

# Glossario

<b>Gruppo</b>	Alt+F4
<b>Data</b>	17 Marzo 2025
<b>Versione</b>	v1.0



## Registro modifiche

Versione	Data	Autore/i	Verificatore/i	Descrizione
v1.0	17 Marzo 2025		Enrico Bianchi	Release
v0.3	27 febbraio 2025	Francesco Savio	Marko Peric	Sezione <b>Glossario</b>
v0.2	28 dicembre 2024	Marko Peric	Francesco Savio	Sezione <b>Indice</b>
v0.1	19 dicembre 2024	Francesco Savio	Marko Peric	Sezioni <b>Introduzione</b> , <b>Glossario</b>

## Indice

- Registro delle modifiche
- Introduzione
- Glossario
  - A
  - B
  - C
  - D
  - E
  - F
  - G
  - H
  - I
  - J
  - K
  - L
  - M
  - N
  - O
  - P
  - Q
  - R
  - S
  - T
  - U
  - V
  - W
  - X
  - Y
  - Z

## Introduzione

Questo glossario contiene termini contenuti nei documenti creati dal gruppo Alt+f4 che potrebbero richiedere una spiegazione più esaustiva, la ricerca può essere fatta tramite la parola specifica utilizzando il comando Ctrl+f, oppure scorrendo il glossario in ordine alfabetico.

## Glossario

### A

**Actual Cost** Rappresenta il costo effettivamente sostenuto per completare il lavoro in un determinato momento o fase di un progetto. L'AC viene utilizzato per monitorare l'andamento del progetto e per confrontare il costo pianificato con il costo effettivamente sostenuto. L'AC è una metrica importante per la gestione dei progetti di sviluppo software, in quanto consente di monitorare l'andamento del progetto e di identificare eventuali scostamenti dal budget.

**Agile** Agile in informatica è un insieme di metodologie e pratiche di sviluppo software che promuovono iterazioni rapide, adattabilità ai cambiamenti e una stretta collaborazione tra il team di sviluppo e i clienti. Il termine Agile si riferisce a un approccio che enfatizza la flessibilità, la trasparenza e il miglioramento continuo, con l'obiettivo di consegnare software di qualità in modo rapido e reattivo alle necessità del cliente.

**Analisi statica** È una tecnica di valutazione del software che viene eseguita senza l'esecuzione del programma. Si basa sull'analisi del codice sorgente, per rilevare errori, vulnerabilità e problemi di qualità prima dell'esecuzione del software.

**Analisi dinamica** È una tecnica di valutazione del software eseguita durante l'esecuzione del programma. Monitora il comportamento del software in tempo reale per identificare errori, problemi di performance, falle di sicurezza e altre anomalie.

**Angular** È un framework open-source per lo sviluppo di applicazioni web single-page (SPA) e front-end.

**API REST** (Application Programming Interface basata su Representational State Transfer) È un'interfaccia che consente la comunicazione tra sistemi software utilizzando il protocollo HTTP e seguendo i principi dell'architettura REST. Le API REST usano URI leggibili e gerarchici per identificare le risorse.

**Artefatti** Sono i sottoprodotti del processo di sviluppo software. Sono i prodotti tangibili e intangibili creati durante lo sviluppo, come documenti e codice sorgente.

Gli artefatti sono importanti perché forniscono una traccia del processo di sviluppo, consentono la comunicazione tra i membri del team e gli stakeholder, e aiutano a garantire la qualità del software.

**Aspell** È un correttore ortografico open-source che è stato impiegato nella creazione della GitHub action per il controllo grammaticale dei file.

## B

**Baseline** È una versione ufficiale e stabile di un sistema, di un software o di un progetto, che viene utilizzata come punto di riferimento per tutte le successive modifiche e sviluppi. Una baseline rappresenta un insieme di specifiche, documentazione, codice o configurazioni che sono stati approvati e che vengono considerati come "fissati" a un certo momento del ciclo di vita del progetto.

**Bertino embedding** Si riferisce alle rappresentazioni vettoriali generate utilizzando il modello BERTino. Questi embedding catturano le caratteristiche semantiche delle parole o delle frasi in italiano, facilitando compiti come il calcolo della similarità tra testi e la ricerca semantica. BERTino offre una soluzione efficiente e specializzata per l'elaborazione del linguaggio naturale in italiano, combinando l'efficacia dei modelli BERT con una struttura più leggera e prestazioni ottimizzate.

**Best practices** Le best practices (in italiano, "migliori pratiche") sono un insieme di metodi, procedure, tecniche o regole riconosciute come le più efficaci e affidabili per ottenere risultati ottimali in un determinato ambito. Queste pratiche si basano su esperienze consolidate, studi, e casi di successo, e servono come guida per migliorare la qualità, l'efficienza e l'affidabilità di un processo o di un'attività.

**BGE 5 embedding** BGE sta per "Bilingual General Embedding", una serie di modelli sviluppati per generare rappresentazioni numeriche (embedding) di testi in modo che concetti semanticamente simili siano più vicini nello spazio vettoriale.

**Budget at Completion** È una metrica utilizzata nella gestione dei progetti per rappresentare il costo totale previsto di un progetto al suo completamento, sulla base del piano iniziale. Il valore del BAC corrisponde alla somma di tutti i costi pianificati per il completamento delle attività previste, senza tenere conto di eventuali variazioni dovute a ritardi, modifiche o imprevisti.

## C

**Capitolato** È un documento formale che definisce in modo dettagliato i requisiti, le specifiche e le condizioni tecniche, funzionali e organizzative di un progetto o di

un sistema da sviluppare, acquisire o implementare. Viene utilizzato come base di riferimento per lo sviluppo, la fornitura o la valutazione di soluzioni informatiche.

**Caption** È un breve testo descrittivo associato a un elemento visivo o multimediale, utilizzato per fornire informazioni contestuali, esplicative o identificative. Le caption aiutano gli utenti a comprendere meglio il contenuto a cui si riferiscono, migliorandone l'accessibilità e l'usabilità.

**Cerimonie** Sono eventi formali e predefiniti che si svolgono durante il ciclo di vita del progetto per favorire la comunicazione, la collaborazione e l'ispezione del lavoro. Le cerimonie Scrum hanno una durata limitata e un obiettivo specifico, e aiutano il team a rimanere focalizzato sul raggiungimento degli obiettivi dello sprint.

**Checklist** È un elenco di elementi, attività o passaggi che devono essere verificati, completati o controllati per assicurarsi che un processo o un'attività sia eseguito correttamente. Le checklist sono utilizzate per garantire che tutte le operazioni necessarie siano state eseguite, riducendo il rischio di errori e dimenticanze.

**Ciclo di vita** Rappresenta l'insieme delle fasi attraverso cui un prodotto software viene ideato, sviluppato, distribuito e mantenuto nel tempo.

**Ciclo di Deming** È un metodo di gestione iterativo in quattro fasi utilizzato per il controllo e il miglioramento continuo dei processi e dei prodotti. È un modello per il miglioramento continuo della qualità in un'ottica a lungo raggio. Serve per promuovere una cultura della qualità che è tesa al miglioramento continuo dei processi e all'utilizzo ottimale delle risorse.

**Codeline** È il termine utilizzato per riferirsi a una sequenza continua di versioni di codice sorgente che viene sviluppata e mantenuta all'interno di un sistema di controllo di versione. Una codeline rappresenta una "linea" di sviluppo del codice, che può evolversi con modifiche, miglioramenti e correzioni nel tempo.

**Committente** È l'organizzazione o l'individuo che commissiona lo sviluppo di un sistema, prodotto o servizio software a un fornitore. Il committente è colui che stabilisce i requisiti, le specifiche e gli obiettivi che il prodotto finale deve soddisfare.

**Configuration Item** È un elemento di un sistema informatico che viene gestito e monitorato come parte di un processo di gestione della configurazione. Un CI può essere qualsiasi componente hardware, software, documentazione, o altro elemento rilevante che influisce sul funzionamento di un sistema o di un progetto. Ogni CI è identificato in modo univoco e contiene informazioni relative alla sua configurazione, stato e versione.

**Console** È un'interfaccia utente che permette di interagire con un sistema operativo o un software tramite comandi testuali. Solitamente è un'applicazione software o a una finestra di terminale che permette agli utenti di inviare comandi al sistema operativo o a un programma e ricevere risposte o output.

**Container** È una tecnologia di virtualizzazione leggera che consente di eseguire applicazioni in ambienti isolati e portabili, chiamati contenitori. A differenza delle macchine virtuali tradizionali, i container condividono il sistema operativo sottostante, ma sono separati a livello di processo, garantendo che le applicazioni all'interno di ciascun container non interferiscano tra loro. Questa isolazione consente di eseguire applicazioni in modo consistente su qualsiasi ambiente.

**Containerizzazione** È una tecnica di virtualizzazione a livello di sistema operativo che consente di eseguire applicazioni in ambienti isolati chiamati container. Un container è un'unità leggera e portabile che include il codice dell'applicazione, le dipendenze e le configurazioni necessarie per l'esecuzione, garantendo coerenza e riproducibilità su diverse infrastrutture.

**Cost Variance** È una metrica che misura la differenza tra il valore del lavoro svolto e il costo effettivamente sostenuto per completarlo in un determinato momento o fase di un progetto. La CV viene utilizzata per monitorare l'andamento del progetto e per identificare eventuali risparmi o sforamenti rispetto al budget. La CV viene calcolata sottraendo l'Actual Cost (AC) dall'Earned Value (EV).

**Cruscotto di valutazione** È uno strumento che fornisce una visualizzazione delle metriche relative al processo di sviluppo e al prodotto software. Il cruscotto di solito include grafici, tabelle e altri elementi visivi che mostrano l'andamento delle metriche nel tempo e consentono al team di sviluppo di monitorare l'andamento del progetto, a identificare le aree critiche che richiedono attenzione e a prendere decisioni informate per migliorare la qualità del processo e del prodotto.

**CSV** È un formato di file utilizzato per memorizzare dati tabulari in forma di testo semplice, in cui i valori sono separati da virgole. È ampiamente usato per lo scambio di dati tra applicazioni come fogli di calcolo, database e software di analisi dati.

**Current working directory** La current working directory è la cartella in cui un programma o sistema operativo opera attualmente e svolge operazioni sui file.

## D

**Dashboard** È un'interfaccia utente grafica che visualizza informazioni chiave relative a un progetto o a un sistema in modo sintetico e facilmente comprensibile. Le

dashboard sono utilizzate per monitorare lo stato di avanzamento, identificare tendenze, individuare potenziali problemi e prendere decisioni informate.

**Database** È un insieme organizzato di dati strutturati, memorizzati e gestiti in modo da consentire un accesso efficiente, una facile manipolazione e un'integrità dei dati ottimale.

**Design Pattern** È una soluzione generale e riutilizzabile a un problema comune nella progettazione del software. Non è un design finito che può essere trasformato direttamente in codice, ma piuttosto una descrizione o un modello su come affrontare un problema di progettazione.

**Diagramma di Gantt** È uno strumento di pianificazione visiva utilizzato per rappresentare graficamente le attività di un progetto nel tempo. Ideato da Henry L. Gantt negli anni 1910, questo diagramma è uno dei più comuni metodi per il controllo e la gestione di progetti, ed è utilizzato principalmente in contesti di gestione dei progetti per visualizzare la sequenza, la durata e il progresso delle attività.

**Draw.io** È un software online gratuito e open source per la creazione di diagrammi di vario tipo, è stato usato dal gruppo per la creazione di diagrammi UML.

**DVCS** DVCS (Distributed Version Control System) è un sistema di controllo di versione distribuito che consente a più sviluppatori di lavorare su un progetto in modo collaborativo, tracciando e gestendo le modifiche al codice sorgente nel tempo. A differenza dei sistemi di controllo di versione centralizzati, in cui esiste un repository centrale, un DVCS consente a ciascun sviluppatore di avere una copia completa del repository, inclusa la cronologia delle modifiche, sul proprio computer.

## E

**Earned Value** Rappresenta il valore del lavoro effettivamente completato in un determinato momento o fase di un progetto. L'EV viene utilizzato per monitorare l'avanzamento del progetto e per confrontare il lavoro pianificato con il lavoro effettivamente svolto.

**Embedding** L'embedding per il confronto tra stringhe è una tecnica che rappresenta le stringhe come vettori numerici in uno spazio continuo, permettendo di calcolare la similarità tra di esse in modo efficiente.

**End value** Rappresenta il valore finale stimato di un progetto al completamento.



**Estimated at Completion** È una metrica utilizzata nella gestione dei progetti per stimare il costo totale previsto al completamento di un progetto, basandosi sui dati di avanzamento attuali e sulle previsioni di spesa future. Consente di anticipare scostamenti di budget e di adottare misure correttive per mantenere il progetto sotto controllo.

**Extend** In UML (Unified Modeling Language), Extend indica una relazione tra due casi d'uso in cui un caso d'uso "estende" il comportamento di un altro caso d'uso. In altre parole, il caso d'uso esteso aggiunge funzionalità o comportamenti opzionali al caso d'uso base.

## F

**Fast-forward** Il termine fast-forward si riferisce a una situazione in cui una serie di operazioni o aggiornamenti viene applicata in modo diretto e senza conflitti, ovvero in modo che il risultato finale "avanzato" nel tempo sia una progressione lineare senza la necessità di risolvere conflitti tra modifiche parallele. Nel contesto del controllo di versione (come Git), fast-forward indica una situazione in cui il ramo (branch) corrente può essere aggiornato semplicemente "avanzando" la sua punta (tip) al commit più recente di un altro ramo, senza dover fare un "merge" (fusione) con risoluzione dei conflitti.

**Feedback** È il processo attraverso il quale i risultati o le risposte di un sistema, applicazione o processo vengono restituiti agli utenti, ai team di sviluppo o ad altri attori coinvolti, per informare, correggere o migliorare il funzionamento del sistema stesso. Il feedback può essere positivo (quando il risultato è conforme alle aspettative) o negativo (quando è necessario apportare modifiche o correzioni).

**File system** È essenziale per la gestione dei dati su dispositivi di memorizzazione, permettendo di archiviare, recuperare e organizzare i file in modo sicuro e strutturato.

**Font** È un insieme di caratteri tipografici che condividono uno stesso stile e dimensione. I font determinano l'aspetto visivo del testo, influenzando l'estetica e la leggibilità di un documento, di un sito web o di un'applicazione. Ogni font include una varietà di simboli, lettere, numeri e segni di punteggiatura, tutti progettati con lo stesso stile grafico.

**Fork** È una copia di un repository di codice sorgente che consente a uno sviluppatore di apportare modifiche al codice senza influenzare il progetto originale. Il fork crea una nuova copia del repository, consentendo allo sviluppatore di sperimentare nuove funzionalità, correggere bug o apportare modifiche sostanziali senza dover ottenere l'approvazione del progetto principale.

**Fornitore** È un'organizzazione o un individuo che fornisce un prodotto software o un servizio software a un'altra organizzazione o individuo (il cliente). Il fornitore è responsabile dello sviluppo, della consegna e, in alcuni casi, della manutenzione del software o del servizio, in base ai requisiti e alle specifiche concordate con il cliente.

**Framework** È un insieme di strumenti, librerie e convenzioni che forniscono una struttura predefinita per lo sviluppo di applicazioni software. Si tratta di una piattaforma che semplifica e standardizza il processo di sviluppo, consentendo agli sviluppatori di concentrarsi sulla logica applicativa piuttosto che scrivere codice da zero per ogni singola funzionalità di base.

## G

**Git** È un sistema di controllo versione distribuito open-source, utilizzato per gestire e tracciare le modifiche apportate a file e progetti, in particolare nel contesto dello sviluppo software. Permette a più sviluppatori di collaborare in modo efficace, tenendo traccia delle modifiche in un progetto e facilitando il lavoro su diverse versioni del codice.

**Gitflow** È una strategia di branching che aiuta a gestire in modo strutturato lo sviluppo di software, organizzando il lavoro in vari rami dedicati a specifici scopi, come lo sviluppo di funzionalità, la preparazione dei rilasci e la gestione delle correzioni di bug.

**Github** È una piattaforma di sviluppo software basata su Git, che fornisce un servizio di hosting per il codice sorgente e supporta il controllo versione distribuito. GitHub permette di collaborare a progetti software in modo efficiente, mantenendo un repository centralizzato per il codice e facilitando la gestione delle versioni, la revisione del codice e la collaborazione tra team.

**GitHub action marketplace** È una piattaforma integrata in GitHub che permette agli utenti di scoprire, acquistare e utilizzare azioni GitHub (GitHub Actions)

**GitHub action** È uno strumento di automazione che consente di eseguire flussi di lavoro (workflows) direttamente all'interno di un repository GitHub. Le GitHub Actions sono utilizzate per automatizzare compiti ripetitivi, come il testing del codice, il deployment (distribuzione), la compilazione, e altre attività legate al ciclo di vita del software. Queste azioni si integrano con GitHub, facilitando l'integrazione continua (CI) e la distribuzione continua (CD) nei progetti software.

## H

**Host** È qualsiasi dispositivo o sistema che fornisce risorse, servizi o capacità di comunicazione in una rete. Un host può essere un computer, un server, un dispositivo mobile, o qualsiasi altro dispositivo che possa connettersi a una rete e interagire con altri dispositivi o sistemi.

## I

**IDE** È un software che fornisce un insieme di strumenti integrati per facilitare lo sviluppo di applicazioni. Gli IDE sono progettati per aiutare i programmatori a scrivere, testare, eseguire e fare il debug del codice in modo più efficiente. Un IDE tipicamente combina diversi strumenti in un'unica interfaccia utente, semplificando il flusso di lavoro durante la scrittura del software.

**IEEE 12207:1996** Noto anche come Software Life Cycle Processes, è uno standard internazionale sviluppato dall'Institute of Electrical and Electronics Engineers (IEEE), in collaborazione con l'ISO/IEC, che definisce un quadro di riferimento completo per i processi coinvolti nel ciclo di vita del software. Questo standard descrive tutte le attività necessarie per lo sviluppo, la gestione, la manutenzione e la dismissione di sistemi software.

**IEEE 828** È uno standard che stabilisce le linee guida per la gestione della configurazione software nei progetti di ingegneria del software e dei sistemi. Il suo obiettivo è garantire controllo, tracciabilità e gestione delle modifiche durante l'intero ciclo di vita del software o di un sistema complesso.

**Include** In UML (Unified Modeling Language), Include è un tipo di relazione utilizzato nei diagrammi di casi d'uso per rappresentare un caso d'uso che include il comportamento di un altro caso d'uso. Questo tipo di relazione implica che un caso d'uso principale esegue un altro caso d'uso come parte del proprio comportamento. La relazione Include viene utilizzata per separare la logica comune che può essere riutilizzata in diversi casi d'uso, evitando la duplicazione del codice o della logica all'interno del sistema. Quando un caso d'uso "incluso" viene eseguito, il caso d'uso "inclusivo" lo invoca sempre.

**Indice di Gulpease** È una metrica di leggibilità progettata per la lingua italiana che misura la facilità di comprensione di un testo in base alla lunghezza delle parole e delle frasi.

**ISO 31000** È uno standard che fornisce un approccio sistematico, integrato e continuo per la gestione del rischio, aiutando le organizzazioni a identificare, valutare e gestire i rischi in modo efficace per raggiungere i propri obiettivi.

**ISO 9126** Lo standard ISO/IEC 9126 fornisce un quadro di riferimento per la valutazione e il miglioramento della qualità del software.

**Issue** È un termine generico utilizzato per riferirsi a un problema, un bug, una richiesta o una questione che deve essere risolta o gestita all'interno di un progetto software. Le issue possono riguardare vari aspetti dello sviluppo software, come difetti nel codice, funzionalità da implementare, miglioramenti da apportare o discussioni relative a decisioni progettuali.

**Issue tracking system** È un software utilizzato per registrare, tracciare, gestire e risolvere le problematiche, i bug, le richieste di funzionalità e altri tipi di "issue" (problemi) all'interno di un progetto o di un'organizzazione. Questi sistemi sono essenziali per monitorare il progresso delle attività, assegnare compiti e garantire che le problematiche vengano risolte in modo tempestivo ed efficiente.

**ITS** Un Issue Tracking System (ITS) è un software utilizzato per registrare, tracciare, gestire e risolvere le problematiche, i bug, le richieste di funzionalità e altri tipi di "issue" (problemi) all'interno di un progetto o di un'organizzazione. Questi sistemi sono essenziali per monitorare il progresso delle attività, assegnare compiti e garantire che le problematiche vengano risolte in modo tempestivo ed efficiente.

## J

**Javascript** È un linguaggio di programmazione interpretato, dinamico e orientato agli oggetti, utilizzato principalmente per lo sviluppo di applicazioni web interattive.

## K

**Kanban board** È uno strumento visivo utilizzato per gestire e monitorare il flusso di lavoro in progetti, tipicamente in contesti di sviluppo software o gestione dei processi aziendali. È una rappresentazione grafica che aiuta i team a visualizzare le attività in corso, a tracciare il progresso e a identificare eventuali colli di bottiglia nel processo.

**Keyword** È una parola o a un termine che ha un significato speciale all'interno di un linguaggio di programmazione o di un sistema. Le parole chiave sono riservate e non possono essere utilizzate come identificatori (come nomi di variabili, funzioni o classi), poiché sono predefinite dal linguaggio per svolgere funzioni specifiche.

## L

**Label** È un identificatore utilizzato per rappresentare, descrivere o categorizzare un elemento, un dato o un'area specifica all'interno di un sistema, un programma o un'interfaccia. Il significato e l'utilizzo di una label variano in base al contesto.

**Large Language Model** È un modello di intelligenza artificiale basato su reti neurali profonde, addestrato su vasti insiemi di dati testuali per comprendere e generare linguaggio naturale.

**Latex** È un linguaggio di markup e un sistema di preparazione di documenti, utilizzato principalmente per la composizione di documenti scientifici, tecnici e accademici. LaTeX è ampiamente preferito in ambiti come la matematica, la fisica, l'ingegneria e l'informatica grazie alla sua capacità di gestire complessi simboli matematici, formule, citazioni bibliografiche e riferimenti incrociati.

## M

**Markdown** È un linguaggio di markup leggero che consente di scrivere testo formattato in modo semplice, con la possibilità di convertirlo facilmente in HTML o altri formati, ed è ampiamente utilizzato per la scrittura di documentazione.

**Markup** Si riferisce a un insieme di simboli o codici usati per annotare o strutturare il contenuto di un documento. Questi simboli o codici, noti come markup tags, indicano come il testo o gli altri contenuti devono essere trattati, visualizzati o interpretati da un programma.

**Metriche** Sono valori utilizzati per valutare e monitorare diverse caratteristiche del processo di sviluppo e del prodotto software. Le metriche forniscono informazioni quantitative che aiutano a comprendere l'efficacia dei processi, la qualità del prodotto e l'efficienza del team di sviluppo.

**Milestone** È un punto di riferimento significativo all'interno di un progetto o di un ciclo di sviluppo software che segna il completamento di una fase importante o di un obiettivo specifico. Le milestone sono utilizzate per monitorare il progresso del progetto, per garantire che vengano rispettati i tempi e gli obiettivi prefissati e per facilitare la pianificazione delle fasi successive.

**Minimum Viable Product** È una prima versione di un prodotto che include solo le funzionalità essenziali per offrire ciò che era stato richiesto dai clienti consumando la quantità minima di risorse possibile. L'obiettivo principale dell'MVP è ottenere un feedback il prima possibile da utenti reali, riducendo il rischio di investire risorse in funzionalità superflue o in uno sviluppo non ottimale.

**Mock** È un oggetto simulato che replica il comportamento di un componente software reale per scopi di test.

**Modello di sviluppo a periodi** Il modello di sviluppo a periodi (o ciclo di vita del software iterativo e incrementale) è un approccio in informatica che suddivide

il processo di sviluppo del software in periodi o iterazioni ripetute, in cui ogni periodo produce una versione parziale ma funzionante del sistema. Ogni ciclo o periodo include fasi di progettazione, sviluppo, testing e revisione, e alla fine di ogni periodo viene rilasciato un incremento delle funzionalità del software.

## N

**Nomic** Sono rappresentazioni vettoriali (embedding) di dati ad alta dimensione. L'obiettivo principale è rendere i dati complessi più interpretabili e visualizzabili in modo intuitivo.

## O

**Obiettivi di qualità** Rappresentano i criteri misurabili che un sistema software deve soddisfare per garantire prestazioni, affidabilità e usabilità conformi alle aspettative degli stakeholder. Questi obiettivi derivano dai requisiti non funzionali e contribuiscono alla valutazione complessiva della qualità del prodotto, influenzando aspetti come la manutenibilità, la sicurezza e l'esperienza utente.

**OpenAPI 3.1** È una specifica standardizzata per descrivere, documentare e definire API REST in modo leggibile sia dagli esseri umani che dalle macchine. È l'evoluzione della specifica OpenAPI 3.0, sviluppata dalla OpenAPI Initiative (OAI), con miglioramenti nella flessibilità e compatibilità con JSON Schema.

**OS** OS sta per Operating System (sistema operativo), che è un software di base fondamentale per la gestione delle risorse hardware e software di un computer o dispositivo. L'OS fornisce un'interfaccia tra l'utente e l'hardware, gestendo operazioni come l'esecuzione di programmi, la gestione della memoria, il controllo dei dispositivi di input/output, e la gestione dei file.

## P

**Packages** Si riferisce a un insieme di file o moduli software organizzati in una struttura di directory e distribuiti come unità singola, destinati a facilitare la gestione, l'installazione e l'uso di componenti software. I pacchetti contengono tipicamente codice sorgente, librerie, configurazioni, documentazione e altre risorse necessarie per l'implementazione di una funzionalità o applicazione specifica.

**Pascal case** È una convenzione di denominazione utilizzata in informatica, soprattutto nella programmazione, in cui ogni parola che compone un nome o identificatore inizia con una lettera maiuscola, senza spazi o altri caratteri separatori. In Pascal Case, la prima lettera di ogni parola, inclusa la prima, è maiuscola, e le parole sono concatenate senza l'uso di underscore o altri simboli.

**PDF** È un formato di file sviluppato da Adobe che consente di visualizzare, stampare e scambiare documenti in modo indipendente dal software, hardware e sistema operativo utilizzato. Il formato PDF è progettato per mantenere la formattazione originale di un documento, inclusi testo, immagini, grafica vettoriale, font e layout, in modo che appaia uguale su qualsiasi dispositivo o piattaforma.

**Personal Access Token** È un tipo di token di autenticazione che viene utilizzato per accedere a risorse protette su piattaforme e servizi, come GitHub, GitLab, o Azure DevOps, al posto di una password tradizionale. I PAT vengono generati dall'utente e possono essere utilizzati per autenticarsi in modo sicuro e automatizzato quando si interagisce con API o si eseguono operazioni come il push di codice o la gestione di repository.

**Pipeline** È un insieme di processi automatizzati che vengono eseguiti in sequenza per costruire, testare e distribuire il software. Le pipeline sono spesso utilizzate in combinazione con la Continuous Integration e Continuous Delivery (CI/CD) per automatizzare il processo di rilascio del software. Nel progetto le pipeline sono implementate tramite GitHub Actions.

**Planned Value** Rappresenta il valore pianificato del lavoro che dovrebbe essere completato in un determinato momento o fase di un progetto. Il PV viene utilizzato per monitorare l'avanzamento del progetto e per confrontare il lavoro pianificato con il lavoro effettivamente svolto.

**PoC** È una realizzazione incompleta di un determinato progetto o metodo per dimostrare la fattibilità o confermare la validità di alcuni principi o concetti fondamentali.

**Project back-og** Il project backlog (o backlog del progetto) è un elenco prioritizzato di attività, requisiti, funzionalità o task che devono essere completati in un progetto. Viene utilizzato principalmente nelle metodologie agili, come Scrum, per gestire e pianificare il lavoro da svolgere durante lo sviluppo di un prodotto o di una soluzione. Ogni voce nel backlog rappresenta un elemento di lavoro che può essere un requisito del cliente, una correzione di bug, una funzionalità nuova o un miglioramento da implementare.

**Pull request** È un meccanismo di collaborazione in cui uno sviluppatore propone modifiche al codice, che vengono esaminate e potenzialmente integrate nel progetto principale, consentendo il controllo della qualità e la revisione collettiva del codice.

**Python** È un linguaggio di programmazione interpretato, dinamico e ad alto livello, caratterizzato da una sintassi chiara e leggibile.

**Q**

**Quality Assurance** Si riferisce all'insieme delle attività pianificate e sistematiche implementate per assicurare che un prodotto software o un servizio soddisfi i requisiti di qualità specificati. Il QA è un processo continuo che si svolge durante tutto il ciclo di vita dello sviluppo software. Il suo obiettivo è prevenire i difetti, migliorare la qualità del software e garantire la soddisfazione del cliente.

**R**

**Radio button** Un radio button (o pulsante di opzione) è un tipo di controllo dell'interfaccia utente utilizzato nei moduli e nelle applicazioni per permettere all'utente di selezionare una sola opzione da un gruppo di scelte predefinite.

**Report** È un documento o una rappresentazione strutturata di dati e informazioni, generato in modo automatico o manuale, con lo scopo di fornire una sintesi chiara e dettagliata su un determinato argomento, processo o sistema.

**Repository** È un luogo centrale dove vengono archiviati e gestiti file, dati, codice sorgente o altre risorse digitali, in modo strutturato e organizzato. Il repository permette di centralizzare e controllare l'accesso a questi file, facilitando il lavoro collaborativo e il versionamento. È uno strumento essenziale per la gestione del ciclo di vita del software e dei progetti, nonché per la conservazione e distribuzione dei dati.

**Requisiti funzionali** Definiscono le funzionalità e i comportamenti che un sistema software deve implementare per soddisfare le esigenze degli utenti e degli stakeholder. Essi descrivono cosa il sistema deve fare, specificando input, elaborazioni e output attesi.

**Requisiti utente** Sono una categoria di requisiti software che descrivono le necessità, gli obiettivi e le aspettative degli utenti finali rispetto a un sistema o applicazione. Rappresentano il punto di partenza per la definizione dei requisiti funzionali e non funzionali del software.

**Requisiti software** Definiscono le caratteristiche, le funzionalità e i vincoli che un sistema software deve soddisfare per rispondere alle necessità degli utenti e degli stakeholder. Costituiscono il punto di riferimento per la progettazione, lo sviluppo, il testing e la manutenzione del software, garantendo che il prodotto finale sia conforme alle specifiche richieste.

**Rischio organizzativo** Indica qualsiasi cosa possa generare insicurezze all'interno di un'organizzazione. Esso può compromettere il raggiungimento degli obiettivi.



strategici, operativi o di conformità. Questo tipo di rischio viene associato a inefficienze nei processi decisionali, mancanza di coordinamento tra componenti organizzativi o problemi gestionali.

**Rischio tecnologico** Indica la possibilità che un'organizzazione subisca perdite economiche, inefficienze o interruzioni durante lo sviluppo a causa di problemi legati alla scelta, l'uso, la sicurezza o la mancanza di supporto delle tecnologie utilizzate. Il rischio può anche derivare da fonti come attacchi informatici, malfunzionamenti hardware o problemi di integrità dei dati.

**Risk Mitigation Rate** È una metrica che misura l'efficacia delle azioni intraprese per mitigare i rischi identificati durante il processo di sviluppo. In altre parole, indica la percentuale di rischi che sono stati mitigati con successo. Aiuta il team di sviluppo a valutare l'efficacia delle proprie strategie di gestione dei rischi e a prendere decisioni informate per migliorare la qualità del processo di sviluppo.

**RTB** In ingegneria del software, RTB sta per "Reason to Believe" (Motivazione per Credere) ed è un concetto utilizzato principalmente nell'ambito della valutazione della qualità e dell'affidabilità di un sistema software. L'RTB rappresenta la giustificazione, la base o le evidenze che supportano la fiducia nella capacità di un software di soddisfare i requisiti stabiliti.

## S

**Schedule Variance** È una metrica che misura la differenza tra il valore del lavoro pianificato e il valore del lavoro effettivamente svolto in un determinato momento o fase di un progetto. La SV viene utilizzata per monitorare l'avanzamento del progetto e per identificare eventuali ritardi o anticipi rispetto al programma. La SV viene calcolata sottraendo il Planned Value (PV) dall'Earned Value (EV).

**SCRUM** È un framework di gestione di progetti Agile che si concentra sulla collaborazione di team, sulla definizione di obiettivi chiari e sulla flessibilità per adattarsi ai cambiamenti. In Scrum, i progetti vengono suddivisi in cicli di lavoro brevi e iterativi chiamati sprint, che in genere durano da una a quattro settimane. Scrum definisce anche un insieme di ruoli, artefatti e cerimonie che aiutano il team a lavorare in modo efficace.

**Spell checkin** È il processo di controllo e correzione automatica degli errori di ortografia all'interno di un testo. Viene comunemente utilizzato in applicazioni come editor di testo, word processor, e-mail e strumenti di programmazione per rilevare parole scritte in modo errato o che non corrispondono a quelle presenti in un dizionario di riferimento.

**Sprint** È un'unità di lavoro a tempo determinato utilizzata nelle metodologie agili, in particolare nel framework Scrum, per sviluppare e rilasciare incrementi di un prodotto o progetto. Lo sprint rappresenta un intervallo di tempo fisso (solitamente da 1 a 4 settimane) durante il quale un team lavora per completare un set specifico di attività o obiettivi definiti nel backlog dello sprint.

**Sprint backlog** È un elenco delle attività, dei task e delle funzionalità che il team di sviluppo si impegna a completare durante uno Sprint. Lo Sprint Backlog deriva dal Product Backlog, che contiene tutte le attività necessarie per completare un intero progetto. Durante la pianificazione dello Sprint, il team seleziona gli elementi dal Product Backlog che possono essere completati entro la durata dello Sprint e li inserisce nello Sprint Backlog.

**Stakeholder** È qualsiasi individuo, gruppo o organizzazione che ha un interesse diretto o indiretto in un progetto (portatore di interesse). Gli stakeholder sono essenziali perché definiscono i requisiti, gli obiettivi e i vincoli di un progetto, influenzandone la direzione e il successo.

**Strategia di branching** Una strategia di branching (o strategia di ramificazione) è una pratica o un insieme di regole utilizzate nella gestione del controllo di versione del codice sorgente, in particolare quando si lavora con sistemi come Git. Essa definisce come e quando creare branch (rami) separati nel codice, in modo che i team di sviluppo possano lavorare in parallelo su diverse funzionalità, correzioni o versioni senza interferire con la versione principale del software.

## T

**TLShell** È un'interfaccia grafica per TeX Live, una delle distribuzioni più utilizzate di LaTeX. Viene fornita con TeX Live per semplificare la gestione della distribuzione, senza dover usare la riga di comando.

## U

**UML** È una rappresentazione grafica utilizzata per modellare, visualizzare, documentare e progettare sistemi software o processi. L'UML è uno standard internazionale che fornisce un linguaggio visuale per descrivere sia la struttura che il comportamento di un sistema, rendendo più chiara e condivisibile la progettazione tra sviluppatori, analisti, e stakeholder.

**Use Case** È una descrizione dettagliata di come un sistema interagisce con gli utenti (o altri sistemi) per raggiungere un obiettivo specifico. È uno strumento fondamentale nell'analisi dei requisiti e aiuta a definire il comportamento del software dal punto di vista dell'utente.

**V**

**Variabile di ambiente** È una variabile dinamica che viene utilizzata dal sistema operativo e dai programmi in esecuzione per memorizzare informazioni di configurazione o parametri specifici relativi all'ambiente di esecuzione. Le variabili di ambiente forniscono un modo per configurare il comportamento di software, applicazioni e script senza dover modificare direttamente il codice.

**Visual Studio Code** È un editor di codice sorgente open-source sviluppato da Microsoft, progettato per supportare una vasta gamma di linguaggi di programmazione e tecnologie. È leggero, altamente estensibile e dotato di funzionalità avanzate per lo sviluppo di software.

**W**

**Walkthrough** È una tecnica di revisione per analizzare e migliorare la qualità di un documento, di un codice sorgente o di un sistema. Si tratta di una revisione guidata e semi-formale, in cui l'autore del documento o del codice presenta il proprio lavoro a un gruppo di revisori per raccogliere feedback, identificare problemi e migliorare il prodotto.

**Way of working** Il termine "way of working" in informatica si riferisce al modo in cui un team, un'organizzazione o un progetto opera, gestisce e coordina le proprie attività per raggiungere i propri obiettivi. Include le metodologie, le pratiche, gli strumenti, le comunicazioni e le procedure seguite nel corso dello sviluppo e della gestione del lavoro.

**Workflow** È una sequenza definita di attività, compiti o processi che vengono eseguiti per completare una specifica operazione o obiettivo. I workflow possono essere automatizzati o manuali e sono utilizzati per coordinare, ottimizzare e monitorare i flussi di lavoro all'interno di un sistema informatico, di un'applicazione o di un'organizzazione.

**X****Y**

**YAML** È un formato di serializzazione dei dati, leggibile dall'uomo, utilizzato per rappresentare informazioni in modo strutturato. Viene spesso impiegato per configurazioni, scambi di dati e definizione di strutture nei contesti di sviluppo software.

**Z**