



## BugBusters

Email: `bugbusters.unipd@gmail.com`

Gruppo: 4

**Università degli Studi di Padova**

Laurea in Informatica

Corso: Ingegneria del Software

Anno Accademico: 2025/2026

# Glossario

<b>Redattori</b>	Alberto Autiero
<b>Verificatori</b>	[Nome Cognome]
<b>Uso</b>	Interno
<b>Destinatari</b>	Prof. Tullio Vardanega, Prof. Riccardo Cardin

## Indice

<b>Versioni del documento</b>	<b>6</b>
<b>Introduzione</b>	<b>7</b>
<b>A</b>	<b>8</b>
ACM (Access Control Mechanism)	8
Accoppiamento	8
Agile	8
Aggregazione	8
AI Assistant Generativo	8
AI Co-Pilot	8
Allocazione delle risorse	8
Amministratore	8
Analisi dei Requisiti	9
Analista	9
Angular	9
Ansible	9
API	9
Approvazione	9
Architettura	9
Associazione	9
Attore	9
Attore principale	10
Attore secondario	10
Audit Trail	10
AUM (Authentication Mechanism)	10
AWS	10
AWS CDK V2	10
<b>B</b>	<b>11</b>
Backend	11
Baseline	11
BERT (Bidirectional Encoder Representations from Transformers)	11
Best Practice	11
BLE (Bluetooth Low Energy)	11
Brainstorming	11
Branch	11
Broker MQTT	11
<b>C</b>	<b>12</b>
Capitolato	12
Caso d'uso	12
Cedolini Massivi	12
Cerimonia	12
CI/CD (Continuous Integration/Continuous Deployment)	12
Classe	12
CNN (Convolutional Neural Network)	12
Committente	12
Composizione	12
Confidence Score	13
Consulente del Lavoro (CdL)	13
Cross-Site Scripting (XSS)	13
Cruscotto/Dashboard	13

<b>D</b>	<b>14</b>
Decisione interna . . . . .	14
Decision tree . . . . .	14
Diagrammi di Gantt . . . . .	14
Diagrammi di PERT . . . . .	14
Diagrammi UML . . . . .	14
DIP (Distribution Information Package) . . . . .	14
Dipendenza . . . . .	14
Discord . . . . .	14
Dispatch . . . . .	14
Distan Writing . . . . .	15
Docker . . . . .	15
Dominio d'uso . . . . .	15
<b>E</b>	<b>16</b>
EasyOCR . . . . .	16
Economicità . . . . .	16
Efficacia . . . . .	16
Efficienza . . . . .	16
EN18031 . . . . .	16
Entity Resolution . . . . .	16
Ereditarietà . . . . .	16
ERP (Enterprise Resource Planning) . . . . .	16
<b>F</b>	<b>17</b>
FAISS (Facebook AI Similarity Search) . . . . .	17
Flutter . . . . .	17
Funzionalità . . . . .	17
<b>G</b>	<b>18</b>
GATT (Generic Attribute Profile) . . . . .	18
GDPR (General Data Protection Regulation) . . . . .	18
Gestione dei rischi . . . . .	18
GitHub . . . . .	18
Glossario . . . . .	18
GPT (Generative Pre-trained Transformer) . . . . .	18
GraphQL . . . . .	18
<b>H</b>	<b>19</b>
Human-in-the-Loop . . . . .	19
<b>I</b>	<b>20</b>
IaC (Infrastructure as Code) . . . . .	20
IDE . . . . .	20
Incapsulamento . . . . .	20
Information Hiding . . . . .	20
IoT (Internet of Things) . . . . .	20
Issue . . . . .	20
<b>K</b>	<b>21</b>
KNX . . . . .	21
KNX IoT . . . . .	21
KPI (Key Performance Indicator) . . . . .	21
<b>L</b>	<b>22</b>
Latex . . . . .	22
LLM (Large Language Model) . . . . .	22

<b>M</b>	<b>23</b>
Machine Learning (ML)	23
Markdown	23
Milestone	23
MongoDB	23
MQTT (Message Queuing Telemetry Transport)	23
Multi-tenant	23
MVP (Minimum Viable Product)	23
<b>N</b>	<b>24</b>
Nest.js	24
NLP (Natural Language Processing)	24
Norme di progetto	24
<b>O</b>	<b>25</b>
OAuth2	25
Obsidian	25
Oggetto	25
OCR (Optical Character Recognition)	25
Ora produttiva vs ora di orologio	25
OWASP (Open Web Application Security Project)	25
<b>P</b>	<b>26</b>
Pair Programming	26
Piano di Qualifica	26
Polimorfismo	26
Post-condizione	26
PostgreSQL	26
Pre-condizione	26
Preventivo	26
Progettista	26
Programmatore	26
Progetto	27
Prometheus	27
Prompt	27
Proponente	27
Processi organizzativi	27
Processi primari	27
Processi di supporto	27
Pull Request	27
<b>R</b>	<b>28</b>
RBAC (Role-Based Access Control)	28
React	28
RED (Radio Equipment Directive)	28
Redattore	28
Repository	28
Requisito	28
Requisiti desiderabili	28
Requisiti funzionali	28
Requisiti non funzionali	28
Requisiti obbligatori	29
Requisiti opzionali	29
Requirements Baseline	29
Requisito software	29

Requisito utente . . . . .	29
Responsabile . . . . .	29
Retention Policy . . . . .	29
RoBERTa (Robustly optimized BERT approach) . . . . .	29
RTB (Requirements and Technology Baseline) . . . . .	29
<b>S</b>	<b>30</b>
Sandbox di Sviluppo . . . . .	30
Scenario . . . . .	30
Scenario alternativo . . . . .	30
Scenario principale . . . . .	30
SCRUM . . . . .	30
SEMAT . . . . .	30
Sei Cappelli per Pensare . . . . .	30
Serverless . . . . .	30
Specializzazione di processi . . . . .	31
SQL Injection . . . . .	31
SSL (Secure Sockets Layer) . . . . .	31
Specifica Tecnica . . . . .	31
Sprint . . . . .	31
Stima dei costi . . . . .	31
<b>T</b>	<b>32</b>
Technology Baseline . . . . .	32
Terraform . . . . .	32
Tesseract OCR . . . . .	32
Test . . . . .	32
TLS (Transport Layer Security) . . . . .	32
Transformer . . . . .	32
TypeScript . . . . .	32
<b>U</b>	<b>33</b>
UML (Unified Modeling Language) . . . . .	33
Use Case (Casi d'uso) . . . . .	33
UWB (Ultra Wide Band) . . . . .	33
<b>V</b>	<b>34</b>
Validazione . . . . .	34
Verifica . . . . .	34
Verificatore . . . . .	34
<b>W</b>	<b>35</b>
Way of Working . . . . .	35
Whisper . . . . .	35
<b>Z</b>	<b>36</b>
Zigbee . . . . .	36

## Versioni del documento

Versione	Data	Descrizione	Redatto	Verificatori	Approvato
0.5.0	[DATA]	Aggiunti i termini: Way of Working, Economicità, Processi primari, Processi di supporto, Processi organizzativi, Specializzazione di processi, Progetto, Preventivo, Ora produttiva vs ora di orologio, Diagrammi di Gantt, Diagrammi di PERT, Allocazione delle risorse, Stima dei costi, Gestione dei rischi, RTB, SEMAT, Ereditarietà, Polimorfismo, Information hiding, Classe, Incapsulamento, Oggetto, Dipendenza, Accoppiamento, Associazione, Aggregazione, Composizione, Attore, Scenario	Alberto Autiero	-	-
0.4.0	28/10/2025	Aggiunte le parole: Brainstorming, Cerimonia, Glossario, Requisiti obbligatori, Requisiti desiderabili, Requisiti opzionali, Technology baseline	Alberto Autiero	-	-
0.3.0	24/10/2025	Aggiunti i seguenti termini: Attore principale, Attore secondario, Dominio d'uso, Piano di qualifica, Post-condizione, Pre-condizione, Requirements baseline, Scenario alternativo, Scenario principale, Use case (casi d'uso), Validazione, Verifica	Alberto Autiero	-	-
0.2.0	23/10/2025	Aggiunti i seguenti termini: Committente, Diagrammi UML, Efficienza, Requisito, Requisito software, Requisito utente	Alberto Autiero	-	-
0.1.0	22/10/2025	Prima stesura del documento	Alberto Autiero	-	-

## Introduzione

Questo documento nasce con lo scopo di evitare qualsiasi tipo di ambiguità o dubbi riguardanti la terminologia adoperata all'interno dei documenti del progetto. Per questo motivo, vengono di seguito esposte le definizioni dei termini specifici, degli acronimi e delle parole ambigue utilizzati nella documentazione, adottando una struttura alfabetica per facilitare la navigazione del documento. Il glossario ha lo scopo di garantire una comprensione univoca dei termini da parte di tutti i membri del team e dei destinatari della documentazione.

## A

### **ACM (Access Control Mechanism)**

Meccanismo di controllo degli accessi che determina quali utenti o sistemi possono accedere a specifiche risorse e con quali privilegi.

### **Accoppiamento**

Grado di interdipendenza tra componenti software. Un accoppiamento stretto indica forte dipendenza, mentre un accoppiamento debole minimizza le dipendenze, favorendo manutenibilità e riutilizzo del codice.

### **Agile**

Metodologia di sviluppo del software iterativa e incrementale che si basa su principi come la collaborazione continua con il cliente, la consegna frequente di software funzionante e la capacità di rispondere ai cambiamenti dei requisiti.

### **Aggregazione**

Relazione strutturale in OOP che rappresenta un legame "parte-tutto" dove gli componenti possono esistere indipendentemente dall'oggetto che li contiene. È una forma di associazione con condivisione dei riferimenti.

### **AI Assistant Generativo**

Sistema di intelligenza artificiale in grado di creare autonomamente contenuti testuali, immagini o altri media a partire da prompt o dati di input, utilizzato per generare comunicati, testi HR e contenuti aziendali.

### **AI Co-Pilot**

Modulo di supporto automatizzato per gli studi dei Consulenti del Lavoro (CdL), che assiste nei processi di gestione, riconoscimento e distribuzione documentale attraverso tecniche di intelligenza artificiale.

### **Allocazione delle risorse**

Processo di assegnazione delle risorse disponibili (persone, tempo, strumenti) alle attività di progetto, cercando di bilanciare carichi di lavoro e ottimizzare l'utilizzo delle risorse stesse.

### **Amministratore**

Figura responsabile della gestione dell'infrastruttura, degli strumenti di sviluppo e dei processi del progetto. Si occupa di configurare e mantenere gli ambienti di lavoro e garantire che il team abbia a disposizione gli strumenti necessari.



## **Analisi dei Requisiti**

Processo sistematico volto a identificare, documentare e validare i bisogni, i vincoli e le esigenze del progetto, trasformandoli in requisiti formali che guideranno lo sviluppo del software.

## **Analista**

Figura professionale specializzata nell'analisi dei requisiti e nella specifica delle funzionalità del sistema. Collabora con gli stakeholder per comprendere le esigenze e tradurle in specifiche tecniche.

## **Angular**

Framework JavaScript open source per lo sviluppo di applicazioni web frontend, mantenuto da Google. Offre un'architettura basata su componenti e strumenti per lo sviluppo di single-page applications.

## **Ansible**

Strumento di automazione IT per il provisioning, la gestione della configurazione e il deployment di applicazioni. Utilizza un linguaggio dichiarativo basato su YAML.

## **API**

Application Programming Interface. Insieme di definizioni, protocolli e strumenti per la costruzione e l'integrazione di software applicativo. Consente a diversi sistemi di comunicare tra loro.

## **Approvazione**

Processo formale attraverso il quale un documento o una funzionalità viene accettata e considerata completa dopo aver superato le verifiche e validazioni necessarie.

## **Architettura**

Struttura fondamentale di un sistema software, comprendente i suoi componenti, le relazioni tra di essi e i principi che ne guidano la progettazione e l'evoluzione.

## **Associazione**

Relazione strutturale in OOP che rappresenta una connessione duratura tra oggetti di classi diverse, dove gli oggetti interagiscono per un periodo di tempo prolungato condividendo riferimenti.

## **Attore**

Ruolo svolto da un utente o sistema esterno che interagisce con il sistema software per raggiungere obiettivi specifici. Gli attori possono essere primari (beneficiari diretti) o secondari (fornitori di servizi).

**Attore principale**

Utente o sistema esterno che interagisce direttamente con il sistema software per raggiungere un obiettivo specifico nel caso d'uso. È il protagonista dell'interazione e trae beneficio diretto dall'esecuzione del caso d'uso.

**Attore secondario**

Utente o sistema esterno che fornisce servizi o supporto all'attore principale durante l'esecuzione di un caso d'uso. Partecipa all'interazione ma non è il beneficiario principale del risultato, spesso fornendo funzionalità accessorie o di supporto.

**Audit Trail**

Registro cronologico delle operazioni svolte nel sistema, utile per verifiche, tracciabilità e compliance, che documenta chi ha fatto cosa e quando.

**AUM (Authentication Mechanism)**

Meccanismo di autenticazione che verifica l'identità di un utente o di un sistema prima di concedere l'accesso a risorse protette.

**AWS**

Amazon Web Services. Piattaforma di cloud computing che offre servizi di calcolo, storage, database e altre funzionalità per supportare lo sviluppo e il deployment di applicazioni.

**AWS CDK V2**

AWS Cloud Development Kit versione 2. Framework di sviluppo che permette di definire l'infrastruttura cloud utilizzando linguaggi di programmazione familiari invece di template.

## B

### Backend

Parte di un'applicazione software che gestisce la logica di business, l'elaborazione dei dati e la comunicazione con il database. Opera sul server ed è inaccessibile direttamente all'utente finale.

### Baseline

Versione approvata e formalmente controllata di un documento o di un componente software che serve come riferimento per sviluppi successivi. Le modifiche successive richiedono procedures formali di controllo.

### BERT (Bidirectional Encoder Representations from Transformers)

Modello di linguaggio bidirezionale pre-addestrato da Google, utilizzato per comprendere il contesto delle parole in una frase e per varie applicazioni di NLP.

### Best Practice

Insieme di tecniche, metodi e procedure che sono state riconosciute come le più efficaci ed efficienti per raggiungere un obiettivo specifico in un determinato contesto.

### BLE (Bluetooth Low Energy)

Protocollo di comunicazione wireless progettato per consumare poca energia, ideale per dispositivi IoT e wearable che richiedono lunga autonomia.

### Brainstorming

Tecnica di gruppo per generare idee creative e soluzioni a problemi specifici, incoraggiando la libera espressione senza critiche immediate.

### Branch

Ramo di sviluppo in un sistema di controllo versione come Git. Permette di lavorare su funzionalità isolate senza influenzare il codice principale fino al completamento e al merge.

### Broker MQTT

Componente centrale nel protocollo MQTT che riceve i messaggi dai publisher e li inoltra ai subscriber interessati, gestendo le sottoscrizioni e la distribuzione dei messaggi.

## C

### Capitolato

Documento contrattuale che specifica i requisiti, le caratteristiche tecniche e le condizioni di un progetto software proposto da un'azienda proponente per il corso di Ingegneria del Software.

### Caso d'uso

Tecnica di specifica dei requisiti che descrive le interazioni tra gli attori (utenti o sistemi esterni) e il sistema software per raggiungere un obiettivo specifico.

### Cedolini Massivi

File contenenti più documenti retributivi aggregati, da suddividere e assegnare ai singoli destinatari tramite riconoscimento automatico.

### Cerimonia

Evento formale o informale nel framework Scrum che segue un agenda prestabilita e ha uno scopo specifico, come la Pianificazione dello Sprint, il Daily Stand-up, la Revisione dello Sprint o la Retrospettiva.

### CI/CD (Continuous Integration/Continuous Deployment)

Insieme di pratiche e strumenti che automatizzano l'integrazione, il testing e il deployment del codice, permettendo rilasci frequenti e affidabili.

### Classe

Costrutto fondamentale nella programmazione orientata agli oggetti che definisce un modello per creare oggetti, fornendo implementazioni per stato (attributi) e comportamento (metodi).

### CNN (Convolutional Neural Network)

Tipo di rete neurale artificiale specializzata nell'elaborazione di dati con struttura a griglia, come le immagini, utilizzando operazioni di convoluzione.

### Committente

Soggetto che commissiona il progetto, definendone gli obiettivi, i vincoli e i requisiti, e che riceve il prodotto finale. Nel contesto del corso di Ingegneria del Software, può essere un'azienda proponente o un docente.

### Composizione

Relazione strutturale in OOP che rappresenta un legame "parte-tutto" forte dove gli componenti non possono esistere indipendentemente dall'oggetto che li contiene. Implica ownership esclusiva e distruzione concomitante.

**Confidence Score**

Misure numerica (0-1 o 0-100%) che indica la probabilità che un riconoscimento automatico sia corretto, utilizzata per valutare l'affidabilità dei risultati di sistemi AI.

**Consulente del Lavoro (CdL)**

Professionista specializzato nella gestione delle risorse umane, della payroll, della compliance normativa e delle relazioni sindacali per aziende e dipendenti.

**Cross-Site Scripting (XSS)**

Vulnerabilità di sicurezza web che permette a un attaccante di iniettare script malevoli in pagine web visualizzate da altri utenti.

**Cruscotto/Dashboard**

Interfaccia utente che presenta in forma grafica e sintetica le metriche, gli indicatori di performance e lo stato corrente del progetto o dell'applicazione.

## D

### **Decisione interna**

Scelta presa dal team di progetto riguardante aspetti tecnici, organizzativi o metodologici, documentata per garantire tracciabilità e coerenza nelle attività successive.

### **Decision tree**

Algoritmo di machine learning che utilizza una struttura ad albero per modellare decisioni e i loro possibili esiti, spesso utilizzato per classificazione e regressione.

### **Diagrammi di Gantt**

Strumento di pianificazione progettuale che visualizza le attività su un asse temporale, mostrando durate, sequenzialità, parallelismo e progressi rispetto alle pianificazioni.

### **Diagrammi di PERT**

(Program Evaluation and Review Technique) Strumento di analisi delle dipendenze temporali tra attività di progetto, utilizzato per identificare cammini critici e margini temporali (slack time).

### **Diagrammi UML**

Rappresentazioni grafiche che utilizzano il linguaggio UML (Unified Modeling Language) per modellare sistemi software, inclusi diagrammi di classe, di sequenza, dei casi d'uso, di attività e altri, che descrivono struttura e comportamento del sistema.

### **DIP (Distribution Information Package)**

Formato standardizzato per la distribuzione di pacchetti di informazioni nella conservazione digitale, contenente metadati e dati oggetto di conservazione.

### **Dipendenza**

Relazione tra componenti software per cui un componente (dipendente) richiede un altro componente (dipenduto) per funzionare correttamente. Le modifiche al componente dipenduto possono influenzare il componente dipendente.

### **Discord**

Piattaforma di comunicazione utilizzata dal team per la coordinazione quotidiana, le riunioni e la condivisione di informazioni in tempo reale.

### **Dispatch**

Processo di distribuzione automatizzata dei documenti verso i destinatari finali attraverso diversi canali (es. app, portale, email, PEC).

**Distan Writing**

Tecnica di scrittura in cui l'utente specifica cosa vuole scrivere ma lascia la scrittura effettiva all'intelligenza artificiale, utilizzata nel "vibe coding" e nella generazione di contenuti.

**Docker**

Piattaforma per lo sviluppo, spedizione ed esecuzione di applicazioni all'interno di container software, che permettono di isolare l'applicazione e le sue dipendenze.

**Dominio d'uso**

Contesto specifico o ambiente operativo in cui il sistema software sarà impiegato, comprendente le caratteristiche degli utenti finali, le condizioni operative, i vincoli tecnologici e le regole di business che definiscono l'ambito di applicazione del prodotto.

## **E**

### **EasyOCR**

Modulo Python per estrarre testo da immagini, supporta oltre 80 lingue e vari script di scrittura, utilizzato per l'OCR in applicazioni di document processing.

### **Economicità**

Principio gestionale che combina efficienza ed efficacia, misurando la capacità di raggiungere obiettivi prefissati (efficacia) impiegando le risorse minime indispensabili (efficienza).

### **Efficacia**

Capacità di raggiungere gli obiettivi prefissati e produrre i risultati attesi, indipendentemente dalle risorse impiegate.

### **Efficienza**

Rapporto tra i risultati ottenuti e le risorse impiegate per conseguirli. Un processo è efficiente quando raggiunge i suoi obiettivi utilizzando il minimo di risorse necessarie.

### **EN18031**

Serie di standard europei che definiscono requisiti di sicurezza informatica per apparecchiature radio connesse (radio equipment) ai fini della conformità alla Radio Equipment Directive (RED). Copre temi come controllo accessi, autenticazione, aggiornamenti sicuri, storage protetto e protezione della privacy.

### **Entity Resolution**

Processo di identificazione e associazione di entità (es. persone o aziende) a partire da dati parziali o duplicati, tramite algoritmi di matching e disambiguazione.

### **Ereditarietà**

Meccanismo della programmazione orientata agli oggetti che permette a una classe (sottoclasse) di acquisire attributi e metodi di un'altra classe (superclasse), favorendo il riutilizzo del codice e le relazioni di generalizzazione.

### **ERP (Enterprise Resource Planning)**

Sistema software integrato che gestisce i processi aziendali fondamentali, come produzione, acquisti, vendite, contabilità e risorse umane.



## **F**

### **FAISS (Facebook AI Similarity Search)**

Libreria sviluppata da Facebook per la ricerca efficiente di similarità e clustering di vettori di grandi dimensioni, comunemente usata in applicazioni di machine learning.

### **Flutter**

Framework di Google per lo sviluppo di applicazioni multiplatforma (iOS, Android, Web) utilizzando un unico codice base scritto in Dart.

### **Funzionalità**

Caratteristica o capacità specifica che un sistema software deve possedere per soddisfare i bisogni degli utenti e gli obiettivi del progetto.

## G

### **GATT (Generic Attribute Profile)**

Protocollo Bluetooth Low Energy che definisce come i dispositivi scambiano dati utilizzando servizi e caratteristiche, standard per la comunicazione tra dispositivi BLE.

### **GDPR (General Data Protection Regulation)**

Regolamento generale sulla protezione dei dati dell'Unione Europea che stabilisce norme per la protezione e la libera circolazione dei dati personali.

### **Gestione dei rischi**

Processo sistematico di identificazione, analisi, pianificazione e controllo dei rischi di progetto, volto a minimizzare la probabilità di occorrenza e l'impatto degli eventi negativi.

### **GitHub**

Piattaforma di hosting per repository Git che offre strumenti per il version control, la collaborazione e la gestione del ciclo di vita del software.

### **Glossario**

Documento che raccoglie e definisce i termini specifici, gli acronimi e le parole ambigue utilizzati nella documentazione di progetto, con lo scopo di garantire una comprensione univoca della terminologia da parte di tutti i membri del team e dei destinatari.

### **GPT (Generative Pre-trained Transformer)**

Serie di modelli di linguaggio sviluppati da OpenAI, progettati per comprendere il contesto, generare contenuti e ragionare attraverso testo, immagini e altro.

### **GraphQL**

Linguaggio di query e runtime per API che permette ai client di richiedere esattamente i dati di cui hanno bisogno, riducendo il over-fetching e under-fetching.

## H

### **Human-in-the-Loop**

Approccio in cui l'intelligenza artificiale e gli esseri umani collaborano, con l'uomo che supervisiona, corregge o fornisce feedback al sistema AI, particolarmente utile quando la confidenza del sistema è bassa.

## I

### **IaC (Infrastructure as Code)**

Approccio alla gestione dell'infrastruttura che utilizza file di configurazione e script per automatizzare il provisioning e la gestione delle risorse, trattando l'infrastruttura come software.

### **IDE**

Integrated Development Environment. Ambiente di sviluppo integrato che fornisce strumenti completi per la scrittura, il testing e il debugging del codice in un'unica interfaccia.

### **Incapsulamento**

Principio della programmazione orientata agli oggetti che consiste nel racchiudere in un'unica entità (classe) dati e metodi che operano su di essi, nascondendo i dettagli implementativi all'esterno.

### **Information Hiding**

Principio di progettazione software che consiste nel nascondere i dettagli implementativi di un modulo, esponendo solo le interfacce necessarie, per ridurre l'accoppiamento e aumentare la manutenibilità.

### **IoT (Internet of Things)**

Rete di dispositivi fisici connessi a Internet, in grado di raccogliere e scambiare dati, come sensori, elettrodomestici smart, veicoli e sistemi di automazione.

### **Issue**

Segnalazione di un problema, un bug o una richiesta di miglioramento nel sistema di tracking del progetto. Ogni issue viene tracciata, assegnata e gestita fino alla risoluzione.

## **K**

### **KNX**

Standard internazionale per l'automazione degli edifici e la domotica, che permette il controllo integrato di illuminazione, riscaldamento, sicurezza e altri sistemi.

### **KNX IoT**

Protocollo di comunicazione standardizzato per l'automazione degli edifici e la domotica, estensione di KNX per dispositivi IoT con interfacce REST per il controllo degli impianti.

### **KPI (Key Performance Indicator)**

Indicatore chiave di performance che misura l'efficacia di un processo, un'attività o un'organizzazione nel raggiungere i propri obiettivi.

## L

### **Latex**

Sistema di composizione tipografica utilizzato per la produzione di documentazione tecnica e scientifica di alta qualità, particolarmente adatto per documenti complessi con formule matematiche.

### **LLM (Large Language Model)**

Modello di linguaggio di grandi dimensioni addestrato su vasti corpus di testo, in grado di generare, comprendere e elaborare linguaggio naturale in modo sofisticato.

## M

### **Machine Learning (ML)**

Campo dell'intelligenza artificiale che si occupa di sviluppare algoritmi che permettono ai computer di apprendere pattern dai dati e migliorare le proprie performance con l'esperienza.

### **Markdown**

Linguaggio di markup leggero per la formattazione di testo, utilizzato per documentazione, readme e messaggi, che può essere convertito in HTML e altri formati.

### **Milestone**

Punto significativo nel ciclo di vita del progetto che segna il completamento di un insieme di attività o il raggiungimento di un obiettivo importante. Serve come punto di verifica del progresso.

### **MongoDB**

Database NoSQL document-oriented che memorizza i dati in formato JSON-like, offrendo flessibilità nello schema e scalabilità orizzontale.

### **MQTT (Message Queuing Telemetry Transport)**

Protocollo di messaggistica leggero e efficiente progettato per dispositivi con risorse limitate e reti a larghezza di banda ridotta, ampiamente usato in IoT.

### **Multi-tenant**

Architettura software in cui una singola istanza dell'applicazione serve multiple organizzazioni (tenant), mantenendo l'isolamento dei dati e della configurazione.

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

Versione minima di un prodotto che include solo le funzionalità essenziali necessarie per soddisfare gli utenti iniziali e raccogliere feedback per sviluppi futuri.

## N

### **Nest.js**

Framework per la costruzione di applicazioni server-side efficienti e scalabili in Node.js, utilizzando TypeScript e ispirato ad Angular.

### **NLP (Natural Language Processing)**

Campo dell'intelligenza artificiale che si occupa dell'interazione tra computer e linguaggio umano, includendo compiti come traduzione, analisi del sentiment e riconoscimento di entità.

### **Norme di progetto**

Insieme di regole, procedures e convenzioni stabilite dal team per garantire coerenza, qualità e uniformità nelle attività di sviluppo e nella documentazione prodotta.



## O

### **OAuth2**

Protocollo standard per l'autorizzazione che permette a terze parti di accedere a risorse protette senza condividere le credenziali dell'utente.

### **Obsidian**

Software per la gestione di note personali basato su file Markdown, con funzionalità di linking tra note e grafo della conoscenza, utilizzato per el personal knowledge management.

### **Oggetto**

Istanza di una classe nella programmazione orientata agli oggetti, caratterizzata da uno stato (valori degli attributi) e un comportamento (metodi) definiti dalla classe di appartenenza.

### **OCR (Optical Character Recognition)**

Tecnologia che converte immagini di testo scritto o stampato in testo digitale machine-readable.

### **Ora produttiva vs ora di orologio**

Distinzione tra il tempo effettivamente dedicato a compiti produttivi (ore produttive) e il tempo totale trascorso (ore di orologio). Il rapporto tra queste due metriche indica l'efficienza nell'utilizzo del tempo.

### **OWASP (Open Web Application Security Project)**

Comunità globale che si occupa di sicurezza delle applicazioni web, nota per la pubblicazione della OWASP Top 10 sulle vulnerabilità web più critiche.

## P

### Pair Programming

Pratica di sviluppo in cui due programmatori lavorano insieme su un'unica workstation, condividendo la scrittura del codice e la risoluzione dei problemi in tempo reale.

### Piano di Qualifica

Documento che definisce le attività, le risorse, le metriche e le procedure per garantire la qualità del processo di sviluppo e del prodotto software. Descrive le strategie di verifica e validazione, gli standard qualitativi di riferimento e le modalità di controllo e misurazione della qualità.

### Polimorfismo

Principio della programmazione orientata agli oggetti che permette a oggetti di classi diverse di rispondere allo stesso messaggio (metodo) in modo specifico per la propria classe, favorendo flessibilità ed estensibilità del codice.

### Post-condizione

Condizione o stato del sistema che deve essere vero dopo il completamento di un caso d'uso. Definisce il risultato atteso e le garanzie che il sistema fornisce al termine dell'esecuzione del caso d'uso.

### PostgreSQL

Sistema di gestione di database relazionale open source estensibile e conforme agli standard SQL, noto per la sua robustezza e funzionalità avanzate.

### Pre-condizione

Condizione o stato del sistema che deve essere vero prima che un caso d'uso possa iniziare. Definisce i prerequisiti necessari per l'esecuzione corretta del caso d'uso.

### Preventivo

Documento di pianificazione che stima i costi e le risorse necessarie per lo svolgimento del progetto, basato sulle attività pianificate e sulle risorse disponibili.

### Progettista

Figura responsabile della progettazione dell'architettura software e delle soluzioni tecniche, garantendo che soddisfino i requisiti e siano realizzabili efficientemente.

### Programmatore

Figura che implementa il codice sorgente secondo le specifiche tecniche, seguendo le best practice e gli standard di qualità definiti nel progetto.

**Progetto**

Insieme di attività che devono raggiungere obiettivi specifici a partire da date specifiche, con un inizio e una fine fissate in calendario, disponendo di risorse limitate (persone, tempo, denaro, strumenti) e consumando risorse nel loro svolgersi.

**Prometheus**

Sistema di monitoraggio e alerting open source progettato per l'affidabilità e la scalabilità, ampiamente utilizzato per il monitoring di applicazioni e infrastrutture.

**Prompt**

Input (testo e/o altri dati) fornito a un modello di intelligenza artificiale per ottenere una risposta o un output specifico.

**Proponente**

Azienda o organizzazione che propone un capitolato d'appalto per il progetto del corso di Ingegneria del Software, definendone requisiti e obiettivi.

**Processi organizzativi**

Processi trasversali rispetto ai singoli progetti che riguardano la gestione dei processi, delle infrastrutture, del miglioramento continuo e della formazione del personale nell'organizzazione.

**Processi primari**

Processi fondamentali che agiscono direttamente sul ciclo di vita del prodotto software, includendo acquisizione, fornitura, sviluppo, operazione e manutenzione.

**Processi di supporto**

Processi che supportano i processi primari, includendo documentazione, gestione della configurazione, accertamento della qualità, verifica, validazione e risoluzione dei problemi.

**Pull Request**

Richiesta di integrazione di modifiche da un branch di sviluppo al branch principale nel sistema di version control, soggetta a revisione e approvazione.

## R

### **RBAC (Role-Based Access Control)**

Modello di controllo degli accessi in cui i permessi sono assegnati a ruoli specifici, e gli utenti ottengono i permessi attraverso l'assegnazione a questi ruoli.

### **React**

Libreria JavaScript open source per la costruzione di interfacce utente, mantenuta da Facebook. Basata su componenti riutilizzabili e un virtual DOM per performance ottimizzate.

### **RED (Radio Equipment Directive)**

Direttiva europea che stabilisce i requisiti per la messa sul mercato di apparecchiature radio, inclusi i dispositivi che utilizzano comunicazioni wireless.

### **Redattore**

Membro del team responsabile della stesura e della produzione dei documenti di progetto, garantendo chiarezza, completezza e conformità alle norme stabilite.

### **Repository**

Archivio centrale in cui vengono memorizzati e versionati i file sorgente, la documentazione e le risorse del progetto utilizzando un sistema di controllo versione.

### **Requisito**

Condizione o capacità che deve essere posseduta da un sistema o componente software per soddisfare un contratto, standard, specifica o altro documento formalmente imposto.

### **Requisiti desiderabili**

Requisiti che sono importanti ma non essenziali per il funzionamento base del sistema. La loro implementazione apporta valore aggiunto ma la loro assenza non compromette il progetto.

### **Requisiti funzionali**

Specificano cosa il sistema deve fare, descrivendo le funzionalità, i comportamenti e le interazioni che il software deve supportare.

### **Requisiti non funzionali**

Definiscono come il sistema deve comportarsi in termini di prestazioni, sicurezza, affidabilità, usabilità e altri attributi di qualità, senza riguardo alle funzionalità specifiche.

**Requisiti obbligatori**

Requisiti che devono essere necessariamente soddisfatti e la cui mancata implementazione comporterebbe il fallimento del progetto. Sono critici per il successo del sistema.

**Requisiti opzionali**

Requisiti che sono utili ma non necessari, e la cui implementazione dipende dalla disponibilità di risorse e tempo. Possono essere considerati per versioni future del prodotto.

**Requirements Baseline**

Insieme dei requisiti concordati e formalmente approvati che costituisce il riferimento per lo sviluppo del progetto. Una volta stabilita, qualsiasi modifica alla baseline dei requisiti deve seguire un processo formale di controllo delle modifiche.

**Requisito software**

Requisito specificato in termini tecnici, destinato agli sviluppatori, che descrive in modo dettagliato e misurabile una funzionalità o un vincolo del sistema software.

**Requisito utente**

Requisito espresso dal punto di vista dell'utente finale, descritto in linguaggio naturale e senza dettagli tecnici, focalizzato su ciò che l'utente si aspetta che il sistema faccia.

**Responsabile**

Figura di riferimento del progetto con compiti di coordinamento, pianificazione, gestione delle risorse e comunicazione con docenti e proponenti.

**Retention Policy**

Politica che stabilisce per quanto tempo i dati o documenti devono essere conservati prima di essere eliminati, in base a requisiti legali o aziendali.

**RoBERTa (Robustly optimized BERT approach)**

Variante ottimizzata del modello BERT per la comprensione del linguaggio naturale, addestrata con strategie più efficaci per migliorare le performance.

**RTB (Requirements and Technology Baseline)**

Milestone progettuale che combina la baseline dei requisiti con la baseline tecnologica, formalizzando l'accordo su cosa sviluppare e con quali tecnologie, dopo aver valutato la fattibilità tecnica.

## S

### **Sandbox di Sviluppo**

Ambiente isolato per testare e validare funzionalità senza influenzare i sistemi di produzione, utilizzato per sviluppare e verificare nuove feature in sicurezza.

### **Scenario**

Sequenza di interazioni tra attori e sistema che descrive un percorso specifico attraverso un caso d'uso. Può essere principale (percorso di successo) o alternativo (variazioni ed eccezioni).

### **Scenario alternativo**

Sequenza di interazioni nel caso d'uso che rappresenta un percorso diverso da quello principale, tipicamente gestendo condizioni eccezionali, errori o scelte alternative dell'utente. Descrive come il sistema reagisce in situazioni non standard.

### **Scenario principale**

Sequenza di interazioni tra l'attore principale e il sistema che descrive il percorso di successo del caso d'uso, dove l'obiettivo viene raggiunto senza intoppi o condizioni eccezionali. Rappresenta il flusso ideale e più frequente di esecuzione.

## **SCRUM**

Framework agile per la gestione dello sviluppo software che enfatizza lo sviluppo iterativo, l'adattamento ai cambiamenti e la consegna incrementale di valore.

## **SEMAT**

(Software Engineering Method and Theory) Iniziativa internazionale per rifondare l'ingegneria del software come disciplina rigorosa, basata su un kernel di elementi essenziali comuni a tutti i metodi di sviluppo software.

### **Sei Cappelli per Pensare**

Tecnica di pensiero laterale sviluppata da Edward De Bono che utilizza sei cappelli metaforici per rappresentare diverse prospettive di pensiero (fatti, emozioni, critica, ottimismo, creatività, controllo).

### **Serverless**

Architettura cloud computing in cui il provider cloud gestisce dinamicamente l'allocazione delle risorse di macchina, permettendo agli sviluppatori di focalizzarsi sul codice senza gestire l'infrastruttura.

**Specializzazione di processi**

Adattamento dei processi standard alle specifiche esigenze del progetto, considerando fattori come dimensione, complessità, rischi, competenze del team e tecnologie utilizzate.

**SQL Injection**

Vulnerabilità di sicurezza che permette a un attaccante di interferire con le query che un'applicazione effettua verso il database, potenzialmente accedendo a dati sensibili.

**SSL (Secure Sockets Layer)**

Vecchio protocollo per connessioni cifrate tra client e server; è stato sostituito da TLS e non è più ritenuto sicuro.

**Specifica Tecnica**

Documento che descrive in dettaglio l'architettura, il design e le scelte implementative del sistema software, guidando le attività di sviluppo.

**Sprint**

Periodo di tempo fisso (tipicamente 2-4 settimane) in Scrum durante il quale il team sviluppa e consegna un incremento di prodotto potenzialmente rilasciabile.

**Stima dei costi**

Processo di previsione dei costi associati alle attività di progetto, considerando risorse umane, strumenti, infrastrutture e altri fattori che influenzano il budget complessivo.

## T

### **Technology Baseline**

Insieme delle tecnologie, framework, librerie e strumenti di sviluppo selezionati e approvati per il progetto. Definisce lo stack tecnologico di riferimento e costituisce la base per le scelte implementative, garantendo coerenza e standardizzazione nell'architettura software.

### **Terraform**

Strumento di Infrastructure as Code (IaC) che permette di definire e provisionare infrastruttura cloud usando un linguaggio dichiarativo (HCL), supportando multiple piattaforme cloud.

### **Tesseract OCR**

Motore OCR open source sviluppato originariamente da HP e successivamente mantenuto da Google, noto per la sua accuratezza nel riconoscimento del testo.

### **Test**

Processo sistematico di verifica che il software soddisfi i requisiti specificati e identifichi difetti, attraverso l'esecuzione controllata di casi di test.

### **TLS (Transport Layer Security)**

Protocollo di sicurezza che fornisce comunicazioni private e integrità dei dati tra due applicazioni che comunicano attraverso una rete, successore di SSL.

### **Transformer**

Architettura di rete neurale basata su meccanismi di attenzione, rivoluzionaria per le attività di elaborazione del linguaggio naturale e altre sequenze.

### **TypeScript**

Superset di JavaScript che aggiunge tipizzazione statica opzionale e altre funzionalità; il codice TypeScript viene compilato in JavaScript eseguibile.



## U

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

Linguaggio di modellazione visuale standardizzato per la specifica, costruzione, visualizzazione e documentazione dei sistemi software, utilizzato per rappresentare diverse viste di un sistema attraverso diagrammi.

### **Use Case (Casi d'uso)**

Tecnica di specifica dei requisiti che descrive le interazioni tra gli attori (utenti o sistemi esterni) e il sistema software per raggiungere un obiettivo specifico. I casi d'uso catturano i requisiti funzionali del sistema descrivendo scenari di utilizzo tipici, alternativi ed eccezionali.

### **UWB (Ultra Wide Band)**

Tecnologia di comunicazione wireless a corto raggio che utilizza una larghezza di banda molto ampia per comunicazioni ad alta velocità e precisione nel posizionamento.

## V

### **Validazione**

Processo che accerta che il prodotto software sviluppato soddisfi le effettive esigenze del cliente e gli obiettivi di business per i quali è stato realizzato. Risponde alla domanda "Stiamo costruendo il prodotto giusto?" e viene tipicamente effettuata attraverso test di accettazione con il cliente.

### **Verifica**

Processo sistematico che determina se i prodotti di lavoro (documenti, codice, componenti) soddisfano i requisiti e le specifiche definite per loro. Risponde alla domanda "Stiamo costruendo il prodotto nel modo giusto?" e include attività come revisioni, ispezioni e test.

### **Verificatore**

Membro del team responsabile di controllare che documenti, codice e altri prodotti di lavoro rispettino gli standard di qualità definiti, le norme di progetto e siano privi di errori, incoerenze o ambiguità.

## W

### **Way of Working**

Insieme di processi, metodologie, strumenti e pratiche adottati dal team per organizzare e svolgere le attività di progetto in modo coordinato ed efficiente. Definisce come il team collabora, comunica e gestisce il lavoro quotidiano.

### **Whisper**

Sistema di riconoscimento vocale automatico sviluppato da OpenAI, in grado di trascrivere, tradurre e identificare la lingua parlata in audio.

## **Z**

### **Zigbee**

Protocollo di comunicazione wireless a basso consumo energetico basato sullo standard IEEE 802.15.4, utilizzato in reti mesh per l'automazione domestica e industriale.