

# Glossario

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

Versione: 1.0.0 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

# Registro dei cambiamenti

Versione	Data	Autore	Descrizione	Verifica
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				
	<b>.</b>			6
•				7
	_			
	•			
Analisi se	mantica			
Analista .				
API (Appl	lication Progra	amming Interface)		7
	_			7
Artefatti .				7
Attore		• • • • • • • • • • • • • • • • • • • •		7
В				8
Backend .				8
Baseline .				8
Best Pract	tices			8
Branch				8
Budget di	Progetto			8
Bug				8
C				9
				9
-				9
U	-			9
Checklist				0

	Codebase	9
	Codifica	9
	Commit	9
	Committente	9
	Configuration items	9
	Consuntivo di Periodo	
	CSV (Comma-Separated Values)	9
D		
	Debugging	10
	Deployment	
	Design	
	Design Pattern	
	Diagramma UML	
	Dipendenza	
	Driver	
E		
_	EAC (Estimated At Completion)	
	Economicità	
	Efficienza	
	Endpoint	
	Entry	
	ETC (Estimated To Completion)	
	EV (Earned Value)	
F	LV (Larried value)	
1 .	Feature	
	Feedback	
	File explorer	
	Filesystem	
	Fix	
	Fornitura	
	Frontend	
$\sim$	Trontend	
U	GitHub	
	Github Issues	
тт	Github Projects	
Ι.	IDE	
	IDE	
	Issue	
_	ITS (Issue Tracking System)	
K	*** 1.1	
	Knowledge	
L		
	Librerie	18

	Linearmente indipendenti	18
	LLM (Large Language Model)	18
M		19
	Mapping	19
	Merge conflicts	19
	Metrica	19
	Mitigazione dei Rischi	19
	Modello di Sviluppo	19
	Moduli software	19
	MPDS (Metriche di Processo e Sviluppo)	19
	MPC (Minimum Predictive Capability)	
	MVP (Minimum Viable Product)	
N	······································	
	NodeJS	20
O	· · · · · · · · · · · · · · · · · · ·	21
	Ollama	21
Ρ.		22
	Package	
	Parametri	
	Pattern	
	Pianificazione	
	Piano di Contingenza	
	Piano di Qualifica	
	Plug-in	
	Postcondizione	
	Precondizione	
	Preventivo	
	Programmatori	
	Progettisti	
	Prompt	
	Proof of Concept (PoC)	
	Pull Request	
	PV (Planned Value)	
$\cap$		
Z	Qualità di Processo	
R	Qualità di l'10cc350	
1	RC (Requirement Coverage)	
	Refactoring	
	Report	
	<del>-</del>	
	Repository	
	•	
	Requisito Desiderabile	
	Requisito di Qualità	
	Requisito Facoltativo	
	Requisito Funzionale	25

	Requisito Non Funzionale	. 25
	Requisito Obbligatorio	. 26
	Requisito Prestazionale	. 26
	Requirement Tracker	. 26
	Responsabile di progetto	. 26
S.		
	Scenario Principale	
	Scenari Alternativi	
	Script	
	Single Responsibility Principle	
	Sprint	
	Stakeholder	
	Stub	
	SV (Schedule Variance)	
т	SV (Schedule Variance)	
1.	Task	
	Team	
	Template	
	Test	
	Test di unità	
	Test funzionali	
	Token	
	Tool	
	Tracciamento dei Requisiti	
	TypeScript	
U		
	UML (Unified Modeling Language)	
	Usabilità	. 29
	Use Case (Caso d'Uso)	
	Utility	. 29
V		. 30
	Versionamento	. 30
	Verifica	. 30
	Vincolo	. 30
	Visual Studio Code (VS Code)	. 30
	VLN (Presenza di Vulnerabilità)	. 30
	Validazione	. 30
W		. 31
	Way of Working	. 31
X		

### 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.

# **Approvatore**

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

### 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.

### $\mathbf{C}$

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

### $\mathbf{D}$

### **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.

### $\mathbf{E}$

## **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.

# 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.

 $\mathbf{H}$ 

# 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 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 runtime per JavaScript lato server.

# 0

# 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.



# 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

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

# 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.

# 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.

### $\mathbf{T}$

### **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.

# $\mathbf{W}$

# Way of Working

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

 $\mathbf{X}$ 

 $\mathbf{Y}$ 

 $\mathbf{Z}$