incident Report

---

*Autori:*

Matteo MEROLA

Carlo BRANCA

Simone SCALABRINO

Giovanni GRANO

*Supervisore:*

Prof. Andrea DE LUCIA

Test Incident Report  
Versione 1.0

30 giugno 2014

# Revision History

Tabella 1: Tabella delle revisioni del documento

Data	Versione	Descrizione	Autori
19/05/2014	1.0	Versione iniziale	Simone Scalabrino

# Indice

<b>1</b>	<b>Introduzione</b>	<b>3</b>
<b>2</b>	<b>Test Incidents</b>	<b>4</b>
2.1	Metrica per il calcolo del numero medio di volte in cui i file di un package hanno subito cambiamenti . . . . .	5
2.2	Metrica per il calcolo del numero medio di bug-fix di un package	6
2.3	Metrica per il calcolo del numero di autori di commit di un package . . . . .	7

# 1. Introduzione

Il Test Incident Report è un documento che descrive dettagliatamente i test che hanno evidenziato problemi del sistema. In particolare verranno qui inserite tutte le informazioni necessarie affinché tali problemi possano essere risolti.

Questo Test Incident Report è relativo al primo incremento di implementazione del sistema. Sono stati eseguiti i casi di test descritti nelle sezioni che vanno dalla 3.1 alla 3.7 del Test Case Specification. Si veda il Test Log annesso per i dettagli sull'esecuzione dei casi di test.

## 2. Test Incidents

In questa sezione verranno indicati tutti i test incident. Si provvederà a fornire le seguenti informazioni per ogni caso di test:

- Identificativo del caso di test (tracciabile nel Test Case Specification e nel Test Plan)
- Risultato prodotto e risultato atteso
- Input che ha portato alla failure
- Note aggiuntive

Per visualizzare i risultati è sufficiente eseguire la query SQL descritta nella figura 2.1.

```
1 SELECT M.name, M.description, M1.value
2 FROM repominer.metrics AS M, repominer.package_metrics AS
   M1
3 WHERE M.id = M1.metric
4
```

Figura 2.1: Query utilizzata per estrarre i valori relativi alle metriche di package

## 2.1. Metrica per il calcolo del numero medio di volte in cui i file di un package hanno subito cambiamenti

Caso di test	Risultato atteso	Risultato ottenuto
TC 2.2	1	0

### TC 2.2

Passi da eseguire per arrivare alla failure:

- Avviare Eclipse
- Pulire il database
- Caricare il file “Vuoto Versioning1.sql” nella cartella “sql” del progetto
- Cliccare sul tasto “Calculate history metrics”
- Eseguire la query che mostra i risultati per le metriche di package

Verrà mostrata la seguente tupla:

name	description	value
mean_NCHANGE	Mean number of time in which files of a package have been modified	0

Il progetto di riferimento ha le seguenti caratteristiche:

- Non ci sono file locali
- È presente un solo cambiamento (cancellazione di un file)

**Note** Il sistema non considera i cambiamenti su file che hanno subito cambiamenti ma che non sono più presenti in locale (perché cancellati).

## 2.2. Metrica per il calcolo del numero medio di bug-fix di un package

Caso di test	Risultato atteso	Risultato ottenuto
TC 4.16	1	0

### TC 4.16

Passi da eseguire per arrivare alla failure:

- Avviare Eclipse
- Pulire il database
- Caricare il file “Bugfix4.sql” nella cartella “sql” del progetto
- Cliccare sul tasto “Calculate history metrics”
- Eseguire la query che mostra i risultati per le metriche di package

Verrà mostrata la seguente tupla:

name	description	value
mean_NFIX	Mean number of time in which files of a package have been fixed for a bug	0

Il progetto di riferimento ha le seguenti caratteristiche:

- È presente un file locale
- È presente un solo cambiamento
- Il cambiamento in questione è un bugfix (messaggio: “Bugfix”)

**Note** Il sistema non identifica il cambiamento presente come bug-fix.

## 2.3. Metrica per il calcolo del numero di autori di commit di un package

Caso di test	Risultato atteso	Risultato ottenuto
TC 5.9	2	1
TC 5.19	2	1

### TC 5.9

Passi da eseguire per arrivare alla failure:

- Avviare Eclipse
- Pulire il database
- Caricare il file “Progetto9.sql” nella cartella “sql” del progetto
- Cliccare sul tasto “Calculate history metrics”
- Eseguire la query che mostra i risultati per le metriche di package

Verrà mostrata la seguente tupla:

name	description	value
NAUTH	Number of authors of changes made on a package	1

Il progetto di riferimento ha le seguenti caratteristiche:

- Sono presenti due file locali
- Sono presenti due cambiamenti
- Gli autori dei cambiamenti sono diversi e non hanno indirizzo email

**Note** Il sistema non considera il caso in cui un autore non ha indirizzo email.



## TC 5.19

Passi da eseguire per arrivare alla failure:

- Avviare Eclipse
- Pulire il database
- Caricare il file “Progetto19.sql” nella cartella “sql” del progetto
- Cliccare sul tasto “Calculate history metrics”
- Eseguire la query che mostra i risultati per le metriche di package

Verrà mostrata la seguente tupla:

name	description	value
NAUTH	Number of authors of changes made on a package	1

Il progetto di riferimento ha le seguenti caratteristiche:

- Sono presenti due file locali
- Sono presenti due cambiamenti
- Gli autori dei cambiamenti sono diversi e non hanno indirizzo email

**Note** Il sistema non considera il caso in cui un autore non ha indirizzo email.