



# Glossario

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

Versione: 1.0.0

26/11/2024

**Redattori**

Malik Giafar Mohamed

**Verifica**

Ion Cainareanu  
Maria Fuensanta Trigueros Hernandez

**Approvazione**

**Uso**

Esterno

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

# Registro dei cambiamenti

Versione	Data	Autore	Descrizione	Verifica
1.0.0	08/03/2025	Marco Perazzolo	Aggiornamento e aggiunta di nuovi termini al glossario	
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 .....	6
A .....	7
API (Application Programming Interface) .....	7
Agile .....	7
Analisi dei Requisiti .....	7
Attore .....	7
Analisi dei Rischi .....	7
Analisi semantica .....	7
Artefatti .....	7
Approvatore .....	7
Analista .....	7
Ad hoc .....	7
AI / IA (Intelligenza Artificiale) .....	7
B .....	8
Baseline .....	8
Best Practices .....	8
Bug .....	8
Branch .....	8
Budget di Progetto .....	8
Backend .....	8
C .....	9
CamelCase .....	9
Capitolato .....	9
Change Request Form .....	9
Checklist .....	9
Codebase .....	9
Codifica .....	9
Commit .....	9
Committente .....	9
Configuration items .....	9
Consuntivo di Periodo .....	9
CSV (Comma-Separated Values) .....	9
D .....	10

Debugging .....	10
Design .....	10
Design Pattern .....	10
Diagramma UML .....	10
Dipendenza .....	10
Deployment .....	10
Driver .....	10
E .....	11
Endpoint .....	11
Entry .....	11
Economicità .....	11
Efficienza .....	11
EAC (Estimated At Completion) .....	11
EV (Earned Value) .....	11
ETC (Estimated To Completion) .....	11
F .....	12
File explorer .....	12
Feature .....	12
Feedback .....	12
Fix .....	12
Fornitura .....	12
Frontend .....	12
Filesystem .....	12
G .....	13
GitHub .....	13
Github Projects .....	13
Github Issues .....	13
H .....	14
I .....	15
IDE .....	15
Issue .....	15
ITS (Issue Tracking System) .....	15
J .....	16
K .....	17
Knowledge .....	17
L .....	18
Librerie .....	18
Linearmente indipendenti .....	18
LLM (Large Language Model) .....	18
M .....	19
Mapping .....	19
Merge conflicts .....	19
MVP (Minimum Viable Product) .....	19
Metrica .....	19
MPC (Minimum Predictive Capability) .....	19
MPDS (Metriche di Processo e Sviluppo) .....	19
Mitigazione dei Rischi .....	19
Modello di Sviluppo .....	19
Moduli software .....	19
N .....	20
NodeJS .....	20

Normazione .....	20
O .....	21
Ollama .....	21
P .....	22
Package .....	22
Parametri .....	22
Pattern .....	22
Plug-in .....	22
Precondizione .....	22
Prompt .....	22
Progettisti .....	22
Programmatori .....	22
Postcondizione .....	22
PV (Planned Value) .....	22
Piano di Contingenza .....	22
Piano di Qualifica .....	22
Pianificazione .....	22
Preventivo .....	22
Proof of Concept (PoC) .....	22
Pull Request .....	22
Q .....	23
Qualità di Processo .....	23
R .....	24
RC (Requirement Coverage) .....	24
Refactoring .....	24
Requisito .....	24
Requisito Obbligatorio .....	24
Requisito Facoltativo .....	24
Requisito Desiderabile .....	24
Requisito Prestazionale .....	24
Requisito di Qualità .....	24
Requisito Funzionale .....	24
Requisito Non Funzionale .....	24
Requirement Tracker .....	24
Report .....	24
Repository .....	24
Responsabile di progetto .....	24
S .....	25
Script .....	25
Stakeholder .....	25
Scenario Principale .....	25
Scenari Alternativi .....	25
Single Responsibility Principle .....	25
Sprint .....	25
Stub .....	25
SV (Schedule Variance) .....	25
T .....	26
Task .....	26
Test .....	26
Test di unità .....	26
Test funzionali .....	26

Team .....	26
Template .....	26
Token .....	26
TypeScript .....	26
Tracciamento dei Requisiti .....	26
Tool .....	26
U .....	27
Usabilità .....	27
Use Case (Caso d'Uso) .....	27
UML (Unified Modeling Language) .....	27
Utility .....	27
V .....	28
Validazione .....	28
Visual Studio Code (VS Code) .....	28
Verifica .....	28
Vincolo .....	28
Versionamento .....	28
VLN (Presenza di Vulnerabilità) .....	28
W .....	29
Way of Working .....	29
X .....	30
Y .....	31
Z .....	32

## **Introduzione**

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

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

## A

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

Interfaccia di programmazione che consente a software diversi di comunicare tra loro.

### **Agile**

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

### **Analisi dei Requisiti**

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

### **Attore**

Persona o sistema che interagisce con il software nei casi d'uso.

### **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, completezza e correttezza.

### **Artefatti**

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

### **Approvatore**

Figura responsabile della validazione finale dei documenti e delle specifiche di progetto.

### **Analista**

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

### **Ad hoc**

Creato specificamente per soddisfare una determinata esigenza.

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

Tecnologie che consentono ai sistemi di svolgere compiti che richiedono intelligenza umana.

## B

### **Baseline**

Versione di riferimento di un prodotto software, indica un punto di arrivo tecnico irreversibile, che viene stabilito come standard.

### **Best Practices**

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

### **Bug**

Errore o malfunzionamento in un software.

### **Branch**

Linea indipendente di sviluppo in un sistema di controllo di versione, utilizzata per implementare nuove funzionalità o correzioni.

### **Budget di Progetto**

Allocazione delle risorse finanziarie necessarie per lo sviluppo del progetto.

### **Backend**

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



## C

### **CamelCase**

Convenzione di denominazione in cui ogni parola inizia con una lettera maiuscola, senza spazi.

### **Capitolato**

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

### **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.

### **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 utilizzato per archiviare dati strutturati, come elenchi di requisiti.

## D

### **Debugging**

Processo di individuazione e correzione degli errori nel software.

### **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.

### **Deployment**

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

### **Driver**

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

## E

### **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 o configurazione).

### **Economicità**

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

### **Efficienza**

Capacità del software di eseguire le operazioni richieste utilizzando un numero minimo di risorse (tempo, memoria, CPU).

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

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

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

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

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

Stima del costo rimanente per completare il progetto

## **F**

### **File explorer**

Strumento che permette all'utente di navigare tra file e cartelle.

### **Feature**

Funzionalità o caratteristica specifica di un software.

### **Feedback**

Informazioni fornite dagli utenti o dal sistema per migliorare il prodotto.

### **Fix**

Correzione di un errore o malfunzionamento.

### **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.

### **Filesystem**

Struttura organizzativa che gestisce la memorizzazione e l'accesso ai file in un computer.

## **G**

### **GitHub**

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

### **Github Projects**

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

### **Github Issues**

Sistema di tracciamento degli errori e delle problematiche su GitHub.

**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**



## 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 predefiniti per supportare lo sviluppo software.

### **Linearmente indipendenti**

Concetto che indica che due elementi non presentano una relazione lineare diretta.

### **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 si tenta di unire modifiche in un repository di versionamento.

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

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

### **Metrica**

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

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

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

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

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

### **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.

## N

### **NodeJS**

Ambiente di runtime per JavaScript lato server.

### **Normazione**

Attività di definizione e aggiornamento delle norme e convenzioni del progetto.

## 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 influenzarne il comportamento.

### Pattern

Soluzione riutilizzabile a problemi ricorrenti nel design del software.

### Plug-in

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

### Precondizione

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

### Prompt

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

### Progettisti

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

### Programmatori

Sviluppatori che scrivono e mantengono il codice sorgente.

### Postcondizione

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

### PV (Planned Value)

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

### 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.

### Pianificazione

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

### Preventivo

Stima dei costi del progetto con allocazione delle risorse economiche necessarie.

### Proof of Concept (PoC)

Implementazione preliminare del progetto per verificarne la fattibilità.

### Pull Request

Proposta di integrazione di modifiche in un repository di versionamento.

## 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.

### **Requisito**

Condizione o capacità necessaria affinché un sistema software soddisfi le esigenze degli stakeholder.

### **Requisito Obbligatorio**

Specifica una caratteristica essenziale del sistema che deve essere implementata affinché il prodotto sia conforme alle specifiche e utilizzabile secondo gli obiettivi previsti.

### **Requisito Facoltativo**

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

### **Requisito Desiderabile**

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

### **Requisito Prestazionale**

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

### **Requisito di Qualità**

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

### **Requisito Funzionale**

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

### **Requisito Non Funzionale**

Definisce caratteristiche e vincoli che il sistema deve rispettare senza specificare comportamenti, come sicurezza, usabilità, prestazioni e scalabilità.

### **Requirement Tracker**

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

### **Report**

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

### **Repository**

Archivio di codice sorgente gestito con strumenti come GitHub

### **Responsabile di progetto**

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



## S

### **Script**

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

### **Stakeholder**

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

### **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.

### **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.

### **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.

## T

### **Task**

Attività specifica da svolgere all'interno del progetto.

### **Test**

Attività per verificare il funzionamento corretto del software.

### **Test di unità**

Test che verificano il funzionamento di singole unità di codice.

### **Test funzionali**

Test che verificano il corretto funzionamento delle funzionalità complessive del software.

### **Team**

Gruppo di persone che collaborano nello sviluppo del progetto.

### **Template**

Modello predefinito per la creazione di documenti o codice.

### **Token**

Unità minima di testo elaborata da un modello LLM.

### **TypeScript**

Linguaggio di programmazione che estende JavaScript con tipi statici.

### **Tracciamento dei Requisiti**

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

### **Tool**

Strumento software utilizzato per supportare lo sviluppo o il testing.

## U

### **Usabilità**

Facilità con cui un utente può interagire con un'interfaccia o un sistema software

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

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

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

Linguaggio standard per la modellazione grafica di sistemi software.

### **Utility**

Strumento ausiliario che semplifica compiti ripetitivi o complessi.

## V

### **Validazione**

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

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

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

### **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.

### **Versionamento**

Processo di gestione delle versioni di un software o documento.

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

Numero di vulnerabilità di sicurezza ancora presenti nel codice.

## **W**

### **Way of Working**

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

**X**

**Y**

**Z**