

# Università degli Studi di Salerno

Dipartimento di Informatica



UNIVERSITÀ DEGLI STUDI DI SALERNO  
**DIPARTIMENTO DI INFORMATICA**  
**DIPARTIMENTO DI ECCELLENZA**

## Corso di Ingegneria Gestione ed Evoluzione del Software

Aniello Florido & Alexander Minichino

July 22, 2019

## Metric 3.0 Web Edition

Test Plan

Coordinatore del progetto:

<b>Nome</b>
Prof.Andrea De Lucia

Partecipanti:

<b>Nome</b>	<b>Matricola</b>
Aniello Florido	0522500625
Alexander Minichino	0522500644

#### Revision History

<b>Data</b>	<b>Versione</b>	<b>Descrizione</b>	<b>Autori</b>
19/03/2019	1.0	Prima stesura	Aniello Florido Alexander Minichino

# Contents

<b>1</b>	<b>Introduzione</b>	<b>4</b>
<b>2</b>	<b>Panoramica del sistema</b>	<b>4</b>
<b>3</b>	<b>Funzionalità da testare</b>	<b>4</b>
<b>4</b>	<b>Approccio</b>	<b>5</b>
<b>5</b>	<b>Test cases</b>	<b>6</b>
<b>6</b>	<b>Specifica dei Test Cases</b>	<b>7</b>
6.1	Autenticazione e Registrazione Nuovo Utente . . . . .	7
6.2	Creazione nuova analisi . . . . .	8
6.3	Richiesta informazioni . . . . .	9

# 1 Introduzione

Lo scopo di questo documento è quello di analizzare, pianificare e gestire lo sviluppo e le attività di testing del progetto Metric 3.0 Web Edition.

Questa sessione di lavoro va a verificare il corretto funzionamento del sistema Metric 3.0 Web Edition in diversi casi, studiati appositamente per testare ogni singola caratteristica e funzionalità del sistema, al fine di ottenere un corretto funzionamento.

Dai risultati ottenuti, qualora fossero presenti delle anomalie, saranno apportate delle modifiche per migliorare la stabilità del sistema.

## 2 Panoramica del sistema

Il tool Metric 3.0 è stato sviluppato presso l'Università degli studi di Salerno. Esso esegue un'analisi dettagliata del codice sorgente Java, ottenendo delle informazioni che vanno a descrivere, attraverso una serie di metriche, la qualità del codice analizzato, così da offrire agli sviluppatori un valido aiuto nella comprensione di progetti con scarsa documentazione.

Il tool è stato implementato utilizzando la libreria JDT Core, la quale permette l'accesso all'AST (Abstract Syntax Tree), fornendo gli strumenti e le strutture dati per analizzarlo.

I nodi dell'AST sono gli elementi presenti nel codice sorgente, come classi, metodi, variabili, ecc.

Metric 3.0 Web Edition è la versione web del tool Metric. E' stata realizzata un'interfaccia che permettesse l'utilizzo delle funzionalità di Metric anche in un browser e non solo in locale.

Inoltre è stata inserita una storicizzazione delle analisi e una sezione per il profilo utente.

## 3 Funzionalità da testare

Nel testing si andranno a considerare le funzionalità principali di Metric 3.0 Web Edition, allo stesso tempo verranno testate anche parti del funzionamento del tool Metric 3.0. LE funzionalità che verranno testate sono:

- **Autenticazione e Registrazione Nuovo Utente:** questa funzione permette all'utente la registrazione al sistema e l'accesso ad esso. Se l'utente non è autenticato, non può avere accesso alle funzionalità del sistema.

- **Creazione di un nuova analisi:** questa funzionalità permette all'utente, dopo aver inserito i dati riguardo la nuova analisi e aver caricato il file contenente il progetto da testare, la creazione di una nuova analisi e la visualizzazione dei risultati forniti dal tool.
- **Richiesta di informazioni:** attraverso questa sezione l'utente potrà inviare una email al team di sviluppo per richiedere delle informazioni sul sistema.

## 4 Approccio

Nella sessione di testing di Metric 3.0 Web Edition verrà utilizzato un approccio di tipo "BLACK BOX", il quale prevede che i test vengano effettuati in modo tale da non scendere nel dettaglio del codice, ma ci si basi sulle specifiche delle funzionalità da testare.

L'approccio alla fase di testing è si divide in tre fasi:

- Creazione di uno script batch, che permetta di testare il funzionamento del vecchio sistema Metric 3.0.
- Testing di integrazione, che va a testare l'integrazione dei sottosistemi.
- Testing di sistema: è un test funzionale che andrà a verificare la funzionalità dell'intero sistema assemblato.

### Utilizzo script batch

Per verificare che il funzionamento del tool Metric 3.0 sia corretto, verrà creato uno script batch che, prendendo in input un progetto in Java, avvierà l'analisi statica del codice del progetto caricato.

In tal modo, si andrà a verificare il corretto funzionamento del tool Metric 3.0 prima di procedere alla creazione della versione web.

**Testing di Integrazione** Nel Testing di integrazione verrà effettuato un controllo sull'integrazione delle varie componenti del sistema. verrà adottata una strategia di tipo "Bottom-up".

Tale strategia va ad integrare i sottosistemi passo passo, per tale motivo potrebbe essere necessario utilizzare dei driver. La lista dei casi di test verrà ricavata dall'applicazione del metodo Category Partition.

### Testing di sistema

Nel testing di sistema si andrà a testare la correttezza dell'intero sistema.

Tale test è considerato il test più critico in quanto, può risultare abbastanza complesso andare alla ricerca di eventuali errori, poichè sono impegnati tutti i sottosistemi.

Per effettuare tale test sarà utilizzato il framework Selenium, il quale fornisce degli strumenti per testare sistemi web-based.

## 5 Test cases

Per sviluppare i test cases sarà utilizzato il metodo del Category Partition.

Questo metodo consiste nell'identificare per ogni funzionalità da testare dei parametri; per ogni parametro verranno individuate delle categorie, le quali poi saranno suddivise in scelte. Alle scelte verrà assegnato un valore.

I test cases verranno definiti nel documento di Test Cases Specification (TCS).

## 6 Specifica dei Test Cases

### 6.1 Autenticazione e Registrazione Nuovo Utente

Parametro = Username	
Categorie	Scelte
lunghezza lUser	1: lunghezza 0-4 [Errore] 2: lunghezza 5-15 [lunghezza accettata] 3: lunghezza 16 [Errore]

Parametro = Password	
Categorie	Scelte
lunghezza lPass	1: lunghezza 0-4 [Errore] 2: lunghezza 5-8 [lunghezza accettata] 3: lunghezza 9 [Errore]
presenza pp	1: dato presente [if lunghezzaUserOk and StringUserOk and lunghezzaPassOk and StringPassOk] 2: dato non presente [errore]

codice	combinazione	esito
TC 0.01	lUser1	errore
TC 0.02	lPass1	errore
TC 0.03	lUser1 e lPass1	errore
TC 0.04	lUser2 lPass2 pp1	accesso

## 6.2 Creazione nuova analisi

Parametro = NameProject	
formato = [0-9 A-Z a-z_]	
Categorie	Scelte
formato fName	1: rispetta il formato / ^[0-9A-Za-z_] 2: non rispetta il formato [errore]
già presente gpe	1: già presente[errore]

Parametro = FileProject	
formato =	
Categorie	Scelte
caricamento file cf	1: file non caricato [errore] 2: file caricato [inizio analisi]

codice	combinazione	esito
TC 1.01	fName2	errore
TC 1.02	fName2	errore
TC 1.03	cf1	errore
TC 1.04	gpe1	errore
TC 1.05	fName1 fc2	inizio nuova analisi



## 6.3 Richiesta informazioni

Parametro = email	
formato = $[A-z0-9\backslash.\backslash+_-] + @[A-z0-9\backslash.\backslash+_-].[A-z]\{2,6\}$	
Categorie	Scelte
valore email ve	1: email non rispetta il formato [errore] 2: email rispetta il formato [email accettata]

Parametro = campoInfo	
formato = $[A-Za-z0-9]^*$	
Categorie	Scelte
valore campo informazioni vci	1: campo riempito [accettato] 2: campo non riempito [errore]

codice	combinazione	esito
TC 2.01	ve1	errore
TC 2.02	vci2	errore
TC 2.03	ve2 vci1	email inviata