



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

Marco Perazzolo

Uso

Esterno

nextsoftpadova@gmail.com

Registro dei cambiamenti

Versione	Data	Autore	Descrizione	Verifica
2.0.0	14/05/2025	Malik Giafar Mohamed	Versione finale	Maria Fuensanta Trigueros Hernandez
1.1.0	13/05/2025	Marco Perazzolo	Aggiunta di nuovi termini al glossario in seguito all'aggiornamento dei documenti	Malik Giafar Mohamed
1.0.1	24/04/2025	Maria Fuensanta Trigueros Hernandez	Aggiornamento e aggiunta di nuovi termini al glossario	Marco Perazzolo
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
Architettura a Strati / Layered Architecture	8
Architettura Client-Server (Deployment)	8
Architettura Esagonale / Hexagonal Architecture	8
Artefatti	8
Attore	8
B	9
Backend	9
Baseline	9
Best Practices	9
Branch	9

Budget	9
Bug	9
Build	9
C	10
CamelCase	10
Cammini Linearmente Indipendenti	10
Capitolato	10
Caso d'Uso / Use Case	10
Checklist	10
Ciclo di Deming / Plan-Do-Check-Act (PDCA)	10
Codebase	10
Codifica	10
Commit	10
Committente	10
Configuration Items	10
Consuntivo di Periodo	10
Container	10
CSV (Comma-Separated Values)	10
D	11
Datasheet	11
Debugging	11
Deployment	11
Design	11
Design Pattern	11
Diagramma UML	11
Dipendenza	11
Discord	11
Docker	11
Dockerfile	11
Driver	11
E	12
EAC (Estimated At Completion)	12
Economicità	12
Efficienza	12
Endpoint	12
Entry	12
ETC (Estimate To Completion)	12
EV (Earned Value)	12
F	13
Feature	13
Feedback	13
File Explorer	13
Filesystem	13
Fix	13
Fornitura	13
Frontend	13
G	14
Gerarchia di Ereditarietà	14
Git	14
GitHub	14
GitHub Issues	14

GitHub Projects	14
GitLab	14
Google Sheets	14
H	15
I	16
IDE	16
Immagine docker (Docker Image)	16
Intelligenza Artificiale / Artificial Intelligence / AI	16
Issue	16
Issue Tracking System (ITS)	16
J	17
JavaScript (JS)	17
Jest	17
K	18
Knowledge	18
L	19
Librerie	19
Linux	19
LLM (Large Language Model)	19
M	20
Mapping Requisiti-Codice	20
Merge Conflicts	20
Metrica	20
Milestone	20
Minor	20
Mitigazione dei Rischi	20
Mock	20
Mocha	20
Modello di Embedding	20
Modello di Sviluppo	20
Moduli Software	20
MVP (Minimum Viable Product)	20
N	21
Node.js	21
npm (Node Package Manager)	21
O	22
Ollama	22
P	23
Package	23
Parametri	23
Pattern	23
Path (Percorso)	23
Pianificazione	23
Piano di Qualifica	23
Plug-in	23
Postcondizioni	23
Precondizioni	23
Preventivo	23
Principio di Responsabilità Singola (SRP)	23
Product Baseline (PB)	23
Programmatori	23

Progettisti	23
Prompt	23
Proof of Concept (PoC)	23
Proponente	23
Pull Request (PR)	24
PV (Planned Value)	24
Q	25
Qualità di Processo	25
R	26
RC (Requirement Coverage)	26
Refactoring	26
Report	26
Repository	26
Requirements and Technology Baseline (RTB)	26
Requisito	26
Requisito Desiderabile	26
Requisito Facoltativo	26
Requisito Funzionale	26
Requisito Obbligatorio	26
Requisito Prestazionale	26
Requisito Qualitativo	26
Responsabile di Progetto	26
Rischi	26
Root Directory	26
S	27
Scenario Principale	27
Scenari Alternativi	27
Script	27
Server	27
Soglia di Accettazione	27
Sprint	27
Stakeholder	27
Stub	27
SV (Schedule Variance)	27
T	28
Task / Attività	28
Team	28
Template	28
Test	28
Test di Unità / Unit Test	28
Timeout	28
Token	28
Tool	28
Tracciamento dei Requisiti	28
TypeScript (TS)	28
Typst	28
U	29
UML (Unified Modeling Language)	29
Usabilità	29
Utility	29
V	30

Validazione	30
Verifica	30
Versionamento	30
Vincolo	30
Visual Studio Code (VS Code / VSC)	30
VLN (Presenza di Vulnerabilità)	30
VSCE - Visual Studio Code Extension Manager	30
VSIX - Visual Studio Code Extension Package	30
W	31
Way of Working	31
WhatsApp	31
X	32
Y	33
YAML (YAML Ain't Markup Language)	33
Z	34
Zoom	34

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

Elemento creato appositamente per una specifica esigenza.

Agile

Metodologia di sviluppo iterativo che privilegia cicli brevi, feedback continui e capacità di adattamento, riducendo il rischio e massimizzando il valore di business.

AI / IA (Intelligenza Artificiale)

Settore dell'informatica che studia e realizza sistemi capaci di replicare funzioni cognitive umane mediante algoritmi e modelli di machine learning.

Analisi dei Requisiti

Processo strutturato che raccoglie, chiarisce e documenta i requisiti del committente; produce il documento di AdR, base per la fase di progettazione.

Analisi dei Rischi

Identificazione e valutazione dei possibili eventi negativi che possono compromettere tempo, costi o qualità di progetto, con strategie di mitigazione associate.

Analisi Semantica

Valutazione di un requisito in termini di completezza, chiarezza, correttezza e non ambiguità di un requisito.

Analista

Figura che studia il dominio applicativo, elabora i casi d'uso e formalizza i requisiti funzionali e non funzionali.

API (Application Programming Interface)

Insieme di regole e definizioni che consente a componenti software diversi di comunicare e scambiarsi dati o funzionalità attraverso chiamate di servizio.

Architettura a Strati / Layered Architecture

Modello architetturale che separa UI, logica di dominio e persistenza in livelli con dipendenze unidirezionali, facilitando test e manutenzione.

Architettura Client-Server (Deployment)

Topologia in cui l'estensione (client) invia richieste a un server remoto (LLM, embedding) e riceve risposte tramite rete.

Architettura Esagonale / Hexagonal Architecture

Pattern che isola la logica di dominio da interfacce esterne tramite porte e adapter, favorendo test e sostituzione di componenti.

Artefatti

Tutti i prodotti tangibili (codice, documenti, diagrammi, report di test, verbali) generati nel ciclo di vita del software.

Attore

Persona o sistema che interagisce con il software in un caso d'uso.

B

Backend

Parte del software che esegue la logica di business, l'accesso ai dati e i servizi 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

Importo economico massimo previsto per completare il progetto, calcolato su ore e costi di ruolo.

Bug

Anomalia che provoca un comportamento errato o inatteso del software; tracciata tramite issue e corretta con debugging.

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 (e.g. nomeMetodo).

Cammini 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.

Capitolato

Documento contrattuale che definisce requisiti e vincoli del sistema richiesto dal proponente.

Caso d'Uso / Use Case

Narrazione strutturata di interazioni attore-sistema che soddisfano un obiettivo di business.

Checklist

Lista di voci da verificare (code review, document inspection) per garantire completezza e qualità.

Ciclo di Deming / Plan-Do-Check-Act (PDCA)

Schema iterativo di miglioramento continuo: pianificare, eseguire, controllare, agire.

Codebase

Collezione completa dei file sorgenti di un progetto software.

Codifica

Scrittura del codice sorgente in conformità a specifiche, standard e best practice.

Commit

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

Committente

Ente o persona che commissiona il progetto e approva gli artefatti consegnati.

Configuration Items

Qualsiasi artefatto soggetto a controllo di configurazione (codice, documenti, immagini).

Consuntivo di Periodo

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

Container

Istanza isolata di un'immagine Docker che esegue applicazioni con dipendenze autoincluse.

CSV (Comma-Separated Values)

Formato testo tabellare in cui i record sono righe e i campi separati da virgole; usato per import/export requisiti.

D

Datasheet

Documento tecnico che dettaglia specifiche di componenti hardware/software.

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

Fase di definizione dell'architettura del sistema (moduli, pattern, interfacce).

Design Pattern

Soluzione collaudata e riutilizzabile a un problema ricorrente di progettazione (GoF, architetturali, ecc.).

Diagramma UML

Rappresentazione visuale (classi, casi d'uso) che documenta struttura o comportamento del sistema.

Dipendenza

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

Discord

Piattaforma VoIP impiegata per la comunicazione sincrona del team.

Docker

Piattaforma di containerizzazione che consente portabilità e riproducibilità degli ambienti di esecuzione.

Dockerfile

Script declarativo che definisce come costruire un'immagine Docker (base image, comandi, copie, entrypoint).

Driver

Software che permette al sistema operativo di comunicare con hardware o servizi specifici.

E

EAC (Estimated At Completion)

Proiezione del costo totale previsto alla fine del progetto, basata sulle performance correnti.

Economicità

Principio di ottimizzazione dei costi rispetto ai benefici ottenuti.

Efficienza

Capacità di svolgere un compito usando il minimo di risorse (tempo, CPU, memoria).

Endpoint

URL o IP esposto da un servizio REST al quale il client invia richieste.

Entry

Singola riga o elemento all'interno di una struttura dati (file, tabella, dizionario).

ETC (Estimate To Completion)

Costi residui previsti per completare il lavoro rimanente.

EV (Earned Value)

Valore del lavoro completato in relazione al budget pianificato, usato nel controllo di progetto.

F

Feature

Funzionalità o caratteristica specifica di un software.

Feedback

Riscontro qualitativo o quantitativo (utente, committente, metriche) usato per migliorare prodotto e processo.

File Explorer

Finestra nativa di selezione e navigazione file/cartelle utilizzata dall'estensione.

Filesystem

Struttura logica che organizza e gestisce file e directory su un dispositivo di memorizzazione.

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

Gerarchia di Ereditarietà

Albero delle relazioni extends/inherits tra classi; profondità e ampiezza influenzano manutenibilità.

Git

Sistema di controllo di versione distribuito che registra la storia delle modifiche e facilita la collaborazione.

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.

GitLab

Alternativa self-hosted/cloud a GitHub con funzionalità DevOps complete.

Google Sheets

Foglio di calcolo online usato per metriche, budget e reporting.

H

I

IDE

Ambiente integrato che fornisce editor, debugger e tooling per lo sviluppo (es. VS Code, IntelliJ).

Immagine docker (Docker Image)

Snapshot immutabile che include codice, runtime e dipendenze necessarie ad avviare un container.

Intelligenza Artificiale / Artificial Intelligence / AI

Tecniche (LLM, embedding) che consentono al plug-in di analizzare testo e codice in modo intelligente.

Issue

Ticket che descrive bug, feature o task all'interno di un sistema di tracking (GitHub, GitLab).

Issue Tracking System (ITS)

Strumento per la gestione, assegnazione e monitoraggio delle issue di progetto.

J

JavaScript (JS)

Linguaggio di scripting dinamico multiparadigma che esegue su runtime V8 (Node.js) e browser.

Jest

Framework di testing per JS / TS con supporto a test unitari, mock e coverage.

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

Collezioni di funzioni o classi riusabili che estendono le funzionalità del linguaggio.

Linux

Sistema operativo open-source usato come host per Docker.

LLM (Large Language Model)

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

M

Mapping Requisiti-Codice

Associazione tra un requisito e i frammenti di codice che lo implementano.

Merge Conflicts

Conflitti di fusione che richiedono intervento manuale quando branch divergenti modificano le stesse righe.

Metrica

Misura quantitativa (es. complessità, copertura test) usata per valutare prodotto o processo.

Milestone

Punto di controllo che segna il completamento di un insieme di deliverable (RTB, PB).

Minor

Release intermedia che introduce fix o miglioramenti non sostanziali (versione Z in X.Y.Z).

Mitigazione dei Rischi

Strategie (evitare, trasferire, ridurre, accettare) per contenere la probabilità o l'impatto del rischio.

Mock

Sostituto controllato di una dipendenza reale usato nei test automatici.

Mocha

Framework JS per test asincroni lato frontend.

Modello di Embedding

Rete neurale che converte testo/codice in vettori numerici per similarità semantica.

Modello di Sviluppo

Approccio organizzativo (Agile, Waterfall, Spiral) adottato per gestire il ciclo di vita del software.

Moduli Software

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

MVP (Minimum Viable Product)

Versione minima funzionante che implementa solo i requisiti obbligatori.

N

Node.js

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

npm (Node Package Manager)

Gestore di pacchetti per Node.js, utilizzato per installare librerie e strumenti, esegue script definiti in `package.json`.

O

Ollama

Server locale/remoto che espone modelli LLM o embedding via REST, configurabile nel plug-in.

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

Schema riutilizzabile di soluzione a un problema ricorrente di design.

Path (Percorso)

Stringa che identifica la posizione di file o directory.

Pianificazione

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

Piano di Qualifica

Documento che elenca strategie, metriche e responsabilità per garantire la qualità.

Plug-in

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

Postcondizioni

Stato del sistema garantito a fine caso d'uso se il flusso si conclude con successo.

Precondizioni

Condizioni che devono valere prima di avviare un caso d'uso.

Preventivo

Stima iniziale di costi e tempi su cui basare budget e contratti.

Principio di Responsabilità Singola (SRP)

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

Product Baseline (PB)

Baseline in cui sono definiti i requisiti e le tecnologie scelte per il progetto, include l'MVP approvato dal proponente.

Programmatori

Sviluppatori che scrivono e mantengono il codice sorgente.

Progettisti

Figure che definiscono architettura e design, producendo modelli e documenti tecnici.

Prompt

Input testuale fornito a un LLM per ottenere una risposta generata.

Proof of Concept (PoC)

Implementazione preliminare del progetto per verificarne la fattibilità.

Proponente

Azienda/ente autore del capitolato che fornisce feedback e approva deliverable.

Pull Request (PR)

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

Livello di efficienza ed efficacia con cui le attività di sviluppo vengono eseguite.

R

RC (Requirement Coverage)

Percentuale di requisiti totalmente soddisfatti rispetto al totale.

Refactoring

Ristrutturazione interna del codice per migliorarne leggibilità e manutenibilità senza alterare il comportamento esterno.

Report

Documento strutturato che presenta risultati di analisi, metriche o verifiche.

Repository

Archivio Git che contiene sorgenti, documenti e cronologia delle modifiche.

Requirements and Technology Baseline (RTB)

Prima milestone che sancisce l'approvazione dei requisiti e delle tecnologie scelte.

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 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 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.

Requisito Qualitativo

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

Responsabile di Progetto

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

Rischi

Eventi potenziali che minacciano tempo, costi o qualità; valutati per probabilità e impatto.

Root Directory

Cartella radice che contiene configurazioni e sorgenti del progetto.

S

Scenario Principale

Flusso ottimale (“happy path”) di azioni che porta al successo del caso d’uso.

Scenari Alternativi

Varianti o eccezioni rispetto allo scenario principale che gestiscono errori o deviazioni.

Script

File di comandi eseguiti da interpreti per automatizzare attività (build, test, deploy).

Server

Host fisico/virtuale che fornisce servizi a client tramite rete.

Soglia di Accettazione

Valore minimo (tipicamente percentuale) oltre il quale una metrica è considerata conforme (“*passed*”).

Sprint

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

Stakeholder

Qualsiasi persona o ente impattato dal progetto (committente, utenti, sviluppatori).

Stub

Implementazione minima sostitutiva usata nei test per isolare un componente.

SV (Schedule Variance)

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

T

Task / Attività

Attività specifica da svolgere all'interno del progetto.

Team

Gruppo di persone che collaborano nello sviluppo del progetto.

Template

Documento o codice pre-strutturato che garantisce uniformità e velocizza la produzione di artefatti.

Test

Esecuzione controllata che verifica se il software soddisfa requisiti e aspettative.

Test di Unità / Unit Test

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

Timeout

Tempo massimo di attesa per una risposta prima di dichiarare l'operazione fallita.

Token

Unità minima (porzione di parola) elaborata da un LLM.

Tool

Strumento software utilizzato per supportare lo sviluppo, il deploy, 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 (TS)

Linguaggio di programmazione che estende JavaScript con tipi statici.

Typst

Linguaggio di markup che combina la potenza di LaTeX con una sintassi moderna e concisa per la generazione di documenti.

U

UML (Unified Modeling Language)

Linguaggio standard per la modellazione grafica di sistemi software.

Usabilità

Indice di facilità con cui utenti possono apprendere, comprendere e utilizzare un sistema.

Utility

Strumento ausiliario che semplifica compiti ripetitivi o complessi.

V

Validazione

Processo che dimostra che il prodotto soddisfa bisogni e aspettative dell'utente finale.

Verifica

Ispezione sistematica che accerta conformità del prodotto ai requisiti specificati.

Versionamento

Gestione strutturata delle diverse versioni di un documento o software.

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 / VSC)

Editor di codice sviluppato da Microsoft, utilizzato per la programmazione e la gestione del codice sorgente, usato come piattaforma per l'estensione Requirement Tracker.

VLN (Presenza di Vulnerabilità)

Conteggio delle vulnerabilità di sicurezza rilevate ma non ancora risolte.

VSCE - Visual Studio Code Extension Manager

CLI ufficiale per creare, impacchettare e pubblicare estensioni .vsix.

VSIX - Visual Studio Code Extension Package

Formato di pacchetto distribuibile/installabile per estensioni VS Code.

W

Way of Working

Insieme di regole, processi e strumenti concordati dal team per garantire coerenza e qualità.

WhatsApp

App di messaggistica istantanea usata per comunicazioni rapide e non formali.

X

Y

YAML (YAML Ain't Markup Language)

Linguaggio di serializzazione dati leggibile dall'uomo, usato in `docker-compose.yml`, CI, configurazioni.

Z

Zoom

Piattaforma di videoconferenza impiegata per meeting con stakeholder.