



# Glossario

## Glossario Tecnico del capitolato di Ing. del Software 2024-2025

Versione: 2.0.0

26/11/2024

---

### Redattori

Malik Giafar Mohamed

Marco Perazzolo

Maria Fuensanta Trigueros Hernandez

---

### Verifica

Ion Cainareanu

Maria Fuensanta Trigueros Hernandez

Stefano Baso

---

### Approvazione

Luca Parise

---

### Uso

Esterno

[nextsoftpadova@gmail.com](mailto:nextsoftpadova@gmail.com)

# Registro dei cambiamenti

Versione	Data	Autore	Descrizione	Verifica
2.0.0	024/04/2025	Maria Fuensanta Trigueros Hernandez	Aggiornamento e aggiunta di nuovi termini al glossario	
1.0.0	08/03/2025	Marco Perazzolo	Aggiornamento e aggiunta di nuovi termini al glossario	Stefano Baso
0.2.0	25/01/2025	Maria Fuensanta Trigueros Hernandez	Stesura iniziale del glossario	Marco Perazzolo
0.1.0	23/11/2024	Malik Giafar Mohamed	Creazione Documento	Ion Cainareanu, Maria Fuensanta Trigueros Hernandez

## Indice

Introduzione .....	7
A .....	8
Ad hoc .....	8
Agile .....	8
AI / IA (Intelligenza Artificiale) .....	8
Analisi dei Requisiti .....	8
Analisi dei Rischi .....	8
Analisi semantica .....	8
Analista .....	8
API (Application Programming Interface) .....	8
Artefatti .....	8
Attore .....	8
B .....	9
Backend .....	9
Baseline .....	9
Best Practices .....	9
Branch .....	9
Budget di Progetto .....	9
Bug .....	9
Build .....	9

C .....	10
CamelCase .....	10
Capitolato .....	10
Change Request Form .....	10
Checklist .....	10
Codebase .....	10
Codifica .....	10
Commit .....	10
Committente .....	10
Configuration items .....	10
Consuntivo di Periodo .....	10
CSV (Comma-Separated Values) .....	10
Container .....	11
D .....	12
Debugging .....	12
Deployment .....	12
Design .....	12
Design Pattern .....	12
Diagramma UML .....	12
Dipendenza .....	12
Driver .....	12
Docker .....	12
Dockerfile .....	12
E .....	13
EAC (Estimated At Completion) .....	13
Economicità .....	13
Efficienza .....	13
Endpoint .....	13
Entry .....	13
ETC (Estimated To Completion) .....	13
EV (Earned Value) .....	13
F .....	14
Feature .....	14
Feedback .....	14
File explorer .....	14
Filesystem .....	14
Fix .....	14
Fornitura .....	14
Frontend .....	14
File System .....	14
G .....	15
GitHub .....	15
Github Issues .....	15
Github Projects .....	15
H .....	16

I .....	17
IDE .....	17
Issue .....	17
ITS (Issue Tracking System) .....	17
J .....	18
JavaScript .....	18
K .....	19
Knowledge .....	19
L .....	20
Librerie .....	20
Linearmente indipendenti .....	20
LLM (Large Language Model) .....	20
M .....	21
Mapping .....	21
Merge conflicts .....	21
Metrica .....	21
Mitigazione dei Rischi .....	21
Modello di Sviluppo .....	21
Moduli software .....	21
MPDS (Metriche di Processo e Sviluppo) .....	21
MPC (Minimum Predictive Capability) .....	21
MVP (Minimum Viable Product) .....	21
N .....	22
NodeJS .....	22
Npm .....	22
O .....	23
Ollama .....	23
P .....	24
Package .....	24
Parametri .....	24
Pattern .....	24
Pianificazione .....	24
Piano di Contingenza .....	24
Piano di Qualifica .....	24
Plug-in .....	24
Postcondizione .....	24
Precondizione .....	24
Preventivo .....	24
Programmatori .....	24
Progettisti .....	24
Prompt .....	25
Proof of Concept (PoC) .....	25
Pull Request .....	25
PV (Planned Value) .....	25
Q .....	26

Qualità di Processo .....	26
R .....	27
RC (Requirement Coverage) .....	27
Refactoring .....	27
Report .....	27
Repository .....	27
Requisito .....	27
Requisito Desiderabile .....	27
Requisito di Qualità .....	27
Requisito Facoltativo .....	27
Requisito Funzionale .....	27
Requisito Non Funzionale .....	27
Requisito Obbligatorio .....	28
Requisito Prestazionale .....	28
Requirement Tracker .....	28
Responsabile di progetto .....	28
Root directory .....	28
S .....	29
Scenario Principale .....	29
Scenari Alternativi .....	29
Script .....	29
Single Responsibility Principle .....	29
Sprint .....	29
Stakeholder .....	29
Stub .....	29
SV (Schedule Variance) .....	29
Server .....	29
Soglia di accettazione .....	29
T .....	30
Task .....	30
Team .....	30
Template .....	30
Test .....	30
Test di unità .....	30
Test funzionali .....	30
Token .....	30
Tool .....	30
Tracciamento dei Requisiti .....	30
TypeScript .....	30
U .....	31
UML (Unified Modeling Language) .....	31
Usabilità .....	31
Use Case (Caso d'Uso) .....	31
Utility .....	31
V .....	32

Versionamento .....	32
Verifica .....	32
Vincolo .....	32
Visual Studio Code (VS Code) .....	32
VLN (Presenza di Vulnerabilità) .....	32
Validazione .....	32
Vsce .....	32
W .....	33
Way of Working .....	33
X .....	34
Y .....	35
Z .....	36
.....	37
.vsix .....	37

## Introduzione

Il presente glossario ha l'obiettivo di definire e chiarire i termini chiave utilizzati nel progetto **Requirement Tracker Plug-in**. Il suo scopo è standardizzare il linguaggio e facilitare la comprensione dei concetti, garantendo una comunicazione chiara ed efficace tra tutte le persone coinvolte nel progetto.

Questo documento servirà come riferimento per evitare fraintendimenti, migliorare la collaborazione e assicurare l'uso corretto della terminologia all'interno del nostro contesto di applicazione. I termini sono organizzati in ordine alfabetico per una ricerca rapida e semplice.

## A

### **Ad hoc**

Creato per soddisfare una determinata esigenza.

### **Agile**

Metodologia di sviluppo iterativo che enfatizza iterazioni brevi, feedback continui e l'adattamento al cambiamento, permettendo un'elevata flessibilità.

### **AI / IA (Intelligenza Artificiale)**

Settore dell'informatica che si occupa di sviluppare software / computer che siano in grado di riprodurre il funzionamento di alcune capacità della mente umana.

### **Analisi dei Requisiti**

Processo che identifica, documenta e gestisce i requisiti di un sistema software.

### **Analisi dei Rischi**

Processo di identificazione e valutazione dei possibili problemi che possono insorgere durante lo sviluppo del progetto.

### **Analisi semantica**

Valutazione del contenuto testuale dei requisiti per verificarne chiarezza, coerenza e correttezza.

### **Analista**

Figura professionale che individua e definisce i casi d'uso ed i requisiti del prodotto software.

### **API (Application Programming Interface)**

Interfaccia di programmazione che consente a software diversi di comunicare tra loro, scambiandosi funzionalità, servizi, tecnologie e dati.

### **Artefatti**

Documenti, codice, e altri output generati durante lo sviluppo del software.

### **Attore**

Persona o sistema che interagisce con il software nell'ambito di un caso d'uso.



## B

### **Backend**

La parte del software che gestisce la logica di business, i dati e le operazioni lato server.

### **Baseline**

Versione di riferimento di un prodotto software, indica un punto di arrivo tecnico irreversibile, che viene stabilito come standard. Usata come base per modifiche o sviluppi successivi.

### **Best Practices**

Insieme di linee guida e metodologie raccomandate per migliorare la qualità e l'efficienza del lavoro.

### **Branch**

Ramo indipendente di sviluppo in un sistema di controllo di versione, utilizzata per implementare nuove funzionalità o correzione di bug o testing senza intaccare il codice principale funzionante.

### **Budget di Progetto**

Stima dei costi necessari per realizzare un progetto.

### **Bug**

Errore o malfunzionamento in un software.

### **Build**

Processo di compilazione e preparazione del codice sorgente in una versione eseguibile o distribuibile.

## C

### **CamelCase**

Pratica di scrivere parole composte o frasi unendo tutte le parole tra loro, ma lasciando le loro iniziali maiuscole, eccetto per la prima lettera assoluta della frase che rimane in minuscolo

### **Capitolato**

Documento ufficiale che descrive i requisiti e le specifiche di un progetto software fornito dal proponente.

### **Change Request Form**

Modulo utilizzato per richiedere modifiche alle specifiche o al progetto.

### **Checklist**

Lista di controllo per verificare il completamento di compiti o il rispetto di standard.

### **Codebase**

Insieme del codice sorgente di un progetto.

### **Codifica**

Processo di scrittura del codice sorgente secondo le specifiche definite nella fase di progettazione.

### **Commit**

Salvataggio di modifiche al codice sorgente in un sistema di controllo di versione, viene creato con il comando `'git commit'`.

### **Committente**

La persona o l'organizzazione che commissiona il progetto.

### **Configuration items**

Elementi di configurazione che devono essere gestiti e controllati nel progetto.

### **Consuntivo di Periodo**

Analisi delle variazioni di pianificazione e costi rispetto alle stime iniziali, con indicazione delle misure correttive adottate.

### **CSV (Comma-Separated Values)**

Formato di file di testo che rappresenta dati tabellari separati da virgole.

**Container**

Ambiente isolato che contiene tutto il necessario per eseguire un'applicazione.

## D

### **Debugging**

Processo di individuazione, analisi e correzione degli errori nel software.

### **Deployment**

Processo di rilascio e installazione del software in un ambiente di produzione.

### **Design**

Processo di definizione dell'architettura e della struttura di un sistema software.

### **Design Pattern**

Soluzione standardizzata a problemi ricorrenti nel design del software.

### **Diagramma UML**

Rappresentazione grafica della struttura e del comportamento di un sistema software.

### **Dipendenza**

Relazione in cui la modifica di un modulo o componente necessita della modifica di un altro modulo o componente del software.

### **Driver**

Software che controlla e gestisce l'interazione con hardware o altri sistemi.

### **Docker**

Piattaforma di containerizzazione che consente di creare, distribuire e gestire ambienti isolati.

### **Dockerfile**

File di testo che contiene istruzioni su come creare un'immagine Docker.

## E

### **EAC (Estimated At Completion)**

Stima del costo totale di completamento del progetto basata sulle prestazioni attuali.

### **Economicità**

Capacità del progetto di essere realizzato ed eseguito a costi contenuti.

### **Efficienza**

Capacità di un software o di un processo di eseguire i compiti richiesti utilizzando il numero minimo di risorse.

### **Endpoint**

Indirizzo (IP o URL) del server a cui vengono inviate le richieste API.

### **Entry**

Singolo elemento o riga di dati in una struttura (es. file, tabella).

### **ETC (Estimated To Completion)**

Stima del costo rimanente per completare il progetto.

### **EV (Earned Value)**

Valore del lavoro effettivamente completato in un determinato momento rispetto alla pianificazione.

## **F**

### **Feature**

Funzionalità o caratteristica specifica di un software.

### **Feedback**

Informazioni e riscontri forniti dagli utenti o dal sistema per migliorare il prodotto.

### **File explorer**

Strumento che permette all'utente di navigare tra file e cartelle del sistema in uso.

### **Filesystem**

Struttura usata dal sistema operativo per organizzare e gestire i file in un dispositivo.

### **Fix**

Correzione di un malfunzionamento o di un errore in un documento o nel codice sorgente.

### **Fornitura**

Processo di consegna del software al cliente, inclusi test finali e rilascio della documentazione.

### **Frontend**

La parte del software che interagisce direttamente con l'utente, tipicamente l'interfaccia grafica. Nel nostro caso, il frontend è rappresentato da Visual Studio Code.

### **File System**

Struttura dati deputata alla gestione e all'archiviazione dei file su un computer.

## G

### **GitHub**

Piattaforma per il versionamento del codice sorgente e per la gestione collaborativa dei progetti software.

### **Github Issues**

Sistema di tracciamento degli errori e delle problematiche presente su GitHub.

### **Github Projects**

Piattaforma per la gestione dei progetti integrata in GitHub, che consente la gestione, creazione e assegnazione di issues.

## H



## **I**

### **IDE**

Ambiente integrato per scrivere, testare e fare debug del codice.

### **Issue**

Ticket utilizzato in strumenti di gestione dei progetti per tracciare problemi, attività o richieste di miglioramento.

### **ITS (Issue Tracking System)**

Sistema utilizzato per il tracciamento delle issue del progetto.

## J

### **JavaScript**

Linguaggio di programmazione multi paradigma orientato agli eventi.

## K

### **Knowledge**

Insieme di informazioni tecniche e documentazione di supporto, come datasheet, manuali e specifiche tecniche, utilizzate per migliorare la comprensione e l'implementazione dei requisiti di un sistema. Nel contesto del plug-in, il termine si riferisce alla capacità di utilizzare tali documenti per garantire un'analisi efficace dei requisiti.

## L

### **Librerie**

Insieme di funzioni, classi e metodi messi a disposizione per supportare lo sviluppo software.

### **Linearmente indipendenti**

In un grafo, due o più cammini sono detti linearmente indipendenti se ciascuno contiene almeno un arco o nodo che non compare negli altri. Nessun cammino, quindi, può essere ottenuto come combinazione degli altri.

### **LLM (Large Language Model)**

Modelli di intelligenza artificiale addestrati per comprendere ed elaborare il linguaggio naturale.

## M

### **Mapping**

Associazione tra requisiti e le relative porzioni di codice che li implementano.

### **Merge conflicts**

Conflitti che sorgono quando due modifiche contrastanti vengono apportate allo stesso file in un repository di versionamento e che richiede un intervento manuale per la risoluzione.

### **Metrica**

Misura quantitativa utilizzata per valutare aspetti del software, come performance, affidabilità o manutenibilità.

### **Mitigazione dei Rischi**

Strategie applicate per ridurre o eliminare l'impatto dei rischi identificati nel progetto.

### **Modello di Sviluppo**

Approccio metodologico utilizzato per organizzare il lavoro e garantire un avanzamento strutturato del progetto.

### **Moduli software**

Componenti o parti indipendenti di un sistema che possono essere sviluppate, testate e mantenute separatamente.

### **MPDS (Metriche di Processo e Sviluppo)**

Insieme di metriche utilizzate per valutare la qualità del software nei suoi vari aspetti.

### **MPC (Minimum Predictive Capability)**

Metrica che valuta la capacità minima di previsione di un modello o sistema.

### **MVP (Minimum Viable Product)**

Versione minima di un prodotto con funzionalità essenziali per testarne la fattibilità.

## N

### **NodeJS**

Ambiente di esecuzione open source per JavaScript, progettato per creare applicazioni scalabili lato server.

### **Npm**

Gestore di pacchetti per Node.js, utilizzato per installare librerie e strumenti.

## O

### **Ollama**

Piattaforma per l'integrazione di modelli LLM (Large Language Models) nel processo di analisi dei requisiti.

## P

### **Package**

Insieme di moduli o librerie raggruppati per facilitare la distribuzione e l'utilizzo.

### **Parametri**

Variabili passate a una funzione o metodo per indurre un comportamento.

### **Pattern**

Soluzione riutilizzabile a problemi ricorrenti nel design del software.

### **Pianificazione**

Processo di suddivisione del progetto in fasi con definizione di risorse, tempistiche e responsabilità.

### **Piano di Contingenza**

Strategia alternativa da applicare in caso di problemi imprevisti nel progetto.

### **Piano di Qualifica**

Documento che definisce i criteri e le attività per la verifica della qualità del progetto.

### **Plug-in**

Estensione di un software che aggiunge nuove funzionalità senza modificarne il core.

### **Postcondizione**

Stato atteso dopo l'esecuzione di un caso d'uso.

### **Precondizione**

Stato richiesto affinché un caso d'uso possa essere eseguito correttamente.

### **Preventivo**

Stima dei costi del progetto in termini economici e di tempo.

### **Programmatori**

Sviluppatori che scrivono e mantengono il codice sorgente.

### **Progettisti**

Figure responsabili della definizione dell'architettura e del design del sistema



**Prompt**

Testo o comando fornito a un modello LLM per generare una risposta.

**Proof of Concept (PoC)**

Implementazione preliminare del progetto per verificarne la fattibilità.

**Pull Request**

Proposta di integrazione di modifiche in un repository di versionamento.

**PV (Planned Value)**

Valore pianificato del lavoro da completare fino a una determinata data di riferimento.

## Q

### **Qualità di Processo**

Misura dell'efficienza e dell'efficacia dei processi seguiti nello sviluppo del software.

## R

### **RC (Requirement Coverage)**

Percentuale di requisiti soddisfatti rispetto al totale.

### **Refactoring**

Processo di ristrutturazione del codice per migliorarne la struttura senza modificarne il comportamento.

### **Report**

Documento che sintetizza i risultati di un'analisi o di un processo.

### **Repository**

Archivio di codice sorgente o di documenti, gestito con strumenti come GitHub

### **Requisito**

Condizioni o capacità necessarie che un sistema o prodotto deve possedere per soddisfare esigenze o vincoli specifici.

### **Requisito Desiderabile**

Requisito non indispensabile, ma la cui implementazione è raccomandata poiché apporta valore aggiunto in termini di usabilità, efficienza o qualità.

### **Requisito di Qualità**

Descrive attributi qualitativi del software, come affidabilità, manutenibilità, sicurezza, usabilità e accessibilità, che ne determinano il livello di accettabilità.

### **Requisito Facoltativo**

Descrive una funzionalità aggiuntiva non essenziale, la cui implementazione può essere omessa senza compromettere il funzionamento principale del sistema.

### **Requisito Funzionale**

Specifica un comportamento o una funzionalità che il sistema deve fornire, descrivendo le interazioni tra l'utente e il software per soddisfare un obiettivo.

### **Requisito Non Funzionale**

Proprietà di tipo qualitativo che un sistema deve soddisfare, come prestazioni, sicurezza o scalabilità, non riguardante direttamente una funzionalità specifica.

**Requisito Obbligatorio**

Specifica una caratteristica essenziale del sistema che deve essere implementata affinché il prodotto sia conforme alle specifiche e alle richieste del proponente.

**Requisito Prestazionale**

Definisce vincoli relativi alle prestazioni del sistema, come tempi di risposta, capacità di elaborazione, consumo di risorse e scalabilità.

**Requirement Tracker**

Plug-in per Visual Studio Code che automatizza il tracciamento dei requisiti nel codice sorgente.

**Responsabile di progetto**

Figura incaricata di gestire e coordinare le attività e le risorse del progetto.

**Root directory**

Cartella principale di un progetto, da cui si parte per organizzare tutti i file e le sottocartelle.

## S

### **Scenario Principale**

Sequenza di passaggi che descrive il normale flusso di esecuzione di un caso d'uso.

### **Scenari Alternativi**

Possibili variazioni nel flusso di esecuzione di un caso d'uso.

### **Script**

Sequenza di comandi in un linguaggio di scripting per automatizzare operazioni.

### **Single Responsibility Principle**

Principio di design che stabilisce che ogni modulo o classe deve avere un solo compito (responsabilità).

### **Sprint**

Intervallo di tempo breve durante il quale viene sviluppata una parte del lavoro in metodologie Agile.

### **Stakeholder**

Persona o gruppo interessato all'andamento e ai risultati del progetto.

### **Stub**

Implementazione minima usata come sostituto per un componente durante il testing.

### **SV (Schedule Variance)**

Differenza tra il valore guadagnato (EV) e il valore pianificato (PV), indica eventuali ritardi o anticipi.

### **Server**

Dispositivo fisico o sistema informatico di elaborazione e gestione del traffico di informazioni, fornisce, a livello logico e fisico, un qualunque tipo di servizio ad altre componenti.

### **Soglia di accettazione**

Valore numerico che definisce il punteggio minimo affinché un requisito venga considerato "passato" dopo un'analisi.

## T

### **Task**

Attività specifica da svolgere all'interno del progetto.

### **Team**

Gruppo di persone che collaborano nello sviluppo del progetto.

### **Template**

Modello predefinito per la creazione di documenti o codice.

### **Test**

Attività per verificare il funzionamento corretto del software.

### **Test di unità**

Test che verificano il funzionamento di singole componenti del codice, come moduli, classi o funzioni.

### **Test funzionali**

Test che verificano il corretto funzionamento delle funzionalità del software.

### **Token**

Unità minima di testo elaborata da un modello LLM.

### **Tool**

Strumento software utilizzato per supportare lo sviluppo, il testing o la produzione di documenti.

### **Tracciamento dei Requisiti**

Processo che collega ogni requisito alla sua implementazione nel codice per garantirne la copertura.

### **TypeScript**

Linguaggio di programmazione che estende JavaScript con tipi statici.

## U

### **UML (Unified Modeling Language)**

Linguaggio standard per la modellazione grafica di sistemi software.

### **Usabilità**

Facilità di utilizzo di un'interfaccia o di un sistema software

### **Use Case (Caso d'Uso)**

Rappresentazione grafica e testuale che descrive come gli utenti interagiscono con un sistema e viceversa.

### **Utility**

Strumento ausiliario che semplifica compiti ripetitivi o complessi.

## V

### **Versionamento**

Processo di gestione delle versioni di un software o documento.

### **Verifica**

Processo per determinare se il prodotto è stato sviluppato correttamente rispetto ai requisiti specificati.

### **Vincolo**

Impone limitazioni progettuali o implementative, come l'uso di tecnologie specifiche, la compatibilità con sistemi esistenti, il rispetto di normative o restrizioni di budget e tempistiche.

### **Visual Studio Code (VS Code)**

Editor di codice sviluppato da Microsoft, utilizzato per la programmazione e la gestione del codice sorgente.

### **VLN (Presenza di Vulnerabilità)**

Numero di vulnerabilità di sicurezza ancora presenti nel codice.

### **Validazione**

Processo che assicura che il prodotto soddisfi i bisogni e le aspettative dell'utente finale.

### **Vsce**

Strumento a riga di comando per creare e gestire pacchetti .vsix per Visual Studio Code.



## W

### **Way of Working**

Modalità di organizzazione e gestione delle attività all'interno di un progetto.

## **X**

## Y

## **Z**

- 

**.vsix**

Formato di pacchetto utilizzato per distribuire estensioni su Visual Studio Code.