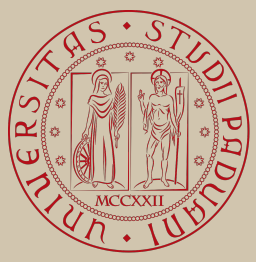
Glossario

2025-02-09

V1.0

sweetenteam@gmail.com

https://sweetenteam.github.io



Destinatari	Prof. Tullio Vardanega Prof. Riccardo Cardin
Redattori	Orlando Ferazzani
Verificatori	Valeri Mihail Belenkov Mouad Mahdi Andrea Santi

Registro delle modifiche

Versione	Data	Autori	Verificatori	Dettaglio
1.0	2025-02-09	Orlando Ferazzani	Mouad Mahdi	Approvazione per RTB
0.3	2024-12-09	Orlando Ferazzani	Valeri Mihail Belenkov	Aggiunte nuove voci al glossario
0.2	2024-11-25	Orlando Ferazzani	Andrea Santi	Aggiunte nuove voci al glossario
0.1	2024-11-18	Orlando Ferazzani	Andrea Santi	Prima stesura del glossario

Indice

1) Introduzione

In questo documento sono raccolti tutti i termini tecnici e le loro definizioni utilizzati all'interno del progetto. Per facilitarne la consultazione, il glossario è disponibile anche sul sito al seguente link. Inoltre, in ogni documento, i termini presenti nel glossario sono indicati dal colore beige e la sottolineatura, oltre ad avere una *G* al pedice.

2) A

2.1) Agile

La programmazione Agile è caratterizzata da uno sviluppo iterativo e incrementale. Si concentra sulla realizzazione di un MVP il più rapidamente possibile, ottenendo un feedback continuo dai clienti e rispondendo ai cambiamenti dei requisiti o della tecnologia.

2.2) Amministratore

Un amministratore è una persona responsabile del controllo e dell'amministrazione dell'ambniente di lavoro utilizzato dal gruppo ed è anche il punto di riferimento per quanto concerne le norme di progetto. Le sue principali mansioni sono: affrontare e risolvere le problematiche associate alla gestione dei processi; gestire l'archiviazione e le versioni della documentazione; gestire la configurazione del prodotto; redigere ed attuare le norme e le procedure per la gestione della qualità; amministrare le infrastrutture e i servizi per i processi di supporto.

2.3) Analisi Dei Requisiti

L'*analisi dei requisiti*$_G$ è una fase fondamentale nello sviluppo del software, che comprende la raccolta, l'analisi, la documentazione e la comprensione delle esigenze e delle specifiche richieste dagli stakeholder per un sistema o un'applicazione software da sviluppare. Durante questa fase si formalizza quanto raccolto e si ottiene un documento dei requisiti, che descrive in modo chiaro e dettagliato le funzionalità del sistema.

2.4) Analista

Figura professionale con competenze avanzate riguardo l'attività di analisi dei requisiti ed il dominio applicativo del problema. Il suo ruolo è quello di identificare, documentare e comprendere a fondo le esigenze e le specifiche del progetto, traducendole in requisiti chiari e dettagliati. Si occupa di: analizzare il contesto di riferimento, definire il problema in esame e stabilire gli obiettivi da raggiungere; comprendere il problema e definire la complessità e i requisiti, redigere il documento Analisi dei requisiti; studiare i bisogni espliciti ed impliciti.

2.5) API

Un'API è un'interfaccia che consente a due sistemi software di comunicare tra loro. Le API definiscono un insieme di regole e protocolli che specificano come i dati devono essere inviati e ricevuti. Sono utilizzate, ad esempio, per collegare frontend e backend, accedere a servizi di terze parti (come mappe, notifiche o dati meteo) e sviluppare applicazioni integrate.

2.6) Artefatti

Documenti, diagrammi, codici o output creati durante lo sviluppo software, utilizzati per supportare il processo di sviluppo e la comunicazione tra i team.

3) B

3.1) Bad Practice

Pratica di sviluppo o gestione software considerata inefficace o dannosa per la qualità del prodotto.

3.2) Baseline

Riferimento fisso in un progetto, come specifiche, piani o documenti, utilizzato per monitorare e confrontare i progressi.

3.3) Best Practice

Un insieme di metodi o approcci riconosciuti come i più efficaci e affidabili per raggiungere un obiettivo. Le best practice sono spesso basate su esperienze consolidate e standard del settore.

3.4) BuddyBot

Assistente virtuale in forma di chatbot capace di reperire informazioni da diverse fonti (GitHub, Confluence e Jira) e di fornire supporto e assistenza ai membri del team.

3.5) Budget at Completion

Costo totale pianificato di un progetto alla sua conclusione.

3.6) Bug

Un errore o malfunzionamento in un programma software che causa risultati imprevisti o indesiderati. I bug possono derivare da errori di programmazione, progettazione o configurazione.

3.7) Branch

Un branch è una linea indipendente di sviluppo all'interno di un repository. Consente di creare una copia separata del codice principale (solitamente chiamato main o master) per lavorare su nuove funzionalità, correzioni di bug o esperimenti, senza modificare il codice stabile.

4) C

4.1) camelCase

Uno stile di scrittura di identificatori simile a PascalCase, ma con la prima parola in lettere minuscole e le successive con iniziali maiuscole. Esempio: `myVariableName`.

4.2) Capitolato

Documento che contiene le specifiche e le condizioni per lo sviluppo di un progetto software. II capitolato viene redatto dal proponente e viene presentato ai fornitori o agli sviluppatori interessati a partecipare all'appalto per la realizzazione del prodotto software.

4.3) Chat

Sistema di comunicazione testuale in tempo reale tra utenti.

4.4) Chatbot

Applicazione software che utilizza algoritmi di intelligenza artificiale per simulare una conversazione umana. I chatbot sono progettati per interagire con gli utenti attraverso messaggi di testo o vocali, rispondendo a domande, eseguendo attività o fornendo supporto in maniera automatica e personalizzata.

4.5) Commit

Operazione di salvataggio nel repository Git che memorizza una versione specifica dei file, permettendo di tracciare le modifiche nel tempo.

4.6) Committente

Il committente è la persona o l'organizzazione che richiede la realizzazione di un progetto software e finanzia il suo sviluppo. Il committente definisce i requisiti e le specifiche del progetto e valuta il prodotto finale rispetto ai propri obiettivi e aspettative.

4.7) Confluence

Confluence è uno strumento collaborativo di gestione della conoscenza e documentazione, sviluppato da Atlassian, progettato per aiutare i team a creare, organizzare e condividere contenuti in un'unica piattaforma centralizzata.

4.8) Content Switch

Un cambiamento dinamico e fluido del contenuto visualizzato in una pagina web o interfaccia, spesso senza ricaricare la pagina. Implementato comunemente con tecnologie come JavaScript e framework come React.

4.9) Context Switch

Processo con cui il gruppo sostituisce un membro impossibilitato a svolgere una precisa task, con un altro membro del team, in modo da garantire la continuità del lavoro.

4.10) Contratto

Un accordo formale tra due o più parti che stabilisce obblighi, diritti e responsabilità reciproche, solitamente in forma scritta e legalmente vincolante. In ambito software, un contratto può riguardare la fornitura di servizi, lo sviluppo di un progetto o l'uso di un prodotto.

4.11) Cruscotto

Un cruscotto è un'interfaccia utente che fornisce una panoramica visiva delle informazioni più importanti, utilizzando grafici, tabelle e altri elementi visivi per rappresentare i dati in modo chiaro e conciso. I cruscotti sono ampiamente utilizzati in diversi contesti, come il monitoraggio delle prestazioni aziendali, la visualizzazione dei dati di analisi e la gestione dei progetti. Nel contesto dello sviluppo software, si riferisce a un'interfaccia grafica o a un'applicazione web che fornisce una panoramica visiva delle metriche e delle statistiche rilevanti per il controllo della qualità del software. Questo cruscotto di solito raccoglie dati da varie fonti, come sistemi di gestione del versionamento del codice, strumenti di test automatizzati, sistemi di monitoraggio delle prestazioni e altri strumenti di analisi. Il suo obiettivo principale è quello di fornire agli sviluppatori, ai tester e ai responsabili un modo rapido ed efficace per valutare lo stato del progetto, identificare eventuali problemi o anomalie e prendere decisioni basate sui dati per migliorare la qualità del software.

5) D

5.1) Database Vettoriale

Un database vettoriale è progettato per archiviare e gestire dati sotto forma di vettori, che sono rappresentazioni numeriche di informazioni come testo, immagini o audio. Questi database sono ottimizzati per eseguire ricerche di similarità su grandi insiemi di dati, ad esempio per trovare elementi simili in base al loro contesto o contenuto. Sono comunemente usati in applicazioni di machine learning, AI e ricerca semantica.

5.2) Dev team

In Scrum, il Development Team si riferisce all'insieme dei membri che svolge un ruolo nello sviluppo e nel supporto del prodotto e può includere ricercatori, architetti, designer, programmatori, e molti altri.

5.3) Design Pattern

Soluzioni riutilizzabili a problemi ricorrenti nella progettazione del software. I design pattern offrono linee guida per scrivere codice più efficiente, leggibile e mantenibile. Esempi includono Singleton, Factory e Observer.

5.4) Diagramma dei casi d'uso

Un diagramma di casi d'uso è uno strumento grafico utilizzato nella modellazione UML (Unified Modeling Language) per rappresentare le interazioni tra gli attori (utenti o sistemi esterni) e un sistema software. Serve a descrivere le funzionalità del sistema dal punto di vista dell'utente e a documentare i requisiti funzionali in modo chiaro e visivo.

5.5) Discord

Piattaforma di comunicazione VoIP e messaggistica istantanea con supporto per server, canali e integrazioni.

5.6) Docker

Docker è una piattaforma che consente di creare, distribuire ed eseguire applicazioni in container. Un container è un ambiente isolato che include tutto il necessario per eseguire un'applicazione, come il codice, le librerie e le dipendenze. Docker facilita il deployment, garantendo che le applicazioni funzionino in modo coerente su diverse piattaforme e infrastrutture.

5.7) Documentation as code

Un approccio alla documentazione che tratta i documenti come codice sorgente, archiviandoli in repository versionati, usando strumenti di automazione per la generazione e manutenzione. Favorisce l'allineamento continuo con il codice.

6) E

7) F

7.1) Feature

Una feature è una caratteristica o una funzionalità specifica di un software o applicazione progettata per soddisfare un bisogno o offrire un vantaggio agli utenti. Una feature può essere un'azione che l'utente può eseguire, un comportamento del sistema o un miglioramento che aggiunge valore al software.

7.2) Feature Branch

Branch di sviluppo dedicato a una nuova funzionalità, creato per isolare le modifiche prima dell'integrazione nel branch principale.

7.3) Feedback

Informazioni o commenti ricevuti su un prodotto, processo o servizio, usati per apportare miglioramenti.

7.4) Framework

Una struttura riutilizzabile e predefinita di codice che fornisce un insieme di strumenti, regole e librerie per sviluppare applicazioni software. I framework riducono il lavoro manuale, offrono standardizzazione e accelerano lo sviluppo.

7.5) Fogli Google

Applicazione web per la creazione e gestione di fogli di calcolo in cloud, parte della suite Google Workspace.

7.6) Fornitura

Processo di consegna di beni o servizi secondo specifiche contrattuali.

8) G

8.1) Git

Sistema di controllo di versione distribuito che consente a più sviluppatori di lavorare su un progetto contemporaneamente, mantenendo un registro delle modifiche apportate al codice.

8.2) GitHub

GitHub è una piattaforma di sviluppo collaborativo basata su Git, un sistema di controllo delle versioni distribuito. È ampiamente utilizzato da sviluppatori di software e team di sviluppo per gestire progetti, tenere traccia delle modifiche al codice sorgente, coordinare il lavoro tra i membri del team e facilitare la collaborazione su larga scala.

8.3) GitHub Organization

Struttura su GitHub che consente la gestione collaborativa di repository e permessi per team di sviluppo.

8.4) Glossario

Elenco strutturato di termini tecnici o specializzati, ognuno corredato dalla propria definizione o spiegazione. Questo strumento aiuta a migliorare la comunicazione tra le varie parti coinvolte in un progetto, riducendo le ambiguità e garantendo una comprensione condivisa dei termini utilizzati in un determinato contesto.

8.5) Gmail

Servizio di posta elettronica di Google basato su cloud.

8.6) Google Calendar

Servizio di gestione e condivisione di eventi e appuntamenti in cloud.

8.7) Google Meet

Piattaforma di videoconferenza di Google, integrata con Google Workspace.

8.8) GroqCloud

Piattaforma AI basata su hardware specializzato (LPU) per inferenza ad alte prestazioni, supporta modelli LLM e integrazione con strumenti AI per elaborazione in tempo reale.

8.9) GUI (Graphical User Interface)

Una GUI (acronimo di Graphical User Interface, in italiano Interfaccia Grafica Utente) è un'interfaccia che consente agli utenti di interagire con un sistema informatico o un software attraverso elementi visivi e grafici, anziché utilizzare solo comandi testuali o linee di comando.

9) H

10) I

10.1) IA

Sistema artificiale (tipicamente un sistema informatico) che simula una generica forma di intelligenza.

10.2) Inspection

Una tecnica formale di revisione del software in cui un team analizza codice o documentazione per identificare errori o incongruenze rispetto agli standard. L'ispezione include solitamente un moderatore, lettori e un protocollo definito.

10.3) Issue

Un GitHub Issue è uno strumento offerto da GitHub per la gestione delle attività e il tracciamento dei problemi all'interno di un progetto software. Ogni issue rappresenta un'unità di lavoro o un problema specifico, e può essere personalizzata tramite assegnatari, che identificano i responsabili della sua risoluzione, label, che ne permettono la categorizzazione e milestone, che raggruppano più issue sotto un obiettivo comune per monitorare il progresso verso una scadenza o un risultato.

10.4) Issue Tracking System

Strumento per registrare, monitorare e gestire problemi, richieste di miglioramento o errori in un progetto software.

11) J

11.1) Jira

Jira è uno strumento software sviluppato da Atlassian, utilizzato per la gestione di progetti, il monito raggio dei problemi e il controllo dello sviluppo agile. È particolarmente popolare tra i team di sviluppo software, specie grazie al servizio di ticketing che offre, ma è anche ampiamente utilizzato in altri ambiti aziendali.

12) K

12.1) Key Performance Indicator (KPI)

I Key Performance Indicator (KPI) sono misure quantitative utilizzate per valutare le prestazioni di un'organizzazione, di un progetto o di un processo rispetto agli obiettivi prefissati. I KPI forniscono un modo oggettivo per monitorare e valutare il successo di un'attività, identificare le aree di miglioramento e prendere decisioni informate per ottimizzare le prestazioni.

13) L

13.1) Label

Tag o etichette usate in GitHub (o altri strumenti di issue tracking) per classificare problemi, richieste di pull o attività secondo criteri specifici.

13.2) LangChain

LangChain è un framework progettato per costruire applicazioni basate su modelli di linguaggio (LLM). Consente di combinare LLM con altre tecnologie come database, API o strumenti di esecuzione per creare flussi di lavoro complessi. È particolarmente utile per la costruzione di chatbot, sistemi di recupero informazioni e strumenti di automazione.

13.3) LaTeX

LaTeX è un linguaggio di markup e un sistema di preparazione di documenti utilizzato principalmente per la produzione di documenti scientifici, accademici e tecnici, ma anche per altri tipi di pubblicazioni come libri, tesi, articoli e presentazioni. È basato su TeX, un sistema di tipografia creato da Donald Knuth negli anni '70, ma LaTeX aggiunge una serie di funzionalità e pacchetti per semplificare la scrittura e la formattazione dei documenti.

13.4) LLM

Un LLM (large language model) è un modello di intelligenza artificiale basato su reti neurali di grandi dimensioni, addestrato su enormi quantità di testo per comprendere e generare linguaggio naturale. Esempi popolari includono GPT (Generative Pre-trained Transformer) e BERT. Questi modelli sono usati in applicazioni come chatbot, traduzione automatica, completamento del testo e analisi del sentiment.

14) M

14.1) Manuale Utente

Documento che spiega agli utenti finali come utilizzare il software, spesso includendo istruzioni dettagliate e risoluzioni comuni ai problemi.

14.2) Merge

Operazione di unione di due branch in un repository Git, combinando le modifiche apportate in entrambi i branch in uno solo.

14.3) Milestone

Le Milestone sono strumenti utilizzati nella gestione dei progetti per segnare punti specifici lungo la timeline di un progetto. Questi punti possono segnalare ancore come una data di inizio e di fine del progetto, o la necessità di una revisione. In molti casi, le milestone, non incidono sulla durata del progetto ma si concentrano invece sui principali punti di avanzamento che devono essere raggiunti per ottenere il successo.

14.4) Minimum Viable Product

È una versione semplificata di un prodotto software che include solo le funzionalità essenziali per soddisfare i bisogni dei primi utenti. L'obiettivo principale di un MVP è testare e validare l'idea di base del prodotto, raccogliere feedback dagli utenti e ridurre al minimo il tempo e le risorse necessarie per il suo sviluppo.

14.5) Modello a V

Un modello di sviluppo software che rappresenta visivamente il ciclo di vita di un progetto, enfatizzando la relazione tra le fasi di sviluppo e quelle di testing. La «V» simboleggia la verifica durante lo sviluppo e la validazione nella fase di testing.

15) N

15.1) Norme di Progetto

Regole e linee guida stabilite all'interno di un progetto per garantire coerenza e qualità nelle attività svolte. Definiscono standard e procedure, come documentazione, gestione delle versioni e criteri di codifica, per assicurare uniformità nell'approccio e nel risultato finale.

16) O

16.1) Onboarding

Processo per integrare nuovi membri in un team o progetto, fornendo loro le informazioni e le risorse necessarie per iniziare a lavorare in modo efficace, efficiente, e in linea con le pratiche e gli standard dell'azienda.

17) P

17.1) PascalCase

Uno stile di scrittura di identificatori in cui ogni parola inizia con una lettera maiuscola e non ci sono spazi o separatori. Esempio: `VariableName`.

17.2) Piano Di Progetto

Documento formale che delinea in dettaglio la pianificazione, l'esecuzione, il monitoraggio e il controllo di tutte le attività coinvolte nella realizzazione di un progetto. Questo documento fornisce una roadmap chiara e organizzata, comprensiva di obiettivi, risorse, scadenze e strategie di gestione dei rischi. Essenziale per la gestione efficace di un progetto, il piano di progetto serve come guida per il team di lavoro e gli stakeholder, fornendo una struttura che facilita il coordinamento delle attività e l'assegnazione delle risorse.

17.3) Piano Di Qualifica

Documento che stabilisce gli standard di qualità, i processi e le attività di testing che saranno implementati durante lo sviluppo di un progetto. Contiene una descrizione dettagliata delle strategie di testing, delle metriche di valutazione e dei criteri di accettazione del prodotto finale. L'obiettivo principale del Piano di Qualifica è garantire che il prodotto soddisfi gli standard di qualità prefissati e che il processo di sviluppo segua procedure coerenti ed efficaci.

17.4) PostgreSQL

Postgres è un sistema di gestione di database relazionale open-source, noto per la sua affidabilità, flessibilità e conformità agli standard SQL. Supporta funzionalità avanzate come transazioni complesse, estensioni personalizzabili, gestione di grandi volumi di dati e supporto per tipi di dati avanzati. È largamente utilizzato in applicazioni web, analisi dei dati e gestione delle informazioni.

17.5) Processo

Un processo è un insieme di attività correlate e coordinate che trasformano input in output, seguendo regole e procedure.

17.6) Product Backlog

Consiste in un elenco prioritizzato di tutte le funzionalità, i requisiti, le correzioni di bug e le modifiche che devono essere fatte a un prodotto software. Il Product Backlog è dinamico e può essere aggiornato in qualsiasi momento per riflettere le esigenze e le priorità del progetto. È gestito dal Product Owner e utilizzato dal team di sviluppo per pianificare e organizzare il lavoro da svolgere.

17.7) Product Baseline

Fase di progetto durante la quale il focus si sposta dall'analisi dei requisiti e dalla progettazione concettuale all'implementazione e alla costruzione effettiva del prodotto software. Le attività principali comprendono la scrittura del codice, i test unitari e l'integrazione di componenti software. L'obiettivo è raggiungere una versione stabile e funzionante del prodotto che rappresenti la base per ulteriori sviluppi e miglioramenti.

17.8) Product Owner

Un product owner supervisiona lo sviluppo di un prodotto software. È il membro di un team di sviluppo Scrum che mantiene la visione di un progetto di sviluppo secondo la metodologia Agile. Le sue responsabilità comprendono il mantenimento di un backlog del prodotto, ovvero un elenco prioritario di opzioni di funzionalità.

17.9) Progettista

Il progettista è la figura di riferimento per quanto riguarda le scelte progettuali partendo dal lavoro dell'analista. Spetta al progettista assumere decisioni di natura tecnica e tecnologica, oltre a supervisionare il processo di sviluppo. Tuttavia, non è responsabile della manutenzione del prodotto. In particolare si occupa di: progettare l'architettura del prodotto secondo specifiche tecniche dettagliate; prendere decisioni per sviluppare soluzioni che soddisfino i criteri di affidabilità, efficienza, sostenibilità e conformità ai requisiti; redige la Specifica Architetturale e la parte pragmatica del Piano di Qualifica.

17.10) Project

Iniziativa temporanea con obiettivi specifici, risorse definite e una data di completamento.

17.11) Project Board

Strumento visivo per la gestione del flusso di lavoro e il monitoraggio dell'avanzamento di un progetto.

17.12) Project Manager

Figura responsabile della pianificazione, esecuzione e monitoraggio di un progetto, garantendone il successo in termini di obiettivi, tempi e budget.

17.13) Proponente

Il proponente è la persona o l'organizzazione che presenta un capitolato d'appalto per la realizzazione di un progetto software. Il proponente definisce i requisiti e le specifiche del progetto e valuta le proposte dei fornitori o degli sviluppatori interessati a partecipare all'appalto.

17.14) Programmatore

Il programmatore è la figura professionale incaricata della scrittura del codice software. Il suo compito primario è implementare il codice conformemente alle specifiche fornite dall'analista e all'architettura definita dal progettista. In particolare, il programmatore: scrive codice manutenibile in conformità con le Specifiche Tecniche; codifica le varie componenti dell'architettura seguendo quanto ideato dai progettisti; realizza gli strumenti per verificare e validare il codice; redige il Manuale Utente.

17.15) Proof Of Concept

Dimostrazione pratica per verificare la fattibilità o la validità di un'idea, di un concetto o di un progetto specifico. Nel contesto dello sviluppo software, un PoC consiste nella creazione di una versione semplificata di un'applicazione per testare un nuovo approccio tecnologico o per dimostrare la fattibilità di una funzionalità specifica.

17.16) Pull Requests

Una Pull Request (PR) è una richiesta che un sviluppatore invia per proporre modifiche al codice di un progetto in un repository, tipicamente su piattaforme di versionamento del codice come GitHub, GitLab

o Bitbucket. La PR permette ad altri membri del team o ai manutentori del progetto di revisionare il codice proposto prima che venga integrato nel ramo principale (di solito chiamato main o master).

18) Q

18.1) QDrant

Qdrant è un motore di database vettoriale open-source progettato per memorizzare e cercare rappresentazioni vettoriali di dati (embeddings). Supporta ricerche di similarità ad alta velocità e applicazioni di machine learning, come motori di raccomandazione e ricerca semantica. È spesso utilizzato con LLM per migliorare l'elaborazione dei dati.

19) R

19.1) Repository

Un repository (o repo) è un archivio digitale che contiene file, cartelle e informazioni relative al progetto, inclusi il codice sorgente, la documentazione e la cronologia delle modifiche. È uno spazio organizzato, solitamente gestito tramite un sistema di controllo versione come Git, dove i file sono archiviati e tenuti sotto controllo, permettendo di monitorare le modifiche nel tempo. I repository sono usati per gestire il ciclo di vita di un progetto software, facilitando la collaborazione tra sviluppatori e il versionamento del codice. I repository possono essere locali (sul proprio computer) o remoti (su piattaforme come GitHub, GitLab, Bitbucket).

19.2) Requisiti

Specifiche dettagliate che descrivono ciò che un sistema o un prodotto deve fare (requisiti funzionali) o le qualità che deve avere (requisiti non funzionali). Possono essere raccolti tramite interviste, analisi e documentazione, e sono fondamentali per la progettazione e lo sviluppo.

19.3) Requisiti di Qualità

Criteri relativi alle prestazioni, usabilità, sicurezza e manutenibilità del software.

19.4) Requisiti di Vincolo

Specificano le condizioni e i limiti che il sistema deve rispettare, come vincoli di budget, tecnologici o legali.

19.5) Requisiti Funzionali

Specificano cosa deve fare un sistema software per soddisfare le esigenze dell'utente.

19.6) Responsabile

Figura fondamentale che coordina il gruppo, funge da punto di riferimento per il committente e per il team, svolgendo il ruolo di mediatore tra le due parti. In particolare si occupa di: gestire le relazioni con l'esterno; pianificare le attività (quali svolgere, data di inizio e fine, assegnazione delle priorità), valutare i rischi associati alle decisioni da prendere, controllare i progressi del progetto, gestire le risorse umane e approvare la documentazione.

19.7) RTB

Fase iniziale e fondamentale del processo di sviluppo di un software. In questa fase, l'obiettivo principale è stabilire e comprendere i requisiti del sistema e definire la base tecnologica sulla quale il progetto si svilupperà. Le tre principali attività di questa fase sono: l'analisi dei requisiti, la definizione della baseline tecnologica e la definizione della baseline di progetto.

19.8) Rischi organizzativi

Problemi legati alla gestione, alla pianificazione o alla comunicazione che possono influire sul successo di un progetto.

19.9) Rischi tecnologici

Problemi legati a una o più tecnologie utilizzate, come software obsoleti, bug critici o incompatibilità.

20) S

20.1) Scrum

Scrum è una struttura Agile di collaborazione tra team, comunemente utilizzata nello sviluppo di software e in altri settori. Scrum prescrive ai team di suddividere il lavoro in obiettivi da completare entro iterazioni a tempo, chiamate sprint.

20.2) Scrum master

Il ruolo di uno Scrum Master è quello di utilizzare la gestione agile dei progetti per sostenere un progetto, i team e i membri del team. Poiché gli Scrum Master possono lavorare in diversi contesti, i compiti e le responsabilità possono variare.

20.3) Software

Insieme di programmi, dati e istruzioni che consentono a un computer di eseguire specifiche operazioni.

20.4) Specifica Tecnica

Un documento che descrive dettagliatamente come un sistema o un componente software deve essere costruito. Include requisiti funzionali, architettura, interfacce e altre informazioni necessarie per lo sviluppo.

20.5) Sprint

Gli sprint sono periodi di tempo che vanno da una settimana a un mese, durante i quali il Product Owner, lo Scrum Master e il Development Team lavorano per completare una specifica aggiunta al prodotto. Durante uno sprint, si lavora per creare nuove funzionalità basate sulle user stories e sul backlog.

20.6) Sprint Review

Un evento Scrum che si tiene alla fine di uno sprint, in cui il team presenta gli incrementi di prodotto completati agli stakeholder per ricevere feedback e verificare che siano in linea con gli obiettivi.

20.7) Stato Avanzamento Lavori (SAL)

Nell'ambito del project management, è una riunione periodica (o attività di confronto analoga) che viene stabilita per garantire e verificare l'avanzamento di un progetto rispetto agli obiettivi prestabiliti.

20.8) Stakeholder

Gli stakeholder sono le persone o le organizzazioni coinvolte o interessate a un progetto o a un'organizzazione. Gli stakeholder possono includere clienti, utenti finali, fornitori, dipendenti, investitori, partner commerciali, organizzazioni non governative e altri soggetti che possono influenzare o essere influenzati dalle decisioni e dalle azioni di un'organizzazione.

21) T

21.1) Task

Un task è un'unità di lavoro o attività specifica che deve essere completata come parte di un progetto o di un processo. I task sono generalmente definiti in modo chiaro e conciso, con obiettivi, scadenze e responsabilità ben definiti. Possono essere assegnati a singoli membri del team o a gruppi di lavoro e sono utilizzati per organizzare e monitorare il lavoro da svolgere.

21.2) Telegram

Telegram è un'applicazione di messaggistica istantanea e di comunicazione basata su cloud, sviluppata da Telegram.

21.3) Test di Accettazione

Verifica finale condotta dall'utente o dal cliente per assicurarsi che il prodotto soddisfi le loro aspettative e sia pronto per l'uso.

21.4) Test di Integrazione

Test che verificano l'interazione tra diverse unità o moduli di codice per garantire che lavorino insieme correttamente.

21.5) Test di Sistema

Valutazione dell'intero sistema software per verificare che soddisfi i requisiti specificati.

21.6) Test di Unità

Verifica di singole unità o componenti del codice (come funzioni o metodi) per assicurarsi che funzionino correttamente in isolamento.

21.7) Text-to-Text

Un chatbot text-to-text è un tipo di intelligenza artificiale conversazionale che utilizza il Natural Language Processing (NLP) e algoritmi di machine learning per comprendere e rispondere a input testuali forniti dagli utenti. Questi chatbot sono progettati per simulare conversazioni simili a quelle umane, offrendo risposte utili e informative alle domande degli utenti.

21.8) Ticket

Un ticket è una registrazione formale di un'attività, problema o richiesta all'interno di un sistema di gestione del lavoro o del supporto tecnico. I ticket sono utilizzati per tracciare, monitorare e gestire le richieste di assistenza, i bug, le nuove funzionalità o qualsiasi altra attività che richiede attenzione. Ogni ticket contiene informazioni come la descrizione del problema o della richiesta, lo stato attuale (ad esempio, «aperto», «in corso», «chiuso»), l'assegnazione a una persona o team responsabile, e altre informazioni utili per risolvere la questione. I sistemi di ticketing sono comunemente usati nei progetti di software, nei servizi di supporto e nei flussi di lavoro aziendali.

21.9) Tracciamento

Processo di monitoraggio e registrazione delle attività di sviluppo, modifiche ai requisiti o avanzamento di un progetto.

21.10) Typst

Typst è un linguaggio di markup e un sistema di composizione tipografica moderno e innovativo, progettato per creare documenti di alta qualità, come articoli, tesi, report e libri, con un approccio più semplice e flessibile rispetto a LaTeX. Typst si distingue per la sua sintassi più intuitiva e per la sua capacità di generare automaticamente un design tipografico professionale. È particolarmente utile per gli utenti che cercano un'alternativa moderna a LaTeX, ma senza rinunciare alla qualità tipografica e alla potenza nella gestione dei documenti complessi. Typst supporta anche l'integrazione di elementi grafici, come immagini e tabelle, e offre un'esperienza di compilazione veloce e fluida.

22) U

22.1) UML

Un linguaggio di modellazione standardizzato utilizzato per visualizzare, specificare, costruire e documentare i sistemi software. Include diagrammi strutturali (es. diagrammi di classi) e comportamentali (es. diagrammi di sequenza).

22.2) User

Entità (persona o sistema) che interagisce con un software o servizio digitale.

22.3) User Experience (UX)

Insieme delle percezioni e risposte di un utente durante l'interazione con un sistema o prodotto software.

22.4) User Story

Una user story è una descrizione informale, in linguaggio naturale, delle caratteristiche di un sistema software.

23) V

23.1) Validazione

La validazione è il processo volto ad assicurare che il prodotto software, modulo o sistema soddisfi le necessità e le aspettative dell'utente finale, indipendentemente dalle specifiche tecniche. Si concentra, quindi, sul garantire che il sistema svolga ciò per cui è stato progettato.

23.2) Verifica

La verifica è il processo volto ad accertare che un prodotto software, un modulo, un sistema o un processo soddisfi i requisiti specificati o i criteri di progettazione. Si concentra sul fatto che il sistema sia stato costruito correttamente, ovvero in conformità alle specifiche tecniche o funzionali.

23.3) Verificatore

La principale responsabilità del verificatore consiste nell'ispezionare il lavoro svolto da altri membri del team per assicurare la qualità del prodotto e la conformità alle attese prefissate. Stabilisce se il lavoro è stato svolto correttamente sulla base delle proprie competenze tecniche, esperienza e conoscenza delle norme. In particolare il verificatore si occupa di: verificare che il lavoro svolto sia conforme alle Norme di Progetto; verificare che il lavoro svolto sia conforme alle Specifiche Tecniche; ricercare ed in caso segnalare eventuali errori; redigere la sezione retrospettiva del Piano di Qualifica, descrivendo le verifiche e le prove effettuate durante il processo di sviluppo del prodotto.

24) W

24.1) Walkthrough

Una revisione informale di codice, documentazione o requisiti, guidata dall'autore, con lo scopo di ottenere feedback iniziale. I walkthrough sono meno strutturati rispetto alle inspection.

24.2) Way of Working (WoW)

Il Way of Working (WoW) si riferisce al modo in cui un team, un'organizzazione o un'azienda svolge le proprie attività quotidiane, gestisce i flussi di lavoro e raggiunge i propri obiettivi. Include le pratiche, i processi, le metodologie, le tecniche e le convenzioni adottate per eseguire il lavoro in modo efficiente e coerente. Il WoW può riguardare aspetti come la comunicazione, la collaborazione, la gestione del tempo, l'uso di strumenti e tecnologie, nonché le modalità di decision-making e il coinvolgimento delle persone. Un WoW ben definito aiuta a garantire l'efficacia operativa, la qualità del lavoro e la soddisfazione del team, adattandosi alle esigenze e alla cultura aziendale.

24.3) Workflow

Un workflow è un insieme strutturato di attività o processi interconnessi che vengono eseguiti in sequenza o in parallelo per raggiungere un obiettivo specifico. In sostanza, rappresenta il flusso di lavoro, ossia come le informazioni, i compiti e le risorse vengono gestiti all'interno di un sistema o di un'organizzazione. Un workflow può essere manuale o automatizzato e viene utilizzato per ottimizzare, monitorare e controllare i processi aziendali, facilitando la cooperazione tra diversi attori o strumenti. I workflow sono fondamentali per la gestione di progetti, l'automazione dei processi aziendali e per garantire l'efficienza e la coerenza nelle operazioni quotidiane.

24.4) Workshop

Un evento collaborativo in cui un gruppo di persone si riunisce per affrontare e risolvere problemi specifici, acquisire conoscenze o creare risultati condivisi. Spesso utilizzato nella raccolta dei requisiti o nel brainstorming.

25) X

26) Y

27) Z