



7zpus.swe@gmail.com

Glossario

Indice

1	A	3
2	B	3
3	C	4
4	D	4
5	E	4
6	F	5
7	G	5
8	H	5
9	I	5
10	J	6
11	K	6
12	L	6
13	M	6
14	N	6
15	O	6
16	P	6

17 Q	7
18 R	7
19 S	8
20 T	9
21 U	9
22 V	9
23 W	9
24 X	10
25 Y	10
26 Z	10

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.
- **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.
- **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.

C

- **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:** È lo stakeholder che acquisisce o procura il prodotto o servizio dal fornitore, finanzia il progetto e ha l'ultima parola sulla validazione (collaudo) per accettare il prodotto finale.
- **Consuntivo di Progetto:** Elenco quantitativo di ciò che è stato fatto nel periodo.

D

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.

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.

G

- **Garanzia della qualità:** Insieme di attività pianificate e sistematiche implementate all'interno di un sistema di qualità per fornire fiducia che un prodotto o servizio soddisferà i requisiti di qualità specificati. La garanzia della qualità si concentra sulla prevenzione dei difetti attraverso processi ben definiti, standard e procedure, piuttosto che sulla rilevazione dei difetti dopo che si sono verificati.
- **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**

- **Ingegneria del Software:** L'Ingegneria del Software è l'applicazione di un approccio sistematico, disciplinato e quantificabile allo sviluppo, all'uso, alla manutenzione e al ritiro del software.
- **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.

K**L****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.
- **Metriche:** Strumenti di misurazione quantitativa che forniscono informazioni oggettive sullo stato e sulle prestazioni di un progetto. Le metriche aiutano a monitorare l'avanzamento, identificare problemi e prendere decisioni informate durante lo sviluppo del software.

N**O****P**

- **PB:** (Project Baseline) un'istantanea (snapshot) del progetto al termine delle attività di pianificazione con il progetto approvato dal committente.
- **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).
- **Plan-Do-Check-Act:** Ciclo di miglioramento continuo che prevede quattro fasi: Pianificare (Plan), Eseguire (Do), Verificare (Check) e Agire (Act). Questo ciclo aiuta

a identificare e risolvere problemi, migliorare processi e garantire la qualità del prodotto.

- **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.
- **Processo:** insieme di attività correlate e coese che trasformano ingressi (bisogni) in uscite (prodotti) secondo regole date, consumando risorse nel farlo.
- **Progetto:** Insieme di attività che:
 - Devono raggiungere dati obiettivi a partire da date specifiche
 - Hanno un inizio e una fine fissate in calendario
 - Dispongono di risorse limitate (persone, tempo, denaro, strumenti)
 - Consumano risorse nel loro svolgersi
- **Proponente:** È colui che esprime le aspettative iniziali ("requisiti utente") che il sistema deve soddisfare. Il proponente soddisfa le condizioni di aver posto i requisiti e aver generato l'opportunità, ma non è necessariamente l'utente finale che userà il 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.

Q

R

- **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).
- **RTB:** (Requirement and Technical Baseline) 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).
- **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.
- **Sprint:** Un periodo di tempo definito durante il quale un team di sviluppo software lavora per completare un insieme specifico di attività o obiettivi. Gli sprint sono una componente chiave della metodologia Agile, in particolare di Scrum, e aiutano a organizzare il lavoro in iterazioni brevi e gestibili.
- **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.
- **Test di Sistema:** Test che verifica la conformità del sistema alle specifiche, con l'obiettivo di identificare eventuali difetti o problemi prima del rilascio. I test di sistema sono eseguiti in un ambiente che simula le condizioni reali di utilizzo del software.
- **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:** Funzionalità di un sistema, che non descrivono però la loro implementazione.
- **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

- **Validazione:** Controllo della soddisfazione delle aspettative (Finale).
- **Verifica:** Controllo che lo sviluppo non introduca errori.

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