



7zpus.swe@gmail.com

Glossario

Indice

A	3
B	3
C	4
D	5
E	5
F	5
G	6
H	6
I	6
J	7
K	7
L	7
M	7
N	8
O	8
P	8

Q	9
R	9
S	10
T	11
U	11
V	12
W	12
X	12
Y	12
Z	12

Tabella di Versionamento

Versione	Data	Autore	Verificatore	Descrizione
1.0	31/10/2025	Vigolo Davide	Gingilino Aaron	Stesura iniziale

A

- **Action:** Una GitHub Action è una componente eseguibile (definita tramite metadati action.yml) che implementa una singola funzione di automazione e che può essere richiamata in un workflow GitHub Actions per automatizzare operazioni come unit testing.
- **Agile:** Approccio alla gestione dei progetti basato su uno sviluppo iterativo e incrementale. È basato su valori e principi che favoriscono la flessibilità e la risposta alle esigenze degli utenti, preferendo cicli di lavoro brevi (sprint) a una pianificazione rigida e lineare.
- **Analista:** Ruolo che si occupa della scrittura dell'analisi dei requisiti e del piano di qualifica.
- **Analisi dei Requisiti:** Fase del processo di sviluppo software in cui vengono raccolti, documentati e analizzati i requisiti, funzionali e non, del sistema da sviluppare. Questa fase è fondamentale per comprendere le esigenze degli utenti e tradurle in un linguaggio tecnico non ambiguo, che permetta e favorisca lo sviluppo successivo del design del sistema software.
- **Amministratore:** Ruolo che si occupa del corretto svolgimento del progetto e la scrittura delle norme di progetto.
- **As Is:** Tecnica per l'Analisi dei Requisiti. Prevede di immaginare il mondo senza la risorsa, con le sue conseguenti limitazioni.
- **Attività di Progetto:** Azioni che portano allo sviluppo del progetto. Componenti dei Processi di progetto.
- **Attore Principale:** Colui che attivamente interagisce con il sistema.
- **Attore Secondario:** Interagisce in modo passivo, viene invocato dal sistema in aiuto per offrire una funzionalità.

B

- **Baseline:** Prodotto che soddisfa i vincoli di una Milestone, non solo per Efficacia ma anche per Efficienza. Una baseline è una versione specifica di un prodotto software che è stata formalmente revisionata e approvata, servendo come punto di riferimento per

lo sviluppo futuro. Le baseline sono utilizzate per tracciare le modifiche e garantire la coerenza del progetto nel tempo.

- **Branch:** Un ramo in Git è una versione parallela del codice sorgente che consente di lavorare su funzionalità o correzioni senza influenzare il ramo principale (di solito chiamato "main" o "master"). I branch possono essere uniti (merged) per incorporare le modifiche nel ramo principale.
- **Brainstorming:** Attività che aiuta a estrarre i requisiti. Necessita di un metodo, una struttura organizzata e quindi è una Cerimonia. È quindi una discussione tra pari che, a partire da una agenda chiara e vigilata da un Maestro di Cerimonia, permette la libera espressione delle idee. Ogni membro però ha un tempo limitato per esprimere le proprie idee. È inoltre necessaria la figura di colui che colleziona i punti salienti che emergono dalla discussione. Risultano utili le Mappe Mentali per schematizzare in modo più comodo i punti di un BrainStorming.
- **Board:** Strumento visivo utilizzato per organizzare e monitorare le attività di un progetto. Le board possono essere suddivise in colonne che rappresentano le diverse fasi del processo di sviluppo, come "Da fare", "In corso" e "Fatto".
- **Build:** Processo di compilazione e assemblaggio del codice sorgente in un eseguibile o in un pacchetto distribuibile.

C

- **Caso d'Uso:** Un caso d'uso è una descrizione formale di un insieme di scenari in cui un attore interagisce con un sistema per ottenere un risultato. Serve per modellare le funzionalità del sistema e definire le interazioni tra utenti e sistema.
- **Classificazione dei Requisiti:** Organizzazione dei requisiti a "cassetti". Un esempio è la classificazione ad albero, distinguendo tra requisiti di Prodotto (System Requirement), sui Processi (Process Requirement) e sul Contratto (Project Requirement).
- **Commit:** Un'operazione in Git che salva le modifiche apportate ai file nel repository locale. Ogni commit rappresenta un'istantanea del progetto in un determinato momento e include un messaggio descrittivo delle modifiche effettuate.
- **Committente:** Soggetto (azienda, ente o privato) che finanzia un progetto e ne definisce i requisiti, le aspettative e gli obiettivi. Il committente è responsabile di fornire le risorse necessarie per lo sviluppo del progetto e di approvare le fasi chiave del processo di sviluppo software.
- **Consuntivo di Progetto:** Elenco quantitativo di ciò che è stato fatto nel periodo.

- **Cruscotto di qualità:** Dashboard che si specializza su tutte quelle metriche che riguardano la qualità del software, come la copertura del codice, la complessità, la presenza di bug o vulnerabilità, e altri indicatori di qualità. Per Dashboard più generiche, si veda la voce "Dashboard/Cruscotto".

D

- **DashBoard/Cruscotto:** Strumento visivo che fornisce una panoramica delle metriche chiave, dei progressi e dello stato del progetto. Le dashboard sono utilizzate per monitorare le attività, identificare eventuali problemi e facilitare la comunicazione tra i membri del team e gli stakeholder.
- **DIPReader:** è il software commissionato dalla proponente su cui si concentra il progetto

E

- **Efficacia:** Raggiungere gli obiettivi prefissati.
- **Efficienza:** Raggiungere gli obiettivi consumando solo le risorse previste per tali obiettivi.
- **Epic:** Un'epic è una grande unità di lavoro che può essere suddivisa in più user story o task più piccoli. Gli epic rappresentano funzionalità o obiettivi significativi all'interno di un progetto di sviluppo software e aiutano a organizzare e pianificare il lavoro in modo più efficiente.
- **Estensione Jira per VSCode:** Un'estensione per Visual Studio Code che consente agli sviluppatori di integrare Jira direttamente nell'ambiente di sviluppo.

F

- **Feature:** Una funzionalità o caratteristica specifica di un sistema software che fornisce valore agli utenti finali. Le feature possono essere descritte in termini di requisiti funzionali e sono spesso suddivise in task più piccoli.
- **Feedback:** Informazioni fornite dagli utenti o dai membri del team che aiutano a migliorare il prodotto o il processo di sviluppo. Il feedback è essenziale per identificare problemi, apportare miglioramenti e garantire che il prodotto soddisfi le esigenze degli utenti.
- **Framework:** Un insieme di strumenti, librerie e best practice che forniscono una struttura di base per lo sviluppo di software. I framework aiutano a semplificare il processo di sviluppo, fornendo soluzioni predefinite per problemi comuni e facilitando la gestione del codice.

- **Fornitore:** Entità che si occupa di fornire il prodotto software alla proponente, seguendo le specifiche e i requisiti definiti. Il fornitore è responsabile dello sviluppo, della consegna e del supporto del prodotto software, lavorando in collaborazione con la proponente per garantire il successo del progetto.

G

- **Git:** Sistema di controllo di versione distribuito, utilizzato per tracciare le modifiche nel codice sorgente durante lo sviluppo del software. Permette a più sviluppatori di collaborare su un progetto in modo efficiente.
- **GitHub:** Piattaforma di hosting per il controllo di versione e la collaborazione che utilizza Git. Consente agli sviluppatori di ospitare i loro repository Git online, facilitando la condivisione del codice, la gestione delle versioni e la collaborazione tra team.
- **Good Practice:** Una buona pratica è una metodologia o tecnica riconosciuta come efficace e raccomandata, basata sull'esperienza e sui risultati ottenuti in progetti precedenti.

H

I

- **Indicizzazione:** Tecnica cruciale per ottimizzare le prestazioni, creando strutture dati (indici) che mappano i termini di ricerca ai dati effettivi, riducendo la necessità di scansioni complete della tabella.
- **Ingegneria del Software:** Disciplina che si occupa di progettare, sviluppare, testare e mantenere software in modo sistematico, disciplinato e quantificabile. L'ingegneria del software mira a garantire la qualità, l'affidabilità e la manutenibilità del software attraverso l'applicazione di principi, metodi e strumenti specifici.
- **Issue:** Un problema, bug o richiesta di funzionalità presente e tracciabile in un sistema di Issue Tracking. Un commit può essere associato a una o più issue.
- **Issue Tracking System (ITS):** Un sistema software utilizzato per gestire e tracciare i problemi, i bug e le richieste di funzionalità durante lo sviluppo del software. Gli ITS aiutano i team di sviluppo a organizzare, assegnare e monitorare lo stato delle issue, migliorando la comunicazione e la collaborazione tra i membri del team.

J

- **Jira:** Strumento di gestione di progetto sviluppato da Atlassian, utilizzato principalmente come Issue Tracking System (ITS), la gestione dei progetti Agile e la pianificazione delle attività di sviluppo software.
- **JQL:** (Jira Query Language) è un linguaggio di query utilizzato in Jira per cercare e filtrare le issue in base a criteri specifici.

K

L

- **LaTeX:** Un sistema di composizione tipografica utilizzato principalmente per la creazione di documenti scientifici e tecnici. LaTeX consente agli utenti di concentrarsi sul contenuto del documento, mentre si occupa della formattazione e dell'impaginazione in modo professionale.
- **LLM:** (Large Language Model) Modello di intelligenza artificiale progettato per comprendere e generare testo in modo simile a come lo farebbe un essere umano. Gli LLM sono addestrati su grandi quantità di dati testuali e utilizzano tecniche di apprendimento automatico per acquisire conoscenze linguistiche e contestuali, consentendo loro di rispondere a domande, completare frasi e generare testo coerente.

M

- **Milestone:** Punti nel tempo e nelle risorse che traccia l'avanzamento del progetto, posizionate a partire dagli obiettivi finali, incrementali ma per piccoli passi, quantitativi e misurabili. Un punto di riferimento significativo in un progetto di sviluppo software che rappresenta il completamento di una fase importante o il raggiungimento di un obiettivo specifico. I milestone aiutano a monitorare i progressi del progetto e a pianificare le attività future.
- **Merge:** L'operazione di unione di due rami (branch) in Git. Quando si esegue un merge, le modifiche apportate in un ramo vengono integrate nel ramo di destinazione, combinando il lavoro di più sviluppatori.

N

O

- **Ore produttive:** Ore di lavoro effettivamente dedicate allo sviluppo del progetto, escludendo le ore non produttive come riunioni, attività amministrative o di studio. Le ore produttive sono un indicatore importante per valutare l'efficienza del team di sviluppo e la progressione del progetto.

P

- **Processo:** Un insieme di attività correlate e coordinate che trasformano input in output, con l'obiettivo di raggiungere un risultato specifico. I processi governano le attività di progetto, organizzando e strutturando il lavoro in modo efficiente.
- **Product Baseline (PB):** un'istantanea (snapshot) del progetto al termine delle attività di pianificazione con il progetto approvato dal committente.
- **Programmatore:** Ruolo che si occupa di codificare il progetto e i rispettivi test.
- **Progettista:** Ruolo che si occupa di creare un progetto basandosi sullo studio dei requisiti degli analisti.
- **Proof of Concept (PoC):** Prodotto software che dimostra la fattibilità di un'idea o di una tecnologia specifica. Il PoC è spesso utilizzato per testare concetti, validare ipotesi o dimostrare la funzionalità di una soluzione prima di investire risorse significative nello sviluppo completo del prodotto.
- **Piano di Qualifica:** Organizzazione di attività di V&V durante lo sviluppo, periodico per arrivare alla Validazione finale garantendo la Validità di tutto il lavoro (Lavorare sempre nel modo giusto per essere nel giusto).
- **Pipeline CI/CD:** Una pipeline CI/CD è un insieme di processi automatizzati che consentono di integrare, testare e distribuire il software in modo continuo.
- **PostCondizioni:** Stato in cui si deve trovare il sistema dopo che l'attore ha utilizzato una funzionalità.
- **PreCondizioni:** Stato in cui si deve trovare il sistema per permettere all'attore di compiere un'azione.
- **PropONENTE:** Soggetto (azienda, ente o privato) che presenta un capitolato d'appalto e manifesta l'esigenza di un prodotto software.

- **Pull request (PR)**: Una richiesta di pull è una funzionalità di GitHub che consente agli sviluppatori di proporre modifiche al codice sorgente di un progetto. Quando un sviluppatore crea una pull request, sta chiedendo al team di revisione del progetto di esaminare le modifiche apportate e, se approvate, di unirle (merging) nel ramo principale del repository.
- **Push**: Un'operazione in Git che invia le modifiche locali al repository remoto. Dopo aver effettuato un commit, è necessario eseguire un push per condividere le modifiche con altri membri del team e aggiornare il repository remoto.

Q

- **Quality Gates**: Punti di controllo che definiscono criteri specifici per valutare la qualità del software durante il processo di sviluppo. I quality gates possono includere metriche come la copertura del codice, la complessità, la presenza di bug o vulnerabilità, e altri indicatori di qualità. Superare un quality gate è spesso necessario per procedere con il rilascio o l'integrazione del software.

R

- **Repository**: Un repository è un archivio digitale che contiene il codice sorgente, la documentazione e altri file correlati a un progetto software. I repository sono utilizzati per gestire le versioni del codice, facilitare la collaborazione tra sviluppatori e tracciare le modifiche nel tempo.
- **Requisiti**: Bisogni, Needs di un progetto.
- **Requisiti Obbligatori**: Grado di Urgenza di un Requisito che lo etichetta. Irrinunciabili per i committenti. Conviene che sia un numero limitato per evitare di rimanere vincolati.
- **Requisiti Desiderabili**: Grado di Urgenza di un Requisito che lo etichetta. Non necessari ma con valore aggiunto che può essere avanzato a Obbligatorio se possibile.
- **Requisiti Opzionali**: Grado di Urgenza di un Requisito che lo etichetta. Relativamente utili e contrattabili più avanti nel progetto.
- **Retrospettiva**: Analisi di ciò che è stato fatto in precedenza con maggiore esperienza e consapevolezza in modo da ottimizzare, crescere in comprensione oltre che nella crescita del prodotto. È ciò che chiude un periodo e prepara al successivo, con l'Ideazione di Misure Correttive e la pianificazione (Consultivo a Finire).
- **Responsabile di progetto**: Ruolo che si occupa di gestire la suddivisione delle attività tra i membri, della scrittura dei verbali, della scrittura del piano di progetto e della comunicazione con la proponente.

- **Requirements and Technology Baseline (RTB)**: Converge l'Analisi dei Requisiti con la risposta Tecnica. Una volta compresi i vincoli tecnologici emerge la necessità che le tecnologie rispecchiano tali vincoli, pesando la scelta in base all'impatto che tali tecnologie hanno sul design (1° Compito della RTB). L'uso di più tecnologie pretende anche che queste eterogenicità possano convivere in un Proof Of Concept (2° Compito della RTB).

S

- **Scenario Alternativo**: Insieme di passi che descrive un Path che porta a PostCondizioni diverse da quelle dello Scenario Principale.
- **Scenario Principale**: L'insieme di passi che descrive l'Happy Path di una funzione (passi per arrivare al risultato).
- **Schedulazione**: Organizzazione temporale delle attività di progetto, che tiene conto delle dipendenze tra le attività, delle risorse disponibili e dei vincoli di tempo. La schedulazione è fondamentale per garantire che il progetto venga completato entro i tempi previsti e con le risorse assegnate.
- **SCIs**: (Software Configuration Items) Componenti software che sono soggetti a controllo di versione e gestione delle configurazioni. Gli SCIs possono includere codice sorgente, documentazione, script di build e altri artefatti correlati al progetto software.
- **Scrum**: Un framework di sviluppo software Agile che si basa su iterazioni brevi e incrementali chiamate sprint. Scrum enfatizza la collaborazione, la trasparenza e l'adattabilità, con ruoli specifici.
- **Slack Time**: Margine tra due attività (sempre maggiore di zero) che attutisce l'impatto di ritardi, in modo da non propagare il ritardo sulle attività successive. Concetto alla base dei diagrammi PERT. Per la valutazione di questi Slack, è necessaria una Gestione dei Rischi accurata formata da Identificazione, Analisi, Pianificazione, Controllo.
- **Smart-commit**: Un commit formattato in modo specifico che include un riferimento a una issue Jira, lo stato della issue, il tempo per completarla e un messaggio descrittivo. Gli smart-commit sono fondamentali per automatizzare il tracciamento delle issue e migliorare la gestione del progetto in Jira.
- **Sprint**: Un periodo di tempo definito (di solito da 1 a 4 settimane) durante il quale un team di sviluppo software lavora per completare un insieme specifico di attività. Gli sprint sono una componente chiave della metodologia Agile, in cui il lavoro viene suddiviso in iterazioni brevi e focalizzate per facilitare la pianificazione, l'esecuzione e la revisione del lavoro.

- **Stato di Avanzamento dei Lavori (SAL):** Controllo dei progressi effettuati in un determinato frangente di tempo. La progressione dei lavori viene comunicata in riunioni periodiche tra il gruppo e la proponente.
- **Stakeholder:** Una persona, gruppo o organizzazione che ha un interesse diretto o indiretto in un progetto o in un sistema software. Gli stakeholder possono includere clienti, utenti finali, sviluppatori, manager e altre parti interessate che influenzano o sono influenzate dal progetto.
- **Subtask:** Un'unità di lavoro più piccola e specifica all'interno di un task più grande. I subtask aiutano a suddividere il lavoro in parti gestibili e facilitano la pianificazione e l'assegnazione delle attività.
- **System Requirement:** Possiamo suddividere i requisiti di sistema: - Constraint (Vincoli legali, fisici, culturale, di design, implementativi, ecc) - Functional Requirement (Funzioni da svolgere del prodotto) - Attribute (Performance Requirement e Specific quality Requirement, aka *-lities*). Questa suddivisione ci permette più agilmente di trovare questi requisiti.

T

- **Task:** Un'unità di lavoro specifica e ben definita all'interno di un progetto di sviluppo software. I task rappresentano attività piccole e gestibili che contribuiscono al completamento di funzionalità o obiettivi più ampi.
- **Time estimate:** Stima del tempo necessario per completare un task o un'attività specifica. Le stime temporali sono essenziali per la pianificazione e la gestione del progetto, aiutando a allocare risorse e a monitorare i progressi.
- **To Be:** Tecnica per l'Analisi dei Requisiti. Prevede di immaginare il mondo con la risorsa, valutando i miglioramenti e le nuove prospettive che si aprono.
- **Tracciabilità:** Serve che i requisiti abbiano un Identificativo unico, che debba anche renderli modificabili in modo semplice, per far fronte a possibili aggiunte e/o cancellazioni.

U

- **Use Case:** Vedi **Caso d'Uso**.
- **User Story:** Tecnica per l'Analisi dei Requisiti che è più immaginativa. È una descrizione breve e semplice di una funzionalità del sistema desiderata dall'utente. La User Story può essere anche interna al gruppo di lavoro.

V

- **V-Model:** Modello di sviluppo software che rappresenta le fasi del processo di sviluppo in forma di "V". La parte sinistra della "V" rappresenta le fasi di pianificazione e progettazione, mentre la parte destra rappresenta le fasi di test e validazione. Il V-Model enfatizza l'importanza della verifica e della validazione in ogni fase del processo di sviluppo.
- **Validazione:** Controllo della soddisfazione delle aspettative (Finale).
- **Verifica:** Controllo che lo sviluppo non introduca errori.
- **VSCode:** (Visual Studio Code) Un editor di codice sorgente sviluppato da Microsoft, noto per la sua leggerezza, velocità e supporto a una vasta gamma di linguaggi di programmazione.

W

- **Way of Working:** Il come si deve lavorare, gli standard, gli obiettivi di qualità garantiti da strumenti ad hoc.
- **Working Copy:** Una copia locale di un repository Git che contiene tutti i file e le cartelle del progetto. Gli sviluppatori lavorano sulla working copy per apportare modifiche prima di inviarle al repository remoto.

X

Y

Z