



# Analisi dei Requisiti

Versione: 0.2.0

04/12/2024

**Redattori**

Malik Giafar Mohamed

**Verifica**

Ion Cainareanu, Maria Fuensanta Trigueros Hernandez

**Approvazione**

Ion Cainareanu, Maria Fuensanta Trigueros Hernandez

**Uso**

Esterno

nextsoftpadova@gmail.com

# Registro dei cambiamenti

Versione	Data	Autore	Descrizione	Verifica	Approvazione
0.2.0	30/12/2024	Ion Cainareanu	Stesura dell'Introduzione e Descrizione	Stefano Baso, Malik Giafar Mohamed	Stefano Baso, Malik Giafar Mohamed
0.1.1	04/12/2024	Luca Parise	Aggiunta indice e creazione struttura tabella per use case	Malik Giafar Mohamed	Malik Giafar Mohamed
0.1.0	23/11/2024	Malik Giafar Mohamed	Creazione Documento	Ion Cainareanu, Maria Fuensanta Trigueros Hernandez	Ion Cainareanu, Maria Fuensanta Trigueros Hernandez

## Indice

1 Introduzione .....	3
1.1 Scopo del documento .....	3
1.2 Scopo del prodotto .....	3
1.3 Glossario .....	3
2 Descrizione .....	3
2.1 Obiettivi del prodotto .....	3
2.2 Funzionalità del prodotto .....	3
2.3 Utenti e caratteristiche .....	4
3 Use Case .....	4
3.1 Obiettivi .....	4
3.2 Attori .....	4
3.3 UC_1 - titolo UC_1 .....	4
3.4 UC_2 - titolo UC_2 .....	4
3.5 UC_3 - titolo UC_3 .....	4
3.5.1 UC_3.1 .....	5
3.6 UC_i .....	5
3.7 UC_n .....	5
4 Requisiti .....	5
5 Elenco delle immagini .....	5
6 Elenco delle tabelle .....	5

# 1 Introduzione

## 1.1 Scopo del documento

Lo scopo del presente documento è fornire una descrizione completa e dettagliata degli obiettivi, delle funzionalità e delle caratteristiche tecniche del progetto **Requirement Tracker - VS Code Plug-in**, con particolare attenzione all'utilizzo dell'UML per la modellazione dei casi d'uso. Il documento funge da riferimento per tutti gli stakeholder coinvolti, descrivendo il contesto operativo, i requisiti funzionali e non funzionali, nonché le linee guida tecnologiche necessarie per lo sviluppo del plug-in. I casi d'uso saranno descritti utilizzando una struttura standardizzata, che includerà il nominativo del caso, gli attori principali, le precondizioni, le postcondizioni, lo scenario principale e gli eventuali scenari alternativi o sottocasi. Questa struttura garantisce chiarezza e coerenza, facilitando la comprensione e la tracciabilità delle funzionalità principali del sistema. Il documento intende inoltre fornire una visione condivisa del progetto, ponendo le basi per una pianificazione e un'implementazione efficaci.

## 1.2 Scopo del prodotto

Lo scopo di **Requirement Tracker - VS Code Plug-in** è affrontare il problema della complessità nella gestione e nel tracciamento dei requisiti nei progetti software di grandi dimensioni. Nei codebase estesi, la verifica manuale della copertura e dell'implementazione dei requisiti è un processo lungo e soggetto a errori, spesso complicato dalla qualità insufficiente con cui i requisiti stessi vengono definiti. Questo può portare a malintesi e problemi durante l'implementazione, compromettendo l'allineamento tra specifiche e funzionalità sviluppate. Il plug-in mira a risolvere queste difficoltà automatizzando il tracciamento dei requisiti nel codice sorgente, migliorando la qualità della loro definizione e semplificando l'identificazione delle aree di mancata o errata implementazione. In particolare, offre strumenti per integrare requisiti tecnici derivati da manuali e datasheet di componenti hardware, fornendo analisi automatizzate e suggerimenti per rendere i requisiti più chiari, specifici e strutturati. Grazie a questo, sviluppatori potranno garantire una gestione più efficace dei requisiti, riducendo errori e aumentando la coerenza tra specifiche e implementazione.

## 1.3 Glossario

Per evitare ambiguità relative alle terminologie utilizzate è stato creato un documento denominato **Glossario**. Questo documento comprende tutti i termini tecnici scelti dai membri del gruppo e utilizzati nei vari documenti con le relative definizioni. Tutti i termini inclusi in questo glossario vengono segnalati all'interno del documento con l'apice <sup>G</sup> accanto alla parola.

# 2 Descrizione

## 2.1 Obiettivi del prodotto

L'obiettivo del progetto è realizzare un plug-in per Visual Studio Code che consenta di tracciare e verificare l'implementazione dei requisiti di progetto, basandosi su analisi automatizzate del codice sorgente e sui requisiti tecnici espressi in documenti di riferimento, mediante l'utilizzo di tecnologie avanzate come modelli LLM<sup>G</sup> di AI<sup>G</sup>. Il plug-in sarà supportato da API REST<sup>G</sup> che si interfacciano con Ollama<sup>G</sup>, fornendo un'infrastruttura flessibile e scalabile per l'integrazione di modelli di AI e garantendo un'elaborazione efficiente e sicura delle analisi richieste.

## 2.2 Funzionalità del prodotto

Il plug-in sarà utilizzato dal programmatore per analizzare i requisiti implementati nel codice sorgente. Sia i requisiti che il codice saranno analizzati da vari modelli LLM reperibili attraverso la piattaforma di Ollama, grazie alle API REST che interagiscono con Ollama.

Le funzionalità implementate nell'applicazione includono:

- Importazione del file dei requisiti in formato CSV<sup>G</sup>
- Richiesta di analisi dei requisiti tramite un modello LLM;
- Valutazione qualitativa dei requisiti;

- Visualizzazione grafica dei risultati dell'analisi;
- Filtraggio dei risultati dell'analisi;
- Possibilità di ripetere l'analisi di un requisito specifico;
- Esportazione dei risultati dell'analisi in formato CSV;
- Verifica dell'implementazione dei requisiti nel codice sorgente;
- Analisi semantica dei requisiti e del codice sorgente;
- Suggerimenti per migliorare la qualità dei requisiti e del codice.

## 2.3 Utenti e caratteristiche

In seguito a un incontro con il proponente, è stato discusso come il plug-in possa essere utilizzato principalmente da un utente che ricopre il ruolo di programmatore. Di conseguenza, si è deciso di focalizzare le funzionalità del plug-in per rispondere alle esigenze di questa categoria di utenti. È stato inoltre specificato che non devono essere fatte assunzioni sulle competenze tecniche dell'utente riguardo all'uso di Visual Studio Code. Pertanto, il plug-in deve essere progettato per essere il più intuitivo possibile, con un processo di installazione semplice e accessibile.

## 3 Use Case

### 3.1 Obiettivi

### 3.2 Attori

### 3.3 UC\_1 - titolo UC\_1

-> qui può essere inserita una immagine utile alla spiegazione dell'use case

codice uc	Titolo uc
Attore primario	
Precondizioni	
Postcondizioni	
Scenario principale	
Estensioni	(Scenari alternativi)

### 3.4 UC\_2 - titolo UC\_2

codice uc	Titolo uc
Attore primario	
Precondizioni	
Postcondizioni	
Scenario principale	
Estensioni	(Scenari alternativi)

### 3.5 UC\_3 - titolo UC\_3

codice uc	Titolo uc
Attore primario	
Precondizioni	
Postcondizioni	
Scenario principale	
Estensioni	(Scenari alternativi)

**3.5.1 UC\_3.1**

<b>codice uc</b>	<b>Titolo uc</b>
Attore primario	
Precondizioni	
Postcondizioni	
Scenario principale	
Estensioni	(Scenari alternativi)

**3.6 UC\_i**

<b>codice uc</b>	<b>Titolo uc</b>
Attore primario	
Precondizioni	
Postcondizioni	
Scenario principale	
Estensioni	(Scenari alternativi)

**3.7 UC\_n****4 Requisiti****5 Elenco delle immagini****6 Elenco delle tabelle**