



Glossario

2024-11-18

V0.1

sweetenteam@gmail.com

<https://sweetenteam.github.io>



Destinatari	Prof. Tullio Vardanega Prof. Riccardo Cardin
Redattori	Orlando Ferazzani
Verificatori	

Registro delle modifiche

Versione	Data	Autori	Verificatori	Dettaglio
0.1	2024-11-18	Orlando Ferazzani		Prima stesura del glossario

Indice

1) Introduzione	5
2) A	6
2.1) Agile	6
2.2) Amministratore	6
2.3) Analisi Dei Requisiti	6
2.4) Analista	6
3) B	7
3.1) Branch	7
4) C	8
4.1) Capitolato	8
4.2) Committente	8
4.3) Confluence	8
4.4) Cruscotto	8
5) D	9
5.1) Dev team	9
5.2) Diagramma dei casi d'uso	9
6) E	10
7) F	11
8) G	12
8.1) GitHub	12
8.2) Glossario	12
8.3) GUI (Graphical User Interface)	12
9) H	13
10) I	14
11) J	15
11.1) Jira	15
12) K	16
12.1) Key Performance Indicator (KPI)	16
13) L	17
13.1) LaTeX	17
14) M	18
14.1) Milestone	18
14.2) Minimum Viable Product (MVP)	18
15) N	19
15.1) Norme di Progetto	19
16) O	20
17) P	21
17.1) Piano Di Progetto	21
17.2) Piano Di Qualifica	21
17.3) Product Backlog	21
17.4) Product Baseline (PB)	21
17.5) Product Owner	21
17.6) Progettista	21
17.7) Proponente	21
17.8) Programmatore	22
17.9) Proof Of Concept (PoC)	22
17.10) Pull Requests	22

18) Q	23
19) R	24
19.1) Repository	24
19.2) RTB (Requirements and Technology Baseline)	24
19.3) Responsabile	24
20) S	25
20.1) Stato Avanzamento Lavori (SAL)	25
20.2) Scrum	25
20.3) Scrum master	25
20.4) Sprint	25
20.5) Stakeholder	25
21) T	26
21.1) Task	26
21.2) Ticket	26
21.3) Typst	26
22) U	27
22.1) User Story	27
23) V	28
23.1) Verificatore	28
24) W	29
24.1) Way of Working (WoW)	29
24.2) Workflow	29
25) X	30
26) Y	31
27) Z	32

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'ambiente 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 è una fase fondamentale nello sviluppo del software, che coinvolge la raccolta, l'analisi, la documentazione e la comprensione approfondita delle esigenze e delle specifiche di un sistema o di un'applicazione software che si intende sviluppare (oltre ad essere un documento all'interno del nostro progetto)

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

3) **B**

3.1) **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) Capitolato

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

4.2) 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

4.3) 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.

4.4) 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 informate sulle azioni da intraprendere per migliorare la qualità del software

5) D

5.1) 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

5.2) 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.

6) E

7) F

8) G

8.1) 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

8.2) 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

8.3) 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

11) J

11.1) Jira

Jira è uno strumento software sviluppato da Atlassian, utilizzato per la gestione di progetti, il monitoraggio dei problemi e il controllo dello sviluppo agile. È particolarmente popolare tra i team di sviluppo software, 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) **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 comandi e pacchetti che ne semplificano l’utilizzo.

14) M

14.1) Milestone

Le Milestone sono strumenti utilizzati nella gestione dei progetti per segnare punti specifici lungo la timeline di un progetto. Questi punti possono segnalare ancora 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. Si concentrano invece sui principali punti di avanzamento che devono essere raggiunti per ottenere il successo

14.2) Minimum Viable Product (MVP)

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

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) Q

17) P

17.1) 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.

17.2) 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.

17.3) 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.

17.4) Product Baseline (PB)

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.

17.5) 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 loro responsabilità comprendono il mantenimento di un backlog del prodotto, ovvero un elenco prioritario di opzioni di funzionalità.

17.6) 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 Architettureale e la parte pragmatica del Piano di Qualifica.

17.7) 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.

17.8) 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.

17.9) Proof Of Concept (PoC)

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.

17.10) 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

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) **RTB (Requirements and Technology Baseline)**

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.

19.3) **Responsabile**

Figura fondamentale che coordinare 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 delle scelte da effettuare; controllare i progressi del progetto; gestire le risorse umane; approvare la documentazione.

20) S

20.1) 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 ai propri obiettivi

20.2) 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

20.3) 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

20.4) 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

20.5) 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) 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.

21.3) 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) User Story

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

23) V

23.1) **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) 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.

24.2) 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.

25) X

26) Y

27) Z