



Il gruppo Avant-Garde

sweavantgarde@gmail.com

---

## GLOSSARIO

---

Informazioni sul documento:

|                     |                     |
|---------------------|---------------------|
| <b>Versione</b>     | 1.0.0               |
| <b>Approvazione</b> | Zaccaria Marangon   |
| <b>Redazione</b>    | Andrea Mangolini    |
| <b>Verifica</b>     | Lorenzo Pasqualotto |
| <b>Uso</b>          | Esterno             |

## Registro delle Modifiche

| Versione | Data     | Nominativo          | Ruolo          | Descrizione   |
|----------|----------|---------------------|----------------|---|
| v0.0.1   | 20-11-23 | Andrea Mangolini    | Programmatore  | Prima redazione, aggiunta dei vocaboli dal documento “Norme di progetto” alla versione 1.0.0  |
| v0.0.2   | 07-01-24 | Jessica Carretta    | Analista       | Aggiunta dei vocaboli del documento “Analisi dei Requisiti” e alcuni del “Piano di progetto”. |
| v0.0.3   | 08-01-24 | Lorenzo Pasqualotto | Amministratore | Aggiunta dei vocaboli del documento “Piano di qualifica”.                                     |
| v0.0.4   | 26-03-24 | Giulio Biscontin    | Amministratore | Aggiunta di tutti i vocaboli del documento “Piano di progetto”.                               |
| v0.1.0   | 26-03-24 | Lorenzo Pasqualotto | Programmatore  | Verifica documento  |
| v1.0.0   | 27-03-24 | Zaccaria Marangon   | Responsabile   | Approvazione del documento alla versione 1.0.0  |

## Indice

|       |    |
|-------|----|
| Sigle | 3  |
| A     | 4  |
| B     | 5  |
| C     | 6  |
| D     | 7  |
| E     | 8  |
| F     | 9  |
| G     | 10 |
| H     | 11 |
| I     | 12 |
| J     | 13 |
| K     | 14 |
| L     | 15 |
| M     | 16 |
| N     | 17 |
| O     | 18 |
| P     | 19 |
| Q     | 20 |
| R     | 21 |
| S     | 22 |
| T     | 23 |
| U     | 24 |
| V     | 25 |
| W     | 26 |
| X     | 27 |
| Y     | 28 |
| Z     | 29 |

## Sigle

**MVP** • Minimum Viable Product

**PB** • Product Baseline

**PoC** • Proof of Concept

**RTB** • Requirements and Technology Baseline

**WoW** • Way of Working

## A

**Architettura software** • L'organizzazione di un sistema software, espressa attraverso le sue componenti, le relazioni tra di loro e con l'ambiente e le proprietà che caratterizzano le componenti e le relazioni.

**Attore** • (*nei casi d'uso*) Un qualsiasi utente, che può essere umano, software o hardware, che interagisce con il sistema per raggiungere un obiettivo o per svolgere un compito. Viene rappresentato graficamente nei diagrammi dei casi d'uso (*Use Case Diagram*) come un uomo stilizzato (*stickman*).

## B

**Bin** • Unità di dimensioni fisse con cui le scaffalature sono misurate e suddivise. Può contenere al massimo un prodotto.

## C

**Capitolato** • Documento prodotto da una azienda proponente con lo scopo di illustrare in maniera discorsiva un particolare problema, esponendo le caratteristiche che un prodotto software deve possedere per poterlo risolvere. Presenta un'opportunità, ma non descrive in maniera dettagliata i requisiti che il prodotto deve soddisfare.

**Caso d'uso** • Tecnica usata in ingegneria del software per la raccolta di requisiti, descrive le interazioni tra un utente e il sistema che consentono all'utente di raggiungere un obiettivo o svolgere un compito.

**Ciclo di vita** • (*del software*) Tutte le fasi in cui viene suddivisa l'attività di realizzazione di prodotti software, il cui obiettivo non è la sola realizzazione del prodotto, ma anche della documentazione ad esso associata; tipicamente, tali sottoattività includono l'analisi dei requisiti, la progettazione, la realizzazione, il testing e la manutenzione.

**Cruscotto** • Pannello di controllo che racchiude tutte le informazioni chiave per comprendere a colpo d'occhio lo stato di un progetto software. Le informazioni vengono solitamente mostrate attraverso grafici e metriche, per garantire una veloce interpretazione e permettere a chi le visualizza di prendere decisioni in maniera veloce ma consapevole.

**Complessità Ciclomatica** • La complessità ciclomatica di una sezione di codice è il numero di cammini linearmente indipendenti attraverso il codice sorgente. La formula della complessità ciclomatica è data da:

$$M=E-N+2P$$

dove:

M è la complessità ciclomatica,

E rappresenta il numero di archi nel grafo,

N è il numero di nodi,

P è il numero di componenti connesse al grafo.

## D

**Deliverable** • Il prodotto software comprensivo di documentazione in una determinata fase interna del progetto.

**Design pattern** • Soluzione progettuale generale ad un problema ricorrente, che può essere adattata a contesti differenti.

**Diagramma di Gantt** • Strumento di supporto alla gestione dei progetti, che permette di visualizzare la pianificazione temporale delle attività di un progetto.

**Dominio applicativo** • L'insieme dei problemi e delle soluzioni che caratterizzano un determinato settore di applicazione.



## E

**Efficacia** • Misura della capacità di raggiungere un obiettivo prefissato.

**Efficienza** • Misura dell'abilità di raggiungere un obiettivo impiegando le risorse minime indispensabili.

## F

**Feature** • Caratteristica o funzionalità di un prodotto software.

**Framework** • Architettura logica di supporto sulla quale un software può essere progettato e realizzato, facilitando lo sviluppo del programmatore. Ha l'obiettivo di creare una infrastruttura generale, lasciando al programmatore lo sviluppo del contenuto vero e proprio dell'applicazione, risparmiandogli la riscrittura di codice già scritto in precedenza per compiti simili.

## G

## H

## I

**Issue** • Unità di lavoro composta da una serie di operazioni atte a raggiungere un obiettivo comune, per avanzare nel progetto software o per realizzare un miglioramento.

**Item** • Prodotto presente all'interno del magazzino creato, che può essere spostato o eliminato.

**J**

## K

## L

**Lista di controllo** • Elenco esaustivo di operazioni da effettuare per completare una attività, presentate generalmente sotto forma di lista con piccole caselle di controllo per inserire il segno di spunta ad operazione completata.

**Libreria** • Con riferimento all'applicativo sviluppato, si intende un'area separata in cui visualizzare in maniera testuale i dati del magazzino, in particolare le informazioni relativi ai prodotti e alle scaffalature.



## M

**Milestone** • Raggiungimento di importanti traguardi intermedi nello svolgimento di un progetto, spesso rappresentato tramite attività con durata di zero o di un giorno.

**Minimum Viable Product** • Prodotto software realizzato durante lo sviluppo di un progetto software che approssima da vicino il prodotto finale atteso, senza però essere definitivo. Permette di valutare la bontà della visione iniziale di prodotto, consente agevole uso esplorativo e aiuta a prendere decisioni ben fondate per il completamento definitivo del prodotto.

N

## O

## P

**Processo software** • Un insieme di attività, metodi, pratiche e trasformazioni che persone coinvolte in un progetto software devono seguire per sviluppare un prodotto software.

**Product Baseline** • Seconda fase di revisione di avanzamento del progetto didattico, ha lo scopo di esaminare la maturità della baseline architetturale del prodotto software e la sua realizzazione. Richiede un design definitivo e un avanzamento sostanziale della codifica tale per cui il prodotto software è dimostrabile in forma di Minimum Viable Product

**Progetto** • Un insieme di attività correlate fra loro ed eseguite al fine di raggiungere uno o più obiettivi in un periodo di tempo fissato e con un numero limitato di risorse.

**Proof of Concept** • Realizzazione parziale ed incompleta di un progetto, allo scopo di determinare la fattibilità dell'idea e delle tecnologie utilizzate. Il Proof of Concept rappresenta sostanzialmente una demo prototipale (spesso usa-e-getta) del software in sviluppo, permettendo la dimostrazione pratica dei funzionamenti di base.

**Proponente** • Entità che propone un capitolato d'appalto, ovvero un'azienda o un ente che ha un problema da risolvere e cerca un'azienda che possa realizzare un prodotto software per risolverlo.

**Piano di contingenza** • Programma operativo che indica preventivamente cosa fare nel caso uno dei rischi si verifichi.

## Q

## R

**Repository** • Archivio digitale centralizzato che contiene tutti i dati di progetto, inclusi codice e documentazione.

**Requirements and Technology Baseline** • Prima fase di revisione di avanzamento del progetto didattico, ha lo scopo di fissare i requisiti da soddisfare, in accordo con il proponente, tramite analisi dei requisiti e di motivare le tecnologie, i framework e le librerie adottate, dimostrandone adeguatezza e compatibilità tramite la realizzazione di un PoC.

**Requisito** • Una condizione o capacità che il software deve possedere per soddisfare un contratto, uno standard, una specifica o un qualsiasi altro documento formalmente specificato.

**Retrospettiva** • Analisi dei processi in corso di progetto, allo scopo di risolvere i problemi emersi e identificare margini di miglioramento. In genere la retrospettiva si realizza tramite incontri del team di progetto alla fine di una iterazione o ad intervalli regolari, ma comunque sempre dopo una serie di attività per capire cosa è andato bene, cosa non è andato bene e come si può migliorare.

## S

**Snake case** • Convenzione sulla scelta dei caratteri da utilizzare per gli identificatori nella programmazione (ad esempio per nomi di variabili, funzioni, tipi) e nella documentazione, secondo la quale ciascun nome deve essere scritto tutto in minuscolo, con le parole separate da un trattino basso ('\_').

**Software item** • Un qualsiasi oggetto di un sistema software, che può essere codice o documentazione, ma anche configurazioni di sistema, apparecchiature hardware e database, che sono identificati in maniera univoca nel sistema software e tracciati dal processo di gestione della configurazione.

**Sprint** • Periodo di tempo, di durata fissa e prefissata, in cui il team di sviluppo lavora per completare un insieme di attività, con l'obiettivo di consegnare un incremento di prodotto software. Per questo progetto, secondo le norme di progetto adottate, la durata di uno sprint è di due settimane.

**Stakeholder** • Si intende una persona esterna coinvolta nel progetto che “porta” interessi legati all'esecuzione o dall'andamento di questo.

## T

**Template** • Modello predefinito che può essere riutilizzato per creare documenti simili, in modo da garantire uniformità e coerenza tra i documenti prodotti.

**Test** • (*del software*) Processo di valutazione e verifica (sia automatica che manuale), del corretto funzionamento di un prodotto software, di norma applicato durante tutto il ciclo di vita del prodotto, anche prima del rilascio, con l'obiettivo di realizzare un prodotto il più possibile privo di errori.



## U

**Unità architetture** • Una componente atomica dell'architettura di un prodotto software.

## V

## W

**Way of working** • Linee guida che descrivono le modalità secondo cui un team lavora, dal punto di vista degli strumenti utilizzati e delle procedure adottate, con particolare attenzione alla collaborazione tra membri del gruppo.

X

Y

## Z