

# ISCA

## Identification of Self-admitted technical debts through Conversational Agent

### Test Plan & Specification - ASATDD

#### Team Members

Alessandro Bergamo  
a.bergamo2@studenti.unisa.it

Rosario Di Palma  
r.dipalma22@studenti.unisa.it

Vincenzo Manserra  
v.manserra@studenti.unisa.it

#### Reviewer

Stefano Lambiase  
slambiase@unisa.it

25 Maggio 2022

## Indice

<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>
3.1	Requisiti funzionali . . . . .	4
<b>4</b>	<b>Approccio</b>	<b>4</b>
<b>5</b>	<b>Testing del modello</b>	<b>5</b>
5.1	Testing sul formato . . . . .	5
5.2	TC_1: Il sistema deve essere in grado di recuperare i commit di una data repository . . . .	5
5.3	TC_2: Il sistema deve essere in grado di istruire il classificatore . . . . .	5
5.4	TC_3: Il sistema deve poter analizzare testualmente i commit message e evidenziare i possibili Self-Admitted Technical Debt . . . . .	6
	<b>References</b>	<b>6</b>

## Revision History

Version	Team Member	Description	Date
0.1	Alessandro Bergamo, Rosario Di Palma, Vincenzo Manserra	Prima stesura del documento	25/05/2022
0.2	Alessandro Bergamo, Rosario Di Palma, Vincenzo Manserra	Aggiornamento del documento	26/05/2022
1.0	Alessandro Bergamo, Rosario Di Palma, Vincenzo Manserra	Test di regressione e stesura finale del documento	27/05/2022

Tabella 1: Revision History

---

## 1 Introduzione

Lo scopo di questo documento è quello di presentare l'attività di testing svolta sul sistema pre-esistente "ASATTD", identificando gli elementi e le funzionalità da testare e le strategie di testing impiegate.

## 2 Panoramica del sistema

Un "Self Admitted Technical Debt" è un modo per segnalare di aver pubblicato del codice incompleto, che richiede un rifacimento, che produce errori o che è stato realizzato per permettere all'intero sistema di poter funzionare, ma con la premessa che saranno necessarie future modifiche.

Questo plugin mira ad identificarli e segnalarli attraverso l'analisi testuale del commit message.

## 3 Funzionalità da testare

### 3.1 Requisiti funzionali

Di seguito vengono esplicitati i requisiti funzionali del sistema ASATTD.

- **FR\_1:** Il sistema deve essere in grado di recuperare i commit di una data repository
- **FR\_2:** Il sistema deve essere in grado di istruire il classificatore
- **FR\_3:** Il sistema deve essere in grado di eseguire l'analisi testuale dei commit message per evidenziare potenziali Self-Admitted Technical Debt

## 4 Approccio

Per la fase di testing si è deciso di effettuare dei casi di test che mirassero sia a testare le vecchie funzionalità, infatti tramite category partition andiamo a testare i nostri requisiti funzionali, ma il focus principale è stato posto sui test dei tipi di formato, sia in input che in output, in modo tale da poterli poi confrontare con i tipi di formati del nuovo sistema con l'obiettivo di ricercare eventuali discrepanze.

## 5 Testing del modello

### 5.1 Testing sul formato

In questa sezione si andrà ad esplicitare i casi di test dei requisiti funzionali; successivamente, si va andrà ad esplicitare i casi di test riguardanti il formato dell'input e dell'output in modo tale da poterli confrontare, in seguito alle modifiche, con il nuovo sistema. In base ai requisiti funzionali, sono stati individuati i seguenti casi di test:

- **TC\_1:** Il sistema deve essere in grado di recuperare i commit di una data repository
- **TC\_2:** Il sistema deve essere in grado di istruire il classificatore
- **TC\_3:** Il sistema deve poter analizzare testualmente i commit message e evidenziare i possibili Self-Admitted Technical Debt

### 5.2 TC\_1: Il sistema deve essere in grado di recuperare i commit di una data repository

Parametro: PATH della repository	
CATEGORIE	SCELTE
Esiste[er]	<ol style="list-style-type: none"> <li>1. Path non valido</li> <li>2. Path valido</li> </ol>

IDENTIFICATIVO	COMBINAZIONE	ESITO
TC_1_1	er1	Path non valido
TC_1_2	er2	Comando eseguito con successo <ul style="list-style-type: none"> <li>• return type: List&lt;Commit&gt;</li> </ul>

### 5.3 TC\_2: Il sistema deve essere in grado di istruire il classificatore

Parametro: TrainingSet	
CATEGORIE	SCELTE
Ben formato [tb]	<ol style="list-style-type: none"> <li>1. Il TrainingSet non esiste</li> <li>2. Il TrainingSet è ben formato</li> </ol>

5.4 TC\_3: Il sistema deve poter analizzare testualmente i commit message e evidenziare i possibili Self-Admitted Technical Debt

IDENTIFICATIVO	COMBINAZIONE	ESITO
TC_2_1	tb1	Il TrainingSet non esiste
TC_2_2	tb2	Comando eseguito con successo <ul style="list-style-type: none"> <li>• return type: Classifier</li> </ul>

5.4 TC\_3: Il sistema deve poter analizzare testualmente i commit message e evidenziare i possibili Self-Admitted Technical Debt

Parametro: List<Commit>	
CATEGORIE	SCelte
Vuota[lv]	<ol style="list-style-type: none"> <li>1. La lista è vuota</li> <li>2. La lista ha almeno un commit</li> </ol>

IDENTIFICATIVO	COMBINAZIONE	ESITO
TC_3_1	lv1	Non ci sono commit da classificare
TC_3_2	lv2	Comando eseguito con successo <ul style="list-style-type: none"> <li>• return type: List&lt;Commit&gt;</li> </ul>