



**Università degli Studi di Padova**  
Laurea: Informatica  
Corso: Ingegneria del Software  
Anno Accademico: 2025/2026

**Gruppo: NullPointers Group**  
Email: nullpointersg@gmail.com

## Glossario

|                      |                           |
|----------------------|---------------------------|
| Stato                | In approvazione           |
| Versione             | 0.1.1                     |
| Data ultima modifica | 10/11/2025                |
| Redattori            | M. Mazzaretto<br>T. Ceron |
| Verificatori         | M. Brunello               |
| Destinatari          | NullPointers Group        |

## Registro delle modifiche

| Vers  | Data  | Autore                    | Verificatore | Descrizione                   |
|-------|-------|---------------------------|--------------|-------------------------------|
| 0.1.1 | 10-11 | M. Mazzaretto             | M. Brunello  | Correzioni estetiche          |
| 0.1.0 | 7-11  | M. Mazzaretto<br>T. Ceron | M. Brunello  | Creazione e stesura documento |

## **Indice**

|                               |           |
|-------------------------------|-----------|
| <b>Introduzione</b> . . . . . | <b>5</b>  |
| <b>A</b> . . . . .            | <b>6</b>  |
| <b>B</b> . . . . .            | <b>7</b>  |
| <b>C</b> . . . . .            | <b>8</b>  |
| <b>D</b> . . . . .            | <b>9</b>  |
| <b>E</b> . . . . .            | <b>10</b> |
| <b>F</b> . . . . .            | <b>11</b> |
| <b>G</b> . . . . .            | <b>12</b> |
| <b>H</b> . . . . .            | <b>13</b> |
| <b>I</b> . . . . .            | <b>14</b> |
| <b>J</b> . . . . .            | <b>15</b> |
| <b>K</b> . . . . .            | <b>16</b> |
| <b>L</b> . . . . .            | <b>17</b> |
| <b>M</b> . . . . .            | <b>18</b> |
| <b>N</b> . . . . .            | <b>19</b> |
| <b>O</b> . . . . .            | <b>20</b> |
| <b>P</b> . . . . .            | <b>21</b> |
| <b>Q</b> . . . . .            | <b>22</b> |
| <b>R</b> . . . . .            | <b>23</b> |
| <b>S</b> . . . . .            | <b>24</b> |
| <b>U</b> . . . . .            | <b>26</b> |
| <b>V</b> . . . . .            | <b>27</b> |
| <b>W</b> . . . . .            | <b>28</b> |
| <b>X</b> . . . . .            | <b>29</b> |

**Y** . . . . . **30**

**Z** . . . . . **31**

## Introduzione

Il documento definisce in modo univoco i termini adottati all'interno del progetto, così da evitare interpretazioni personali o variazioni di significato.

Le definizioni sono presentate in ordine alfabetico per rendere la consultazione diretta e stabile nel tempo.

Quando un termine compare nei documenti con un significato vincolato dal Glossario, viene segnalato, solo alla prima comparsa, con la notazione:

termine<sup>G</sup>

## A

### **Amministratore**

L'Amministratore si occupa della configurazione e gestione dell'infrastruttura IT di supporto ad un progetto. Il suo ruolo è particolarmente importante nelle fasi iniziali e durante il deployment, dove raggiunge il picco di impegno per garantire un deploy corretto.

### **Analisi dei Requisiti**

Documento contenente i Casi d'Uso identificati da un Team per i requisiti di un determinato progetto.

### **Analista**

Si occupa di identificare e chiarire i requisiti, interpretando le esigenze degli utilizzatori finali per garantire una corretta definizione delle funzionalità.

### **API**

Acronimo di *Application Programming Interface*, è un'interfaccia utile a permettere o facilitare la comunicazione tra diversi software o parti di un singolo software.

### **Attore**

Un'entità esterna e non controllabile dal progetto, ma che interagisce con esso, con attività o obiettivi specifici da soddisfare.

### **Automazione**

Esecuzione deterministica e ripetibile di processi (build, test, deploy, verifiche, provisioning) attivati da eventi del repository o da pianificazioni, definita tramite workflow dichiarativi in YAML, eseguita in ambienti isolati, senza intervento manuale.

### **Azione**

L'insieme di processi intrapresi per mettere in pratica una determinata Decisione.

**B****Backend**

La parte di un progetto che si occupa del funzionamento a livello logico del sistema.

**Baseline**

Stato fissato di artefatti (requisiti, design, codice, documentazione) considerato stabile e sufficientemente maturo, dal quale retrocedere comporterebbe costo o rischio elevato.

**Branch**

Uno specifico ramo di lavoro all'interno di una repository, atta a mantenere una distinzione tra ramo "principale" e rami di lavoro in cui ciascun membro del gruppo può lavorare in maniere asincrona senza rischio di conflitto.

## C

### **Capitolato**

Documento fornito dalla proponente, che espone le principali specifiche e condizioni di un progetto. Stabilisce delle regole e delle aspettative che il fornitore deve rispettare.

### **Caso d'uso**

Insieme di azioni all'interno di uno scenario, con in comune uno scopo finale per un Attore.

## D

### **Decisione**

Un accordo all'interno del Team, riguardante metodologie o convenzioni da adottare per soddisfare un determinato bisogno.

## E

### **Efficacia**

Indica il modo con cui un insieme di attività raggiunge gli obiettivi attesi.

### **Efficienza**

Rapporto tra risultati ottenuti e risorse impiegate per ottenerli.

**F****Frontend**

La parte di un progetto che si occupa dell'interazione diretta con l'utilizzatore finale.

**Funzionalità**

Una caratteristica offerta da un Sistema, spesso con lo scopo di soddisfare un bisogno.

## G

### **GitHub**

Servizio cloud che permette di hostare una o più repository git, offre strumenti che semplificano la collaborazione e l'automazione di processi.

### **GitHub Action**

Strumento messo a disposizione da GitHub per effettuare automazioni all'interno di una repository.

## H

# I

## ISO

Ente di standardizzazione internazionale che definisce norme tecniche condive per garantire interoperabilità, qualità e uniformità tra prodotti, processi e sistemi.

## Issue

All'interno dell'ecosistema GitHub, rappresenta un'attività che richiede modifiche al codice sorgente per essere svolta, può trattarsi di un bug da risolvere, una feature da aggiungere, di una correzione minore o anche di qualcosa definito dagli sviluppatori.

**J**

## K

**L****LaTeX**

Linguaggio di markup utilizzato dal gruppo per produrre tutti i documenti.

**LLM**

Large Language Models. Indica i modelli addestrati su enormi quantità di testo come ChatGPT e DeepSeek.

## M

### **Major**

Una modifica atta ad approvare un documento in una sua versione "pronta al rilascio". In un sistema di versionamento x.y.z, la Major è rappresentata dalla lettera x, e ogni modifica di tipo Major aggiorna un file dalla versione x.y.z alla versione (x+1).0.0.

### **Manutenzione**

Insieme di attività atte a risolvere i problemi e migliorare le funzionalità di un prodotto.

### **Merge**

Processo con il quale, in un sistema di versionamento come git, un ramo di lavoro viene fatto confluire in un altro, generalmente in un grado gerarchico più elevato.

### **Milestone**

Periodo di tempo entro il quale il team si pone il raggiungimento di determinati obiettivi.

### **Minimum Viable Product**

La prima versione di un prodotto che assolve le funzionalità minime per rispettare i requisiti obbligatori stabiliti dalla proponenti.

### **Minor**

Una modifica atta ad aggiungere migliorie o funzionalità significative. In un sistema di versionamento x.y.z, la Minor è rappresentata dalla lettera y, e ogni modifica di tipo Minor aggiorna un file dalla versione x.y.z alla versione x.(y+1).0.

### **MVP**

Acronimo di *Minimum Viable Product*, per la definizione vedere 'Minimum Viable Product'

## N

### **Norme di Progetto**

Documento che descrive il Way of Working del gruppo. Contiene le regole che NullPointers Group rispetta per garantire un lavoro efficace ed efficiente.

# O

**P****Patch**

Una modifica atta a risolvere ortografici o di struttura. In un sistema di versionamento x.y.z, la patch è rappresentata dalla lettera z, e ogni modifica di tipo patch aggiorna la versione di un file da x.y.z a x.y.(z+1).

**PB**

Acronimo di *Product Baseline*, per la definizione vedere 'Product Baseline'

**PoC**

Acronimo di *Proof of Concept*, per la definizione vedere 'Proof of Concept'

**PR**

Acronimo di *Pull Request*, per la definizione vedere 'Pull Request'

**Product Baseline**

Fase finale in un progetto, che prevede la realizzazione di un Minimum Viable Product, a seguito del superamento dei test concordati nella Requirements and Technology Baseline.

**Proof of Concept**

Prodotto dimostrativo atto a presentare la fattibilità dell'idea.

**Pull Request**

All'interno dell'ecosistema GitHub, uno strumento che permette di sospendere momentaneamente la pubblicazione di una modifica nel ramo principale della repository, in attesa di approvazione e/o workflow automatici (e.g. GitHub Action).

## Q

## R

### **Redattore**

Colui che scrive il documento. Viene identificato come "Autore" nel registro delle modifiche.

### **Repository**

Uno spazio condiviso tra tutti i membri del Team, in cui viene memorizzato e versionato tutto il codice sorgente prodotto.

### **Requisito funzionale**

Definiscono cosa il sistema deve fare, descrivendo le sue azioni e funzioni specifiche. Sono le capacità che l'utente si aspetta, come "aggiungere un prodotto al carrello" in un negozio online.

### **Requisito non funzionale**

Specificano come il sistema deve comportarsi mentre svolge quelle funzioni. Si concentrano su qualità come le prestazioni, la sicurezza o l'affidabilità, ad esempio "il sistema deve supportare 1000 utenti contemporanei senza rallentamenti".

### **Responsabile**

Il Responsabile coordina le attività del gruppo e garantisce una pianificazione efficace.

### **RTB**

Baseline iniziale e vincolante di requisiti e tecnologie approvate per il progetto. Comprende requisiti funzionali, non funzionali, di sicurezza e conformità, insieme a strumenti, linguaggi, piattaforme e infrastrutture scelte. Costituisce il riferimento stabile per verificare coerenza e avanzamento. Qualsiasi modifica richiede revisione e approvazione formale.

**S****SM**

Codice stabilito dal Team, rappresenta un'attività, con corrispettiva issue in GitHub all'interno della Repository 'SmartOrder', indica dunque un'attività da svolgere relativa al progetto e al *PoC*. Il codice è seguito da un numero incrementale che permette di distinguere univocamente una singola attività.

**SMD**

Codice stabilito dal Team, rappresenta un'attività, con corrispettiva issue in GitHub all'interno della Repository 'Documentazione', indica dunque un'attività da svolgere relativa alla documentazione di progetto. Il codice è seguito da un numero incrementale che permette di distinguere univocamente una singola attività.

**T****TD**

Acronimo di *To Do*, indica un'attività futura decisa in un incontro interno/esterno, presenta uno o più assegnatari. Il codice è seguito da un numero incrementale che permette di distinguere univocamente una singola attività.

**Test**

Un processo dello sviluppo software che si occupa di verificare la funzionalità di una o più parti del prodotto.

## **U**

**V****Validazione**

Insieme di controlli per assicurarsi che un prodotto (o una parte di esso) soddisfi in modo congruo ciò per cui è stato creato.

**Verifica**

Insieme di processi atti a garantire che una funzione all'interno di un prodotto soddisfi determinati requisiti.

**Verificatore**

Il Verificatore si occupa di assicurare la qualità dei prodotti e dei processi adottati, effettuando revisioni e test.

**VI**

Acronimo di *Verbale Interno*, viene utilizzato per tracciare nel tempo decisioni e azioni intraprese in un incontro interno. Il codice è seguito da un numero incrementale che permette di distinguere univocamente una singola attività.

**VE**

Acronimo di *Verbale Esterno*, viene utilizzato per tracciare nel tempo decisioni e azioni intraprese in un incontro esterno. Il codice è seguito da un numero incrementale che permette di distinguere univocamente una singola attività.

## **W**

### **Way of Working**

Insieme di metodologie e strutture organizzative approvate dall'intero Team per svolgere in modo coerente i propri compiti.

## X

## Y

**Z**