

# 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

November 16, 2020

## Metric 3.0 Web Edition

Documentazione di manutenzione:  
Analisi

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>
01/02/2019	1.0	Prima stesura	Aniello Florido Alexander Minichino
30/10/2020	1.1	Correzioni e integrazioni	Aniello Florido Alexander Minichino

# Contents

<b>1</b>	<b>Scopo del documento</b>	<b>4</b>
<b>2</b>	<b>Panoramica del sistema attuale</b>	<b>4</b>
2.1	Funzionamento di Metric3.0 . . . . .	4
2.1.1	Parametri . . . . .	4
2.1.2	Metriche prodotte dal tool . . . . .	6
2.2	Architettura del sistema . . . . .	7
2.2.1	Reverse Engineering . . . . .	7
2.2.2	Tool di reverse engineering . . . . .	7
2.3	Testing . . . . .	8
<b>3</b>	<b>Analisi della modifica richiesta</b>	<b>8</b>
<b>4</b>	<b>Individuazione della soluzione progettuale</b>	<b>8</b>
4.1	Problematiche Affrontate . . . . .	8
4.2	Soluzione individuata . . . . .	10
<b>5</b>	<b>Impact Analysis</b>	<b>12</b>
5.1	Interface layer . . . . .	12
5.2	Application layer . . . . .	12
5.3	Starting impact set . . . . .	14
5.4	Candidate impact set . . . . .	14
5.5	Sequence Diagrams . . . . .	15
5.5.1	Sistema corrente . . . . .	15
5.5.2	Change request . . . . .	15
5.6	Use case change request . . . . .	21
<b>6</b>	<b>Report delle modifiche</b>	<b>29</b>
6.1	Componenti esistenti . . . . .	29
6.2	Integrazione di nuove componenti . . . . .	29
<b>7</b>	<b>Studio di fattibilità</b>	<b>35</b>
7.1	Identificazione, descrizione e valutazione dei costi . . . . .	35
7.2	Identificazione, descrizione e classificazione dei benefici . . . . .	38

# 1 Scopo del documento

In questo documento saranno descritti gli obiettivi del processo di migrazione sul web del progetto Metric 3.0, in riferimento al documento di "Identificazione e classificazione delle modifiche richieste".

Si andrà ad analizzare come tale modifica impetterà sugli artefatti del sistema esistente, osservando il rapporto costi/benefici e i possibili rischi derivati dalla fase di progettazione.

Questo documento includerà anche uno studio di fattibilità.

## 2 Panoramica del sistema attuale

Il tool Metric 3.0 è stato sviluppato presso l'Università degli studi di Salerno. Metric 3.0 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.

### 2.1 Funzionamento di Metric3.0

Attualmente Metric3.0 è un sistema batch eseguibile esclusivamente da linea di comando.

Metric non prevede una profilazione dell'utente e di conseguenza non è possibile effettuare una storicizzazione delle analisi effettuate.

Inoltre, in Metric 3.0 è stato implementato successivamente un modulo sperimentale con l'obiettivo di studiare la relazione tra differenti tipi di commento presenti all'interno del codice e presenza di difetti. Tale modulo è risultato utile allo studio in questione e si presta bene per ulteriori studi ed approfondimenti futuri, tuttavia essendo ancora in via sperimentale, non verrà preso in considerazione in questo processo di migrazione.

#### 2.1.1 Parametri

Essendo un sistema batch, l'interazione con l'utente è ridotta al solo avvio del sistema con la possibilità di indicare dei parametri.

Tra questi parametri vi sono, sia parametri essenziali al funzionamento che parametri opzionali.

I parametri possono alterare l'esecuzione del sistema, permettendo ad esempio

di analizzare una parte del sistema anziché l'intero sistema, oppure di utilizzare una diversa versione di Java per l'analisi.

I parametri esprimibili sotto forma di argomenti passati al main sono:

- Parametri necessari:
  - **path** [STRING]: Una stringa contenente la path del progetto JAVA da analizzare;
  - **sourcePath** [STRING]: Permette di specificare la cartella dove risiede il codice sorgente nel progetto (tipicamente src)
- Parametri opzionali:
  - **java** [INTEGER]: Versione di java da utilizzare (di default 9);
  - **classfilter** [BOOLEAN]: Abilita il class filter, una funzionalità che permette di analizzare solo le classi selezionate
  - **class** [STRING]: Se abilitato “classfilter” permette di specificare i singoli path delle classi separati da virgola
  - **commentsGUI** [BOOLEAN]: Abilita una visualizzazione grafica dei commenti presenti nel progetto
- Parametri opzionali del modulo sperimentale:
  - **reportFilePath** [STRING]: Path report file(xlsx or xls)
  - **createDataset** [BOOLEAN]: Parametro che indica al sistema se creare un nuovo dataset di addestramento
  - **predictBugPresence** [BOOLEAN]: Abilita il modulo di defect prediction

### 2.1.2 Metriche prodotte dal tool

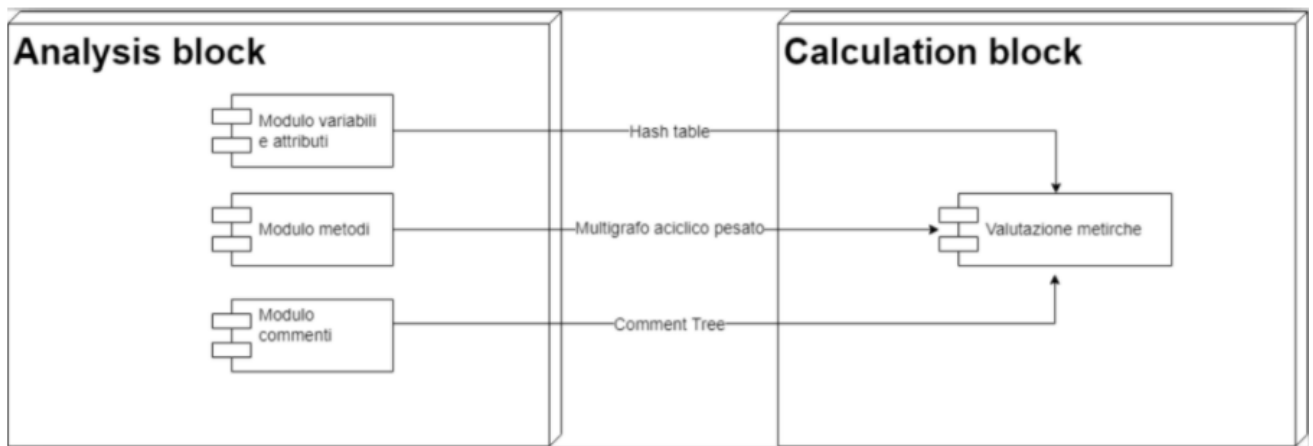
Le metriche utilizzate all'interno del tool Metric 3.0 per l'analisi del codice sorgente sono:

- **DIT** (Depth of Inheritance Tree)
- **WMC** (Weighted Method Class)
- **NOC** (Number of Childern)
- **RFC** (Response For a Class)
- **CBO** (Coupling Between Object Classes)
- **LCOM** (Lock of Choisen of Methods)
- **LOC** (Lines of Code)
- **NC** (Number of Comments)
- **FI** (Fan-in Fan-out)
- **NMS** (Number of Messages)
- **NM** (Number of Methods)
- **NCp** (Number of Classes)
- **CC** (Concrete Classes)
- **AC** (Abstract Classes)
- **Ca** (Coupling Afference)
- **Ce** (Coupling Efference)
- **A** (Abstractiveness)
- **I** (Instability)
- **DFP** (Distance From Parent)
- **NOW** (Number of Words)

## 2.2 Architettura del sistema

Il sistema è organizzato in due blocchi principali: il blocco di analisi e il blocco di calcolo.

Questi due blocchi comunicano tra di loro mediante la condivisione di tre strutture dati che vengono prodotte dal blocco di analisi ed utilizzate dal blocco di calcolo.



Per ulteriori chiarimenti, fare riferimento ai documenti *Reverse\_SDD\_metric* e *Reverse\_ODD\_metric*.

### 2.2.1 Reverse Engineering

Dato che il sistema era sprovvisto di documentazione formale si è deciso di intraprendere un'attività di reverse engineering con lo scopo di ricostruire il modello progettuale del sistema metric 3.0.

Tale attività ha avuto inizio dall'analisi statica del codice del tool metric e ha permesso di ottenere delle informazioni importanti che sono state utilizzate per la stesura di una documentazione più strutturata e dettagliata.

Partendo da un livello più basso di astrazione, dove sono stati studiati le singole componenti e artefatti, si è arrivati ad un livello di astrazione maggiore che ha permesso di ottenere una visione più ampia del sistema e di definire i requisiti funzionali del sistema.

Per maggiori approfondimenti fare riferimento ai documenti *Reverse\_ODD\_metric* e *Reverse\_SDD\_metric*.

### 2.2.2 Tool di reverse engineering

Come tool a supporto del processo di reverse engineering è stato utilizzato *UML* presente all'interno dell'IDE IntelliJ IDEA.

## 2.3 Testing

Il tool attuale è stato fornito sprovvisto di qualsiasi artefatto legato al testing. Quindi, non avendo nulla che attesta l'avvenuto testing, Metric 3.0 viene considerato come sistema non testato.

## 3 Analisi della modifica richiesta

Il processo di migrazione del Metric 3.0 sul web prevede che i principali requisiti di Metric siano resi fruibili attraverso un'applicazione web.

L'obiettivo principale consiste nel creare un interfaccia web che permetta un utilizzo più semplificato e intuitivo del tool da parte dell'utente.

Nell'ottica di semplificare l'utilizzo del sistema da parte dell'utente, sono state effettuate delle scelte progettuali.

Nello specifico, si è deciso di ridurre al minimo indispensabile l'inserimento di parametri da parte dell'utente.

In questo modo, l'utente del nuovo sistema non dovrà indicare null'altro se non il path del progetto da analizzare.

## 4 Individuazione della soluzione progettuale

### 4.1 Problematiche Affrontate

**Issue 1:** Quali funzionalità verranno migrate nel nuovo sistema?

#### Resolution

Verrà effettuata una migrazione del core del sistema Metric 3.0, tralasciando i moduli inerenti la GUI.

Verranno, quindi, migrate le funzionalità legate a :

- Parsing del codice sorgente Java e analisi dell'AST
  - Parsing e analisi di variabili e attributi
  - Parsing e analisi delle catene di invocazione dei metodi
  - Parsing e analisi dei commenti
- calcolo delle metriche
  - Metriche code size
  - Metriche Chidamber e Kemerer (Object Oriented)
  - Metriche di pacchetto
  - Metriche relative ai commenti



**Issue 2:** Quali funzionalità del sistema hanno priorità maggiore?

**Resolution**

Le funzionalità del sistema che hanno priorità maggiore sono quelle legate al parsing del codice sorgente e quelle legate all'analisi dell' AST attraverso JDTCore.

**Issue 3:** Sono necessarie nuove funzionalità?

**Resolution**

Verrà implementato un nuovo modulo che rappresenterà un'interfaccia che permetterà la comunicazione tra il sistema esistente e il nuovo sistema. Nello specifico saranno implementate delle classi (serverlet Java) che avranno il ruolo di dispatcher delle richieste effettuate dalle nuove pagine JSP (Java Servlet Page). Rispetto alla versione precedente è prevista una profilazione degli utenti e inoltre vi sarà anche la possibilità di storicizzare i risultati delle analisi effettuate, le quali verranno associate agli utenti che le effettuano.

**Issue 4:** che tipologia di interfaccia grafica adottare?

**Resolution**

Si andrà ad implementare una GUI semplice ed intuitiva, che permetterà di sfruttare a pieno le funzionalità del tool Metric 3.0. Inoltre, l'interfaccia sarà resa responsive così da garantire una resa ottimale su tutti i dispositivi.

**Issue 5:** che modello architetturale verrà utilizzato?

**Resolution**

Come modello architetturale si è scelto di utilizzare il Pattern Facade. L'implementazione di di tale pattern permette, attraverso un'interfaccia più semplice, l'accesso a sottosistemi che espongono interfacce complesse e molto diverse tra loro, nonché a blocchi di codice complessi.

**Issue 6:** che web server o web container utilizzare?

Resolution
Come Web Server si è scelto Apache Tomcat v.9 in quanto si tratta del web server più utilizzato e diffuso per l'implementazione di sistemi client-server basati su Java.

**Issue 7:** che tipo di DBMS verrà utilizzato?

Resolution
Per la persistenza dei dati si è scelto di non utilizzare un DBMS, ma di salvare i dati direttamente su un file. Tale scelta è stata presa poichè non sussiste la necessità di creare relazioni tra i dati prodotti dal tool (risultati analisi), tranne quella che vi è tra l'analisi effettuata e l'utente che l'ha richiesta.

**Issue 8:** che tipologia di linguaggio di programmazione utilizzare per lo sviluppo del nuovo sistema?

Resolution
Il linguaggio scelto per l'implementazione è JAVA così da garantire una piena compatibilità con il sistema esistente e il Web Server Apache Tomcat. Inoltre verranno utilizzati linguaggi di markup (come HTML e XML), di formattazione (CSS), di scripting (Javascript) e di rappresentazione dei dati (Json) per l'implementazione delle JSP.

## 4.2 Soluzione individuata

La soluzione individuata consiste nell'unione di tutte le resolutions presenti nel paragrafo precedente, di conseguenza la soluzione individuata avrà le seguenti caratteristiche:

- Migrazione del core del sistema tralasciando i moduli inerenti la GUI.
- Verrà implementata un'interfaccia che permetterà la comunicazione tra il sistema esistente e il nuovo sistema.
- Sarà permessa la storicizzazione dei dati, associandoli agli utenti che svolgono le analisi.

- Verrà implementata una GUI che sia il più possibile intuitiva e semplice.
- Il modello architetturale scelto è il Pattern Facade.
- Il Web Server scelto è Apache Tomcat v.9
- I dati verranno salvati all'interno di un file.
- I linguaggi utilizzati sono Java, per garantire una totale compatibilità con il vecchio sistema. Inoltre, verranno utilizzati linguaggi di markup (HTML e XML), di formattazione (CSS), di scripting (Javascript) e di rappresentazione dei dati (JSON) per l'implementazione delle pagine JSP.

## 5 Impact Analysis

La soluzione individuata non andrà a modificare molti artefatti del sistema precedente.

Secondo una prima analisi, le componenti che dovrebbero essere interessate dalla modifica sono le componenti con cui avviene maggiore interazione con l'utente.

Essendo una migrazione verso una versione fruibile da web vi è la necessità di intervenire sulle modalità in cui il tool riceve i comandi e fornisce gli output attesi. Di seguito sono riportati nel dettaglio i componenti interessati dalla modifica.

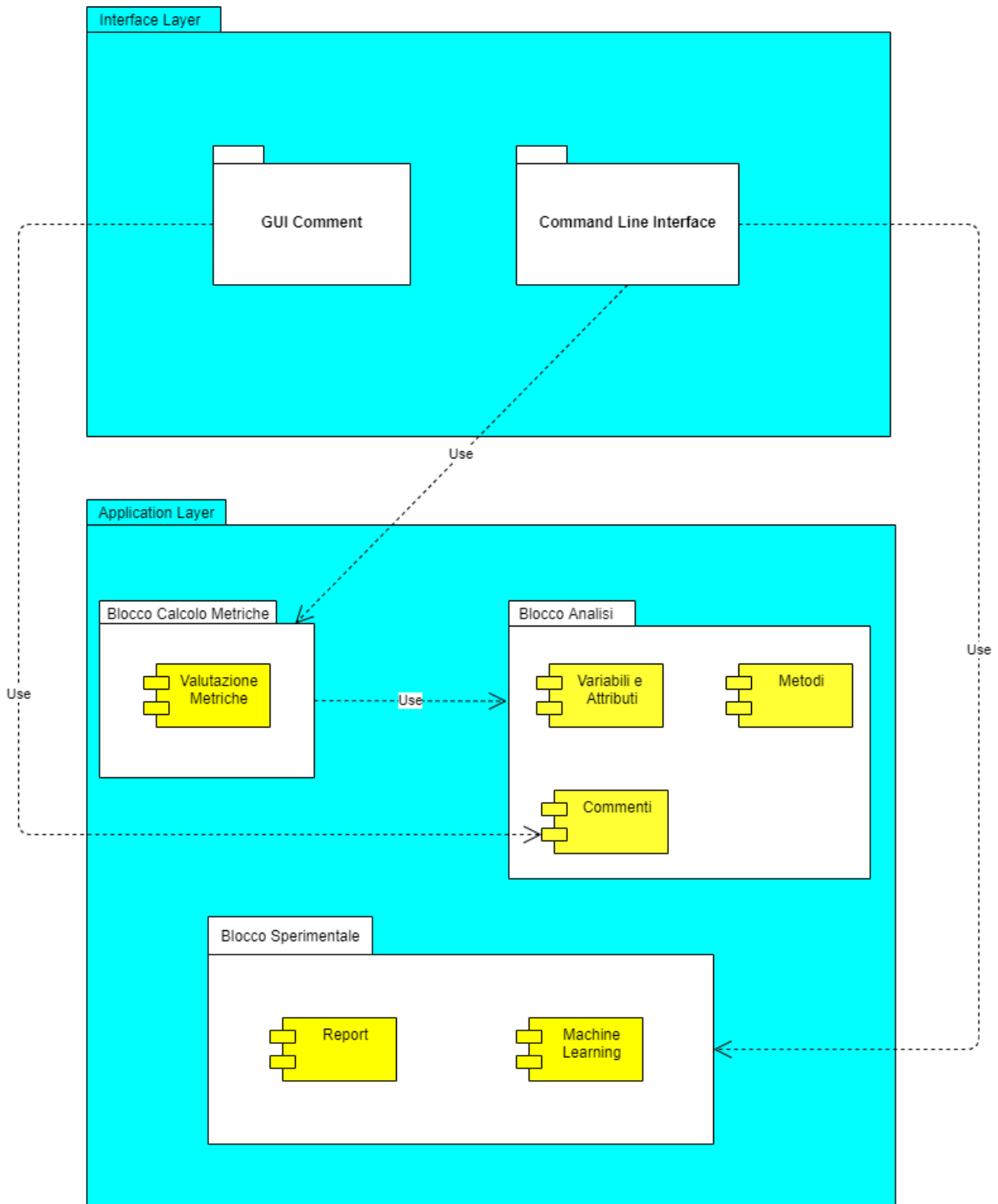
### 5.1 Interface layer

Come prima cosa vanno analizzati i componenti legati all'interfaccia utente, nello specifico il sistema attuale ha un'interfaccia testuale, la quale è legata alle funzionalità core del sistema. Il sistema attuale presenta anche un'interfaccia grafica legata principalmente al modulo dei commenti, si è però deciso che tale interfaccia non rappresenta una funzionalità chiave del tool e quindi non sarà interessata dal processo di migrazione.

### 5.2 Application layer

Essendo un progetto ben strutturato e con una chiara distinzione dei compiti dei vari componenti, il livello applicativo di Metric 3.0 non dovrà necessariamente essere oggetto di modifiche sostanziali. Tale layer potrà essere utilizzato da parte di nuovi moduli che utilizzano le interfacce usate dai moduli della versione attuale. L'idea di base è quella di riutilizzare il più possibile le componenti del sistema originale senza modificarne la forma ma creando nuove componenti che si adattano ad esse.

Di seguito è riportato una schema ad alto livello dell'attuale sistema :



Come anticipato, le componenti interessate dalle modifiche sono principalmente quelle strettamente legate all'interfaccia testuale, e una piccola parte di quelle correlate al calcolo delle metriche. Di seguito sono mostrate le classi che potrebbero essere interessate dal processo di modifica.

- Classe **Core.java**: La classe che contiene il main del progetto e dalla quale vengono richiamate tutte le funzionalità. Si tratta della classe che riceve i parametri dati in input dall'utente e si occupa di fornire i risultati allo stesso, motivo per il quale si ritiene necessario intervenire sulla classe.

- Classe **Constants.java**: Classe utilizzata per racchiudere tutte le costanti utilizzate nel programma. Tali Costanti potrebbero dover rispondere a nuove esigenze.
- Classe **MetricEvaluation.java**: Raccoglie informazioni dalle strutture dati e calcola il valore di diverse metriche rispettivamente ai domini in cui queste metriche sono definite (metodi, classi, packages, project). Classe che si interfaccia direttamente con la classe Core.

Le classi individuate fanno tutte parte del package principale del progetto: *it.unisa.metric*.

### 5.3 Starting impact set

Come già detto, l'obiettivo è quello di modificare componenti del tool originale solo se strettamente necessario per il processo di migrazione. Pertanto le classi individuate saranno modificate affinché supportino nuovi meccanismi di invocazione e affinché si prestino all'integrazione in un web server.

Pertanto, lo starting impact set risulta essere composto degli elementi descritti in precedenza.

### 5.4 Candidate impact set

A seguito di ulteriori analisi per il processo di migrazione si è stabilito che il candidate impact set rispecchia lo starting impact set.

La tabella che segue descrive il Candidate Impact Set individuato, ovvero l'insieme di tutti gli artefatti che potrebbero essere modificati durante la fase di manutenzione. Inoltre, accanto ad ogni artefatto è associato un livello di impatto, tale livello fa riferimento al modo in cui la modifica va ad impattare sull'artefatto stesso.

Per individuare le componenti del sistema impattate dalla modifica si è utilizzato un approccio topdown, ovvero a partire dai documenti di alto livello (ricostruiti a seguito della fase di reverse engineering) si sono individuati gli artefatti di livello più basso che potranno essere modificati durante la fase di manutenzione.

L'impatto della modifica è valutato utilizzando tre categorie:

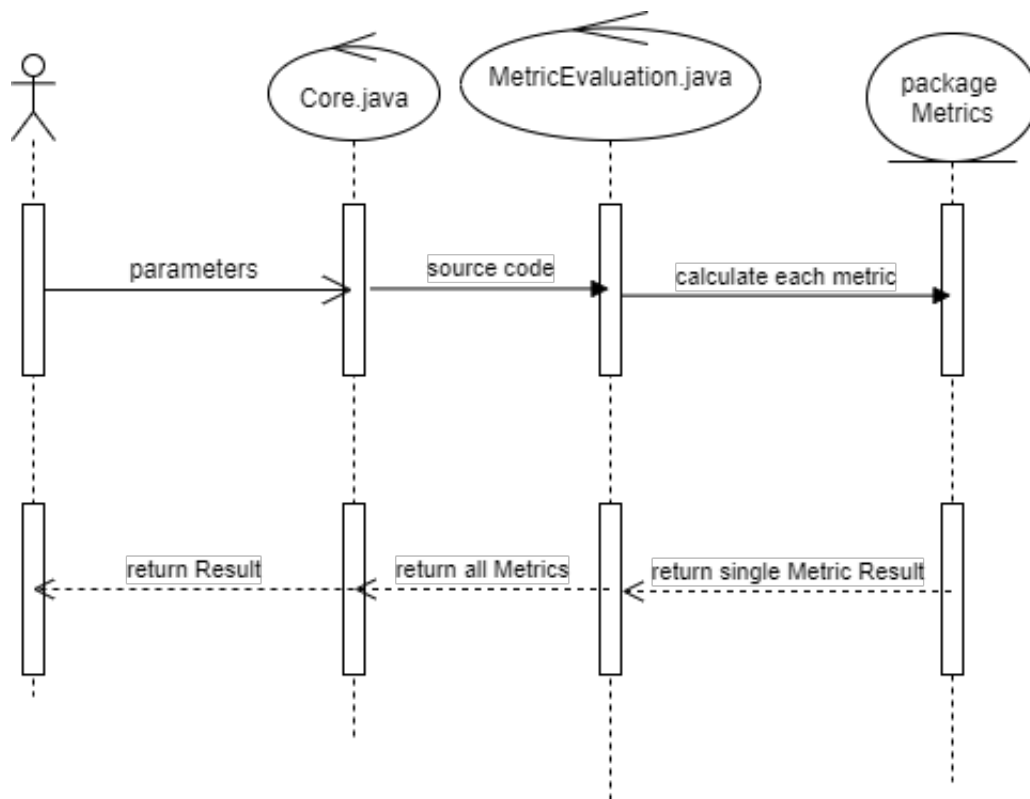
- FORTE: se saranno necessarie pesanti modifiche nell'artefatto o se l'artefatto dovrà essere completamente sostituito;
- MEDIO: se saranno necessarie sostanziali modifiche all'artefatto, non facendo cambiare però la sua struttura in maniera eccessiva;
- DEBOLE: se saranno necessarie solo modifiche marginali;

Artefatto	Impatto	Descrizione
Classe Core.java	Basso	Vedi paragrafo precedente
Classe Constants.java	Medio	Vedi paragrafo precedente
Classe MetricEvaluation.java	Medio	vedi paragrafo precedente

## 5.5 Sequence Diagrams

### 5.5.1 Sistema corrente

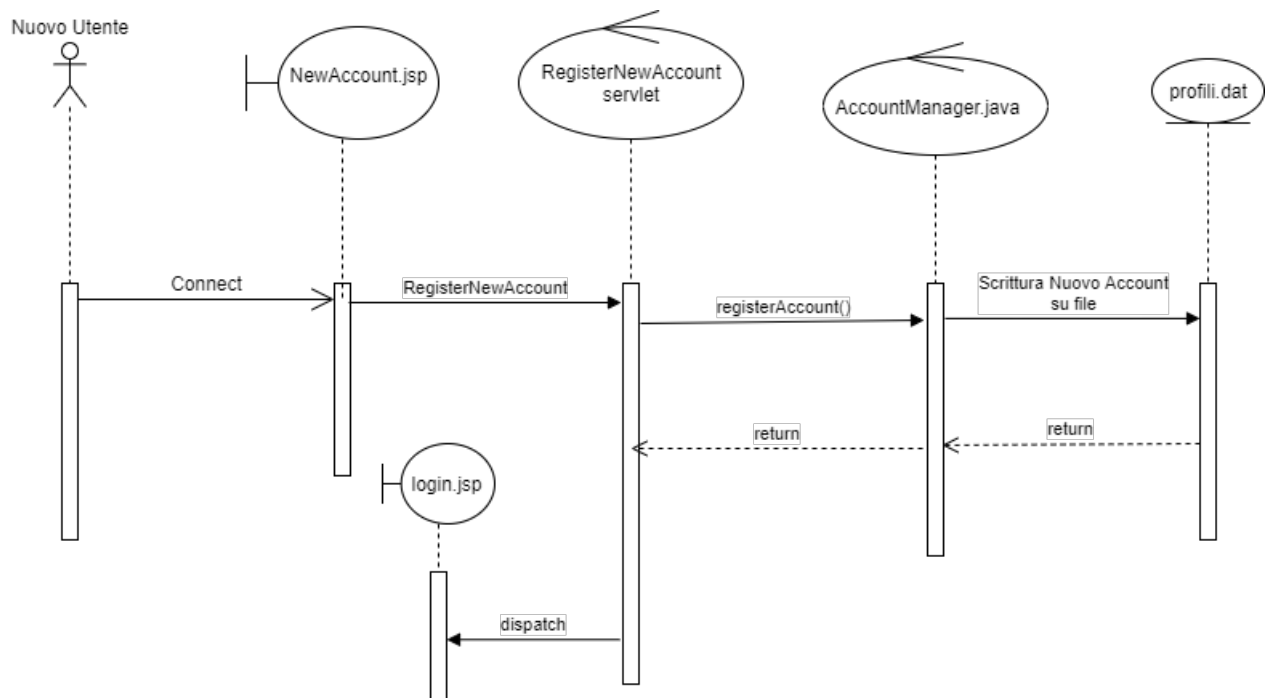
Di seguito è riportato il Sequence diagram del sistema corrente.



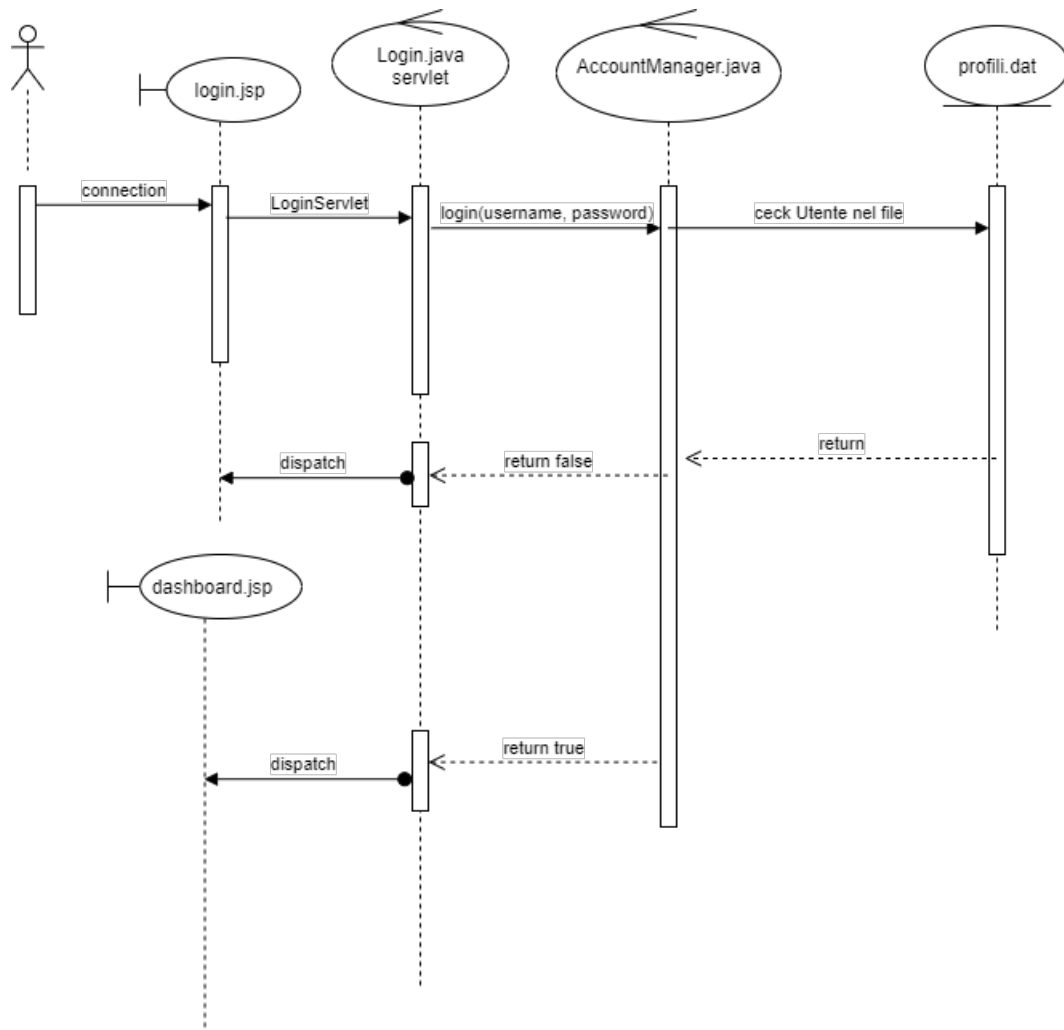
### 5.5.2 Change request

Di seguito vengono riportati i Sequence diagrams relativi al sistema dopo aver apportato le modifiche indicate nella CR.

- Registrazione nuovo utente

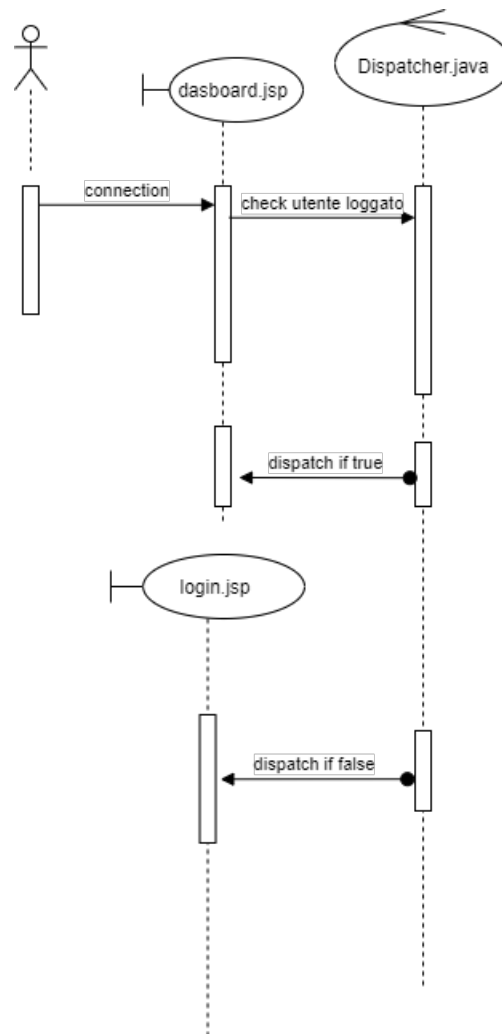


- Login

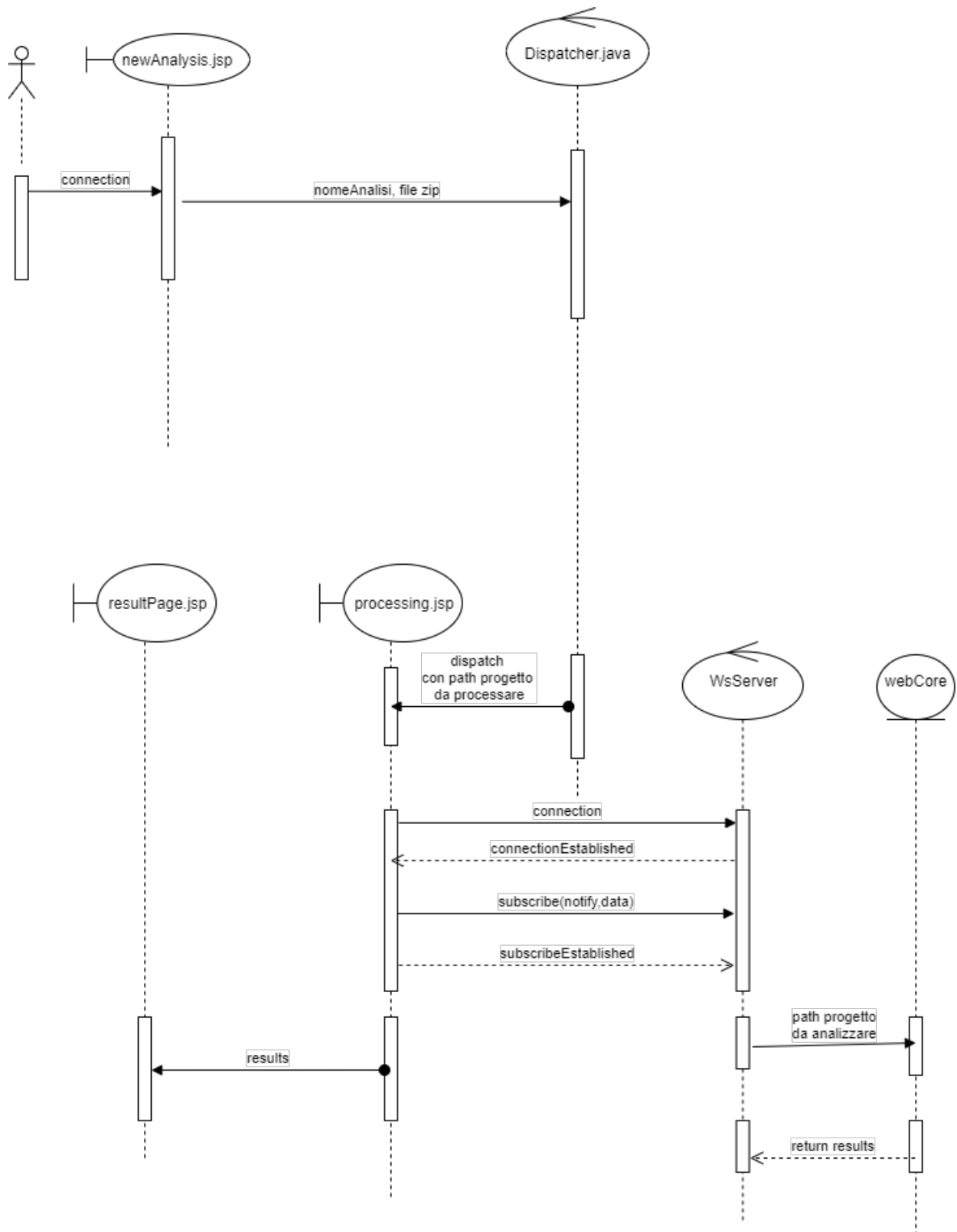




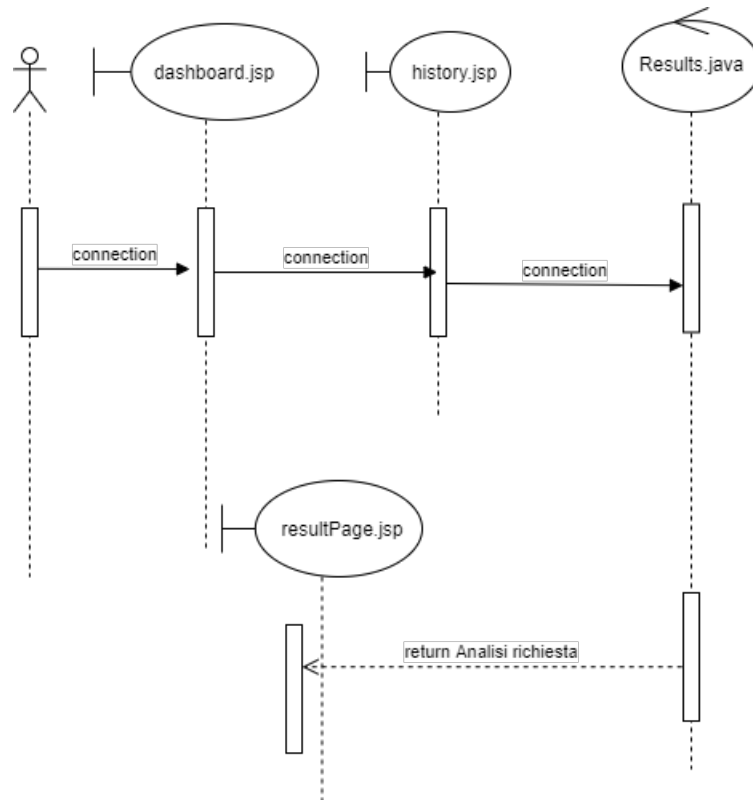
- Visualizza Dashboard



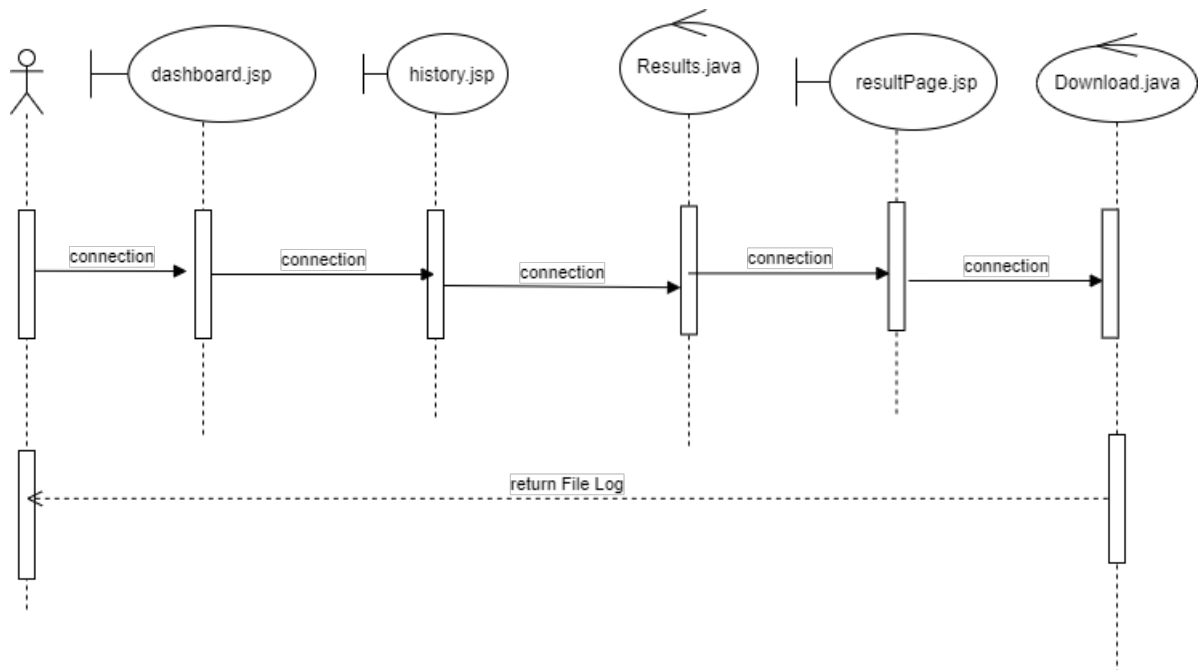
- Creazione Nuova Analisi



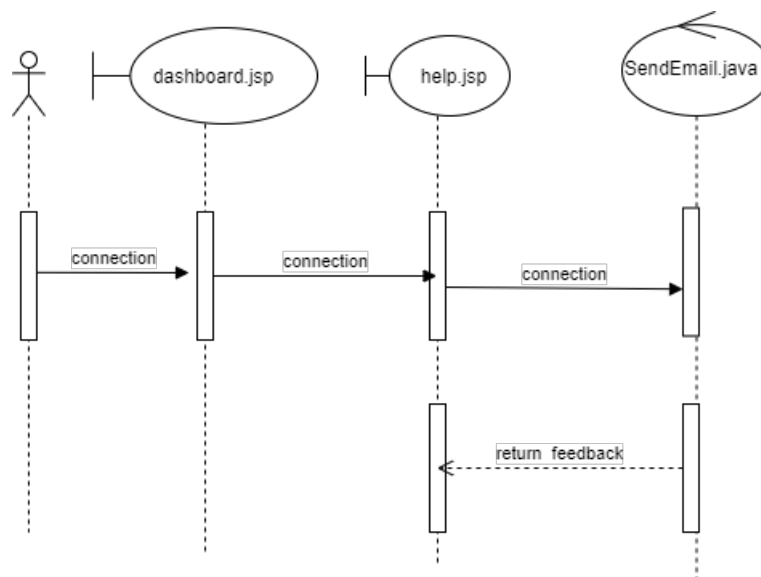
- visualizza Analisi



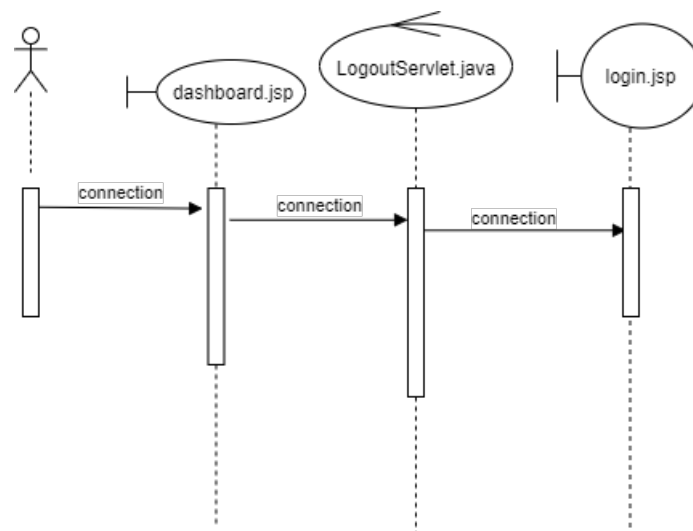
- Download File di log dell'analisi



- Invio Email di supporto



- Logout



## 5.6 Use case change request

Di seguito vengono riportati i casi d'uso.

- Registrazione nuovo utente

Identificativo  UC_CR_1	Registrazione utente	Data	25/10/2020
		Vers.	1.0
Descrizione	L'utente può effettuare una nuova registrazione		
Attore Principale	Utente non registrato  L'utente è intenzionato a creare un profilo per accedere al sistema.		
Attori secondari	NA		
Entry Condition	L'utente non loggato si trova nella login page e clicca sul link di registrazione		
Exit condition  On success	Il sistema inserisce il nuovo profilo utente		
Exit condition  On failure	NA		
Rilevanza/User Priority	ALTA		
Frequenza stimata	bassa/giorno		
Extension point	NA		
Generalization of	NA		
FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO			
1	Utente:	l'utente si trova nella login page e clicca sul pulsante registrazione.	
2	Sistema:	Effettua il redirect alla pagina di registrazione	
3	Utente:	Compila tutti i campi richiesti	
4	Sistema:	Effettua l'inserimento del profilo ed effettua il redirect sulla login page	
Special Requirements			
1		Il sistema deve garantire l'affidabilità	

- Login

Identificativo  UC_CR_2	Login utente	Data	25/10/2020
		Vers.	1.0
Descrizione	L'utente può effettuare il login al sistema		
Attore Principale	Utente registrato  L'utente è intenzionato a ad accedere al sistema		
Attori secondari	NA		
Entry Condition	L'utente non loggato si trova nella login page, inserisce le proprie credenziali e clicca sul pulsante login		
Exit condition  On success	Se i dati sono corretti il sistema crea una sessione per l'utente ed effettua il redirect alla index		
Exit condition  On failure	Se i dati non sono corretti il sistema effettua il redirect alla login page		
Rilevanza/User Priority	ALTA		
Frequenza stimata	alta/giorno		
Extension point	NA		
Generalization of	NA		
FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO			
1	Utente:	Si trova nella login page, inserisce le credenziali e clicca sul pulsante login.	
2	Sistema:	Crea una sessione per l'utente ed effettua il redirect alla index	
Special Requirements			
1		Il sistema deve garantire l'affidabilità	

- Visualizza Dashboard

Identificativo  UC_CR_3		Visualizza dashboard	Data	25/10/2020
			Vers.	1.0
Descrizione		L'utente visualizza la dashboard		
Attore Principale		Utente loggato  L'utente vuole visualizzare la pagina principale del sistema		
Attori secondari		NA		
Entry Condition		L'utente loggato che visita la index del sistema		
Exit condition  On success		L'utente clicca sul link per aprire una pagina diversa dalla dashboard		
Exit condition  On failure		NA		
Rilevanza/User Priority		ALTA		
Frequenza stimata		alta/giorno		
Extension point		NA		
Generalization of		NA		
FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO				
1	Utente:	L'utente visita la pagina principale del sistema		
2	Sistema:	Il sistema mostra la dashboard		
Special Requirements				
1		Il sistema deve garantire l'affidabilità		

- Creazione Nuova Analisi

Identificativo  UC_CR_4	Crea nuova analisi	Data	25/10/2020
		Vers.	1.0
Descrizione	L'utente effettua una nuova analisi		
Attore Principale	Utente loggato  L'utente vuole effettuare una nuova analisi		
Attori secondari	NA		
Entry Condition	L'utente loggato che visita clicca sul link "new analysis"  L'utente effettua l'upload del progetto da analizzare		
Exit condition  On success	Il sistema effettua l'analisi ed effettua il redirect alla pagina che mostra il risultato		
Exit condition  On failure	NA		
Rilevanza/User Priority	ALTA		
Frequenza stimata	alta/giorno		
Extension point	NA		
Generalization of	NA		
FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO			
1	Utente:	L'utente visita la pagina principale "new analysis"	
2	Sistema:	Il sistema mostra la form di upload	
3	Utente:	L'utente effettua l'upload del progetto da analizzare e inserisce un nome per l'analisi	
4	Sistema:	Il sistema effettua l'analisi e mostra i risultati ottenuti	
Special Requirements			
1		Il sistema deve garantire l'affidabilità	



- visualizza Analisi

Identificativo  UC_CR_5	Visualizza risultato analisi	Data	25/10/2020
		Vers.	1.0
Descrizione	L'utente visualizza il dettaglio di un'analisi effettuata in precedenza		
Attore Principale	Utente loggato  L'utente vuole visualizzare i dettagli di un'analisi		
Attori secondari	NA		
Entry Condition	L'utente è loggato e si trova nella pagina degli storici analisi  L'utente individua un record relativo ad un'analisi e clicca sul pulsante "show more"		
Exit condition  On success	Il sistema mostra la pagina contenente i dettagli dell'analisi effettuata		
Exit condition  On failure	NA		
Rilevanza/User Priority	ALTA		
Frequenza stimata	alta/giorno		
Extension point	NA		
Generalization of	NA		
FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO			
1	Utente:	L'utente visita la pagina principale "History"	
2	Sistema:	Il sistema mostra la pagina contenente la lista degli storici	
3	Utente:	L'utente individua un record e clicca sul pulsante "show more"	
4	Sistema	Il sistema mostra i dettagli dell'analisi effettuata	
Special Requirements			
1		Il sistema deve garantire l'affidabilità	

- Download File di log dell'analisi

Identificativo  UC_CR_7	Scarica risultati	Data	25/10/2020
		Vers.	1.0
Descrizione	L'utente visualizza il dettaglio di un'analisi effettuata in precedenza e ne scarica i risultati sotto forma di file testuale		
Attore Principale	Utente loggato  L'utente vuole scaricare i dettagli di un'analisi		
Attori secondari	NA		
Entry Condition	L'utente è loggato e si trova nella pagina di dettaglio di un'analisi effettuata in precedenza  L'utente clicca sul pulsante "Download"		
Exit condition  On success	Il sistema forza il download del file di interesse		
Exit condition  On failure	NA		
Rilevanza/User Priority	ALTA		
Frequenza stimata	alta/giorno		
Extension point	NA		
Generalization of	NA		
FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO			
1	Utente:	L'utente visita la pagina principale "History"	
2	Sistema:	Il sistema mostra la pagina contenente la lista degli storici	
3	Utente:	L'utente individua un record e clicca sul pulsante "show more"	
4	Sistema:	Il sistema mostra i dettagli dell'analisi effettuata	
5	Utente:	L'utente clicca sul pulsante "Download"	
6	Sistema:	Il sistema forza il download del file	
Special Requirements			
1		Il sistema deve garantire l'affidabilità	

- Invio Email di supporto

Identificativo  UC_CR_8	Contatta assistenza	Data	25/10/2020
		Vers.	1.0
Descrizione	L'utente invia una richiesta di supporto tramite l'apposita form		
Attore Principale	Utente loggato L'utente richiede assistenza		
Attori secondari	NA		
Entry Condition	L'utente è loggato e clicca sul pulsante "Help center"		
Exit condition On success	L'utente sottomette la form di contatto e il sistema invia la segnalazione agli admin		
Exit condition On failure	NA		
Rilevanza/User Priority	ALTA		
Frequenza stimata	bassa/giorno		
Extension point	NA		
Generalization of	NA		
FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO			
1	Utente:	L'utente clicca sul pulsante "Help center"	
2	Sistema:	Il sistema mostra la form di contatto	
3	Utente:	L'utente compila la form con i campi richiesti e sottomette la richiesta	
4	Sistema:	Il sistema invia la segnalazione agli admin	
Special Requirements			
1		Il sistema deve garantire l'affidabilità	

- Logout

Identificativo  UC_CR9		Logout utente	Data	25/10/2020
			Vers.	1.0
Descrizione		L'utente effettua il logout distruggendo la sessione in uso		
Attore Principale		Utente loggato L'utente richiede il logout		
Attori secondari		NA		
Entry Condition		L'utente è loggato e clicca sul pulsante "logout"		
Exit condition On success		Il sistema distrugge la sessione corrente		
Exit condition On failure		NA		
Rilevanza/User Priority		ALTA		
Frequenza stimata		media/giorno		
Extension point		NA		
Generalization of		NA		
FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO				
1	Utente:	L'utente clicca sul pulsante "logout"		
2	Sistema:	Il sistema distrugge la sessione relativa all'utente		
Special Requirements				
1		Il sistema deve garantire l'affidabilità		

## 6 Report delle modifiche

La nuova interfaccia Web deve consentire all'utente di effettuare le principali operazioni che vengono offerte dal tool Metric 3.0.

Per tale motivo non vi sono sostanziali modifiche ai requisiti funzionali.

### 6.1 Componenti esistenti

Dopo un'attenta analisi si è deciso di non apportare nessuna modifica alle classi esistenti, bensì di crearne una versione alternativa che si prestasse bene all'integrazione con le nuove componenti.

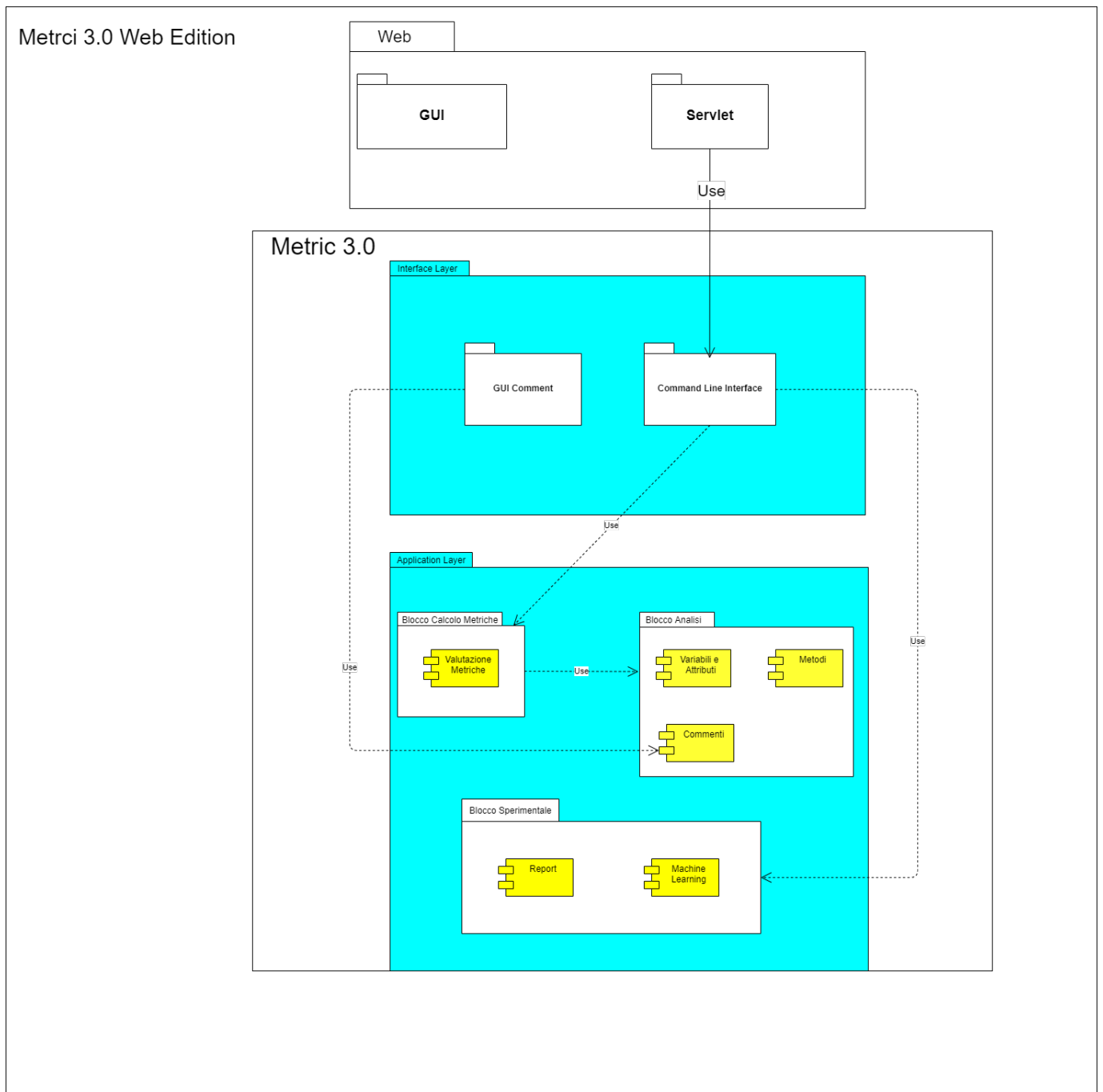
Nello specifico, per quanto riguarda le classi **Constants.java** e **MetricEvaluation.java** si è deciso, sfruttando l'ereditarietà, di fare l'override dei soli metodi che necessitavano di essere modificati. Per quanto riguarda la classe **Core.java**, si è creata una copia della stessa che facesse uso delle sottoclassi create ad-hoc (**WebConstants** e **MetricEvaluationWeb**) anziché le classi originali.

Come si può evincere, le componenti del vecchio sistema interessate da modifiche sono minime, l'idea di sfruttare il più possibile le vecchie componenti ha permesso di attuare il processo di migrazione senza stravolgere il vecchio sistema.

### 6.2 Integrazione di nuove componenti

Come già accennato, per la migrazione verso una versione fruibile da web sarà necessario introdurre nuove componenti che interagiscano con quelle esistenti.

Tali componenti saranno progettate per adattarsi al meglio alle componenti di metric 3.0. Di seguito uno schema ad alto livello che descrive quanto detto:



Nel dettaglio, il processo di migrazione ha visto la creazione di un nuovo package “web” contenente sia le componenti modificate del vecchio sistema che le nuove componenti. Di seguito sono riportate le classi e i package contenuti nel package web.

Classe	Descrizione
Analysis	Rappresenta un’istanza di analisi

Classe	Descrizione
HystoryItem	Rappresenta un elemento dello storico analisi

Classe	Descrizione
MetricEvaluationWeb	Versione modificata della classe MetricEvaluation

Enum	Descrizione
MetricLevel	Enumeratore che rappresenta i livelli di metrica (PACKAGE, CLASS, METHOD)

Classe	Descrizione
MetricResult	Rappresenta il risultato di una metrica

Classe	Descrizione
MetricResultsMean	calcola i valori medi per una data metrica

Classe	Descrizione
SimpleCommentNode	rappresenta un nodo di tipo commento e detiene le informazioni importanti contenute anche nella classe CommentNode

Classe	Descrizione
WebConstants	versione modificata della classe Constants

Classe	Descrizione
WebCore	versione modificata della classe Core

Classe	Descrizione
WebPrinter	Si occupa di effettuare il redirect dello standard output per ottenere uno flusso continuo di feedback in fase di analisi

Package *it.unisa.metric.web.servlets*

Classe	Descrizione
Dispatcher	Gestisce tutte le richieste inoltrandole alla servlet di competenza

Classe	Descrizione
Download	Si occupa di gestire le richieste di download dei file di analisi

Classe	Descrizione
History	Gestisce le richieste legate allo storico delle analisi competenza

Classe	Descrizione
Login	Gestisce le richieste di accesso al sistema

Classe	Descrizione
Results	Provvede al reperimento dei risultati di un'analisi effettuata

Classe	Descrizione
SendEmail	Provvede all'invio di una richiesta di supporto

Classe	Descrizione
Upload	Gestisce le richieste di upload di file contenenti i progetti da analizzare

Package *it.unisa.metric.web.socket*

Classe	Descrizione
CustomSocketConfigurator	Classi di configurazione per il WebSocket

Enum	Descrizione
SocketEventType	Rappresenta gli stati di analisi che vengono comunicati al client tramite socket web

Classe	Descrizione
SocketResponse	Rappresenta un'istanza di messaggio veicolato tramite socket web

Classe	Descrizione
WsServer	espone il servizio WebSocket e si occupa di inviare un flusso di feedback all'utente in fase di analisi

Package *it.unisa.metric.web.utils*

Classe	Descrizione
EmailUtility	Classe di utility che espone servizi per l'invio di email



Classe	Descrizione
FileUtils	Espone metodi per trattare files

Classe	Descrizione
MD5	Espone un metodo per il calcolo dell'hash di una data stringa

Classe	Descrizione
ZipUtils	Espone i metodi per trattare file compressi in formato zip

Package *it.unisa.metric.web.utils.login*

Classe	Descrizione
Account	Rappresenta un'istanza di un profilo utente

Classe	Descrizione
AccountManager	Espone i metodi per trattare oggetti di tipo Account

Classe	Descrizione
LogoutServlet	Provvede alla distruzione della sessione corrente

Classe	Descrizione
RegisterAccountServlet	Espone i metodi per la registrazione di un nuovo account

Oltre al nuovo package *it.unisa.metric.web*, sono stati introdotti dei file *Java Servlet Page* per la creazione dell'interfaccia grafica.

Di seguito sono riportati i file della cartella pages:

File	Descrizione
dashboard.jsp	fornisce l'interfaccia grafica per la visualizzazione della dashboard

File	Descrizione
help.jsp	fornisce l'interfaccia grafica per la visualizzazione della form di richiesta assistenza

File	Descrizione
history.jsp	fornisce l'interfaccia grafica per la visualizzazione della lista delle analisi effettuate in precedenza

File	Descrizione
login.jsp	fornisce l'interfaccia grafica per la visualizzazione della form di accesso al sistema
File	Descrizione
newAccount.jsp	fornisce l'interfaccia grafica per la visualizzazione della form di registrazione al sistema
File	Descrizione
newAnalysis.jsp	fornisce l'interfaccia grafica per la visualizzazione della form di sottomissione di una nuova analisi
File	Descrizione
noData.jsp	fornisce l'interfaccia grafica per la visualizzazione di un messaggio che avvisa l'utente quando viene visitata la pagina degli storici senza elementi da visualizzare
File	Descrizione
processing.jsp	fornisce l'interfaccia grafica per la visualizzazione dell'avanzamento dell'analisi del sistema Java in esame
File	Descrizione
resultPage.jsp	fornisce l'interfaccia grafica per la visualizzazione del risultato ottenuto dall'analisi di un progetto
File	Descrizione
error.jsp	fornisce l'interfaccia grafica per la visualizzazione di un messaggio di errore all'utente qualora dovesse verificarsi una condizione di errore
File	Descrizione
dispatcher.jsp	si occupa di caricare l'interfaccia grafica per la pagina richiesta.
File	Descrizione
index.jsp	si occupa di reindirizzare l'utente verso la dashboard

## 7 Studio di fattibilità

### 7.1 Identificazione, descrizione e valutazione dei costi

Identificazione	Valutazione	Motivazioni
Migrazione dei requisiti funzionali	Debole	Verrà effettuata una migrazione graduale del core del tool, così come specificato nella Issue 1.
Aggiunta di nuove funzionalità	Media	Le modifiche richieste prevedono l'aggiunta della persistenza dei dati con relativa gestione dei profili utenti a cui vengono abbinate le analisi. La creazione di un'interfaccia Web per interagire con il tool.
Alta somiglianza con l'interfaccia esistente	Forte	L'obiettivo principale del processo di migrazione prevede lo sviluppo dei moduli che si occupano dell'interfaccia grafica.
L'architettura usata è il Pattern Facade	Medio	Tale pattern prevede che il lavoro venga concentrato sulla presentazione dei dati con la creazione di una nuova interfaccia, mentre le classi che compongono il tool vengono mantenute così come sono.
Utilizzo del Web Server Apache e di Tomcat	Debole	Il Web Server Apache come Tomcat sono gratuiti pertanto non risulta disponibile a costo zero.

Implementazione con linguaggi di programmazione client-side HTML, CSS, JavaScript e serverside Java.	Debole	Il team ha un'ottima conoscenza di tali tecnologie.
Livello di complessità dell'interfaccia utente.	Media	L'utilizzo combinato di HTML, CSS e Javascript permette la creazione di un'interfaccia utente che possa essere quanto più avanzata. Inoltre verranno implementati alcuni controlli presenti nelle tradizionali applicazioni standalone.
Compatibilità con diversi tipi e versioni di browser.	Forte	Il sistema, funzionando sul web, verrà implementato con framework e linguaggi che possano essere facilmente interpretati da qualsiasi browser, così da avere una piena compatibilità.
Velocità di accesso alle risorse.	Debole	Le tecnologie attuali permettono di sorvolare gli aspetti di ottimizzazione dei tempi di risposta da parte del server.
Mantenimento dello stato dell'applicazione.	Media	Il modello di interazione di base del Web è privo del concetto di connessione, per cui l'esecuzione di un'applicazione web consiste di una serie di interazioni disconnesse. Pertanto si prevede l'utilizzo di informazioni codificate nella richiesta HTTP.

Organizzazione del lavoro nel team di sviluppo.	Forte	I componenti del team di sviluppo condividono poche ore settimanali per lo sviluppo del nuovo sistema. Pertanto verranno effettuate riunioni extra il sabato e/o la domenica.
Testing funzionale	Forte	Data la mancata presenza di artefatti correlati al testing e l'elevata complessità del sistema. Si è scelto di testare il sistema solo attraverso il testing di sistema, così da testare in un'unica volta l'intero sistema.

## 7.2 Identificazione, descrizione e classificazione dei benefici

Facilità di distribuzione e manutenzione.	Forte	L'applicazione web si trova interamente sul server, per cui la pubblicazione o l'aggiornamento di risorse sul nodo server è automaticamente reso disponibile a tutti i nodi client.
Accesso multi-attaforma.	Forte	L'accesso all'applicazione web è indipendente dall'hardware e dal sistema operativo utilizzato dagli utenti.
Riduzione del costo di gestione.	Forte	L'uso di Internet come infrastruttura per un'applicazione web riduce notevolmente sia i costi di connettività che i costi di gestione dei client.