



SWEvenTeam

E-Mail

sweventeam@outlook.it

GLOSSARIO

Informazioni documento

Versione	1.0.0
Redazione	Alessio Barraco Alessandro Damiani Yuri Lunardon Matteo Mazzotti Valentina Schivo Alessio Turetta
Verifica	Alessio Barraco Alessandro Damiani Yuri Lunardon Matteo Mazzotti Valentina Schivo Alessio Turetta
Approvazione	Valentina schivo

Storia del documento

Versione	Data	Autori	Verificatori	Descrizione
1.0.0	2025-03-21	Valentina Schivo	-	Approvazione RTB
0.3.0	2025-03-14	Yuri Lunardon	Matteo Mazzotti, Alessio Turetta	Controllo finale
0.2.2	2025-02-26	Alessandro Damiani	Alessio Barraco	Migliorie finali documento
0.2.1	2025-02-10	Alessio Turetta	Yuri Lunardon	Aggiunti nuovi termini
0.2.0	2025-01-29	Alessio Barraco	Alessandro Damiani	Miglioramento definizioni
0.1.1	2024-12-23	Alessandro Damiani	Alessio Barraco	Aggiunti termini specifici
0.1.0	2024-12-10	Yuri Lunardon, Alessio Turetta	Valentina Schivo	Inseriti termini principali
0.0.3	2024-11-22	Matteo Mazzotti	Alessandro Damiani	Scrittura primi termini
0.0.2	2024-11-16	Alessio Barraco	Alessandro Damiani	Introduzione documento
0.0.1	2024-11-12	Alessandro Damiani	Alessio Turetta	Definizione struttura

Indice

1	Introduzione	3
2	Glossario	4
	A	4
	B	5
	C	6
	D	7
	E	8
	F	9
	G	10
	H	11
	I	12
	J	13
	K	14
	L	15
	M	16
	N	17
	O	18
	P	19
	Q	20
	R	21
	S	22
	T	23
	U	24
	V	25
	W	26
	X	27
	Y	27
	Z	27

1 Introduzione

Il glossario si pone l'obiettivo di minimizzare le incomprensioni tra i membri del gruppo, l'azienda e i professori su ciò che viene scritto e in particolare sulla terminologia specifica, utilizzata sia nelle discussioni del gruppo che nella scrittura della documentazione. Ha quindi lo scopo di perseguire chiarezza e unicità.

Il glossario viene strutturato con un ordine alfabetico, per rendere la navigazione agevole, e ogni termine ritenuto degno di una definizione chiara viene riportato nella sezione alfabetica corrispondente, con la relativa definizione accettata da tutti i membri del gruppo.

La presenza di un termine all'interno del Glossario negli altri documenti viene indicata applicando questo stile_G.

2 Glossario

A

- **Allucinazione:** Fenomeno in cui un modello linguistico genera informazioni false, inesatte o non supportate dai dati di addestramento, presentandole come se fossero corrette;
- **Amministratore:** Si occupa del controllo e della gestione dell'ambiente di lavoro, assicurando il rispetto delle norme di progetto. I suoi principali compiti sono gestire e risolvere le problematiche legate ai processi, gestire il versionamento della documentazione, gestire la configurazione del prodotto e migliorare l'ambiente di lavoro fornendo gli strumenti necessari;
- **Analisi dei Requisiti:** Attività a cui viene accompagnato un documento di specifiche, che si pone l'obiettivo di delineare tutte le funzionalità e i requisiti che il prodotto deve rispettare per essere giudicato conforme e aderente alle specifiche richieste dal capitolato e dall'azienda. Il documento di analisi dei requisiti ha anche lo scopo di rendere chiare le azioni che l'utente può svolgere utilizzando l'applicazione stessa;
- **Analista:** Ruolo presente nelle fasi iniziali del progetto, ha il compito di identificare, documentare e comprendere a fondo le esigenze del progetto trasformandole in requisiti chiari e dettagliati. I suoi principali compiti sono: analizzare il contesto di riferimento e stabilire gli obiettivi, comprendere la complessità del problema e definire i requisiti, raccogliere e analizzare i bisogni espliciti e impliciti dei committenti e redigere il documento "Analisi dei Requisiti";
- **API:** Interfaccia software che consente a due applicazioni di comunicare tra loro, facilitando lo scambio di dati o l'esecuzione di funzionalità predefinite. Nel contesto dell'informatica e dei modelli linguistici di grandi dimensioni (LLM), le API permettono di integrare e utilizzare i servizi offerti dai modelli, come la generazione di testo o l'elaborazione del linguaggio naturale, all'interno di applicazioni personalizzate. Le API sono tipicamente strutturate attraverso endpoint e possono utilizzare protocolli come REST per inviare e ricevere richieste e risposte;
- **Architettura:** L'organizzazione fondamentale di un sistema, definita dai suoi componenti principali, dalle loro interazioni e dalle linee guida che ne governano il design e l'evoluzione;
- **Artefatto:** Elemento generato o prodotto durante il processo di sviluppo del software. Rappresenta risultati intermedi o finali di tale processo e può includere file, documenti, codice sorgente o compilato, oppure qualsiasi altro elemento creato durante lo sviluppo o la fase di build di un'applicazione software;
- **Attività:** Un'azione o un insieme di specifiche azioni che contribuiscono al completamento di un processo. Le attività sono generalmente focalizzate su un singolo obiettivo e dovrebbero produrre risultati misurabili;
- **Attore:** Nei casi d'uso, utente che si interfaccia con il sistema per assolvere ad un'azione, definito dalle precondizioni, a uno stato definito dalle postcondizioni.

B

- **Backend:** La parte di un'applicazione o sistema che gestisce la logica di business, l'elaborazione dei dati e la comunicazione con il database;
- **Baseline:** Una versione fissa e approvata di un prodotto di lavoro, che funge da punto di riferimento per ulteriori sviluppi o modifiche. Rappresenta uno stato stabile del sistema, utilizzato per garantire il controllo delle modifiche e la tracciabilità durante il ciclo di vita del progetto. Sostanzia il raggiungimento di una data milestone;
- **Best Practice:** Raccomandazioni, metodologie e approcci consolidati che guidano lo sviluppo, la gestione e la manutenzione del software in modo efficace ed efficiente. Le best practice si basano su esperienze comprovate e mirano a ottimizzare la qualità del prodotto, ridurre i rischi, migliorare la produttività e garantire la sostenibilità del codice nel tempo;
- **Board:** In GitHub, una Board è uno strumento di gestione dei progetti che consente di organizzare e tracciare le attività attraverso una visualizzazione tipo "kanban". Le board permettono di creare e gestire issue e pull request, suddividendole in colonne per monitorare lo stato di avanzamento delle attività;
- **Branch:** Su GitHub, è una linea separata di sviluppo all'interno di un repository, usata per lavorare su funzionalità o correzioni senza modificare direttamente il ramo principale (main o master);
- **Budget At Completion (BAC):** Costo complessivo del progetto pianificato per eseguire tutte le attività e consegnare il prodotto finale. Rappresenta il costo originario previsto per completare il lavoro;
- **Bug:** Malfunzionamento che causa un comportamento anomalo e non previsto. La causa è in genere attribuibile ad errori nel codice.

C

- **Capitolato:** Documento formale che definisce i requisiti, le specifiche tecniche, i vincoli e gli obiettivi di un progetto software. Il capitolato funge da accordo tra il cliente e il fornitore, stabilendo cosa deve essere realizzato, come deve essere sviluppato e quali standard devono essere rispettati. È un documento essenziale e stabilisce il primo approccio tra l'azienda proponente e il gruppo di lavoro;
- **Caso d'uso:** Possibile circostanza in cui l'attore, colui che si interfaccia con il sistema, può ritrovarsi. Prevede un attore, un obiettivo, una preconditione, uno scenario che descrive i passi del caso d'uso, e una postcondizione;
- **Checklist:** Un elenco di voci o attività da verificare o completare, utilizzato per assicurarsi che tutti i passi necessari siano stati eseguiti correttamente;
- **Code coverage:** Misura della quantità di codice che è stato testato. Maggiore è la code coverage, più ampio è il numero di linee di codice che è stato testato;
- **Commit:** Operazione che salva le modifiche apportate ai file nel repository, creando uno snapshot del progetto, ossia una fotografia istantanea dello stato attuale del repository. Ogni commit include un messaggio descrittivo delle modifiche effettuate;
- **Committente:** La persona o entità che commissiona un progetto o una parte di esso, definendo gli obiettivi, i requisiti e le aspettative. In ambito software, il committente è generalmente il cliente che richiede lo sviluppo di un'applicazione o di una soluzione specifica, e che ha il ruolo di approvare o validare il lavoro svolto;
- **Consuntivo:** Rendiconto di quanto svolto in un periodo di attività preciso. Viene spesso confrontato con le previsioni iniziali, per valutarne le performance;
- **Cruscotto:** Un'interfaccia che visualizza in tempo reale informazioni e metriche chiave sullo stato di un sistema o progetto, facilitando il monitoraggio e la gestione.

D

- **Database:** Un sistema organizzato per raccogliere, archiviare e gestire grandi quantità di dati in modo strutturato. I dati sono memorizzati in tabelle, che possono essere interrogate e manipolate tramite linguaggi come SQL;
- **Deployment:** Processo di distribuzione e rilascio di un'applicazione o di una nuova versione del software in un ambiente di produzione o su un server, rendendola accessibile agli utenti finali;
- **Discord:** Piattaforma dove gli utenti possono comunicare attraverso videochiamate e messaggi. Permette di creare server, organizzazioni, dove gli utenti possono riunirsi. Utilizzata per riunioni interne;
- **Diagramma dei casi d'uso:** Una rappresentazione grafica che evidenzia gli attori e i servizi del sistema. È costituito da un grafo in cui i nodi rappresentano gli attori e i casi d'uso, mentre gli archi indicano la comunicazione tra attori e i casi d'uso e le relazioni tra i casi d'uso stessi;
- **Diagramma delle classi:** Diagramma UML per rappresentare la struttura statica di un sistema software. Mostra le classi di un'applicazione, i loro attributi e metodi, oltre alle relazioni tra di esse, come associazioni, ereditarietà e dipendenze;
- **Django:** Un web framework basato su Python, open source, progettato per lo sviluppo di applicazioni web, seguendo il paradigma "Model-Template-View";
- **Docker:** Una piattaforma open-source che automatizza il processo di sviluppo, distribuzione e esecuzione di applicazioni all'interno di contenitori (container). Questi contenitori permettono di isolare l'applicazione e le sue dipendenze dal sistema operativo sottostante, garantendo portabilità e consistenza tra ambienti di sviluppo, test e produzione. Docker semplifica la gestione delle applicazioni scalabili e la gestione di microservizi.

E

- **Economicità:** Misura che mette in relazione l'efficacia con l'efficienza, indica la capacità di raggiungere obiettivi prefissati perseguendo efficienza, ovvero ottimizzando al meglio le risorse disponibili;
- **Efficacia:** La capacità di raggiungere gli obiettivi prefissati o di produrre il risultato desiderato;
- **Efficienza:** Capacità di raggiungere un obiettivo ottimizzando al meglio le risorse che si hanno a disposizione;
- **Embedding:** Organizzazione di parole in strutture semantico-sintattiche a forma di spazi vettoriali, dove i vettori parola sono tanto più vicini quanto la relazione semantica tra le due parole. Viene utilizzato dagli LLM per rappresentare il linguaggio in una forma tale da essere processata;
- **Endpoint:** Un punto di accesso a un servizio o a un'API che permette la comunicazione tra diversi sistemi o applicazioni. Rappresenta un'URL o un indirizzo specifico a cui inviare richieste per eseguire operazioni o ottenere dati;
- **Estimate at Completion (EAC):** Stima del costo totale previsto per completare il progetto software. Può variare nel tempo, riflettendo problemi, ritardi o miglioramenti nella produttività.

F

- **Fornitore:** Ente, azienda o individuo che fornisce un servizio, prodotto o componente software a un altro ente, azienda o individuo che l'ha richiesto. Si impegna a rispettare i requisiti stabiliti con il committente;
- **Framework:** Raccolta di componenti software riutilizzabili, come librerie, funzionalità già pronte, strumenti e linee guida per sviluppo di applicazioni, con l'obiettivo di semplificare e velocizzare il lavoro dei programmatori;
- **Frontend:** Complementare al backend, è la parte di prodotto software che permette agli utenti di interfacciarsi e interagire con il sistema. Presenta e rende fruibili all'utente i risultati del backend, attraverso un'interfaccia grafica;
- **Funzionalità:** Caratteristica di un sistema software che soddisfa una esigenza dell'utente o un requisito del progetto.

G

- **Gemini:** Famiglia di modelli di intelligenza artificiale sviluppata da Google DeepMind, progettata per compiti avanzati di comprensione del linguaggio naturale, ragionamento e multi-modalità;
- **Gestione del progetto:** Insieme di attività, metodi, strumenti e competenze utilizzati per pianificare, organizzare, monitorare e completare un progetto, garantendo il raggiungimento degli obiettivi prefissati nei tempi e nei costi stabiliti;
- **Git:** Software di controllo di versione, gratuito e open source, tra i più utilizzati, che permette di gestire il tracciamento delle modifiche nel codice sorgente durante lo sviluppo software. Fondamentale per la gestione del controllo delle versioni, risulta utile per lo sviluppo collaborativo in progetti di gruppo;
- **GitHub:** Piattaforma online di versionamento e collaborazione che utilizza Git per il controllo della versione del codice. Consente agli sviluppatori di archiviare, gestire e condividere progetti software, facilitando il lavoro di squadra grazie a funzionalità come repository, issue tracking, pull request e integrazione continua;
- **GitHub Action:** Funzionalità di GitHub che consentono in risposta a determinati eventi, di avviare un'automazione che svolge alcuni flussi di lavoro all'interno di una repository, ad esempio, il test, il build e il deploy del codice;
- **Google Meet:** Piattaforma di Google progettata per le videochiamate. Utilizzata per confrontarsi con l'azienda.

H

- **Hosting:** In ambito LLM, ospitare gli LLM su infrastrutture adeguate, che possono includere server locali o servizi cloud. L'hosting efficace degli LLM assicura che gli utenti possano interagire con i modelli in modo efficiente, sfruttando appieno le loro capacità di generazione e comprensione del linguaggio naturale.

I

- **IEC:** Acronimo di International Electrotechnical Commission, è un'organizzazione internazionale che studia e pubblica standard per ogni tecnologia elettrica, elettronica e correlate;
- **ISO:** Acronimo di International Organization for Standardization, organizzazione di riferimento a livello mondiale per le normative di qualità, definite come standard nel settore;
- **Issue:** Strumento per segnalare, discutere e gestire attività o problemi legati a un progetto. Ogni issue include una sezione commenti che consente ai team e ai collaboratori di comunicare e lavorare insieme, semplificando il monitoraggio e l'organizzazione delle attività nello sviluppo software;
- **Issue Tracking System (ITS):** Strumento software progettato per contenere, monitorare e gestire problemi, richieste o attività all'interno di un progetto. Permette al gruppo di lavoro di tenere tracciato lo stato di ogni issue aperta, di discutere sulla issue e facilita la collaborazione e una precisa organizzazione del lavoro.

J

- **JSON**: Formato standard per lo scambio di dati basato su coppie chiave-valore. Si distingue per la sua compatibilità con vari linguaggi di programmazione, la sua semplicità e la sua facile leggibilità.

K

- **Know how:** Insieme delle conoscenze, esperienze e abilità consolidate in un certo ambito, tali da poter essere usate per risolvere attività specifiche e problemi.

L

- **LangChain:** Software per facilitare l'integrazione di grandi LLM all'interno di un prodotto software;
- **LaTeX:** Linguaggio di marcatura usato per la preparazione di testi e la scrittura di documentazione professionale;
- **Libreria:** Insieme di funzioni, classi, o moduli predefiniti e riutilizzabili, progettati per semplificare lo sviluppo di software. Le librerie forniscono codice già scritto per compiti specifici, consentendo agli sviluppatori di integrare queste funzionalità nei propri progetti senza doverle scrivere da zero;
- **LLM (Large Language Model):** Modello di intelligenza artificiale, basato su reti neurali di grandi dimensioni, e addestrato su grandi quantità di dati. Viene progettato per generare testo in linguaggio naturale, comprensibile all'uomo. Analizza una richiesta ed è in grado di generare testo tale da rispondere alla richiesta;
- **LLM di test:** Nel nostro progetto il LLM che genererà delle risposte alla serie di domande che gli verranno fornite. Rappresenta LLM di cui bisognerà poi valutare la correttezza della risposta fornita;
- **LLM di valutazione:** LLM più potente rispetto a quello da testare, che riceverà le risposte generate dal LLM di test e ne confronterà la coerenza con le risposte standard o con le risposte inserite dall'utente;
- **Log:** Registrazione sequenziale e cronologica delle azioni svolte in un sistema informatico, ad esempio un server o un client.

M

- **Manuale Utente:** Manuale destinato all'utente pensato per guidare l'utente all'utilizzo dell'app. Deve essere un supporto per un utente in difficoltà e viene scritto con un linguaggio semplice e chiaro;
- **Manuale Sviluppatore:** Manuale destinato agli sviluppatori che si occuperanno di mantenere o estendere l'applicazione. Utilizza un linguaggio tecnico e puntuale e deve contenere tutte le informazioni utili sul lato tecnologico dell'applicazione, fornendo un quadro completo della struttura e composizione del prodotto software;
- **Manutenzione:** Processo finale del ciclo di vita del software, che si occupa di estendere il prodotto secondo le volontà degli stakeholders, oppure risolvere bug che si sono riscontrati o verranno riscontrati. Rappresenta una delle fasi più costose e importanti del ciclo di vita del prodotto software;
- **Metrica:** Misura quantitativa che identifica quanto bene sta andando un lavoro e quanto è buono, ben sviluppato, un aspetto software;
- **Milestone:** Rappresenta un punto temporale nel progetto dove si sono raggiunti determinati obiettivi;
- **Merge:** Operazione in Git che si pone l'obiettivo di unificare e combinare i cambiamenti effettuati in due o più branch di un repository. Il merge può generare conflitti che vanno risolti manualmente;
- **Mock:** Oggetto simulato, utilizzato nei test, che replica il comportamento di un componente reale di sistema. Solitamente utilizzato per lo studio del comportamento delle dipendenze di un oggetto;
- **MVP (Minimum Viable Product):** Versione base di un prodotto, che soddisfa solo i requisiti minimi. Viene utilizzata per ottenere feedback da parte di stakeholders, rimanendo pronta per futuri processi di miglioramento;
- **MySQL:** Sistema di gestione di database relazionale (RDBMS) open-source, basato sul linguaggio SQL (Structured Query Language). MySQL è noto per essere scalabile, veloce e affidabile, e supporta molteplici utenti e accessi simultanei.

N

- **Next.js:** Framework open-source basato su React, che fornisce funzionalità avanzate, garantendo applicazioni più performanti e scalabili, per lo sviluppo di applicazioni web;
- **Norme di progetto:** Regole e linee guida formulate per un preciso progetto, con lo scopo di regolamentare e uniformare i processi.

O

- **Ollama:** Piattaforma che fornisce API per la creazione, esecuzione e gestione di LLM sul dispositivo dell'utente, eliminando la necessità di connessioni cloud o di inviare dati a server esterni;
- **Open source:** Licenza software che permette agli utenti di accedere, modificare e distribuire liberamente il codice sorgente di un'applicazione.

P

- **Piattaforma:** Ambiente hardware e software che fornisce risorse e servizi per sviluppare, eseguire e gestire applicazioni o servizi;
- **Postcondizione:** Parte di un caso d'uso. Stato in cui si trovano l'attore e il sistema dopo che è avvenuta l'interazione specificata dallo stesso caso d'uso;
- **Precondizione:** Parte di un caso d'uso. Stato in cui si trovano l'attore e il sistema prima che avvenga l'interazione specificata dallo stesso caso d'uso;
- **Preventivo:** Stima che quantifica le risorse necessarie e i costi di una determinata attività o di un periodo;
- **Processo:** Insieme strutturato di attività correlate e interdipendenti, progettate per raggiungere un obiettivo specifico;
- **Procedura:** Sequenza dettagliata e standardizzata di passi da seguire per completare un'attività in modo coerente ed efficiente;
- **Product baseline (PB):** Seconda revisione obbligatoria del progetto didattico, che mira a valutare la maturità della baseline architetturale del prodotto;
- **Programmatore:** Colui che mette in pratica le direttive scelte dal progettista e scrive codice per un sistema software;
- **Progettista:** Persona responsabile di definire l'architettura e il design di un sistema software;
- **Progetto:** Insieme organizzato di attività e processi con lo scopo di sviluppare un prodotto;
- **Proponente:** L'individuo, l'ente o l'azienda che presenta un'iniziativa, un progetto o un'idea da sviluppare;
- **Proof of Concept (PoC):** Artefatto sotto forma di demo eseguibile, con l'intento di dimostrare la fattibilità tecnologica di un progetto;
- **Protocollo:** Insieme di regole di comunicazione tra due dispositivi elettronici;
- **Pull:** In Git, operazione di recupero dati dal repository remoto aggiornando il repository locale;
- **Pull request:** Proposta da parte di un collaboratore di un repository di apportare le modifiche nel branch principale;
- **Push:** In Git, operazione conseguente ai commit per caricare le modifiche nel repository remoto.

Q

- **Qualità:** Insieme delle caratteristiche di un'entità che ne determinano la capacità di soddisfare esigenze sia espresse che implicite. Misura in cui un prodotto software soddisfa i requisiti specificati e le aspettative implicite di funzionamento, affidabilità, efficienza, manutenibilità e usabilità;
- **Quantizzazione:** Tecnica utilizzata all'interno dei modelli linguistici di grandi dimensioni (LLM) per convertire pesi e valori di attivazione di dati ad alta precisione, in dati a precisione inferiore. Può comportare una lieve perdita di precisione nelle risposte del modello, ma porta vantaggi significativi in termini di velocità ed efficienza, soprattutto su hardware con risorse limitate.

R

- **RDBMS**: Sistema di gestione di database basato sul modello relazionale, in cui ogni tabella è collegata alle altre tramite relazioni di chiave primaria e chiave esterna;
- **React**: Libreria JavaScript open-source per la creazione di interfacce utente (UI) interattive e dinamiche. Utilizza un approccio basato su componenti riutilizzabili, che consentono di suddividere l'interfaccia in parti modulari e gestibili;
- **Redattore**: Colui che si occupa della scrittura di una parte o dell'intero documento;
- **Responsabile**: Chi gestisce il progetto e guida il gruppo. Ha la responsabilità delle scelte, delle approvazioni e della rappresentanza del team verso l'esterno;
- **Repository**: Spazio di archiviazione digitale utilizzato per conservare e gestire i file di un progetto, in particolare nel contesto del controllo delle versioni del software;
- **Requirements and Technology Baseline (RTB)**: Prima revisione obbligatoria del progetto didattico, in cui vengono fissati i requisiti da soddisfare e si dimostra, attraverso il PoC, la fattibilità tecnologica;
- **Requisito**: Descrizione di una funzionalità che il sistema deve implementare o di un vincolo che deve rispettare;
- **Rischio**: Evento incerto che potrebbe verificarsi durante il corso di un progetto e compromettere il suo normale andamento;
- **Risposta (fornita dall'utente)**: Risposta inserita dall'utente, considerata corretta, che risponde a una delle domande test poste al LLM;
- **Risultato (elaborazione LLM)**: Risposta generata dal LLM a una delle domande di test fornite dall'utente;
- **Requisiti funzionali**: Specificano le funzionalità e i comportamenti che un sistema deve possedere per soddisfare le esigenze degli utenti. Descrivono cosa il sistema deve fare e includono operazioni, calcoli, gestione dei dati e altri aspetti operativi;
- **Requisiti soluzione**: Definiscono i vincoli tecnici e le caratteristiche che la soluzione deve rispettare per soddisfare i requisiti funzionali e non funzionali. Possono includere architettura del sistema, tecnologie utilizzate e standard da seguire;
- **Requisiti utente**: Descrivono i bisogni e le aspettative degli utenti finali nei confronti del sistema. Sono espressi in un linguaggio naturale e rappresentano i desideri e gli obiettivi dell'utente, senza entrare nei dettagli tecnici della soluzione.

S

- **Servizio:** Un servizio si riferisce a una funzionalità o a un'opzione offerta da un sistema;
- **Sistema:** Il termine sistema si riferisce a un insieme di componenti interconnessi che lavorano insieme per raggiungere uno scopo comune;
- **Sprint:** Periodo di tempo fisso in un ciclo di sviluppo continuo, in cui il team completa il lavoro programmato;
- **Stakeholder:** Tutti coloro che a vario titolo hanno influenza sul prodotto e sul progetto;
- **Standard di qualità:** Cataloghi di processi già formati, accettati e seguiti dall'industria, che aiutano a raggiungere l'economicità;
- **Standard ISO/IEC 12207:1995:** Standard che definisce un quadro di riferimento per i processi del ciclo di vita del software, stabilendo attività e compiti necessari per l'acquisizione, lo sviluppo, il funzionamento, la manutenzione e la dismissione dei sistemi software.

T

- **Telegram:** Applicazione di messaggistica istantanea e piattaforma di comunicazione;
- **Test:** Attività di verifica e validazione di un sistema o componente per assicurarsi che funzioni correttamente, soddisfi i requisiti specificati e sia privo di errori;
- **Temperatura:** Parametro che regola il grado di casualità (o “creatività”) nelle risposte di un modello di linguaggio. Valori più bassi (es. 0) rendono le risposte più deterministiche e focalizzate, mentre valori più alti (es. 1) aumentano la variabilità e la creatività, ma possono introdurre incoerenze.

U

- **UML**: Linguaggio di modellazione visuale, utilizzato nel contesto dello sviluppo software per rappresentare e documentare la progettazione di un sistema software, senza servirsi del linguaggio naturale, in modo tale da usare un linguaggio universale e chiaro, piuttosto che un linguaggio che può risultare ambiguo;
- **URL**: Indirizzo univoco che identifica una risorsa su una rete, tipicamente su Internet.

V

- **Validazione dei requisiti:** Attività che consiste nell'accertare che il prodotto corrisponda alle attese;
- **Verifica dei requisiti:** Attività che consiste nell'accertare che lo svolgimento delle attività di sviluppo non introduce errori;
- **Versionamento:** Attività di gestione e controllo delle diverse versioni di un prodotto, documento o software durante il suo ciclo di vita;
- **Verificatore:** La persona che verifica che quanto svolto dagli altri ruoli, venga svolto seguendo le norme di progetto e il way of working stabilito. Verifica inoltre che il sistema o le componenti, rispettino i requisiti specificati e soddisfino gli standard di qualità previsti.

W

- **Way of Working:** Metodologie e strumenti adottati e sviluppati da un gruppo nel corso di un progetto affinché si raggiunga un modo di lavorare efficiente, efficace e professionale, “allo stato dell’arte”;
- **Workflow:** Un workflow rappresenta il flusso o la sequenza di attività, processi o azioni che vengono eseguite in un particolare ordine per completare un compito o raggiungere un obiettivo specifico.

X

Nessun termine.

Y

Nessun termine.

Z

Nessun termine.