

Università degli Studi di Padova

DIPARTIMENTO DI MATEMATICA "TULLIO LEVI-CIVITA "

CORSO DI LAUREA IN INFORMATICA



**Un plugin Maven per l'automatizzazione
della pubblicazione di documentazione
software**

Tesi di laurea triennale

Relatore

Prof. Paolo Baldan

Laureanda

Laura Cameran

ANNO ACCADEMICO 2018-2019

“I made a discovery today. I found a computer.
Wait a second, this is cool. It does what I want it to.
If it makes a mistake, it’s because I screwed it up. Not because it doesn’t like me.”

— The Mentor

Sommario

Il documento corrente descrive il lavoro svolto durante il periodo di stage, della durata di trecentoventi ore, dalla laureanda Laura Cameran presso l'azienda Finantix Pro Unipersonale S.r.l.

L'obiettivo principale da raggiungere era lo sviluppo di un plugin Maven al fine di automatizzare la pubblicazione di documentazione di software sul sistema documentale Atlassian Confluence. Per realizzare tale compito era richiesto inoltre lo studio di API RESTful, mezzo con cui interagire con il sistema aziendale.

Indice

1	Introduzione	1
1.1	Il progetto	1
1.2	Principali problematiche	1
1.2.1	Soluzioni scelte	1
1.3	Strumenti utilizzati	1
1.4	Il prodotto ottenuto	2
1.5	Organizzazione del testo	2
2	Analisi dei requisiti	3
2.1	Casi d'uso	3
2.2	Requisiti	4
2.2.1	Tracciamento dei requisiti	4
2.2.2	Requisiti di funzionalità	5
2.2.3	Requisiti di qualità	6
3	Progettazione	7
3.1	Tecnologie e strumenti	7
3.2	Ciclo di vita del software	7
3.3	Progettazione	7
3.4	Design Pattern utilizzati	7
4	Realizzazione e testing	9
5	Conclusioni	11
5.1	Risultato ottenuto	11
5.2	Analisi critica del prodotto e del lavoro di stage	11
5.2.1	Il prodotto è utilizzato?	11
5.2.2	Valutazione degli strumenti utilizzati	11
5.2.3	Possibili punti di insoddisfazione	11
5.2.4	Possibili estensioni	11
A	Appendice A	13
	Bibliografia	17

Elenco delle figure

2.1	Use Case - UC0: Scenario principale	3
-----	---	---

Elenco delle tabelle

1.1	Tabella di tecnologie utilizzate durante il progetto e loro scopo.	2
2.1	Elenco dei requisiti di funzionalità (1)	5
2.2	Elenco dei requisiti di qualità.	6

Capitolo 1

Introduzione

1.1 Il progetto

Finantix è un'azienda di informatica che vende prodotti software. Il suo prodotto principale è suddiviso in moduli. Ognuno di questi moduli prevede una propria documentazione delle API Java (un archivio zip contenente documentazione in formato Javadoc) e la documentazione della API RESTful (un archivio zip contenente documentazione in formato Open API). Questa documentazione viene manualmente caricata sulla piattaforma Confluence ove cui è consultata.

Il plugin Maven nasce dalla necessità di automatizzare la pubblicazione di questa documentazione su Confluence, in modo da semplificare e velocizzare notevolmente questo processo. Infatti, una volta configurato correttamente il plugin in tutti i progetti relativi ai moduli software, il caricamento avviene direttamente durante la build dei progetti, senza richiedere ulteriore intervento umano.

1.2 Principali problematiche

Le principali problematiche riscontrate

1.2.1 Soluzioni scelte

.....

1.3 Strumenti utilizzati

Qui di seguito viene riportata una tabella con tutte le tecnologie utilizzate e a quale scopo.

Tecnologie	Scopo
Eclipse	Ambiente di sviluppo
Maven	Build automation per la gestione di progetti
Confluence	Pubblicazione, creazione e consultazione di documentazione
Jira	Issue tracking system
Jenkins	Continuous integration
Sonarqube	Analisi statica del codice
Bitbucket e GitKraken	Controllo di versione
JUnit	Test di unità
Visual Studio Code	Editor di codice
SequenceDiagram.org	Creazione dei diagrammi di sequenza
ObjectAid UML Explorer	Creazione dei diagrammi delle classi
Meecrowave	Creazione di server

Tabella 1.1: Tabella di tecnologie utilizzate durante il progetto e loro scopo.

1.4 Il prodotto ottenuto

.....

1.5 Organizzazione del testo

Il secondo capitolo comprende l'analisi dettagliata dei requisiti del prodotto con casi d'uso e il relativo tracciamento dei requisiti individuati.

Il terzo capitolo descrive la progettazione del software.

Il quarto capitolo approfondisce la realizzazione del plugin e come è stata effettuata l'attività di testing.

Il quinto capitolo corrisponde al capitolo conclusivo. Esso riassume il risultato finale ottenuto e attua una valutazione critica del prodotto.

Riguardo la stesura del testo, relativamente al documento sono state adottate le seguenti convenzioni tipografiche:

- * gli acronimi, le abbreviazioni e i termini ambigui o di uso non comune menzionati vengono definiti nel glossario, situato alla fine del presente documento;
- * per la prima occorrenza dei termini riportati nel glossario viene utilizzata la seguente nomenclatura: *parola*^[g];
- * i termini in lingua straniera o facenti parti del gergo tecnico sono evidenziati con il carattere *corsivo*.

Capitolo 2

Analisi dei requisiti

Tale capitolo ha l'obiettivo di esporre e analizzare i requisiti espliciti e impliciti per la realizzazione del plugin Maven per la pubblicazione di documentazione software. L'attività di analisi ha funto da base per la fase di progettazione del software, in modo che il prodotto fosse conforme alle richieste dell'azienda.

2.1 Casi d'uso

Per lo studio dei casi di utilizzo del prodotto sono stati creati dei diagrammi. I diagrammi dei casi d'uso (in inglese *Use Case Diagram*) sono diagrammi di tipo [Unified Modeling Language \(UML\)](#) dedicati alla descrizione delle funzioni o servizi offerti da un sistema, così come sono percepiti e utilizzati dagli attori che interagiscono col sistema stesso. Essendo il progetto finalizzato alla creazione di un tool per l'automazione di un processo, le interazioni da parte dell'utilizzatore devono essere ovviamente ridotte allo stretto necessario. Per questo motivo i diagrammi d'uso risultano semplici e in numero ridotto.

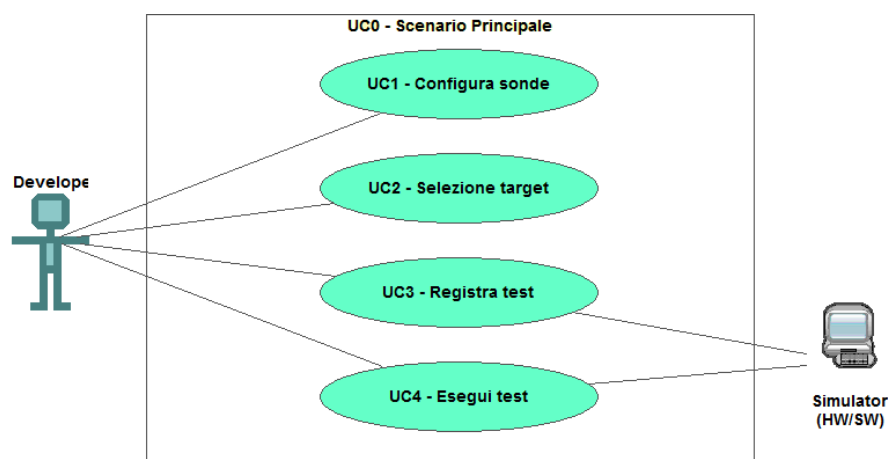


Figura 2.1: Use Case - UC0: Scenario principale

UC0: Scenario principale

Attori Principali: Sviluppatore applicativi.

Precondizioni: Lo sviluppatore è entrato nel plug-in di simulazione all'interno dell'IDE.

Descrizione: La finestra di simulazione mette a disposizione i comandi per configurare, registrare o eseguire un test.

Postcondizioni: Il sistema è pronto per permettere una nuova interazione.

UC1: Configurazione

Attori Principali: Sviluppatore.

Precondizioni: Lo sviluppatore sta utilizzando il progetto.

Descrizione: Lo sviluppatore configura come .

Postcondizioni: Il sistema è pronto per l'esecuzione.

2.2 Requisiti

Ad ogni requisito viene assegnato il codice identificativo univoco:

R[Numero] [Tipo] [Priorità]

in cui ogni parte ha un significato preciso:

- * **R**: requisito.
- * **Numero**: numero progressivo che segue la struttura dei documenti.
- * **Tipo**: la tipologia di requisito che può essere di:
 - **F**: funzionalità.
 - **Q**: qualità.
 - **V**: vincolo.
- * **Priorità**: indica il grado di urgenza di un requisito di essere soddisfatto, come:
 - **0**: opzionale.
 - **1**: desiderabile.
 - **2**: obbligatorio.

Esempio: R2Q1 indica il secondo requisito di qualità ed è desiderabile.

2.2.1 Tracciamento dei requisiti

Da un'attenta analisi dei requisiti e dei casi d'uso effettuata sul progetto, è stata stilata la tabella che traccia tutti i requisiti in rapporto alle loro fonti.

2.2.2 Requisiti di funzionalità

Codice	Requisito	Fonte
R1F2	Redmine deve poter inviare una segnalazione al Producer Redmine	Interno UC1-PR
R1.1F2	Redmine deve poter inviare la segnalazione di apertura issue al Producer Redmine	Interno UC1.1-PR
R1.2F2	Redmine deve poter inviare la segnalazione di modifica issue al Producer Redmine	Interno UC1.2-PR
R1.3F2	Redmine deve poter inviare la segnalazione di commento di una issue al Producer Redmine	Interno UC1.3-PR
R2F2	GitLab deve essere in grado di inviare una segnalazione al Producer GitLab	Interno UC2-PG
R2.1F2	GitLab deve essere in grado di segnalare l'apertura di issue al Producer GitLab	Interno UC2.1-PG
R2.2F2	GitLab deve essere in grado di segnalare la modifica issue al Producer GitLab	Interno UC2.2-PG
R2.3F2	GitLab deve essere in grado di segnalare il commento di una issue al Producer GitLab	Interno UC2.3-PG
R2.4F2	GitLab deve poter segnalare un evento di push al Producer GitLab	Interno UC2.4-PG
R2.5F2	GitLab deve poter segnalare un evento di commento di commit al Producer GitLab	Interno UC2.5-PG
R3F2	Il Producer Redmine deve essere in grado di inviare un messaggio al Gestore Personale	Interno UC3-GP
R3.1F2	Il Producer Redmine deve essere in grado di inviare un messaggio di apertura issue al Gestore Personale	Interno UC3.1-GP
R3.2F2	Il Producer Redmine deve essere in grado di inviare un messaggio di modifica issue al Gestore Personale	Interno UC3.2-GP
R3.3F2	Il Producer Redmine deve essere in grado di inviare un messaggio di commento issue al Gestore Personale	Interno UC3.3-GP

Tabella 2.1: Elenco dei requisiti di funzionalità (1)

2.2.3 Requisiti di qualità

Codice	Requisito	Fonte
R1Q1	La copertura dei test deve essere almeno pari al 70% del codice	Interno
R2Q1	Ogni commit effettuato deve rispettare la formattazione descritta nella wiki	Interno
R3Q1	Ogni documento deve attraversare tutte le fasi previste dal suo ciclo di vita	Interno QPR008
R4Q1	L'intera progettazione del progetto non deve contenere un numero di pattern più alto di quanto segnalato nel pdq	Interno QPR012
R5Q2	Le norme presenti sulla wiki aziendale devono essere rispettate	Interno
R6Q2	Deve essere redatto un manuale utente	Interno
R7Q2	Deve essere redatto un manuale dello sviluppatore	Capitolato
R8Q1	Deve essere redatta una pagina di utilizzo Maven ..	Interno

Tabella 2.2: Elenco dei requisiti di qualità.

Capitolo 3

Progettazione

Breve introduzione al capitolo

3.1 Tecnologie e strumenti

Di seguito viene data una panoramica delle tecnologie e strumenti utilizzati.

Tecnologia 1

Descrizione Tecnologia 1.

Tecnologia 2

Descrizione Tecnologia 2

3.2 Ciclo di vita del software

3.3 Progettazione

3.4 Design Pattern utilizzati

Capitolo 4

Realizzazione e testing

Capitolo 5

Conclusioni

5.1 Risultato ottenuto

5.2 Analisi critica del prodotto e del lavoro di stage

.....

5.2.1 Il prodotto è utilizzato?

.....

5.2.2 Valutazione degli strumenti utilizzati

.....

5.2.3 Possibili punti di insoddisfazione

.....

Relativi miglioramenti

.....

5.2.4 Possibili estensioni

.....

Appendice A

Appendice A

Citazione

Autore della citazione

Bibliografia