



**Università degli Studi di Padova**

Laurea: Informatica

Corso: Ingegneria del Software

Anno Accademico: 2025/2026



**Gruppo RubberDuck**

email: [GroupRubberDuck@gmail.com](mailto:GroupRubberDuck@gmail.com)

**Norme di Progetto**

<b>Stato</b>	Approvato
<b>Versione</b>	1.0.0
<b>Autori</b>	Davide Lorenzon Aldo Bettega Filippo Guerra Ana Maria Draghici
<b>Verificatori</b>	Ana Maria Draghici Davide Lorenzon Aldo Bettega Filippo Guerra Felician Mario Neculescu
<b>Uso</b>	Interno
<b>Destinatari</b>	Tutto il gruppo

Vers.	Data	Autore	Verificatore	Descrizione
1.0.0	2026-04-01	Felician Mario Necsulescu	Aldo Bettega	Approvazione
0.13.1	2026-03-20	Davide Lorenzon	Filippo Guerra	<p>Aggiornamento della sezione relativa alla gestione dell'infra-struttura</p> <p><b>Sezione 4.3</b></p> <ul style="list-style-type: none"> <li>• <a href="#">Numerazione automatica</a>,</li> <li>• <a href="#">Tracciamento automatico</a>,</li> <li>• <a href="#">Plot diagrammi</a>,</li> <li>• <a href="#">Uso di script python</a>,</li> </ul>
0.13.0	2026-03-20	Ana Maria Draghici	Felician Mario Necsulescu	Aggiunta alla <b>Sezione 3.5</b> la descrizione delle varie strategie di testing (Processo di Qualifica)
0.12.0	2026-02-05	Davide Lorenzon	Filippo Guerra	Aggiunte descrizioni delle metriche di qualità
0.11.0	2026-01-14	Ana Maria Draghici	Davide Lorenzon	Completata prima scrittura Norme di Progetto
0.10.0	2026-01-10	Ana Maria Draghici	Davide Lorenzon	Rivisto processo di supporto <b>Sezione 3.1</b> , aggiunte sottosezioni <b>Sezione 3.4</b> e <b>Sezione 3.5</b>
0.9.0	2026-01-10	Ana Maria Draghici	Davide Lorenzon	Rivisti processi primari <b>Sezione 2.1</b> , sottosezioni <b>Sezione 2.2</b> e <b>Sezione 2.3</b>

Vers.	Data	Autore	Verificatore	Descrizione
0.8.4	2025-12-15	Davide Lorenzon	Ana Maria Draghici	Stesura della sezione del Processo di miglioramento <b><u>Sezione 4.4</u></b> e processo di formazione <b><u>Sezione 4.5</u></b>
0.8.3	2025-12-14	Davide Lorenzon	Ana Maria Draghici	Stesura della sezione Gestione dell'infrastruttura <b><u>Sezione 4.3</u></b>
0.8.2	2025-12-13	Davide Lorenzon	Ana Maria Draghici	Rivista introduzione, approfondita Gestione del processo <b><u>Sezione 4.2</u></b>
0.8.1	2025-12-10	Davide Lorenzon	Ana Maria Draghici	Aggiunta Rendicontazione delle ore <b><u>Sezione 4.2.4.2</u></b>
0.8.0	2025-12-09	Filippo Guerra	Davide Lorenzon	Aggiunto Processo di fornitura <b><u>Sezione 2.2</u></b>
0.7.0	2025-12-04	Aldo Bettega	Davide Lorenzon	Aggiunta scrittura commit <b><u>Sezione 3.4.4.1</u></b> , e sezione su workflow documentale <b><u>Sezione 3.2.4.3</u></b> , aggiornata definition of done <sub>G</sub> <b><u>Sezione 3.5.3.5</u></b>
0.6.1	2025-12-02	Davide Lorenzon	Davide Lorenzon	Apportate modifiche di ordine nella documentazione <b><u>Sezione 3.2</u></b>
0.6.0	2025-12-02	Ana Maria Draghici	Aldo Bettega	Completata struttura documenti <b><u>Sezione 3.2.6.3</u></b> , aggiunto «Configurazione» <b><u>Sezione 3.3</u></b>

Vers.	Data	Autore	Verificatore	Descrizione
0.5.0	2025-11-30	Ana Maria Draghici	Aldo Bettega	Aggiunto «Definizione of Done <sub>G</sub> » <b>Sezione 3.5.3.5</b> e «Issue <sub>G</sub> tracking System» <b>Sezione 4.2.4.4</b>
0.4.0	2025-11-29	Guerra Filippo	Ana Maria Draghici	Aggiunto «ruolo-documento» <b>Sezione 4.2.4.3</b>
0.3.0	2025-11-11	Ana Maria Draghici	Davide Lorenzon	Aggiunta struttura Analisi Requisiti <b>Sezione 3.2.6.4</b>
0.2.0	2025-11-11	Davide Lorenzon	Ana Maria Draghici	Aggiunta struttura dei documenti <b>Sezione 3.2.6.3</b>
0.1.0	2025-11-	Davide Lorenzon	Ana Maria Draghici	Stesura iniziale

## Indice

1) Introduzione .....	1
1.1) Scopo del documento .....	1
1.2) Scopo del prodotto .....	1
1.3) Glossario .....	1
1.4) Riferimenti .....	1
1.4.1) Riferimenti normativi .....	1
1.4.2) Riferimenti informativi .....	2
2) <b>Processi Primari</b> .....	3
2.1) Introduzione ai processi primari .....	3
2.2) <b>Processo di fornitura</b> .....	3
2.2.1) Introduzione .....	3
2.2.2) Scopo del processo .....	3
2.2.3) Attività del processo .....	4
2.2.4) Procedure operative .....	4
2.2.4.1) Metodi di comunicazione .....	4
2.2.5) Documenti principali .....	5
2.2.6) Strumenti a supporto .....	6
2.3) <b>Processo di sviluppo</b> .....	6
2.3.1) Introduzione .....	6
2.3.2) Scopo del processo .....	6
2.3.3) Attività del processo .....	7
2.3.4) Inquadramento del processo nelle Baseline di progetto .....	8
2.3.5) Procedure operative .....	8
2.3.6) Documenti principali .....	10
2.3.7) Strumenti a supporto .....	10
3) <b>Processi di Supporto</b> .....	11
3.1) Introduzione ai processi di supporto .....	11
3.2) <b>Processo di documentazione</b> .....	11
3.2.1) Introduzione .....	11
3.2.2) Scopo del processo .....	11
3.2.3) Attività del processo .....	12
3.2.3.1) Impegno del gruppo .....	12
3.2.3.2) Attività principali .....	12
3.2.4) Procedure operative .....	12
3.2.4.1) Identificazione dei documenti .....	12
3.2.4.2) Progettazione dei documenti .....	13
3.2.4.3) Workflow documentale <sub>G</sub> .....	13
3.2.4.4) Stati del documento .....	13
3.2.4.5) Procedura di avanzamento tra stati .....	14
3.2.4.6) Procedura di archiviazione e tracciabilità .....	14

3.2.4.7)	Pubblicazione e Distribuzione della Documentazione .....	15
3.2.5)	Inquadramento del processo nelle Baseline di progetto .....	15
3.2.6)	Documentazione fornita .....	16
3.2.6.1)	Elenco dei documenti .....	16
3.2.6.2)	Informazioni comuni .....	16
3.2.6.3)	Struttura specifica .....	16
3.2.6.4)	Analisi dei Requisiti <sub>G</sub> .....	17
3.2.6.4.1)	Struttura principale .....	17
3.2.6.5)	Piano di Qualifica <sub>G</sub> .....	17
3.2.6.5.1)	Struttura principale .....	17
3.2.6.6)	Piano di Progetto .....	18
3.2.6.6.1)	Struttura principale .....	18
3.2.6.7)	Verbali .....	19
3.2.6.7.1)	Struttura principale .....	19
3.2.6.8)	Diario di Bordo .....	20
3.2.6.9)	Norme di Progetto .....	20
3.2.6.9.1)	Struttura principale .....	20
3.2.6.10)	Glossario .....	21
3.2.7)	Strumenti di supporto .....	21
3.3)	<b>Processo di Gestione delle Configurazioni</b> .....	22
3.3.1)	Introduzione .....	22
3.3.2)	Scopo del processo .....	22
3.3.3)	Attività del processo .....	22
3.3.4)	Procedure operative .....	22
3.3.4.1)	Codice di versione .....	22
3.3.4.1.1)	Regole di incremento .....	23
3.3.4.2)	Versionamento automatico tramite Typst <sub>G</sub> .....	23
3.3.4.3)	Registro modifiche .....	24
3.3.4.4)	Formato nome dei verbali .....	24
3.3.5)	Strumenti principali utilizzati .....	24
3.4)	<b>Processo di Accertamento della Qualità</b> .....	24
3.4.1)	Introduzione .....	24
3.4.2)	Scopo del processo .....	25
3.4.3)	Attività del processo .....	25
3.4.4)	Procedure operative .....	25
3.4.4.1)	Scrittura dei commit .....	25
3.4.5)	Strumenti .....	26
3.4.6)	Metriche .....	26
3.5)	<b>Processo di Qualifica</b> .....	26
3.5.1)	Introduzione .....	26
3.5.2)	Scopo del processo .....	26

3.5.3)	Attività del processo .....	26
3.5.3.1)	Analisi statica .....	27
3.5.3.1.1)	Checklist di ispezione .....	27
3.5.3.1.2)	Quando usare quale metodo .....	28
3.5.3.2)	Analisi dinamica .....	28
3.5.3.3)	Classificazione dei test .....	28
3.5.3.3.1)	Stati dei test .....	29
3.5.3.3.2)	Test di Unità <sub>G</sub> .....	29
3.5.3.3.3)	Test di Integrazione <sub>G</sub> .....	30
3.5.3.3.4)	Test di Sistema <sub>G</sub> .....	30
3.5.3.3.5)	Test di Accettazione <sub>G</sub> .....	30
3.5.3.3.6)	Test di Regressione .....	30
3.5.3.4)	Validazione <sub>G</sub> .....	31
3.5.3.4.1)	Attività di validazione <sub>G</sub> .....	31
3.5.3.5)	Definition of Done <sub>G</sub> .....	31
3.5.4)	Strumenti a supporto .....	32
3.5.5)	Documentazione a supporto .....	32
4)	<b>Processi Organizzativi</b> .....	34
4.1)	Introduzione .....	34
4.2)	<b>Processo di gestione del processo</b> .....	34
4.2.1)	Introduzione .....	34
4.2.2)	Scopo del processo .....	34
4.2.3)	Attività del processo .....	35
4.2.4)	Procedure operative .....	36
4.2.4.1)	Ruoli di Progetto .....	36
4.2.4.2)	Rendicontazione delle ore .....	37
4.2.4.3)	Assegnazione Ruolo-Documento .....	37
4.2.4.4)	Issue <sub>G</sub> Tracking System .....	38
4.2.4.4.1)	Creazione e struttura di un'issue <sub>G</sub> .....	39
4.2.4.4.2)	Flusso operativo .....	40
4.3)	<b>Processo di Gestione dell'infrastruttura</b> .....	41
4.3.1)	Introduzione .....	41
4.3.2)	Scopo del processo .....	41
4.3.3)	Attività del processo .....	41
4.3.4)	Procedure operative .....	41
4.3.4.1)	Implementazione del processo .....	41
4.3.4.2)	Creazione .....	42
4.3.4.3)	Manutenzione <sub>G</sub> .....	44
4.3.4.3.1)	Project board .....	45
4.3.4.3.2)	Milestone <sub>G</sub> .....	45
4.3.4.4)	GitHub Actions <sub>G</sub> .....	45

4.3.4.5) Script e automazioni .....	45
4.3.4.6) Discord .....	46
4.3.4.7) Strumenti Google .....	46
4.4) <b>Processo di miglioramento</b> .....	46
4.4.1) Introduzione .....	46
4.4.2) Attività del processo .....	46
4.4.3) Procedure operative .....	47
4.4.3.1) Ciclo PDCA <sub>G</sub> .....	47
4.4.3.1.1) Guida alla Retrospettiva .....	47
4.5) <b>Processo di formazione</b> .....	48
4.5.1) Introduzione .....	48
4.5.2) Scopo del processo .....	48
4.5.3) Attività del processo .....	48
4.5.4) Procedure operative .....	48
4.5.4.1) Apprendimento libero .....	48
4.5.4.2) Apprendimento in gruppo .....	49
4.5.4.3) Trasferimento dei ruoli .....	49
4.5.4.4) Rendicontazione delle ore .....	49
5) <b>Metriche</b> .....	50
5.1) Introduzione .....	50
5.1.1) Nomenclatura delle Metriche .....	52
6) <b>Metriche di Qualità del Processo</b> .....	52
6.1) Processi primari .....	52
6.1.1) Fornitura .....	53
6.1.2) Sviluppo .....	57
6.2) Processi di supporto .....	58
6.2.1) Documentazione .....	58
6.2.2) Verifica <sub>G</sub> .....	60
6.3) Processi organizzativi .....	60
6.3.1) Gestione dei processi .....	60
7) <b>Metriche di Qualità del Prodotto</b> .....	62
7.1) Funzionalità .....	62
7.2) Affidabilità .....	64
7.3) Usabilità .....	65
7.4) Efficienza .....	66
7.5) Manutenibilità .....	68

## Elenco delle Tabelle

Tabella 1	Metodologie di testing .....	28
Tabella 2	Tipi di test .....	29
Tabella 3	Stati dei test .....	29
Tabella 4	Tipi di test d integrazione .....	30
Tabella 5	Strumenti utilizzati .....	42
Tabella 6	Metriche processo di Fornitura .....	53
Tabella 7	Metriche processo di Sviluppo .....	57
Tabella 8	Metriche processo di Documentazione .....	58
Tabella 9	Metriche processo di Verifica <sub>G</sub> .....	60
Tabella 10	Metriche processo di Gestione dei Processi .....	60
Tabella 11	Metriche di funzionalità del prodotto .....	62
Tabella 12	Metriche di affidabilità del prodotto .....	64
Tabella 13	Metriche di usabilità del prodotto .....	65
Tabella 14	Metriche di efficienza del prodotto <sup>1</sup> .....	66
Tabella 15	Metriche manutenibilità del prodotto .....	68

---

<sup>1</sup>I valori soglia di questa sezione sono espressi in termini assoluti, non come percentuali relative.

# 1) Introduzione

## 1.1) Scopo del documento

In questo documento il gruppo definisce e tiene traccia del proprio way of working, ovvero l'insieme di pratiche e convenzioni adottate per organizzare e svolgere il lavoro di progetto.

La stesura del documento avviene in modo incrementale, evolvendosi parallelamente all'avanzamento delle attività. Nel corso del progetto potrà essere soggetto ad aggiunte, modifiche o rimozioni, derivate dal processo di apprendimento e sperimentazione del team.

Tali aggiornamenti nascono dall'analisi e dall'evoluzione delle best practices comuni nell'ingegneria del software, che il gruppo studia, applica e adatta in funzione delle esigenze del team e del contesto progettuale.

## 1.2) Scopo del prodotto

Il prodotto sviluppato è un'applicazione software progettata per verificare in modo automatico la conformità<sub>G</sub> alla norma EN 18031<sub>G</sub>, lo standard tecnico europeo che definisce i requisiti di sicurezza informatica per i dispositivi radio wireless (es. Wi-Fi<sub>G</sub>, LTE<sub>G</sub>, Bluetooth<sub>G</sub>, IoT<sub>G</sub> wireless).

L'obiettivo dell'applicazione è supportare e guidare l'utente nella valutazione<sub>G</sub> dei requisiti di cybersecurity<sub>G</sub>, eseguendo in autonomia i percorsi decisionali tramite decision tree<sub>G</sub> (alberi di decisione). Questo approccio permette di accelerare, standardizzare e rendere più affidabile il processo di verifica<sub>G</sub> della conformità<sub>G</sub>, con la generazione automatica della documentazione tecnica richiesta a supporto dell'assessment.

## 1.3) Glossario

Per garantire precisione terminologica senza compromettere la leggibilità, in questo documento viene adottato il seguente approccio per la gestione dei riferimenti al Glossario:

I termini tecnici vengono marcati con **pedice "G" (esempio<sub>G</sub>)**.

Questo sistema consente di mantenere il documento tecnicamente rigoroso, chiaro e facilmente navigabile, favorendo la consultazione mirata del Glossario solo quando necessario.

## 1.4) Riferimenti

### 1.4.1) Riferimenti normativi

- Capitolato d'appalto C1 - Automated EN18031 Compliance Verification<sub>G</sub> di BlueWind Srl

<https://www.math.unipd.it/~tullio/IS-1/2025/Progetto/C1.pdf<sub>G</sub>>

*Ultima consultazione: 14 gennaio 2025*

- Regolamento del progetto  
<https://www.math.unipd.it/~tullio/IS-1/2025/Dispense/PD1.pdf>  
*Ultima consultazione: 11 dicembre 2025*

#### 1.4.2) Riferimenti informativi

- Standard ISO/IEC/IEEE 12207:2017  
<https://www.iso.org/standard/63712.html>  
*Ultima consultazione: 10 gennaio 2026*
- Software Engineering, Ian Sommerville  
[https://galileodiscovery.unipd.it/permalink/39UPD\\_INST/1j3btfv/alma9938989417806046](https://galileodiscovery.unipd.it/permalink/39UPD_INST/1j3btfv/alma9938989417806046)  
*Ultima consultazione: 20 febbraio 2026*
- Standard ISO/IEC 9126  
[https://en.wikipedia.org/wiki/ISO/IEC\\_9126](https://en.wikipedia.org/wiki/ISO/IEC_9126)  
*Ultima consultazione: 10 gennaio 2026*
- Standard ISO/IEC/IEEE 12207:1995  
[https://www.math.unipd.it/~tullio/IS-1/2009/Approfondimenti/ISO\\_12207-1995.pdf](https://www.math.unipd.it/~tullio/IS-1/2009/Approfondimenti/ISO_12207-1995.pdf)  
*Ultima consultazione: 10 gennaio 2026*
- Glossario, versione 1.0.0  
<https://grouprubberduck.github.io/Documentazione/glossario.html>  
*Ultima consultazione: 11 marzo 2026*

## 2) Processi Primari

### 2.1) Introduzione ai processi primari

Per sviluppare un sistema software di qualità, la semplice scrittura di codice e l'esecuzione dei test non sono sufficienti: il prodotto deve essere **affidabile, utilizzabile da un ampio numero di utenti e facilmente manutenibile nel tempo**.

Per raggiungere questi obiettivi, è necessario adottare un modello di sviluppo che definisca **processi chiari e strutturati**, capaci di guidare in modo sistematico le attività del progetto.

Secondo lo **standard ISO/IEC 12207**, i principali processi primari sono:

- **Fornitura**, che gestisce i rapporti con il committente o proponente esterno;
- **Sviluppo**, che si occupa della realizzazione del prodotto software vero e proprio.

### 2.2) Processo di fornitura

#### 2.2.1) Introduzione

Il processo di fornitura descrive le attività necessarie per **instaurare, gestire e mantenere il rapporto tra il gruppo e la proponente esterna (BlueWind S.r.l.)** durante lo svolgimento del progetto.

Seguendo le linee guida dettate dai principi espressi dallo [Standard ISO 12207:2017](#) — che definisce il processo di supply come l'insieme delle attività volte alla definizione degli accordi contrattuali, alla pianificazione, al monitoraggio e alla consegna dei prodotti software — il gruppo si impegna a stabilire modalità **organizzate, verificabili e tracciabili** per comunicare, consegnare documentazione e garantire trasparenza verso la proponente.

#### 2.2.2) Scopo del processo

Lo scopo del processo di fornitura è garantire che la relazione tra gruppo e proponente sia **strutturata, efficiente e basata su comunicazioni chiare e documentate**.

Gli obiettivi principali includono:

- Definire i **canali ufficiali di comunicazione** tra gruppo e proponente;
- Stabilire **frequenza, modalità e tempi dei confronti**;
- Assicurare che le richieste della proponente siano **comprese, tracciate e integrate** nel flusso di lavoro;
- Produrre **documentazione coerente** per revisioni e fasi di progetto;
- Garantire **trasparenza, tracciabilità e rispetto delle scadenze** previste da capitolato e milestone.

### 2.2.3) Attività del processo

Il processo di fornitura comprende le seguenti attività principali:

#### 1. Inizializzazione

Analisi delle richieste della proponente, identificazione dei requisiti contrattuali e valutazione<sub>G</sub> dei vincoli organizzativi.

#### 2. Preparazione delle risposte

Realizzazione di eventuali contro-proposte basate sull'Analisi dei Requisiti<sub>G</sub> (AdR).

#### 3. Contrattazione

Confronto con la proponente per formalizzare requisiti, scadenze e modalità di lavoro.

#### 4. Pianificazione

Definizione dell'organizzazione, del ciclo di vita del software, assegnazione delle risorse e gestione dei rischi.

#### 5. Esecuzione e controllo

Implementazione delle attività pianificate, monitoraggio della qualità e del progresso del lavoro.

#### 6. Revisione e valutazione<sub>G</sub>

Raccolta di feedback dalla proponente per eventuali correzioni o aggiornamenti del lavoro.

#### 7. Consegna e completamento

Rilascio dei prodotti software e della documentazione, garantendo supporto post-consegna.

### 2.2.4) Procedure operative

Per garantire continuità e tracciabilità nelle comunicazioni, il gruppo utilizza modalità **sincrone e asincrone**:

#### 2.2.4.1) Metodi di comunicazione

**Comunicazione sincrona (meeting):**

- **Riunioni con la proponente (BlueWind S.r.l.):**

- Programmate periodicamente secondo le esigenze del progetto.
- Documentate tramite verbali esterni.
- Decisioni, richieste e dubbi vengono registrati come issue<sub>G</sub> su GitHub.

- **Riunioni interne al gruppo:**

- Programmate per coordinare attività e aggiornare lo stato del progetto.

- Documentate tramite verbali interni.
- Decisioni operative e compiti interni tracciati come issue<sub>G</sub> su GitHub.

### Comunicazione asincrona:

- **Con la proponente:** email ufficiale, messaggistica (Telegram) per chiarimenti rapidi.
- **Internamente al gruppo:** chat interne (Discord/WhatsApp) per coordinamento e aggiornamenti.

### Tracciamento

- Verbali esterni e interni;

### Disponibilità

Il gruppo si impegna a fornire aggiornamenti regolari e materiali richiesti entro le scadenze di sprint<sub>G</sub> e milestone<sub>G</sub>. Il processo di fornitura produce documentazione fondamentale per la tracciabilità e la trasparenza del progetto.

### 2.2.5) Documenti principali

- **Verbali esterni:** resoconto dei meeting con BlueWind;
- **Verbali interni:** documentazione di riunioni interne;
- **Norme di Progetto (NdP):** descrizione dei processi adottati e dell'organizzazione interna;
- **Piano di Progetto (PdP):** pianificazione delle attività, disponibilità delle risorse, avanzamento del lavoro;
- **Analisi dei Requisiti<sub>G</sub> (AdR):** raccolta e formalizzazione dei requisiti della proponente;
- **Piano di Qualifica<sub>G</sub> (PdQ):** criteri di verifica<sub>G</sub> e validazione<sub>G</sub>, test effettuati e risultati;
- **Glossario:** definizione dei termini tecnici e concetti chiave utilizzati nel progetto;
- **Dichiarazione degli Impegni:** stima dei costi del progetto, ore per ruolo e responsabilità dei componenti del gruppo;
- **Lettera di Presentazione:** presentazione formale del gruppo, fornisce una panoramica delle risorse consumate, rimanenti e metodologie usate;
- **Valutazione<sub>G</sub> dei Capitolati:** analisi dei capitolati disponibili, punti di forza, criticità e motivazioni della scelta effettuata dal gruppo.

### 2.2.6) Strumenti a supporto

Per svolgere le attività del processo di fornitura, il gruppo utilizza strumenti sia interni sia esterni:

#### Strumenti interni

- **GitHub**: gestione backlog, ticketing, pubblicazione dei documenti;
- **Google Calendar**: gestione appuntamenti e scadenze;
- **Discord / Whatsapp**: coordinamento interno e riunioni del gruppo.

#### Strumenti verso la proponente

- **Google Mail**: comunicazioni ufficiali scritte;
- **Zoom**: riunioni sincrone remote;
- **Telegram**: indicata dalla proponente per chiarimenti rapidi.

## 2.3) Processo di sviluppo

### 2.3.1) Introduzione

Il processo di sviluppo descrive l'insieme delle attività necessarie alla realizzazione del prodotto software, a partire dall'analisi dei requisiti<sub>G</sub> fino alla consegna e accettazione del sistema finale.

Secondo lo [Standard ISO 12207:2017](#), il processo di sviluppo comprende tutte le attività tecniche volte a trasformare i requisiti concordati con il committente in un **prodotto software funzionante, verificato e conforme agli obiettivi di qualità stabiliti**. Tali attività includono l'analisi dei requisiti<sub>G</sub>, la progettazione dell'architettura, la codifica, l'integrazione, il testing e la validazione<sub>G</sub> del sistema.

Nel contesto del progetto, il processo di sviluppo è adottato per garantire che ogni fase di realizzazione del software sia svolta in modo **sistematico, tracciabile e coerente** con le specifiche definite, assicurando la **qualità, l'affidabilità e la manutenibilità** del prodotto nel tempo.

### 2.3.2) Scopo del processo

Lo scopo del processo di sviluppo è garantire la corretta realizzazione del prodotto software in conformità<sub>G</sub> ai requisiti funzionali, di qualità e di vincolo definiti nel capitolato e nell'Analisi dei Requisiti<sub>G</sub>.

In particolare, il processo di sviluppo ha l'obiettivo di:

- **Trasformare i requisiti approvati in una soluzione software** progettata e implementata correttamente;
- **Assicurare la tracciabilità** tra requisiti, componenti progettuali, codice e test;

- Garantire che il software **soddisfi gli standard di qualità** stabiliti;
- Individuare e correggere tempestivamente eventuali difetti attraverso **attività di verifica<sub>G</sub> e validazione<sub>G</sub>**;
- Produrre un sistema software **affidabile, manutenibile e conforme alle aspettative** della proponente.

Il processo di sviluppo costituisce quindi il fulcro tecnico del progetto e guida in modo strutturato tutte le attività necessarie alla costruzione del prodotto finale.

### 2.3.3) Attività del processo

Il processo di sviluppo è articolato in un insieme di attività tra loro correlate, definite in conformità<sub>G</sub> allo **standard ISO/IEC 12207**, che guidano la realizzazione del prodotto software lungo l'intero ciclo di vita.

Le principali attività previste sono le seguenti:

#### 1. Raccolta e Analisi dei Requisiti<sub>G</sub>

Attività volta all'identificazione, analisi e formalizzazione dei requisiti funzionali, di qualità e di vincolo del sistema. I requisiti vengono raccolti a partire dal capitolato e dal confronto con la proponente, documentati nell'Analisi dei Requisiti<sub>G</sub> e resi tracciabili per le successive fasi di progettazione, implementazione e verifica<sub>G</sub>.

#### 2. Progettazione dell'architettura

Definizione della struttura generale del sistema, individuando le componenti principali, le loro responsabilità e le interazioni tra di esse, al fine di soddisfare i requisiti individuati.

#### 3. Progettazione di dettaglio

Definizione dettagliata delle singole componenti software e delle unità che le compongono, fornendo le informazioni necessarie alla fase di codifica.

#### 4. Codifica

Implementazione del software secondo quanto definito in fase di progettazione, adottando convenzioni di stile e buone pratiche di programmazione per garantire leggibilità, manutenibilità e qualità del codice.

#### 5. Test e Integrazione

Verifica<sub>G</sub> del corretto funzionamento tramite i test di sistema<sub>G</sub> e successiva integrazione delle componenti, accompagnata da test di integrazione<sub>G</sub> per individuare eventuali difetti.

#### 6. Installazione e supporto all'accettazione

Consegna del prodotto software nelle modalità concordate e supporto alla proponente nelle attività di accettazione, al fine di verificare il soddisfacimento dei requisiti contrattuali.

### 2.3.4) Inquadramento del processo nelle Baseline di progetto

Il processo di sviluppo è strettamente collegato alle baseline previste dal progetto:

#### Requirements and Technology Baseline (RTB)

Comprende principalmente le attività di:

- Analisi dei Requisiti<sub>G</sub>;
- Prime attività di codifica e prototipazione nella forma di Proof of Concept<sub>G</sub>.

#### Product Baseline<sub>G</sub> (PB)

Comprende principalmente le attività di:

- Progettazione dell'architettura software;
- Codifica completa del prodotto.

### 2.3.5) Procedure operative

Le attività del processo di sviluppo vengono svolte seguendo procedure operative standardizzate, al fine di garantire coerenza, tracciabilità e qualità del prodotto software.

Ogni procedura è supportata e documentata tramite i principali artefatti di progetto: Analisi dei Requisiti<sub>G</sub> (AdR), Piano di Progetto (PdP), Piano di Qualifica<sub>G</sub> (PdQ) e Norme di Progetto (NdP).

#### Analisi dei requisiti<sub>G</sub>

- Analizzare il capitolato e la documentazione fornita dalla proponente;
- Raccogliere eventuali chiarimenti tramite comunicazioni ufficiali;
- Identificare gli attori del sistema e le principali interazioni con il software;
- Individuare e descrivere i casi d'uso, specificando per ciascuno
  1. attori coinvolti;
  2. scenario principale<sub>G</sub>;
  3. scenari alternativi ed eccezioni;
- Assegnare a ogni caso d'uso un identificativo univoco secondo una nomenclatura coerente;
- Derivare dai casi d'uso e dal capitolato i requisiti del sistema;
- Classificare i requisiti in:
  1. requisiti funzionali;
  2. requisiti di qualità;
  3. requisiti di vincolo;
- Assegnare a ciascun requisito<sub>G</sub>:
  1. un identificativo univoco;
  2. una priorità (obbligatorio, desiderabile, opzionale);

- Formalizzare casi d'uso e requisiti nel documento di Analisi dei Requisiti<sub>G</sub> (AdR);
- Associare ogni requisito<sub>G</sub> a uno o più casi d'uso;
- Garantire la tracciabilità dei requisiti verso le successive attività di progettazione, codifica e verifica<sub>G</sub>, come riportato in AdR e PdQ.

### **Progettazione**

- Definire l'architettura generale del sistema sulla base dei requisiti approvati;
- Individuare le componenti software e le loro responsabilità;
- Definire le interfacce tra le componenti;
- Aggiornare la documentazione di progetto in base alle decisioni architetturali;
- Verificare la coerenza tra requisiti definiti in AdR e le scelte progettuali;
- Allineare le attività progettuali alla pianificazione definita nel Piano di Progetto (PdP).

### **Codifica**

- Implementare le funzionalità secondo la progettazione approvata;
- Applicare le convenzioni di stile e le buone pratiche definite nelle Norme di Progetto;
- Utilizzare il sistema di versionamento per la gestione del codice;
- Suddividere il lavoro in attività pianificate nel PdP e tracciate tramite issue<sub>G</sub>;
- Eseguire controlli statici, formattazione automatica e verifiche preliminari del codice;
- Garantire la tracciabilità tra codice implementato e requisiti definiti in AdR.

### **Verifica<sub>G</sub> e test**

- Definire i casi di test in conformità<sub>G</sub> al Piano di Qualifica<sub>G</sub> (PdQ);
- Eseguire test di unità<sub>G</sub> sulle singole componenti software;
- Eseguire test di integrazione<sub>G</sub> tra le componenti;
- Registrare l'esito dei test, le non conformità<sub>G</sub> e le eventuali correzioni nel PdQ;
- Verificare la copertura dei requisiti definiti in AdR attraverso i test eseguiti.

### **Integrazione e rilascio**

- Integrare progressivamente le componenti software secondo la pianificazione del PdP;
- Verificare il corretto funzionamento del sistema nel suo insieme;
- Eseguire i test di qualifica di sistema previsti dal PdQ;
- Preparare il materiale di rilascio e la documentazione associata;
- Effettuare la consegna secondo le modalità e le scadenze concordate con la proponente.

### **Gestione delle modifiche**

- Registrare le richieste di modifica come issue<sub>G</sub>;
- Valutare l'impatto delle modifiche sui requisiti (AdR) e sulla pianificazione (PdP);
- Aggiornare la documentazione di progetto e il codice;
- Eseguire nuovamente le verifiche previste dal PdQ;
- Garantire il mantenimento della tracciabilità tra requisiti, codice e test.

### 2.3.6) Documenti principali

Durante il processo di sviluppo, le attività sono supportate dalla documentazione di progetto già definita nel processo di fornitura (AdR, PdP, PdQ, NdP), con l'aggiunta di riferimenti specifici alle fasi di implementazione e verifica<sub>G</sub> software.

### 2.3.7) Strumenti a supporto

Per lo sviluppo del software, il gruppo utilizza strumenti mirati a garantire qualità, tracciabilità e collaborazione:

- **Linguaggio e ambiente di sviluppo:** Python 3.x come linguaggio principale per le componenti software.
- **Versionamento del codice:** Git/GitHub per gestione dei repository, branch<sub>G</sub>, commit, issue<sub>G</sub> e pull request<sub>G</sub>.
- **Formattazione e controllo del codice:** Il progetto adotta controlli automatici di qualità del codice e della documentazione tramite pipeline CI, includendo formattazione, verifica<sub>G</sub> di leggibilità (indice Gulpease) e generazione automatica della documentazione.
- **Gestione attività e tracciamento:** GitHub Issues per assegnazione, monitoraggio e gestione delle modifiche.
- **Comunicazione e collaborazione interna:** Discord o WhatsApp per coordinamento rapido, aggiornamenti sullo stato di avanzamento e chiarimenti tra membri del gruppo.
- **Comunicazione verso la proponente:** email ufficiale, Zoom per riunioni sincrone e Telegram per chiarimenti rapidi.

## 3) Processi di Supporto

### 3.1) Introduzione ai processi di supporto

I processi di supporto hanno lo scopo di garantire **l'efficace gestione, controllo e qualità delle attività del ciclo di vita del progetto<sub>G</sub>**, fornendo strumenti, procedure e linee guida per supportare i processi primari (analisi, progettazione, sviluppo, verifica<sub>G</sub>).

In conformità<sub>G</sub> agli standard ISO di riferimento, i processi di supporto considerati nel presente documento includono:

#### 1. Documentazione:

Processo volto a registrare, organizzare e rendere disponibili tutte le informazioni prodotte durante il ciclo di vita del progetto<sub>G</sub>, assicurando tracciabilità, trasparenza e coerenza dei contenuti.

#### 2. Gestione delle configurazioni:

Processo finalizzato a identificare, controllare, monitorare e aggiornare tutti gli elementi del progetto, garantendo la corretta gestione delle versioni e delle modifiche.

#### 3. Accertamento qualità:

Processo che comprende attività sistematiche di verifica<sub>G</sub> volte a controllare che processi, documenti e prodotti intermedi siano conformi ai requisiti, agli standard e alle metriche di qualità definiti durante lo svolgimento del progetto.

#### 4. Processo di Qualifica:

Processo che comprende attività di verifica<sub>G</sub> e validazione<sub>G</sub> finalizzate a stabilire se un prodotto o deliverable può essere considerato concluso e idoneo al rilascio, garantendo la conformità<sub>G</sub> alle specifiche tecniche e funzionali.

### 3.2) Processo di documentazione

#### 3.2.1) Introduzione

Il processo di documentazione definisce le **modalità con cui il team raccoglie, organizza e gestisce i documenti** prodotti durante il progetto.

Fornisce un quadro chiaro dei **flussi documentali, delle responsabilità e degli strumenti utilizzati**, garantendo che le informazioni siano aggiornate, accessibili e coerenti con gli obiettivi del progetto.

#### 3.2.2) Scopo del processo

Il processo di documentazione ha lo scopo di **registrare, organizzare e rendere disponibili tutte le informazioni** prodotte durante il ciclo di vita del progetto<sub>G</sub>.

Esso garantisce:

- **Tracciabilità** delle decisioni;
- **Coerenza** tra le attività svolte;

- **Trasparenza** verso gli stakeholder<sub>G</sub> e supporto del lavoro collaborativo del team.

La documentazione prodotta costituisce un **riferimento chiaro e accessibile** per tutte le fasi del progetto, riducendo ambiguità e fraintendimenti.

### 3.2.3) Attività del processo

#### 3.2.3.1) Impegno del gruppo

Il gruppo si impegna a fornire alla proponente e ai docenti tutta la documentazione necessaria a supportare le attività di analisi, progettazione, sviluppo e verifica<sub>G</sub> del progetto. Tale documentazione ha lo scopo di garantire **trasparenza, tracciabilità e qualità del lavoro** svolto, oltre a costituire un **riferimento chiaro per tutti gli stakeholder<sub>G</sub>** coinvolti.

#### 3.2.3.2) Attività principali

Le principali attività che compongono questo processo sono:

1. **Identificazione dei documenti**

Individuazione dei documenti necessari e definizione delle relative responsabilità.

2. **Progettazione dei documenti**

Applicazione delle norme e del workflow stabilito.

3. **Pubblicazione e Distribuzione della Documentazione**

Generazione del documento nel formato previsto e distribuzione ai destinatari autorizzati.

4. **Manutenzione<sub>G</sub>**

Aggiornamento continuo dei contenuti e gestione delle modifiche e delle revisioni.

5. **Archiviazione e tracciabilità**

Gestione del versionamento, conservazione dei documenti e garanzia dell'accessibilità nel tempo.

### 3.2.4) Procedure operative

#### 3.2.4.1) Identificazione dei documenti

Si tratta di un'attività di pianificazione in cui ogni documento viene definito secondo le seguenti caratteristiche principali:

- Titolo;
- Scopo;
- Destinatari;
- Procedure e responsabilità nella redazione e gestione del documento;
- Pianificazione per le versioni e i loro contenuti.

Per la redazione dei documenti è corretto e necessario fare riferimento a fonti autorevoli e aggiornate, quali standard ISO, università e organizzazioni ufficialmente riconosciute, materiale didattico e link di approfondimento forniti durante il corso. Eventuali informazioni provenienti da altre fonti, se ritenute utili, devono essere obbligatoriamente verificate per accertarne la correttezza e l'affidabilità.

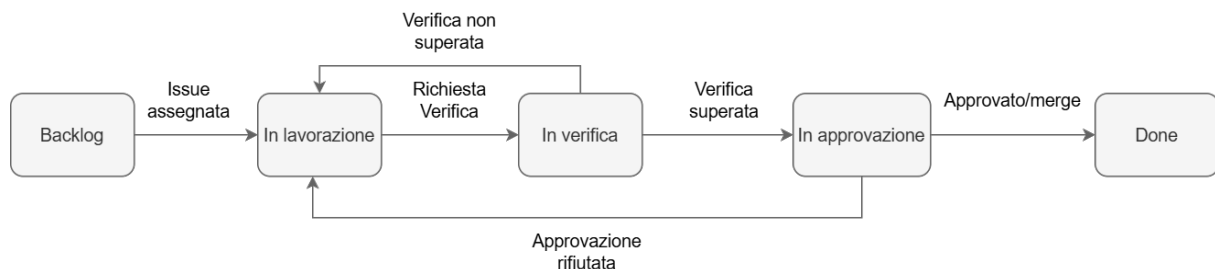
### 3.2.4.2) Progettazione dei documenti

Ogni documento identificato all'interno dello sviluppo software deve rispettare alcuni standard di documentazione uguali per tutti:

- Essere in formato A4;
- I contenuti inseriti devono essere coerenti con lo scopo del documento stesso;
- Tutti i documenti devono includere un indice dei contenuti e delle relative sotto-sezioni visibile all'inizio (ad eccezione del diario di bordo, che ne è esentato);
- Ogni pagina deve contenere nel header e nel footer:
  1. La sezione corrente del documento (in alto a sinistra);
  2. Il nome del gruppo (in alto a destra);
  3. Il titolo del documento (in basso a sinistra);
  4. Il numero della pagina : espresso in numeri romani per la prefazione e in numeri arabi per le pagine del corpo del documento.

### 3.2.4.3) Workflow documentale<sub>G</sub>

All'interno dell'ambito documentale si è optato per il seguente modello per descrivere e modellare le attività necessarie a produrre un documento:



### 3.2.4.4) Stati del documento

- **Backlog<sub>G</sub>**: magazzino delle attività da svolgere, ogni documento inizia in questo stato.
- **In lavorazione**: il documento è stato preso in carico dal componente del gruppo nel ruolo previsto.
- **In verifica<sub>G</sub>**: Il lavoro dell'autore è terminato. Il documento deve ora essere revisionato oppure corretto, nel caso in cui non sia stato approvato durante la fase di validazione<sub>G</sub>.
- **In approvazione**: il lavoro del revisore è finito. Il documento va valutato per l'approvazione oppure respinto, fornendo le opportune motivazioni accompagnate da un elenco delle correzioni da apportare.

- **Done:** il documento è stato approvato.

#### 3.2.4.5) Procedura di avanzamento tra stati

- Da **Backlog<sub>G</sub>** a **In lavorazione:** l'Amministratore crea la `issueG` predisponendo l'ambiente tecnico; contestualmente, in accordo con il Responsabile, viene determinato e impostato l'assegnatario (componente del gruppo nel ruolo pertinente) che prende in carico l'attività.
- Da **In lavorazione** a **In verifica<sub>G</sub>**: l'assegnatario dichiara conclusa la propria attività, trasferendo la `issueG` in revisione e assegnandola al Verificatore (deciso a priori) che verrà notificato automaticamente.
- Da **In verifica<sub>G</sub>** a **In lavorazione** (Rifiuto del Verificatore): il Verificatore rileva che non è conforme e riassegna la `issueG` all'esecutore con un report delle anomalie da sanare.
- Da **In verifica<sub>G</sub>** a **In approvazione:** il Verificatore accerta la correttezza dell'incremento e sposta la `issueG` in approvazione, assegnandola al Responsabile per il controllo finale di coerenza.
- Da **In approvazione** a **In lavorazione** (Rifiuto del Responsabile): il Responsabile rileva incongruenze di alto livello o mancanze strategiche. La `issueG` viene riassegnata all'assegnatario originale per la correzione, informando il Verificatore della svista.
- Da **In approvazione** a **Done:** il Responsabile accetta la revisione proposta e chiude la `issueG` con

```
git commit -m "commento. Close #numero_issue"
```

#### 3.2.4.6) Procedura di archiviazione e tracciabilità

I documenti sono salvati sull'apposito repository.

Il path relativo è ricavabile nel seguente modo:

(Fanno eccezione i diari di bordo, in quanto fanno parte delle regole di progetto, ma non supportano alcun processo primario, perciò sono salvati nella cartella `./src/DiariDiBordo`)

`./<Type>/<Baseline>/<Destinatari>/<Cartella del Documento>`

##### Type:

- **src** per i file in formato `typstG`.
- **output** per i `pdfG`.

##### Baseline:

- **RTB** per i documenti allo stato della RTB.
- **PB** per i documenti allo stato della PB.

##### Destinatari:

- `DocumentazioneInterna` per i documenti ad uso interno.

- Documentazione Esterna per i documenti ad uso esterno.

**Cartella del Documento:**

- Verbali Interni.
- Verbali Esterni.
- Per documenti complessi coincide con il nome del documento<sup>2</sup>.

**3.2.4.7) Pubblicazione e Distribuzione della Documentazione**

I documenti vengono ricompilati automaticamente in PDF<sub>G</sub> tramite GitHub Action e sono consultabili da tutti i membri del team. Sono disponibili nella repository del gruppo dedicata alla [Documentazione](#). La posizione ufficiale dei file è indicata nel README.md del repository.

Per facilitarne la consultazione è inoltre possibile accedere al [sito web ufficiale](#) del gruppo, creato appositamente per visualizzare e navigare i documenti in modo più immediato.

La versione del documento e la tracciabilità delle modifiche sono gestite tramite il Registro delle Modifiche, integrato direttamente all'interno di ciascun documento.

**3.2.5) Inquadramento del processo nelle Baseline di progetto**

La documentazione fornita in corrispondenza della fase RTB (Requirements and Technology Baseline) comprende sia materiali tecnici sia documenti operativi e organizzativi, al fine di garantire una valutazione<sub>G</sub> completa e trasparente dello stato del progetto.

In particolare, vengono prodotti:

**Documenti tecnici :**

- Analisi dei Requisiti<sub>G</sub> (AdR);
- Piano di Progetto (PdP);
- Piano di Qualifica<sub>G</sub> (PdQ);
- Preventivo dei Costi e delle risorse impiegate;

**Documenti operativi e organizzativi**

- Norme di Progetto (NdP);
- Verbali interni;
- Verbali esterni;
- Glossario;
- Diario di bordo.

---

<sup>2</sup>Per motivi di manutenibilità e facilità di aggiornamento i contenuti del file typst<sub>G</sub> sono stati divisi in più file quando una loro sezione diventa eccessivamente corposa.

### 3.2.6) Documentazione fornita

#### 3.2.6.1) Elenco dei documenti

Documento 1 Analisi dei Requisiti <sub>G</sub> .....	17
Documento 2 Piano di Qualifica <sub>G</sub> .....	17
Documento 3 Piano di Progetto .....	18
Documento 4 Verbali .....	19
Documento 5 Diario di Bordo .....	20
Documento 6 Norme di Progetto .....	20
Documento 7 Glossario .....	21

#### 3.2.6.2) Informazioni comuni

Ogni documento presenta una sezione iniziale standardizzata per tutti i membri del team. Questa sezione viene generata utilizzando un apposito template centrale e unico, al fine di garantire coerenza e facilitare la compilazione.

La sezione iniziale è composta dai seguenti elementi:

- 1. Pagina di copertina** : contenente il titolo del documento, il nome e il logo del gruppo e le relative informazioni di contatto.
- 2. Tabella dello stato** : che riassume lo stato del documento e informazioni generali quali versione, autori, verificatori, uso e destinatari.
- 3. Registro delle modifiche** : costituito da una tabella contenente le informazioni sul versionamento e sulla tracciabilità.
- 4. Indice dei contenuti** : aggiornato automaticamente tramite sintassi Typst<sub>G</sub>.
- 5. Indice delle immagini e delle tabelle** : presente solo nei documenti che ne contengono.

#### 3.2.6.3) Struttura specifica

Di seguito viene riportata la struttura standard dei documenti principali, le rispettive sezioni, il loro scopo, i destinatari, e le metodologie adottate per la scrittura e la revisione, al fine di mantenere coerenza e uniformità all'interno del gruppo.

### 3.2.6.4) Analisi dei Requisiti<sub>G</sub>

L'Analisi dei Requisiti<sub>G</sub> ha il compito di descrivere in modo completo, chiaro e verificabile tutte le **funzionalità** che il sistema deve offrire, includendo sia **requisiti funzionali** sia **non funzionali**.

Il documento fornisce inoltre i principali **casi d'uso, con attori e scenari associati**, e garantisce la **tracciabilità tra requisiti**, casi d'uso ed eventuali **estensioni future**. Rappresenta un riferimento stabile per sviluppatori, tester e manutentori durante tutte le fasi del progetto.

**Destinatari** : stakeholder<sub>G</sub> interni ed esterni al progetto (BlueWind S.r.l., docenti e gruppo interno)

#### 3.2.6.4.1) Struttura principale

Il documento comprende:

- Definizione formale dei requisiti funzionali e non funzionali;
- Modellazione dei casi d'uso con attori e flussi narrativi;
- Matrice di tracciabilità requisiti-casi d'uso;
- Eventuali vincoli tecnici, operativi o di contesto.

Documento 1: Analisi dei Requisiti<sub>G</sub>

### 3.2.6.5) Piano di Qualifica<sub>G</sub>

Il Piano di Qualifica<sub>G</sub> ha l'obiettivo di garantire che il prodotto sviluppato rispetti elevati standard di qualità. Definisce processi, metriche, strategie di testing e criteri di valutazione<sub>G</sub> necessari a verificare la qualità del software e del processo di sviluppo. Fornisce inoltre strumenti operativi per la misurazione e la validazione<sub>G</sub> dei risultati.

**Destinatari**: stakeholder<sub>G</sub> interni ed esterni al progetto (BlueWind S.r.l., docenti e gruppo interno)

#### 3.2.6.5.1) Struttura principale

Il documento è articolato nelle seguenti componenti:

- Metriche di qualità del prodotto e del processo;
- Strategie, livelli e tecniche di testing;
- Piano delle verifiche e validazioni;
- Cruscotto qualità con indicatori e soglie di accettazione.

Documento 2: Piano di Qualifica<sub>G</sub>

### 3.2.6.6) Piano di Progetto

Il Piano di Progetto definisce la pianificazione complessiva delle attività, descrivendo l'approccio adottato dal gruppo.

Il documento fornisce una visione strutturata dell'**organizzazione del lavoro**, includendo la definizione degli obiettivi, la gestione delle risorse, l'assegnazione dei ruoli, la pianificazione temporale e l'analisi dei rischi.

La sua funzione principale è garantire un monitoraggio costante dell'**avanzamento** del progetto attraverso **revisioni periodiche** e **rendicontazioni** relative ai vari sprint<sub>G</sub>.

Tale monitoraggio consente al gruppo di valutare l'efficienza del workflow, individuare tempestivamente eventuali criticità e adattare la pianificazione quando necessario.

**Destinatari** : stakeholder<sub>G</sub> interni ed esterni al progetto (BlueWind S.r.l., docenti e gruppo interno)

#### 3.2.6.6.1) Struttura principale

La struttura del documento comprende:

- Ambito e obiettivi del progetto;
- Analisi dei rischi e piani di mitigazione;
- Preventivo iniziale e disponibilità delle risorse;
- Pianificazione di lungo periodo;
- Sezione dedicata alle revisioni, che per ogni sprint<sub>G</sub> comprende:
  - **Attività pianificate** → obiettivi e task previsti per il periodo di sprint<sub>G</sub>;
  - **Gestione dei rischi** → analisi proattiva dei rischi attesi e dei rischi emersi, con indicazione delle relative strategie di mitigazione e piani di contingenza;
  - **Preventivo ore per ruolo** → stima dell'effort pianificato, suddiviso per responsabilità;
  - **Retrospektiva del gruppo** → riflessioni su apprendimento, workflow ed efficacia della collaborazione;
  - **Consuntivo ore effettive** → rendicontazione delle ore realmente impiegate dal gruppo nello sprint<sub>G</sub> per aggiornare il budget residuo.

Documento 3: Piano di Progetto

### 3.2.6.7) Verbali

I verbali sono suddivisi in due categorie principali :

- Verbali interni -> documentano riflessioni e confronti avvenuti esclusivamente tra i membri del gruppo.
- Verbali esterni -> vengono redatti in corrispondenza di riunioni o confronti con l'azienda di riferimento (Bluewind).

Ogni verbale si conclude con una **riflessione finale del gruppo**, dalla quale emergono decisioni operative che vengono successivamente formalizzate tramite la creazione di **issue<sub>G</sub> GitHub**, che il gruppo si impegna a completare.

#### 3.2.6.7.1) Struttura principale

Ogni verbale deve avere la seguente suddivisione numerata:

1. **Informazioni comuni della Sezione 3.2.6.2** (standard condivisi di documento)
2. **Informazioni generali**
  - Data e luogo della riunione;
  - Orario di inizio/fine;
  - Partecipanti;
  - Tipologia (interno/esterno);
  - Motivo (principalmente per verbali esterni);
  - Scriba (responsabile del verbale in quel momento).
3. **Ordine del giorno** : scaletta dei temi da discutere, raccolti e organizzati del responsabile sulla base dei contributi dei membri del gruppo o dei referenti aziendali.
4. **Riassunto della riunione** : sintesi breve e oggettiva dei punti discussi.
5. **Decisioni** : azioni o obiettivi (anche ad alto livello) che il gruppo deve intraprendere per dare seguito alla riunione.
6. **TODO** : attività specifiche derivate dalle decisioni. Una singola decisione può essere suddivisa in più TODO, che complessivamente consentono di raggiungere l'obiettivo stabilito.

Documento 4: Verbali

### 3.2.6.8) Diario di Bordo

Il Diario di Bordo è un'attività prevista dal Prof. Tullio Vardanega all'interno del progetto di Ingegneria del Software. Rappresenta un momento di condivisione in cui ciascun gruppo espone il proprio stato di avanzamento, con particolare attenzione a dubbi o problematiche emerse durante lo svolgimento delle attività.

Composto principalmente da:

**Titolo:** Diario di Bordo seguito dal numero progressivo associato.

**Scopo:** Fornire ai gruppi un feedback sulle attività svolte e consentire di portare all'attenzione comune eventuali dubbi relativi al processo di lavoro.

**Destinatari:** Prof. Tullio Vardanega e gli altri gruppi coinvolti nel progetto.

Documento 5: Diario di Bordo

### 3.2.6.9) Norme di Progetto

Le Norme di Progetto definiscono l'insieme di regole, convenzioni e standard adottati dal gruppo al fine di garantire coerenza, qualità e uniformità nella produzione della documentazione, del codice e dei deliverable. Il documento stabilisce inoltre procedure condivise per redazione, revisione, versionamento, gestione dei file e comunicazione interne, riducendo il rischio di errori, fraintendimenti o incoerenze operative tra i membri del team.

**Destinatari :** Tutti i membri del gruppo di progetto (interno)

#### 3.2.6.9.1) Struttura principale

Il documento delle Norme di Progetto è organizzato secondo i tipi di processo presenti nel progetto:

- **Processi primari:** attività direttamente legate alla realizzazione del prodotto software.
- **Processi di supporto:** attività che garantiscono qualità, tracciabilità e gestione della documentazione.
- **Processi organizzativi:** attività relative alla gestione del team, pianificazione e coordinamento.

Documento 6: Norme di Progetto

### 3.2.6.10) Glossario

Il Glossario ha l'obiettivo di garantire **chiarezza e uniformità nella terminologia** utilizzata nei documenti di progetto, sia verso l'esterno (proponente, docenti) sia all'interno del gruppo di lavoro.

Raccoglie i **termini ritenuti non banali, abbreviazioni e acronimi**, fornendo definizioni precise per ridurre ambiguità interpretative e favorire una comprensione condivisa dei concetti chiave.

Il documento è soggetto ad **aggiornamento continuo**, in modo da riflettere l'evoluzione del progetto e mantenere allineata la terminologia. Oltre alla versione ufficiale inclusa nei documenti di progetto, il team mantiene un glossario interno aggiornato disponibile sul sito web della documentazione, che funge da riferimento centralizzato sempre accessibile.

Per **favorire la tracciabilità e la consultazione immediata**, all'interno dei singoli documenti tutte le parole presenti nel glossario vengono evidenziate. Questa evidenziazione consente agli utenti di riconoscere rapidamente i termini definiti formalmente e di rimandare al glossario in caso di dubbi o possibili ambiguità.

#### Documento 7: Glossario

### 3.2.7) Strumenti di supporto

**Typst<sub>G</sub>**: Linguaggio di markup moderno per la composizione e la tipografia di documenti, pensato come alternativa più semplice e veloce a LaTeX.

- Sintassi intuitiva;
- Supporto ad automazioni tramite template, funzioni e regole di stile riutilizzabili;
- Preview istantanea del documento.

**GitHub**: Strumento scelto dal gruppo per la condivisione del lavoro e la gestione delle attività tramite **issue<sub>G</sub> tracking**.

- Utilizzo di GitHub Actions<sub>G</sub> per la compilazione automatica dei documenti.
- Documentazione disponibile nel repository [GitHub](#).
- [Sito web](#) predisposto tramite GitHub Pages per facilitare la consultazione della documentazione.

**TurboScribe AI<sub>G</sub>**: Per velocizzare il processo di documentazione, il gruppo utilizza il tool di intelligenza artificiale [TurboScribe](#), che consente di trascrivere registrazioni audio in formato testuale. Lo strumento è utilizzato in particolare per il supporto alla redazione dei verbali, permettendo il riascolto rapido di momenti specifici delle riunioni tramite selezione testuale.

### 3.3) Processo di Gestione delle Configurazioni

#### 3.3.1) Introduzione

Il processo di Gestione delle Configurazioni ha lo scopo di garantire la corretta identificazione, controllo, monitoraggio e aggiornamento di tutti gli elementi del progetto, siano essi documenti, codice, deliverable o altri artefatti di progetto.

Questo processo assicura coerenza, tracciabilità e qualità lungo tutto il ciclo di vita del progetto<sub>G</sub>, supportando le attività dei processi primari (sviluppo, fornitura) e riducendo rischi di incongruenze o perdita di informazioni.

#### 3.3.2) Scopo del processo

Il processo di gestione delle configurazioni mira a:

1. Garantire la **tracciabilità completa delle modifiche** apportate a documenti, codice e artefatti.
2. **Controllare e gestire le versioni** in modo chiaro e uniforme.
3. Fornire un **registro delle modifiche** consultabile da tutti i membri del gruppo.
4. **Supportare la revisione, approvazione e validazione<sub>G</sub>** dei deliverable.
5. **Ridurre ambiguità e conflitti** dovuti a modifiche non controllate.

#### 3.3.3) Attività del processo

Le principali attività della Gestione delle Configurazioni includono:

1. Identificazione degli elementi di configurazione.
2. Applicazione del codice di versione X.Y.Z (stabile, feature, patch).
3. Aggiornamento automatico della versione in tutti i documenti tramite Typst<sub>G</sub>/GitHub.
4. Inserimento di ogni modifica nel Registro delle Modifiche, comprensivo di: data, autore, verificatore, descrizione e nuova versione.
5. Verifica<sub>G</sub> delle modifiche da parte dei revisori.
6. Approvazione e rilascio di versioni stabili.
7. Conservazione dei documenti e del codice in repository centralizzati.
8. Backup per garantire sicurezza e recuperabilità dei dati.

#### 3.3.4) Procedure operative

##### 3.3.4.1) Codice di versione

Ogni modifica apportata a un documento genera automaticamente una nuova versione, identificata tramite un codice nel formato:

**X.Y.Z**

dove ciascuna componente rappresenta uno stato diverso del processo di validazione<sub>G</sub>:

- **X – Versione stabile approvata**

Indica l'ultima versione ufficialmente approvata dal Responsabile. Il suo incremento segnala una revisione sostanziale o una modifica di grande rilievo. L'incremento di X comporta l'azzeramento automatico di Y e Z.

- **Y – Feature**

Rappresenta l'ultima approvazione da parte di un Verificatore. Il suo incremento indica l'introduzione o modifica di una nuova funzionalità o sezione del documento. L'incremento di Y comporta l'azzeramento di Z.

- **Z – Patch**

Indica l'ultima modifica di dettaglio verificata (correzioni minori, refusi, aggiustamenti formali). L'incremento di Z rappresenta cambiamenti minori.

### 3.3.4.1.1) Regole di incremento

1. Ogni approvazione genera un incremento della cifra di versione stabile.
2. Nel versionamento X.Y.Z, maggiore è il cambiamento, più significativa è la cifra che viene incrementata: X ha un peso maggiore di Y, e Y maggiore di Z
3. L'incremento di una cifra comporta sempre l'azzeramento delle cifre alla sua destra, mantenendo coerenza nella progressione delle versioni.

### 3.3.4.2) Versionamento automatico tramite Typst<sub>G</sub>

Il processo di gestione delle configurazioni utilizza uno snippet Typst<sub>G</sub> per gestire automaticamente la versione corrente del documento e collegarla ai metadati. Lo snippet è il seguente:

```
#let versionNumber=currentVersion.values().map(n=>{str(n)}).join(".")
#metadata(versionNumber)<versionNumber>
#let doc="Norme di Progetto"
#let versionNumber=currentVersion.values().map(n=>{str(n)}).join(".")
```

- Converte le componenti numeriche della versione (X, Y, Z) in stringhe e le unisce nel formato X.Y.Z. Il risultato viene salvato nella variabile versionNumber, che rappresenta la versione corrente del documento.
- Inserisce la versione corrente nei metadati del documento.
- Permette di rendere il numero di versione accessibile in tutto il documento, per esempio in header, footer o tabella del registro modifiche, senza doverlo aggiornare manualmente.
- Definisce il nome del documento come variabile, utile per riferimenti automatici all'interno del registro modifiche o per generare link e riferimenti interni.

### 3.3.4.3) Registro modifiche

Il registro modifiche è la tabella principale presente in ogni documento che tiene traccia della versione, degli autori, dei verificatori e delle modifiche effettuate. Ogni voce del registro include:

1. **Vers.** – Versione corrente del documento (X.Y.Z);
2. **Data** – Data di aggiornamento della versione;
3. **Autore** – Persona che ha redatto o modificato il documento;
4. **Verificatore** – Persona che ha revisionato o approvato la modifica;
5. **Descrizione** – Sintesi delle modifiche apportate.

Vers.	Data	Autore	Verificatore	Descrizione
0.9.0	2026-01-10	Nome di chi ha modificato	Nome di chi ha verificato	Stesura Sezione 4.2, Gestione dell'infrastruttura

### 3.3.4.4) Formato nome dei verbali

Al fine di avere ordine cronologico all'interno della cartella dei verbali, è stato deciso di adottare il seguente standard per la nomina dei verbali. Di questi documenti interessa data e versione, dunque saranno nel formato:

YYYY-MM-DD\_Verbale-vX.Y.Z.typ

### 3.3.5) Strumenti principali utilizzati

Per la gestione delle configurazioni, delle versioni e della documentazione il gruppo utilizza i seguenti strumenti:

- **Typst<sub>G</sub>**: compilazione dei documenti, template riutilizzabili, gestione dei metadati e preview istantanea.
- **GitHub**: repository centralizzato, controllo versioni, gestione issue<sub>G</sub> e collaborazione asincrona.
- **GitHub Actions<sub>G</sub>**: compilazione automatica dei documenti in PDF<sub>G</sub> e aggiornamento della versione corrente.
- **Registro delle Modifiche** (integrato nei documenti tramite Typst<sub>G</sub>): tiene traccia di autori, verificatori, date e descrizione di ogni modifica.

## 3.4) Processo di Accertamento della Qualità

### 3.4.1) Introduzione

Il processo di Accertamento Qualità ha il compito di verificare che documenti, deliverable e processi siano conformi agli standard, alle procedure e alle metriche definite dal progetto. Tale processo accompagna l'intero ciclo di vita del progetto<sub>G</sub> e fornisce un controllo continuo sulla qualità del lavoro svolto.

### 3.4.2) Scopo del processo

Assicurare che il lavoro del team sia:

- **Corretto e coerente** con le specifiche definite;
- **Conforme agli standard e alle metriche** di qualità stabilite;
- **Tracciabile e verificabile**;
- Adeguatamente **preparato per le successive attività** di Qualifica.

### 3.4.3) Attività del processo

- Revisione della documentazione e dei deliverable;
- Verifica<sub>G</sub> del rispetto delle Norme di Progetto e degli standard adottati;
- Individuazione e segnalazione di eventuali non conformità<sub>G</sub>;
- Raccolta e archiviazione delle evidenze di controllo.

### 3.4.4) Procedure operative

#### 3.4.4.1) Scrittura dei commit

I commit devono avere un **tipo** ed una **descrizione**:

- Il **tipo** indica qual è l'obiettivo del commit (ad esempio feat, fix, docs, ecc.);
- La **descrizione** aiuta il lettore a comprendere meglio quali cambiamenti sono stati effettuati.

Le regole generali sono:

- Iniziare il commit con il tipo seguito da :
- Lasciare uno spazio tra tipo e descrizione
- Iniziare la descrizione con lettera maiuscola
- Limitare la descrizione a massimo 50 caratteri
- Indicare alla fine del commit la issue<sub>G</sub> a cui ci si sta riferendo

```
git commit -m "tipo: descrizione. IssueG #1 "
```

Nel caso sia necessario modificare un commit (ad esempio in caso di errori) si utilizza il seguente comando<sup>3</sup>:

```
git commit --amend
```

---

<sup>3</sup>È consigliato l'utilizzo del comando per modificare commit in locale prima di fare push nella repository condivisa. È preferibile astenersi dal modificare commit che sono già stati resi pubblici.

### 3.4.5) Strumenti

- **Typst<sub>G</sub>** → Calcolo automatico di alcune metriche principali (es. indice Gulpease) tramite script Typst<sub>G</sub>;
- **GitHub** → repository condiviso, issue<sub>G</sub> tracking e versionamento delle modifiche.
- **Automazioni** → script e workflow per calcolare metriche di qualità, generare report automatici, notificare approvazioni.

### 3.4.6) Metriche

Le metriche relative all'Accertamento Qualità sono riportate nella sezione **Sezione 5.1** "Metriche di Qualità" del documento, con valori accettabili e ottimali.

Oppure consultare il seguente documento per approfondimenti : [Piano di Qualifica<sub>G</sub> \(PdQ\)](#).

## 3.5) Processo di Qualifica

### 3.5.1) Introduzione

Il processo di Qualifica coincide con l'insieme delle attività di Verifica<sub>G</sub> e Validazione<sub>G</sub> pianificate nel progetto. Il suo scopo è dimostrare, tramite evidenze oggettive e misurabili, che il prodotto sia conforme alle specifiche software (Verifica<sub>G</sub>) e alle attese dell'utente (Validazione<sub>G</sub>). In termini operativi, la Qualifica non è solo una fase finale, ma un processo continuo regolato dal Piano di Qualifica<sub>G</sub>, che definisce gli obiettivi quantitativi, e monitorato attraverso un cruscotto di valutazione<sub>G</sub>.

### 3.5.2) Scopo del processo

- Assicurare la **conformità<sub>G</sub> dei prodotti** alle specifiche tecniche e funzionali.
- **Ridurre errori e incoerenze** attraverso attività di verifica<sub>G</sub> e validazione<sub>G</sub>.
- Documentare in modo chiaro l'esito delle **attività di verifica<sub>G</sub> e validazione<sub>G</sub>**.
- Definire criteri oggettivi di completamento tramite la **Definition of Done<sub>G</sub> (DoD)**.

### 3.5.3) Attività del processo

Le attività principali del processo di Qualifica comprendono:

- **Revisione finale dei deliverable** rispetto ai requisiti e agli standard;
- **Validazione<sub>G</sub>** dei documenti e dei prodotti software;
- **Verifica<sub>G</sub> del soddisfacimento** dei criteri di completamento;
- **Approvazione finale** dei deliverable qualificati.

### 3.5.3.1) Analisi statica

L'analisi statica è una tecnica di verifica<sub>G</sub> applicata senza eseguire il codice o il prodotto. Il suo scopo è individuare problemi di sintassi, logica o conformità<sub>G</sub> agli standard prima che si manifestino durante l'esecuzione.

Il gruppo adotta due metodi di lettura a seconda del contesto:

- **Walkthrough:** revisione libera e approfondita dell'intero artefatto, condotta senza una lista di controllo predefinita. Parte dall'ipotesi che esista un difetto, ma senza conoscerne la natura o la posizione. È indicata per revisioni critiche o per artefatti nuovi, dove non è ancora disponibile esperienza pregressa sugli errori tipici. A causa dell'elevato costo in termini di tempo e risorse, viene applicata selettivamente, ad esempio in occasione delle revisioni RTB e PB o per la prima stesura di documenti strutturali.
- **Ispezione:** verifica<sub>G</sub> guidata da una checklist predefinita, focalizzata su errori ricorrenti e criteri di qualità specifici. È meno approfondita del walkthrough, ma facilmente automatizzabile e ripetibile. Viene adottata come metodo principale per la verifica<sub>G</sub> sistematica di documenti e codice, in particolare per verbali, analisi dei requisiti<sub>G</sub> e sezioni già consolidate. Un esempio di checklist adottata è la **Sezione 3.5.3.5** (DOD).

#### 3.5.3.1.1) Checklist di ispezione

La checklist utilizzata dal gruppo copre i seguenti aspetti:

- **Documenti:** ortografia e grammatica, rispetto del template, presenza e correttezza del registro delle modifiche, coerenza dei riferimenti interni, nomenclatura conforme alle Norme di Progetto.
- **Codice:** rispetto delle convenzioni di nomenclatura, assenza di codice morto o commentato, presenza di commenti significativi, gestione degli errori, conformità<sub>G</sub> alle linee guida del linguaggio adottato.

### 3.5.3.1.2) Quando usare quale metodo

Contesto	Metodo	Motivazione
Prima stesura di un documento critico	Walkthrough	Nessuna checklist disponibile, alto rischio di errori strutturali
Verifica <sub>G</sub> verbali e documenti ricorrenti	Ispezione	Formato standardizzato, errori prevedibili
Revisione del codice in pull request <sub>G</sub>	Ispezione	Automatizzabile, ripetibile, integrata nel workflow Git
Revisione pre-RTB / pre-PB	Walkthrough	Qualità critica, revisione completa necessaria

Tabella 1: Metodologie di testing

### 3.5.3.2) Analisi dinamica

L'analisi dinamica verifica<sub>G</sub> il software eseguendolo con test specifici, al fine di misurarne la qualità funzionale e individuare errori nel comportamento a runtime. Ogni test è definito da uno stato iniziale, un insieme di input e gli output attesi, e deve produrre risultati riproducibili indipendentemente da chi lo esegue o quando.

I test devono essere ripetibili e automatizzabili, così da poter essere eseguiti in modo continuo durante l'intero ciclo di vita del prodotto, in particolare ad ogni modifica rilevante del codice.

### 3.5.3.3) Classificazione dei test

Tutti i test pianificati, eseguiti e i loro esiti sono tracciati nel Piano di Qualifica<sub>G</sub>. I test sono classificati in categorie gerarchiche, ciascuna identificata da un codice univoco nel formato:

*TipoTest\_XX*

dove **TipoTest** indica la categoria e **XX** è un numero progressivo che identifica univocamente il test all'interno di essa.

Le categorie previste sono:

Codice	Descrizione
TU	Test di Unità <sub>G</sub> – verifica <sub>G</sub> il corretto funzionamento di una singola unità software in isolamento
TI	Test di Integrazione <sub>G</sub> – verifica <sub>G</sub> l'interazione corretta tra più unità o componenti
TS	Test di Sistema <sub>G</sub> – verifica <sub>G</sub> il comportamento del sistema nella sua interezza rispetto ai requisiti
TA	Test di Accettazione <sub>G</sub> – verifica <sub>G</sub> che il prodotto soddisfi i criteri concordati con il committente

Tabella 2: Tipi di test

### 3.5.3.3.1) Stati dei test

Per consentire un monitoraggio efficace dell'avanzamento, ogni test assume uno dei seguenti stati:

Stato	Descrizione
S	Superato – il test è stato eseguito e ha prodotto l'output atteso
I	Implementato – il test è stato scritto ma non ancora eseguito
NI	Non Implementato – il test è pianificato ma non ancora realizzato
F	Fallito – il test ha prodotto un risultato diverso da quello atteso

Tabella 3: Stati dei test

### 3.5.3.3.2) Test di Unità<sub>G</sub>

I test di unità<sub>G</sub> rappresentano il livello di granularità più basso e verificano il corretto funzionamento delle singole unità software nella loro forma più elementare: funzioni, metodi o classi.

Si distinguono due approcci:

- **Test funzionali:** analizzano il comportamento dell'unità osservando esclusivamente input e output, senza considerare la logica interna di implementazione. Poiché non esaminano i percorsi di esecuzione interni, non sono sufficienti da soli a garantire la correttezza dell'unità e devono essere affiancati da test strutturali.
- **Test strutturali:** analizzano la logica interna del codice, verificandone i diversi percorsi di esecuzione e la relativa copertura. Consentono di individuare rami non raggiungibili, condizioni mal gestite o percorsi di errore non testati.

I due approcci sono complementari: i test funzionali verificano il contratto esterno dell'unità, quelli strutturali ne garantiscono la solidità interna.

### 3.5.3.3.3) Test di Integrazione<sub>G</sub>

I test di integrazione<sub>G</sub> verificano che le componenti del sistema, già testate singolarmente, interagiscano e comunichino correttamente tra loro. L'obiettivo è individuare errori che emergono solo dalla combinazione di moduli: incompatibilità di interfacce, gestione scorretta dei dati condivisi o comportamenti inattesi nelle sequenze di chiamata.

Il gruppo adotta uno dei due approcci seguenti a seconda della struttura del componente da integrare:

Approccio	Descrizione	Quando usarlo
Top-Down	Si parte dalle componenti di livello superiore, utilizzando stub per simulare quelle non ancora integrate	Quando la logica di controllo ad alto livello è prioritaria o le componenti di base non sono ancora disponibili
Bottom-Up	Si parte dalle componenti di base, utilizzando driver per simulare le chiamate dei livelli superiori	Quando le componenti fondamentali sono stabili e si vuole verificarne il comportamento prima di procedere verso l'alto

Tabella 4: Tipi di test d integrazione

### 3.5.3.3.4) Test di Sistema<sub>G</sub>

I Test di Sistema<sub>G</sub> verificano il comportamento del sistema nella sua interezza, valutandone la conformità<sub>G</sub> rispetto ai requisiti funzionali e non funzionali definiti nel documento di Analisi dei Requisiti<sub>G</sub>. Consentono di validare aspetti quali la correttezza funzionale, l'affidabilità, la robustezza e la gestione degli errori in condizioni reali o simulate di utilizzo.

### 3.5.3.3.5) Test di Accettazione<sub>G</sub>

I Test di Accettazione<sub>G</sub> verificano che il sistema soddisfi le aspettative e i requisiti richiesti dalla proponente BlueWind Srl. Vengono eseguiti al termine del ciclo di sviluppo e coinvolgono direttamente la proponente nella valutazione<sub>G</sub> del prodotto consegnato. Il loro esito positivo costituisce condizione necessaria per la validazione<sub>G</sub> finale.

### 3.5.3.3.6) Test di Regressione

I Test di Regressione verificano che le modifiche apportate al sistema — siano esse correzioni di difetti o aggiunta di nuove funzionalità — non abbiano compromesso comportamenti precedentemente testati e corretti.

In tal caso il processo da seguire è:

1. Analizzare il problema e identificarne la causa;

2. Sviluppare e codificare la soluzione;
3. Rieseguire il test fallito per verificare che il problema sia stato risolto;
4. Rieseguire **l'intera suite di test** per assicurarsi che la correzione non abbia introdotto regressioni.

Quest'ultimo punto è fondamentale: una modifica apparentemente localizzata può avere effetti inattesi su funzionalità già verificate. Per questo motivo non ci si limita al solo test fallito, ma si rieseguono tutti i test disponibili.

#### 3.5.3.4) Validazione<sub>G</sub>

Il processo di Validazione<sub>G</sub> ha lo scopo di accertare che quanto realizzato soddisfi le esigenze di BlueWind Srl. Si distingue dalla verifica<sub>G</sub> per il suo focus sul risultato finale: mentre la verifica<sub>G</sub> accerta che il prodotto sia costruito correttamente («*Are we building the system right?*»), la validazione<sub>G</sub> accerta che sia stato costruito il prodotto corretto («*Are we building the right system?*»). Dunque, pur avendo responsabilità diverse, la verifica<sub>G</sub> prepara il successo della validazione<sub>G</sub>. Sebbene la verifica<sub>G</sub> si applichi ai singoli prodotti intermedi e la validazione<sub>G</sub> al sistema nel suo complesso, la prima fornisce il supporto e l'evidenza<sub>G</sub> oggettiva necessari per sostenere la successiva conclusione che il software sia validato e pienamente conforme alle attese del committente.

##### 3.5.3.4.1) Attività di validazione<sub>G</sub>

La Validazione<sub>G</sub> si basa sull'analisi degli esiti dei test di accettazione<sub>G</sub> e sul tracciamento dei requisiti definiti in accordo con BlueWind Srl. L'obiettivo è confermare che ogni requisito<sub>G</sub> obbligatorio sia implementato e che il sistema si comporti correttamente in relazione a ciascuno di essi.

##### 3.5.3.5) Definition of Done<sub>G</sub>

La **Definition of Done<sub>G</sub> (DoD)** è un elemento molto importante nello sviluppo software, perché definisce le azioni che devono essere completate affinché i requisiti — espressi tramite un **Product Backlog<sub>G</sub> Item (PBI)** — siano considerati conclusi.

I criteri che la compongono devono essere concreti, verificabili e di dimensione ridotta, e hanno l'obiettivo di garantire un livello minimo di qualità per ogni rilascio o incremento del prodotto.

Di seguito viene riportata la Definition of Done<sub>G</sub> per ogni prodotto documentale:

- Controllare a livello semantico e grammaticale che tutto sia corretto (grammatica, punteggiatura, sintassi, rivedere frasi ripetute/ mal espresse);
- Controllare di aver incluso tutte le sezioni definite del WoW nel documento su cui si lavora;
- Controllare di aver aggiornato nella tabella riepilogativa dello stato del documento:
  - stato;
  - versione;

- ruoli.
- Controllare di aver aggiunto le ultime modifiche anche sulla “tabella delle modifiche del documento”;
  - Nei verbali**, controllare che tutte le decisioni corrispondano a issue<sub>G</sub> specifiche nell’issue<sub>G</sub> template;
  - L’avvenuto completamento della attività documentale deve essere sancito tramite un commit di chiusura che referenzi la issue<sub>G</sub> corrispondente con

```
git commit -m "commento. Close #numero_issue"
```

Verificare poi effettivamente la chiusura nel Projects Board.

Una volta verificati tutti i criteri precedenti, il Responsabile approva il lavoro svolto spostando le relative issue<sub>G</sub> nello stato di “Done”. Solo quando tutti i PBI previsti per lo Sprint<sub>G</sub> risultano completati, il Responsabile autorizza il merge<sub>G</sub> del branch<sub>G</sub> develop nel branch<sub>G</sub> main.

La seguente **Definition of Done<sub>G</sub>** non è statica, ma dinamica: evolve in base alle esigenze del team di sviluppo.

#### 3.5.4) Strumenti a supporto

- **Typst<sub>G</sub>** → gestione dei documenti, versionamento e inserimento automatico di metadati.
- **GitHub** → repository condiviso, issue<sub>G</sub> tracking e tracciamento delle modifiche.
- **Automazioni** → script e workflow per calcolare metriche di qualità, generare report automatici, notificare approvazioni.

#### 3.5.5) Documentazione a supporto

I processi di verifica<sub>G</sub> e validazione<sub>G</sub> si appoggiano ai seguenti documenti:

- **Analisi dei Requisiti<sub>G</sub>**: definisce i requisiti funzionali e non funzionali concordati con BlueWind Srl. Costituisce la base di riferimento per la progettazione dei test di sistema<sub>G</sub> e di accettazione, e per il tracciamento della copertura dei requisiti.

[Riferimento all’Analisi dei Requisiti<sub>G</sub>.](#)

- **Piano di Qualifica<sub>G</sub>**: raccoglie le metriche di qualità adottate, i test pianificati ed eseguiti e i loro esiti. È il documento operativo di riferimento per il monitoraggio dell’avanzamento delle attività di verifica<sub>G</sub>.

[Riferimento al Piano di Qualifica<sub>G</sub>](#)

- **Piano di Progetto:** definisce la pianificazione temporale delle attività, incluse quelle di verifica<sub>G</sub>. Consente di contestualizzare i risultati dei test rispetto agli sprint<sub>G</sub> in cui sono stati eseguiti.

[Riferimento al Piano di Progetto.](#)

- **Norme di Progetto:** definisce le modalità operative di verifica<sub>G</sub> e validazione<sub>G</sub> adottate dal gruppo, incluse le checklist di ispezione, la classificazione dei test e i criteri di accettazione.

[Riferimento alle Norme di Progetto.](#)

- **Verbali esterni:** documentano le decisioni e i requisiti concordati con BlueWind Srl nel corso delle riunioni. Costituiscono riferimento per verificare la corrispondenza tra quanto richiesto e quanto implementato.

[Riferimento ai Verbali esterni](#)

## 4) Processi Organizzativi

### 4.1) Introduzione

I processi organizzativi definiscono il quadro operativo entro cui il progetto viene pianificato, gestito e migliorato nel tempo. Essi supportano i processi primari e di supporto garantendo **coerenza, continuità operativa e qualità complessiva**, in accordo con gli standard di riferimento per il ciclo di vita del software.

Nel presente progetto sono stati individuati i seguenti processi organizzativi:

#### 1. Processo di gestione del processo

Stabilisce e governa le modalità operative del progetto, definendo ruoli, responsabilità, procedure e regole condivise per l'esecuzione delle attività;

#### 2. Processo di gestione dell'infrastruttura

Assicura la disponibilità, l'adeguatezza e il corretto utilizzo degli strumenti software e delle risorse tecnologiche necessarie allo svolgimento del progetto;

#### 3. Processo di miglioramento

Volto a valutare periodicamente l'efficacia dei processi adottati e ad apportare azioni correttive e migliorative sulla base di metriche, evidenze e risultati ottenuti;

#### 4. Processo di formazione

Finalizzato a mantenere e accrescere le competenze del team, garantendo che ogni membro disponga delle conoscenze necessarie per svolgere i compiti assegnati.

L'adozione strutturata di tali processi consente al gruppo di operare in modo ordinato, ripetibile e orientato al miglioramento continuo, riducendo i rischi organizzativi e aumentando l'affidabilità del progetto.

### 4.2) Processo di gestione del processo

#### 4.2.1) Introduzione

Il Processo di Gestione del Processo definisce il quadro organizzativo entro cui vengono pianificate, coordinate e monitorate tutte le attività di progetto.

Esso fornisce le linee guida per governare l'evoluzione del progetto, assicurando coerenza tra i processi adottati, le risorse disponibili e gli obiettivi stabiliti, e supporta l'interazione tra i diversi processi organizzativi, primari e di supporto.

#### 4.2.2) Scopo del processo

Lo scopo del processo di Gestione del Processo è:

- **Definire e strutturare le attività** necessarie allo svolgimento del progetto;
- **Assegnare responsabilità e ruoli** in modo chiaro e coerente;
- **Pianificare e monitorare l'avanzamento** delle attività;

- **Garantire una comunicazione efficace** sia interna al gruppo sia verso gli stakeholder<sub>G</sub> esterni;
- Assicurare il controllo del progetto nel **rispetto di tempi, costi e standard di qualità** stabiliti.

#### 4.2.3) Attività del processo

Lo standard [ISO/IEC/IEEE 12207:1995](#) definisce la gestione del processo come un insieme di attività organizzate che permettono di pianificare, controllare e valutare l'esecuzione dei processi di progetto.

Nel contesto del progetto, tali attività coprono l'intero ciclo di vita del processo e sono strutturate in fasi successive, ciascuna con obiettivi e responsabilità ben definite. Esse consentono di garantire il corretto avanzamento delle attività, il rispetto delle risorse disponibili e la conformità<sub>G</sub> agli standard di qualità stabiliti.

Le attività individuate dallo standard e adottate dal gruppo sono:

##### 1. Inizializzazione

Rappresenta la prima fase del processo. In questa fase vengono definiti i requisiti del processo oggetto di analisi. Una volta stabiliti, il responsabile ne valuta la fattibilità in relazione alle risorse disponibili.

##### 2. Pianificazione

Il responsabile pianifica le attività del processo.

I piani definiscono le attività e i task associati, nonché le caratteristiche del prodotto software.

Ogni piano deve includere le seguenti informazioni:

- tabella di marcia per il completamento dei task;
- stima dello sforzo richiesto;
- risorse necessarie;
- assegnazione dei compiti;
- assegnazione delle responsabilità (si veda la sezione **Sezione 4.2.4.1**);
- quantificazione dei rischi;
- metriche di controllo della qualità;
- costi associati all'esecuzione del processo;
- fornitura dell'ambiente e delle infrastrutture.

##### 3. Esecuzione e controllo

Il responsabile avvia le attività del processo in conformità<sub>G</sub> a quanto stabilito nella fase di pianificazione e ne monitora l'andamento.

- **Controllo interno:** verifica<sub>G</sub> il progresso delle attività e ne mantiene la tracciabilità. Eventuali criticità o impedimenti che possano causare ritardi devono essere segnalati tempestivamente al responsabile.

- **Comunicazione esterna:** gestisce i flussi comunicativi con la **proponente** e il **committente**, garantendo chiarezza e coerenza informativa.

#### 4. Revisione e valutazione<sub>G</sub>

Al completamento delle attività, il verificatore controlla che i risultati ottenuti siano conformi alle metriche di qualità e agli standard definiti.

#### 5. Chiusura

Un'attività si ritiene completa dopo aver superato l'attività di verifica<sub>G</sub>. Si veda la Definition of Done<sub>G</sub> (**Sezione 3.5.3.5**) per maggiori dettagli. La chiusura delle issue<sub>G</sub> legate alle attività avviene tramite merge<sub>G</sub> sul main a intervalli prefissati.

### 4.2.4) Procedure operative

#### 4.2.4.1) Ruoli di Progetto

La seguente sezione descrive la distribuzione delle responsabilità all'interno del team. Per garantire coerenza, efficienza e qualità, ogni ruolo assolve compiti specifici all'interno dei processi previsti dal Way of Working.

Coerentemente con gli obiettivi del corso di Ingegneria del Software, la struttura organizzativa permette una rotazione tale per cui ogni membro del team ricopra ciascun ruolo almeno una volta.

La tabella sottostante riassume le responsabilità e l'apporto di ogni figura:

Ruolo	Compiti	Presenza
Responsabile	<ul style="list-style-type: none"> <li>- Coordinamento piani e scadenze</li> <li>- Approvazione release</li> <li>- Comunicazione col committente</li> <li>- Uso efficiente delle risorse</li> <li>- Approvazione/redazione documenti</li> </ul>	Tutto il progetto
Amministratore	<ul style="list-style-type: none"> <li>- Garanzia efficienza strumenti</li> <li>- Gestione tecnologie di supporto</li> <li>- Verifica<sub>G</sub> procedure secondo norme</li> </ul>	Tutto il progetto
Verificatore	<ul style="list-style-type: none"> <li>- Testing e validazione<sub>G</sub></li> <li>- Controllo qualità deliverable</li> <li>- Conformità<sub>G</sub> ai requisiti</li> </ul>	Tutto il progetto
Analista	<ul style="list-style-type: none"> <li>- Analisi dei requisiti<sub>G</sub></li> <li>- Definizione bisogni del sistema</li> <li>- Redazione specifiche funzionali</li> </ul>	Massimo nella definizione dei requisiti; continuo per l'affinamento
Progettista	<ul style="list-style-type: none"> <li>- Progetta architettura sistema</li> <li>- Design e modellazione</li> <li>- Traduzione dei requisiti in struttura tecnica</li> </ul>	Progettazione architetture e di dettaglio
Programmatore	<ul style="list-style-type: none"> <li>- Codifica software</li> <li>- Implementazione design</li> <li>- Sviluppo funzionalità</li> </ul>	Implementazione

#### 4.2.4.2) Rendicontazione delle ore

La pianificazione e il monitoraggio delle ore produttive del progetto sono gestiti tramite il documento «Piano di Progetto», in cui vengono registrate sia le ore previste sia quelle effettivamente svolte per ciascun ruolo e membro del gruppo.

La ripartizione oraria è accompagnata dai relativi costi, consentendo una visione completa delle risorse economiche impiegate. Al fine di rendere rigorosa, oggettiva e verificabile la rendicontazione oraria, il team ha implementato un foglio di calcolo Google Sheets avanzato dedicato al tracciamento dell'effort. Per alimentare questo documento, è stato sviluppato uno script automatizzato che estrae i dati dal repository e genera dei file CSV<sub>G</sub> dedicati per ogni singolo sprint<sub>G</sub>, i quali vengono poi importati nel foglio. Questo strumento aggrega i dati operativi con lo storico e la gestione delle issue<sub>G</sub> (dimensione, tipologia e scadenze). Tale sistema integrato permette non solo di mappare con precisione l'impegno effettivo di ciascun membro, ma anche di classificare le ore in produttive e non produttive, scalandole dinamicamente dal budget preventivato per i vari ruoli di progetto. Questa organizzazione strutturata favorisce una gestione chiara, trasparente e collaborativa della distribuzione dei ruoli all'interno del team.

#### 4.2.4.3) Assegnazione Ruolo-Documento

La seguente sezione chiarisce i documenti associati a ciascun ruolo.

L'assegnazione viene rappresentata tramite una **legenda** e una **tabella riassuntiva**.

##### Legenda :

Azioni:

- R = Redazione.
- V = Verificag.
- A = Approvazione.
- C = Contribuente.

Ruoli:

- Responsabile = RESP.
- Amministratore = AMM.
- Analista = ANL.
- Progettista = PRG.
- Programmatore = PGRmm.
- Verificatore = VRF.

Documento	RESP	AMM	ANL	PRG	PGRmm	VRF
Norme di Progetto (NdP)	R/A	R	-	-	-	V
Analisi dei Requisiti (AdR)	A	-	R	-	-	V

Documento	RESP	AMM	ANL	PRG	PGRmm	VRF
Piano di Progetto (PdP)	R/A	-	C (supporto rischi)	-	-	V
Piano di Qualifica <sub>G</sub> (PdQ)	A	C (configurazione dei tool automatici)	-	-	-	R/V (Verificatore A redige, il Verificatore B verifica <sub>G</sub> )
Specifica Tecnica	A	-	C (per coerenza requisiti)	R	C	V
Manuale Utente (MU)	A	-	-	R	C	V
Verbali interni	R/A	-	-	-	-	V
Verbali esterni	R/A	-	-	-	-	V

#### 4.2.4.4) Issue<sub>G</sub> Tracking System

L'Issue<sub>G</sub> Tracking System è lo strumento utilizzato dal team per pianificare, assegnare, monitorare e controllare l'avanzamento delle attività di progetto. Il sistema è accessibile tramite la repository GitHub e si basa su un template di issue<sub>G</sub> condiviso, così da garantire uniformità, tracciabilità e chiarezza operativa.

Ogni issue<sub>G</sub> rappresenta un'attività, un task o una modifica da realizzare e segue un flusso di stato predefinito:

- **Backlog<sub>G</sub>**: attività pianificate ma non ancora avviate;
- **In lavorazione**: attività attualmente in esecuzione;
- **In verifica<sub>G</sub>**: attività completate e sottoposte a controllo;
- **In approvazione**: attività in attesa di validazione<sub>G</sub> finale;
- **Done**: attività concluse e integrate.

Le issue<sub>G</sub> possono essere organizzate in modo gerarchico tramite **relazioni parent/child**, consentendo di suddividere attività complesse in sotto-attività più semplici e gestibili.

La chiusura di un'issue<sub>G</sub> avviene esclusivamente al soddisfacimento dei criteri definiti nella Definition of Done<sub>G</sub> (**Sezione 3.5.3.5**), che garantisce il rispetto degli standard di qualità e di completezza stabiliti.

Inoltre, il sistema di gestione delle issue<sub>G</sub> prevede:

- La **definizione delle milestone<sub>G</sub>**, utilizzate per raggruppare attività e monitorare il raggiungimento degli obiettivi intermedi;

- L'uso della **retrospettiva**, al termine di iterazioni o sprint<sub>G</sub>, per analizzare le attività completate, individuare criticità e definire azioni di miglioramento per le fasi successive.

Questo approccio consente una gestione strutturata, trasparente e verificabile dell'intero ciclo di vita delle attività di progetto.

#### 4.2.4.4.1) Creazione e struttura di un'issue<sub>G</sub>

La creazione delle issue<sub>G</sub> avviene a seguito di riunioni interne o incontri con la proponente, durante i quali il gruppo individua le attività su cui concentrarsi. L'**amministratore** è responsabile della creazione delle issue<sub>G</sub> nel sistema, utilizzando l'apposito **template** definito dal gruppo.

Ogni nuova issue<sub>G</sub> deve includere le seguenti informazioni:

##### 1. **Assegnatario/i**

Generalmente l'issue<sub>G</sub> viene assegnata a una singola persona. In casi particolari, come attività di formazione o di esercitazione (**palestra**), l'issue<sub>G</sub> può essere assegnata a più membri o all'intero gruppo.

##### 2. **Descrizione**

Una descrizione chiara, dettagliata e non ambigua delle attività da svolgere.

##### 3. **Scopo**

Specifica il risultato atteso al termine dell'issue<sub>G</sub> e indica dove tale risultato deve essere documentato (ad esempio un documento, una sezione specifica o una modifica al repository).

##### 4. **Autore**

Il membro del gruppo incaricato di svolgere l'issue<sub>G</sub>.

##### 5. **Verificatore**

Il membro incaricato di verificare il corretto completamento dell'issue<sub>G</sub> secondo i criteri definiti nella **Definition of Done<sub>G</sub>** (**Sezione 3.5.3.5**). Per garantire l'oggettività della verifica<sub>G</sub> e l'assenza di conflitti di interesse, il verificatore deve essere necessariamente una figura diversa dall'autore.

##### 6. **Label (ambito/destinazione)<sup>4</sup>**

Le label permettono di classificare le issue<sub>G</sub> in base al loro ambito all'interno del progetto, facilitandone l'organizzazione e la ricerca. Le principali label adottate sono:

- Analisi dei Requisiti<sub>G</sub>,
- Piano di Progetto,
- Piano di Qualifica<sub>G</sub>,

---

<sup>4</sup>Le label possono essere aggiornate nel corso del progetto: label non più utili possono essere rimosse e nuove label introdotte in base alle esigenze.

- Norme di Progetto,
- Verbale,
- Diario di Bordo,
- Glossario,
- Generale → attività non direttamente riconducibili ai documenti principali (studio di materiale, gestione repository, sito web, attività varie).

#### 7. **Tipo di issue<sub>G</sub> (Type)**

Consente di distinguere la natura dell'attività:

- **Palestra** → attività formative non rendicontate;
- **Produttivo** → attività rendicontate che producono risultati concreti;
- **Bug** → individuazione e risoluzione di errori o malfunzionamenti;
- **Correzione** → modifiche e miglioramenti a materiali o documenti esistenti.

#### 8. **Priorità** (Bassa, Media, Alta)

La priorità ha un duplice scopo:

- supportare la valutazione<sub>G</sub> dell'importanza dell'issue<sub>G</sub>;
- comunicare all'assegnatario il livello di urgenza dell'attività.

#### 9. **Dimensione** (ExtraSmall, Small, Medium, Large)

Fornisce una stima indicativa della quantità di lavoro necessaria per completare l'issue<sub>G</sub>.

#### 10. **Data di scadenza**

Generalmente coincide con la fine dello sprint<sub>G</sub> di riferimento o con una milestone<sub>G</sub> pianificata.

#### 4.2.4.4.2) Flusso operativo

Il flusso operativo standard per la gestione di un'issue<sub>G</sub> è il seguente:

1. Il responsabile individua le attività necessarie al raggiungimento degli obiettivi; l'amministratore le traduce operativamente creando le issue<sub>G</sub> sul repository tramite i template predisposti;
2. Vengono assegnati autore/i e verificatore/i;
3. Vengono compilati tutti i campi richiesti (descrizione, scopo, label, tipo, priorità, dimensione, scadenza);
4. L'issue<sub>G</sub> viene inserita nello stato iniziale Backlog<sub>G</sub>;
5. L'issue<sub>G</sub> avanza attraverso gli stati del workflow fino al completamento;
6. La chiusura definitiva dell'issue<sub>G</sub> avviene tramite un commit di chiusura e lo spostamento manuale in Done nel Project Board, operazione che spetta al responsabile come atto di approvazione finale dell'attività.

## 4.3) Processo di Gestione dell'infrastruttura

### 4.3.1) Introduzione

Questo processo definisce, realizza e mantiene l'insieme delle infrastrutture e degli strumenti di supporto utilizzati dal team durante il ciclo di vita del progetto<sub>G</sub>.

### 4.3.2) Scopo del processo

Esso ha lo scopo di garantire un ambiente di lavoro affidabile, coerente e condiviso, che consenta lo svolgimento efficace delle attività di pianificazione, sviluppo, verifica<sub>G</sub> e gestione. L'adozione e la manutenzione<sub>G</sub> controllata degli strumenti assicurano disponibilità, aggiornamento continuo e conformità<sub>G</sub> agli standard di progetto, riducendo il rischio di inefficienze operative e problemi organizzativi.

### 4.3.3) Attività del processo

Questa sezione descrive le macro-attività del processo di infrastruttura, indipendentemente dagli strumenti specifici.

Il processo di infrastruttura comprende le seguenti attività principali:

#### 1. Implementazione del processo

Definizione e adozione degli strumenti necessari a supportare le attività del progetto, con particolare attenzione alla comunicazione, alla collaborazione e all'automazione.

#### 2. Creazione

Configurazione iniziale degli strumenti selezionati, predisposizione degli ambienti di lavoro e definizione delle strutture organizzative (repository, template, workflow).

#### 3. Manutenzione<sub>G</sub>

Aggiornamento e gestione continua dell'infrastruttura al fine di garantirne affidabilità, coerenza e conformità<sub>G</sub> agli standard di progetto.

Tutti i task relativi alle attività di creazione e manutenzione<sub>G</sub> sono di competenza dell'**Amministratore**.

### 4.3.4) Procedure operative

#### 4.3.4.1) Implementazione del processo

Per facilitare il lavoro del gruppo, in particolare la comunicazione asincrona e la collaborazione distribuita, sono stati adottati i seguenti strumenti:

**Discord** – Piattaforma di messaggistica e videoconferenza per riunioni interne da remoto.

**Zoom** – Piattaforma di videoconferenza per riunioni con la proponente BlueWind.

**WhatsApp** – Messaggistica istantanea per comunicazioni interne asincrone.

**Telegram** – Messaggistica asincrona per comunicazioni con la proponente.

**Typst<sub>G</sub>** – Linguaggio di markup per la redazione dei documenti, con compilazione automatizzata tramite script.

**Tinymist Typst<sub>G</sub>** – Estensione<sub>G</sub> di Visual Studio Code per il supporto sintattico e la live preview.

**Script bash** – Automazioni eseguite tramite GitHub Actions<sub>G</sub>.

**Git** – Sistema di versionamento distribuito per codice e documenti.

**GitHub** – Piattaforma cloud per repository, issue<sub>G</sub> tracking, integrazione continua e hosting.

**Google Docs** – Editor collaborativo per attività di brainstorming asincrono.

**Google Sheets** – Fogli di calcolo per il tracciamento di dati e metriche.

**Google Drive** – Sistema di file sharing per materiali di progetto.

**Python** – Linguaggio di programmazione usato per gestire alcuni script di automazione.

#### 4.3.4.2) Creazione

In questa sezione viene descritto il processo di creazione e configurazione degli strumenti ritenuti significativi per il supporto alle attività di progetto.

##### Typst<sub>G</sub>

L'ambiente di lavoro **Typst<sub>G</sub>** è stato personalizzato per supportare in modo efficiente la redazione della documentazione. Typst<sub>G</sub> consente la definizione di **template**, utilizzati in modo analogo alle chiamate di funzione di un linguaggio di programmazione.

I template principali svolgono le seguenti funzioni:

##### **Impaginazione:**

Impostazione uniforme di header e footer e definizione delle regole di stile comuni a tutti i documenti<sup>5</sup>.

##### **Tabelle:**

Predisposizione di template per la creazione semplificata di tabelle.

##### **Sprint<sub>G</sub>:**

Automazione del riepilogo degli sprint<sub>G</sub>, comprensiva dei calcoli necessari e del layout dedicato.

---

<sup>5</sup>Ad esempio, i link non sono visualizzati in blu e sottolineati di default; tale regola stilistica viene ereditata automaticamente dal contenuto del documento.

**Marcatatura automatica dei termini del Glossario:**

Funzionalità attivabile tramite un apposito flag booleano; se lasciata sempre attiva può impattare negativamente le prestazioni della live preview.

**Separazione tra contenuto e layout del Glossario:**

I termini e le definizioni del Glossario sono mantenuti in un file separato sotto forma di dizionario, consentendo l'ordinamento automatico e la generazione di viste personalizzate in formato PDF<sub>G</sub> e HTML.

**Gestione automatica della numerazione :**

È stata implementata una gestione automatica della numerazione sia in fase di creazione sia in fase di referenziamento, anche tra documenti diversi. La referenziazione avviene tramite funzioni apposite che accettano in input una stringa rappresentante il nome che identifica l'elemento target.

Gli elementi finora sottoposti a tale numerazione automatica sono:

- Casi d'uso;
- Requisiti;
- Test.

**Tracciamento automatico<sub>G</sub>:**

Sono state predisposte apposite funzioni per la costruzione automatica delle seguenti tabelle di tracciamento:

- Tracciamento casi d'uso - requisiti funzionali;
- Tracciamento test di sistema<sub>G</sub> - requisiti funzionali.

**Plot diagrammi dei casi d'uso:**

È stata predisposta una funzione per il plot dei diagrammi di attività in modo da poter applicare l'automazione già predisposta per la gestione del numbering.

**Git**

L'intero progetto utilizza **Git** come sistema di versionamento. Sono stati definiti due branch<sub>G</sub> principali:

- **Main**, destinato ai rilasci ufficiali;
- **Develop**, utilizzato per lo svolgimento delle attività di progetto.

È stato inoltre predisposto un file `.gitignore` per evitare la pubblicazione di file indesiderati.

**GitHub**

È stata creata una [GitHub Organization \(https://github.com/GroupRubberDuck\)](https://github.com/GroupRubberDuck) dedicata alle attività di progetto e i seguenti repository:

- [Documentazione https://github.com/GroupRubberDuck/Documentazione](https://github.com/GroupRubberDuck/Documentazione)
- [Proof of Concept<sub>G</sub> https://github.com/GroupRubberDuck/PoC](https://github.com/GroupRubberDuck/PoC)

**GitHub Actions<sub>G</sub>:**

Sono state configurate delle GitHub Actions<sub>G</sub> per la compilazione automatica dei file Typst<sub>G</sub> e per l'aggiornamento automatico del sito web.

**GitHub Pages:**

È stata attivata la funzionalità **GitHub Pages** per l'hosting del sito web del progetto.

**GitHub Issue<sub>G</sub> Tracking System:**

Il gruppo ha deciso di avvalersi dell'issue<sub>G</sub> tracking system offerto da GitHub; maggiori dettagli sono disponibili nella sezione dedicata alla guida operativa **Sezione 4.2.4.4**.

**Strumenti di comunicazione**

Nessuno degli strumenti di comunicazione ha richiesto operazioni significative durante la fase di creazione.

**Strumenti Google**

È stata creata una mail dedicata alle attività di progetto. Google Drive e Google Docs non richiedono particolari operazioni di configurazione, se non il caricamento e la condivisione del materiale. Google Sheets richiede invece operazioni più complesse per l'implementazione delle metriche e degli indicatori stabiliti.

**Script python**

Insieme alla numerazione automatica realizzata tramite funzioni Typst<sub>G</sub> è stato predisposto uno script python per mantenere ordinati e coerenti con la numerazione i file all'interno della cartella di lavoro. Inoltre crea eventuali file mancanti partendo da un template configurabile e aggiorna il file index usato per l'aggregazione dei singoli elementi.

Tabella 5: Strumenti utilizzati

**4.3.4.3) Manutenzione<sub>G</sub>**

Nel corso del progetto è necessario garantire il costante aggiornamento e il corretto funzionamento dell'infrastruttura e degli strumenti adottati. Le attività di manutenzione<sub>G</sub> hanno lo scopo di assicurare la continuità operativa, l'affidabilità e l'allineamento agli standard definiti.

Le operazioni di manutenzione<sub>G</sub> possono essere classificate in base alla frequenza:

- **Manutenzione<sub>G</sub> ordinaria:** aggiornamenti periodici e interventi ricorrenti legati alla normale evoluzione delle attività di progetto;
- **Manutenzione<sub>G</sub> non ordinaria:** aggiunte, modifiche o rimozioni di strumenti, configurazioni o risorse rese necessarie da cambiamenti nei requisiti, criticità emerse o decisioni organizzative straordinarie.

#### 4.3.4.3.1) Project board

Le issue<sub>G</sub> vengono organizzate all'interno di una **Project Board** dedicata, che consente di avere una visione d'insieme sullo stato di avanzamento delle attività e di monitorare il carico di lavoro del team.

#### 4.3.4.3.2) Milestone<sub>G</sub>

Le **milestone<sub>G</sub>** rappresentano obiettivi intermedi fondamentali nella pianificazione di medio e lungo periodo.

L'assegnazione delle issue<sub>G</sub> alle milestone<sub>G</sub> è responsabilità dell'amministratore, che provvede inoltre al loro aggiornamento in caso di variazioni nella pianificazione o nelle priorità di progetto.

#### 4.3.4.4) GitHub Actions<sub>G</sub>

Le **GitHub Actions<sub>G</sub>** costituiscono il sistema di automazione adottato dal gruppo per supportare le attività ricorrenti del progetto, in particolare quelle legate alla compilazione e al rilascio della documentazione.

Le azioni sono definite tramite file di configurazione in formato **YAML** e sono memorizzate nella cartella `.github/workflows` del repository. La loro manutenzione<sub>G</sub> e modifica rientra nelle responsabilità dell'amministratore di progetto.

#### 4.3.4.5) Script e automazioni

Il mantenimento degli script e delle automazioni è di competenza dell'**amministratore**.

Tali attività di manutenzione<sub>G</sub>, siano esse ordinarie o straordinarie, si rendono necessarie nei seguenti scenari:

- **Evoluzione:** introduzione di nuove automazioni per supportare processi emergenti;
- **Ottimizzazione:** dismissione di procedure divenute obsolete o ridondanti;
- **Correzione:** risoluzione di malfunzionamenti o errori logici negli script esistenti.

Le automazioni strettamente legate alla redazione dei documenti in **Typst<sub>G</sub>**, inclusi i template, sono salvate nella cartella `src/TypstTemplate` all'interno del repository [Documentazione](#).

Gli script di uso generale, non direttamente collegati a **Typst<sub>G</sub>**, sono invece salvati nella cartella `scripts`.

#### 4.3.4.6) Discord

Per quanto riguarda la piattaforma **Discord**, le attività operative rilevanti si limitano alla:

- Creazione di nuovi canali, qualora necessario;
- Moderazione di base per garantire ordine e chiarezza nelle comunicazioni interne.

Non sono previste configurazioni avanzate o attività di manutenzione<sub>G</sub> complesse.

#### 4.3.4.7) Strumenti Google

Di seguito sono elencate le principali attività operative associate agli strumenti della suite Google utilizzati nel progetto.

##### **Google Drive:**

Gestione dello spazio condiviso, inclusa la creazione di nuove cartelle e l'eliminazione di file obsoleti.

##### **Google Docs:**

Predisposizione, ove necessario, di layout coerenti per i documenti condivisi e di supporto alle attività di progetto.

##### **Google Sheets:**

Implementazione e aggiornamento delle metriche stabilite, sfruttando le funzionalità offerte dai fogli di calcolo.

### 4.4) Processo di miglioramento

#### 4.4.1) Introduzione

Il **processo di miglioramento** ha lo scopo di stabilire, valutare, misurare, controllare e migliorare in modo continuo i processi del ciclo di vita del software adottati dal progetto.

Attraverso un approccio sistematico e iterativo, il processo consente di incrementare l'efficacia e l'efficienza delle attività svolte, garantendo l'allineamento agli standard di qualità e alle esigenze del progetto.

#### 4.4.2) Attività del processo

Il processo di miglioramento è articolato nelle seguenti attività principali:

1. **Inizializzazione del processo** - Questa attività si occupa di definire e documentare i processi organizzativi adottati dal gruppo. Il presente documento assolve a questo compito, fungendo da riferimento ufficiale per la definizione dei processi.
2. **Valutazione<sub>G</sub> del processo** - Questa attività consiste nello sviluppo, nella documentazione, nell'utilizzo e nella storicizzazione<sup>6</sup> delle procedure di valutazione<sub>G</sub> dei

---

<sup>6</sup>Tramite registrazione.

processi. Al fine di garantirne efficacia ed efficienza, tali procedure sono sottoposte a revisioni periodiche basate sui dati raccolti durante l'esecuzione del progetto.

3. **Miglioramento del processo** - In seguito alle attività di valutazione<sub>G</sub>, vengono pianificate, attuate e documentate le azioni di miglioramento dei processi. A tal fine vengono raccolti e analizzati i seguenti dati: 1. Storico del processo; 2. Dati tecnici; 3. Risultati delle valutazioni; 4. Performance del processo.

Tali attività sono eseguite ciclicamente durante l'intero svolgimento del progetto e costituiscono la base del miglioramento continuo dei processi organizzativi.

#### 4.4.3) Procedure operative

##### 4.4.3.1) Ciclo PDCA<sub>G</sub>

Le attività del processo di miglioramento si inseriscono all'interno del ciclo di miglioramento continuo **PDCA (Plan-Do-Check-Act)**:

- **Plan**: individuazione delle criticità nei processi attuali tramite retrospettive e definizione di nuovi standard o modifiche alle norme per risolverle;
- **Do**: applicazione sperimentale delle nuove procedure definite durante il normale svolgimento delle attività operative;
- **Check**: misurazione delle prestazioni tramite le metriche e confronto dei dati raccolti con lo storico precedente per valutare se il cambiamento ha portato benefici;
- **Act**: aggiornamento formale delle **Norme di Progetto** e consolidamento delle procedure che si sono rivelate efficaci, scartando quelle inefficienti.



Figura 1: Rappresentazione del ciclo PDCA<sub>G</sub> applicato ai processi organizzativi

##### 4.4.3.1.1) Guida alla Retrospettiva

Il nostro gruppo ha ritenuto fondamentale l'uso della **retrospettiva**, soprattutto nell'ambito della pianificazione e del monitoraggio delle attività descritte nel [Piano di Progetto](#).

1. A turno, ogni membro del gruppo condivide le attività svolte e segnala eventuali problemi riscontrati.
2. Si discutono le criticità o i dubbi emersi durante le attività.

3. Alla fine della riunione, con l'assegnazione dei ruoli, le task vengono ridistribuite considerando eventuali difficoltà o problematiche segnalate.

Tutti i membri partecipano attivamente alla definizione delle azioni di miglioramento, sia sulle attività di documentazione sia sulla gestione delle issue<sub>G</sub> e delle infrastrutture.<sup>7</sup>

## 4.5) Processo di formazione

### 4.5.1) Introduzione

Il **processo di formazione** ha lo scopo di garantire che tutti i membri del team acquisiscano le competenze necessarie per contribuire efficacemente alle attività del progetto. Si tratta di un percorso continuo che combina sessioni strutturate e apprendimento autonomo, permettendo ai partecipanti di affrontare le sfide reali del progetto con consapevolezza e autonomia.

### 4.5.2) Scopo del processo

Il processo mira a:

- Facilitare l'acquisizione di competenze tecniche e organizzative;
- Favorire la comprensione dei documenti, degli strumenti e delle procedure del progetto;
- Promuovere l'apprendimento collaborativo e la condivisione delle conoscenze tra i membri del team.

### 4.5.3) Attività del processo

Le attività principali del processo di formazione comprendono:

1. **Sessioni formative guidate su strumenti e metodologie** del progetto;
2. **Studio dei documenti di progetto e dei materiali** forniti dal docente;
3. **Esercitazioni pratiche** per l'applicazione delle conoscenze acquisite;
4. **Apprendimento autonomo e auto-valutazione<sub>G</sub>** delle competenze.

### 4.5.4) Procedure operative

#### 4.5.4.1) Apprendimento libero

Non sempre un programma di apprendimento rigoroso riesce a rispondere in modo efficace alle esigenze reali e immediate che un membro del team può incontrare nello svolgimento di un task. Perciò, il processo prevede la possibilità di **formazione autonoma**, lasciando libertà ai membri del gruppo di approfondire tematiche o strumenti specifici secondo le proprie necessità.

---

<sup>7</sup>il gruppo adotta una filosofia di miglioramento continuo, accettando che all'inizio alcune procedure possano non essere perfette o contenere errori, ma mantenendo la massima trasparenza per favorire l'apprendimento e l'ottimizzazione dei processi.

#### 4.5.4.2) Apprendimento in gruppo

Poiché alcuni processi possono essere molto complessi e richiedere tempi lunghi, il gruppo può decidere di **riunirsi in sottogruppi** o dividere il task tra i membri<sup>8</sup>.

Questi sottogruppi possono essere impiegati anche per:

- Lo sviluppo del **Proof of Concept**<sub>G</sub>;
- Lo studio di nuove tecnologie;
- La realizzazione di prototipi per capire quali tecnologie funzionano meglio.

Lo **studio in gruppo** ha il vantaggio di:

- Favorire il confronto tra i membri;
- Migliorare la comprensione dei contenuti complessi;
- Velocizzare i tempi di apprendimento e di completamento dei task.

#### 4.5.4.3) Trasferimento dei ruoli

Al momento del **cambio dei ruoli**, il membro che cede il ruolo ha la responsabilità di trasmettere conoscenze e competenze acquisite al nuovo titolare del ruolo.

In particolare:

- Se il membro uscente ritiene di aver sviluppato tecniche o conoscenze importanti, deve **formare inizialmente il nuovo membro**;
- Passare il **testimone e le conoscenze**, rendendo più semplice l'ingresso del nuovo responsabile nel ruolo;
- Garantire continuità operativa, riducendo errori dovuti alla mancata familiarità con le procedure e gli strumenti del progetto.

Questo approccio consente di **mantenere l'efficienza del team** e di valorizzare l'esperienza accumulata durante il progetto.

#### 4.5.4.4) Rendicontazione delle ore

Essendo un progetto svolto per la prima volta, ogni attività formativa viene registrata anche in termini di ore non produttive. Tali ore comprendono:

- **Studio dei documenti e dei materiali forniti** dal docente;
- **Preparazione dei documenti di progetto** nel modo più accurato possibile;
- **Apprendimento** delle procedure interne e gestione degli strumenti.

Questa organizzazione permette di monitorare l'impegno formativo dei membri del team e di garantire che ciascuno possa acquisire competenze adeguate per svolgere le proprie attività in modo efficace.

---

<sup>8</sup>Ad esempio, questa opzione è stata utilizzata per studiare il materiale fornito dall'azienda.

## 5) Metriche

### 5.1) Introduzione

Le metriche rappresentano strumenti fondamentali per misurare, controllare e migliorare sia il processo di sviluppo sia i prodotti realizzati.

Il tracciamento delle metriche viene effettuato all'interno del documento: [Piano di Qualifica<sub>G</sub>](#).

Nelle Norme di Progetto sono considerate due principali categorie di metriche:

1. **Metriche di qualità di processo** – Consentono di valutare l'efficacia, l'efficienza e la correttezza dei processi di sviluppo, monitorando la pianificazione, l'esecuzione e la gestione delle attività.
2. **Metriche di qualità di prodotto** – Misurano le caratteristiche intrinseche dei deliverable.

In particolare

#### **Funzionalità:**

Valuta la capacità del software di fornire correttamente le funzionalità richieste dai requisiti, assicurando completezza e coerenza rispetto alle specifiche definite. In particolare sono considerate:

- **Adeguatezza:**  
Presenza di funzioni appropriate per i compiti che deve svolgere.
- **Accuratezza:**  
Capacità di fornire risultati nel modo stabilito nei requisiti.
- **Interoperabilità:**  
La capacità del prodotto di interagire con altri sistemi definiti.
- **Conformità<sub>G</sub>:**  
Il prodotto aderisce a determinati standard di dominio.
- **Sicurezza:**  
Non vi sono falle di sicurezza che espongono dati sensibili degli utenti.

#### **Affidabilità:**

Misura la capacità del software di operare senza guasti in condizioni previste, garantendo comportamenti consistenti e riducendo al minimo malfunzionamenti.

- **Maturità:**  
Capacità dell'applicazione di evitare blocchi a seguito di errori nel software.
- **Tolleranza agli errori:**  
Capacità di mantenere determinati livelli di prestazione in caso di errori.
- **Recuperabilità:**  
Capacità del software di ripristinare il proprio livello di prestazioni e di recuperare i dati direttamente interessati a seguito di un guasto o di un errore.

- **Aderenza :**

Capacità di aderire a standard di affidabilità.

**Efficienza:**

Indica l'ottimizzazione delle risorse e la rapidità di risposta del software alle richieste, valutando tempi di esecuzione, throughput e utilizzo delle risorse disponibili.

- **Comportamento temporale:**

Capacità di soddisfare le richieste con un tempo di risposta adeguato.

- **Utilizzo delle risorse:**

Capacità di usare le risorse in modo proporzionale alle richieste.

- **Conformità<sub>G</sub>:**

Aderenza del prodotto a standard riguardanti l'efficienza.

**Usabilità:**

Rileva quanto il software sia intuitivo e facile da utilizzare, considerando la semplicità delle interazioni, la facilità di apprendimento e la correttezza delle operazioni da parte degli utenti.

- **Comprensibilità:**

Facilità di comprensione del prodotto.

- **Apprendibilità:**

Facilità di apprendimento delle funzionalità.

- **Operabilità:**

Semplicità di utilizzo del prodotto.

- **Attrattività:**

Capacità di fornire un'esperienza utente gradevole.

- **Conformità<sub>G</sub>:**

Aderenza a standard di usabilità.

**Manutenibilità:**

Misura quanto facilmente il software può essere modificato o esteso senza introdurre errori, tenendo conto della complessità del codice, della modularità e della facilità di intervento sugli artefatti.

- **Analizzabilità:**

Facilità di ispezione del codice con lo scopo di cercare errori

- **Modificabilità:**

Facilità nell'apportare modifiche.

- **Stabilità:**

Capacità del prodotto di limitare effetti indesiderati derivanti da modifiche o aggiunte al prodotto.

- **Testabilità:**

Semplicità di testing del prodotto.

**Portabilità:**

Rappresenta la facilità con cui il prodotto può essere spostato da un ambiente di lavoro ad un altro.

**• Adattabilità:**

Facilità di adattamento del prodotto ad ambienti diversi.

**• Installabilità:**

Facilità di installazione del prodotto su un ambiente di sviluppo.

**• Conformità<sub>G</sub>:**

Aderenza alle convenzioni sulla portabilità.

**• Sostituibilità:**

Facilità di migrazione da un prodotto precedente al prodotto oggetto del progetto.

L'adozione sistematica delle metriche permette di identificare aree di miglioramento, garantire trasparenza e mantenere elevati standard di qualità durante l'intero ciclo di vita del progetto<sub>G</sub>.

**5.1.1) Nomenclatura delle Metriche**

La nomenclatura adottata è la seguente:

**TIPO\_METRICA-##**

dove:

- TIPO\_METRICA =
  - MPC per le metriche di qualità del processo
  - MPD per le metriche di qualità del prodotto
- ## è un contatore progressivo

**6) Metriche di Qualità del Processo****6.1) Processi primari**

I processi primari riguardano le attività di sviluppo del software, come requisiti, progettazione, implementazione, integrazione e manutenzione<sub>G</sub>. Per valutarne andamento ed efficacia si utilizzano metriche di processo che permettono di monitorare tempi, costi e progressi, evidenziando eventuali scostamenti rispetto agli obiettivi.

## 6.1.1) Fornitura

Codice	Metrica	Valore accettabile	Valore ottimo
MPC01	Planned Value <sub>G</sub>	$\geq 0$	$\leq BAC$
MPC02	Earned Value <sub>G</sub>	$\geq PV * 0.75$	$\geq PV$
MPC03	Actual Cost <sub>G</sub>	$0 \leq AC \leq 1.2 * EV$	$\leq EV$
MPC04	Schedule Performance Index <sub>G</sub>	$\geq 0.9$	$\geq 1.0$
MPC05	Cost Performance Index <sub>G</sub>	$\geq 0.9$	$\geq 1.0$
MPC06	Estimate at Completion <sub>G</sub>	$\leq 1.1 * BAC$	$\leq BAC$
MPC07	To Complete Performance Index <sub>G</sub>	$\sim 1.0$	$\leq 1.0$
MPC08	Estimate to Complete <sub>G</sub>	$\leq (BAC - AC) * 1.1$	$\leq BAC - AC$

Tabella 6: Metriche processo di Fornitura

**Planned Value<sub>G</sub>**

- **Codice:** MPC01

- **Descrizione:**

Rappresenta il valore del lavoro che avrebbe dovuto essere completato entro una certa data, secondo il piano di progetto approvato.

- **Formula:**

$$PV = BAC * \% \text{ di lavoro pianificata}$$

*BAC* Sta per budget at completion

- **Interpretazione:**

- Valore accettabile:  $\geq 0$
- Valore ottimo:  $\leq BAC$

- **Calcolo:**

Si può rappresentare con il totale di ore produttive previste

**Earned Value<sub>G</sub>**

- **Codice:** MPC02
- **Descrizione:**  
Rappresenta il valore del lavoro completato rispetto al budget previsto.  
Molto utile per monitorare l'andamento effettivo delle attività di progetto.
- **Formula:**

$$EV = BAC * \% \text{ di lavoro completata}$$

*BAC* Sta per budget at completion

- **Interpretazione:**
  - Valore accettabile:  $\geq PV * 0.75$
  - Valore ottimo:  $\geq PV$
- **Calcolo:**  
Si può rappresentare con il totale di ore produttive effettive

**Actual Cost<sub>G</sub>**

- **Codice:** MPC03
- **Descrizione:**  
Rappresenta il costo effettivo sostenuto.  
Risulta utile nel verificare che il lavoro svolto sia in linea con le aspettative.
- **Formula:**

$$AC = \text{Costo sostenuto nello sprint}_G$$

- **Interpretazione:**
  - Valore accettabile:  $0 \leq AC \leq 1.2 * EV$
  - Valore ottimo:  $\leq EV$

**Schedule Performance Index<sub>G</sub>**

- **Codice:** MPC04
- **Descrizione:**  
Rappresenta il rapporto tra il valore del lavoro completato e il lavoro pianificato.
- **Formula:**  
$$\text{Schedule Performance Index}_G = \frac{\text{Earned Value}_G}{\text{Planned Value}_G}$$
- **Interpretazione:**
  - Valore accettabile:  $\geq 0.9$
  - Valore ottimo:  $\geq 1.0$

**Cost Performance Index<sub>G</sub>**

- **Codice:** MPC05
- **Descrizione:**  
Rappresenta il rapporto tra il valore del lavoro completato e il costo effettivamente sostenuto.
- **Formula:**  
$$\text{Cost Performance Index}_G = \frac{\text{Earned Value}_G}{\text{Actual Cost}_G}$$
- **Interpretazione:**
  - Valore accettabile:  $\geq 0.9$
  - Valore ottimo:  $\geq 1.0$

**Estimate at Completion<sub>G</sub>**

- **Codice:** MPC06
- **Descrizione:**  
Rappresenta il costo da sostenere per il completamento delle attività sulla base della performance attuali.
- **Formula:**  
$$\text{Estimate at Completion}_G = \frac{\text{Budget at completion}}{\text{Cost Performance Index}_G}$$
- **Interpretazione:**
  - Valore accettabile:  $\leq 1.1 * \text{BAC}$
  - Valore ottimo:  $\leq \text{BAC}$

**To Complete Performance Index<sub>G</sub>**

- **Codice:** MPC07
- **Descrizione:**  
Rappresenta l'efficienza richiesta in futuro per completare il lavoro entro il termine previsto.
- **Formula:**  
$$\text{Estimate at Completion}_G = \frac{\text{Budget at Completion} - \text{Earned Value}_G}{\text{Budget at Completion} - \text{Actual Cost}_G}$$
- **Interpretazione:**
  - Valore accettabile:  $\sim 1.0$
  - Valore ottimo:  $\leq 1.0$

**Estimate To Complete<sub>G</sub>**

- **Codice:** MPC08
- **Descrizione:**  
Rappresenta il costo ancora da sostenere per il completamento delle attività
- **Formula:**  
Estimate To Complete<sub>G</sub> =  
Estimate at Completion<sub>G</sub> – Actual Cost<sub>G</sub>
- **Interpretazione:**
  - Valore accettabile:  $\leq (BAC - AC) * 1.1$
  - Valore ottimo:  $\leq BAC - AC$

**6.1.2) Sviluppo**

<b>Codice</b>	<b>Metrica</b>	<b>Valore accettabile</b>	<b>Valore ottimo</b>
MPC09	Requirements Stability Index <sub>G</sub>	$\geq 0.7$	1.0

Tabella 7: Metriche processo di Sviluppo

**Requirements Stability index<sub>G</sub>**

- **Codice:** MPC09
- **Descrizione:**  
Misura i cambiamenti apportati ai requisiti nel tempo.
- **Formula:**  

$$\text{Requirements Stability Index}_G = \frac{\text{NRI} - (\text{NC} + \text{NRC} + \text{NRA})}{\text{NRI}}$$
- **Legenda:**
  - ▶ Numero Requisiti Iniziali = NRI.
  - ▶ Numero di Cambiamenti = NC.
  - ▶ Numero di Requisiti Cancellati = NRC.
  - ▶ Numero di Requisiti Aggiunti = NRA.
- **Interpretazione:**
  - ▶ Valore accettabile:  $\geq 0.7$
  - ▶ Valore ottimo: 1.0

**6.2) Processi di supporto**

Questi includono attività che garantiscono controllo, tracciabilità e affidabilità del processo stesso, come la verifica<sub>G</sub>, la validazione<sub>G</sub>, la gestione della configurazione, la documentazione tecnica e l'assicurazione qualità. Questi processi consentono di monitorare e ridurre gli scostamenti rispetto agli standard pianificati. Le metriche associate ai processi di supporto sono definite per consentire una valutazione<sub>G</sub> oggettiva della conformità<sub>G</sub> e del controllo delle attività di supporto rispetto ai processi e alle procedure stabilite.

**6.2.1) Documentazione**

Codice	Metrica	Valore accettabile	Valore ottimo
MPC10	Indice di Gulpease <sub>G</sub>	$\geq 60$	$\geq 70$
MPC11	Correttezza ortografica	$\leq 1$	= 0

Tabella 8: Metriche processo di Documentazione

**Indice di Gulpease<sub>G</sub>**

- **Codice:** MPC10

- **Descrizione:**

L'Indice di Gulpease<sub>G</sub> misura la leggibilità di un testo.

Valuta il grado di istruzione necessario alla comprensione del contenuto.

- **Formula:**

$$IG = 89 + \frac{300 * (\text{numero di frasi}) - 10 * (\text{numero di lettere})}{\text{numero di parole}}$$

- **Interpretazione:**

- Valore accettabile:  $\geq 60$
- Valore ottimo:  $\geq 70$

- **Calcolo:**

Per il calcolo il gruppo ha sviluppato una propria automazione sfruttando le funzionalità di Typst<sub>G</sub>. Questo permette una forte integrazione e analisi di dettaglio più fine.

**Correttezza Ortografica**

- **Codice:** MPC11

- **Descrizione:**

Misura la qualità della documentazione

- **Formula:**

$$\text{Correttezza Ortografica} = 1000 * \frac{\text{numero di errori ortografici}}{\text{Numero di parole}}$$

- **Interpretazione:**

- Valore accettabile:  $\leq 1$
- Valore ottimo:  $= 0$

6.2.2) Verifica<sub>G</sub>

Codice	Metrica	Valore accettabile	Valore ottimo
MPC12	Test Success Rate	≥ 90%	100%

Tabella 9: Metriche processo di Verifica<sub>G</sub>**Test Success Rate**

- **Codice:** MPC12
- **Descrizione:**  
Rappresenta la percentuale di test automatizzati che vengono superati.
- **Formula:**

$$\text{Test success rate} = \frac{\text{Numero di Test superati}}{\text{Numero totale dei test}}$$

- **Interpretazione:**
  - Valore accettabile: ≥ 90%
  - Valore ottimo: 100%

## 6.3) Processi organizzativi

Riguardano il miglioramento continuo del processo, la definizione degli standard interni, la gestione della qualità complessiva e lo sviluppo delle competenze del personale. Questi processi assicurano la sostenibilità e la maturità del modello di sviluppo nel tempo. Le metriche associate ai processi organizzativi sono definite per consentire una valutazione<sub>G</sub> oggettiva della conformità<sub>G</sub> e dell'efficacia dei processi di gestione e governance interna.

## 6.3.1) Gestione dei processi

Codice	Metrica	Valore accettabile	Valore ottimo
MPC13	Time Efficiency <sub>G</sub>	≥ 80%	≥ 100%
MPC14	Process Lead Time	≥ 90%	= 100%

Tabella 10: Metriche processo di Gestione dei Processi

**Time Efficiency<sub>G</sub>**

- **Codice:** MPC13

- **Descrizione:**

Rappresenta il rapporto tra le ore previste e le ore effettivamente impiegate, misurato in modo cumulativo  $\text{sprint}_G$  per  $\text{sprint}_G$ . Un valore inferiore a 1 indica che il team ha impiegato più ore del pianificato.

- **Formula:**

$$\text{Time Efficiency}_G = \frac{\text{Ore previste cumulative}}{\text{Ore effettive cumulative}}$$

- **Interpretazione:**

- Valore accettabile:  $\geq 80\%$
- Valore ottimo:  $\geq 100\%$

**Process Lead Time**

- **Codice:** MPC14

- **Descrizione:**

Stima la durata finale del progetto in settimane, basandosi sulla velocità reale del team misurata tramite lo Schedule Performance Index<sub>G</sub>.

- **Formula:**

$$\text{TimeEAC} = \frac{\text{Settimane Pianificate}}{\text{SPI}}$$

- **Interpretazione:**

- Valore accettabile:  $\geq 90\%$
- Valore ottimo:  $= 100\%$

## 7) Metriche di Qualità del Prodotto

### 7.1) Funzionalità

Codice	Metrica	Valore accettabile	Valore ottimo
MPD01	Requisiti obbligatori soddisfatti	100%	100%
MPD02	Requisiti desiderabili soddisfatti	≥ 50%	≥ 75%
MPD03	Requisiti opzionali soddisfatti	≥ 0%	≥ 50%

Tabella 11: Metriche di funzionalità del prodotto

#### Requisiti obbligatori soddisfatti

- **Codice:** MPD01
- **Descrizione:**  
Rappresenta la percentuale di requisiti obbligatori soddisfatti.  
È utile a monitorare il grado di soddisfacimento dei requisiti essenziali.
- **Formula:**

$$RObbS = \frac{\text{Numero di requisiti obbligatori soddisfatti}}{\text{Numero di requisiti obbligatori}}$$

- **Interpretazione:**
  - Valore accettabile:100%
  - Valore ottimo:100%

**Requisiti Desiderabili soddisfatti**

- **Codice:** MPD02
- **Descrizione:**  
Rappresenta la percentuale di requisiti desiderabili soddisfatti.  
È utile a monitorare il grado di soddisfacimento dei requisiti desiderabili.
- **Formula:**

$$RDesS = \frac{\text{Numero di requisiti desiderabili soddisfatti}}{\text{Numero di requisiti desiderabili}}$$

- **Interpretazione:**
  - Valore accettabile:  $\geq 50\%$
  - Valore ottimo:  $\geq 75\%$

**Requisiti opzionali soddisfatti**

- **Codice:** MPD03
- **Descrizione:**  
Rappresenta la percentuale di requisiti opzionali soddisfatti.  
È utile a monitorare il grado di soddisfacimento dei requisiti opzionali.
- **Formula:**

$$ROpzS = \frac{\text{Numero di requisiti opzionali soddisfatti}}{\text{Numero di requisiti opzionali}}$$

- **Interpretazione:**
  - Valore accettabile:  $\geq 0\%$
  - Valore ottimo:  $\geq 50\%$

## 7.2) Affidabilità

Codice	Metrica	Valore accettabile	Valore ottimo
MPD04	Failure Density	$\leq 0.5$	$\leq 0.2$
MPD05	Statement Coverage <sub>G</sub>	$\geq 80\%$	$\geq 95\%$
MPD06	Branch <sub>G</sub> Coverage	$\geq 70\%$	$\geq 90\%$

Tabella 12: Metriche di affidabilità del prodotto

**Failure Density**

- **Codice:** MPD04
- **Descrizione:**  
Numero di failure per 1000 linee di codice (KLOC)
- **Formula:**

$$\text{Failure Density} = \frac{\text{Numero di Failure}}{\text{KLOC}}$$

- **Interpretazione:**
  - Valore accettabile:  $\leq 0.5$
  - Valore ottimo:  $\leq 0.2$

**Statement Coverage<sub>G</sub>**

- **Codice:** MPD05
- **Descrizione:**  
Percentuale di istruzioni del codice coperte da test automatizzati.
- **Formula:**

$$\text{Statement Coverage}_G = \frac{\text{Istruzioni Testate}}{\text{Istruzioni Totali}}$$

- **Interpretazione:**
  - Valore accettabile:  $\geq 80\%$
  - Valore ottimo:  $\geq 95\%$

**Branch<sub>G</sub> Coverage**

- **Codice:** MPD06
- **Descrizione:**  
Percentuale di rami di codice coperte da test automatizzati.
- **Formula:**

$$\text{Branch}_G \text{ coverage} = \frac{\text{Numero di Branch}_G \text{ coperti da test}}{\text{Numero di Branch}_G \text{ totali}}$$

- **Interpretazione:**
  - Valore accettabile:  $\geq 70\%$
  - Valore ottimo:  $\geq 90\%$

**7.3) Usabilità**

Codice	Metrica	Valore accettabile	Valore ottimo
MPD07	User Error Rate	$\leq 3\%$	$\leq 1\%$
MPD08	Time to Complete Task	10 minuti	5 minuti

Tabella 13: Metriche di usabilità del prodotto

**User Error Rate**

- **Codice:** MPD07
- **Descrizione:**  
Misura quanto spesso un utente fa errori durante l'uso del prodotto.
- **Formula:**

$$\text{User error rate} = \frac{\text{Errori Totali}}{\text{Azioni Totali}}$$

- **Interpretazione:**
  - Valore accettabile:  $\leq 3\%$
  - Valore ottimo:  $\leq 1\%$

**Time To Complete Task**

- **Codice:** MPD08

- **Descrizione:**

Tempo medio per completare un'attività.

- **Formula:**

$$\frac{\text{Tempo Ottimista} + 4 * \text{Tempo Probabile} + \text{Tempo Pessimista}}{6}$$

- **Interpretazione:**

- Valore accettabile:10 minuti
- Valore ottimo:5 minuti

**7.4) Efficienza**

Codice	Metrica	Valore accettabile	Valore ottimo
MPD09	Response Time	≤ 3 sec	≤ 1 sec
MPD10	CPU Utilization	≤ 35%	≤ 20%
MPD11	Memory Utilization	≤ 4,0 GB	≤ 1,5 GB

Tabella 14: Metriche di efficienza del prodotto<sup>9</sup>

**Response Time**

- **Codice:** MPD09

- **Descrizione:**

Misura il tempo medio impiegato dal prodotto per rispondere a una richiesta.

- **Interpretazione:**

- Valore accettabile:≤ 3 sec
- Valore ottimo:≤ 1 sec

<sup>9</sup>I valori soglia di questa sezione sono espressi in termini assoluti, non come percentuali relative.

**CPU Utilization**

- **Codice:** MPD10
- **Descrizione:**  
Misura l'utilizzo di CPU lato client.
- **Interpretazione:**
  - Valore accettabile:  $\leq 35\%$
  - Valore ottimo:  $\leq 20\%$
- **Calcolo:**  
Le percentuali fanno riferimento ai seguenti processori:
  - Intel Core Ultra 5
  - AMD Ryzen 5 5500

**Memory Utilization**

- **Codice:** MPD11
- **Descrizione:**  
Misura l'utilizzo di memoria RAM lato client.
- **Interpretazione:**
  - Valore accettabile:  $\leq 4,0$  GB
  - Valore ottimo:  $\leq 1,5$  GB

## 7.5) Manutenibilità

Codice	Metrica	Valore accettabile	Valore ottimo
MPD12	Cyclomatic Complexity <sub>G</sub>	$\leq 10$	$\leq 8$
MPD13	Instability Index <sub>G</sub>	$I \geq 0.7$ $\vee$ $I \leq 0,30$	$I \geq 0.85$ $\vee$ $I \leq 0,15$
MPD14	Coefficient of Coupling <sub>G</sub>	$\leq 0.4$	$\leq 0.2$
MPD15	Code Smell	$\leq 10$	$\leq 5$
MPD16	Code Coverage	$\geq 80\%$	$\geq 90\%$

Tabella 15: Metriche manutenibilità del prodotto

**Cyclomatic Complexity<sub>G</sub>**

- **Codice:** MPD12
- **Descrizione:**  
Misura la complessità del codice in base al numero di percorsi linearmente dipendenti.
- **Formula:**

$$\text{Cyclomatic complexity}_G = E - N + 2P$$

***E*:**

Numero di archi nel grafo di controllo

***N*:**

Numero di nodi nel grafo di controllo

***P*:**

Numero di componenti connesse

- **Interpretazione:**
  - Valore accettabile:  $\leq 10$
  - Valore ottimo:  $\leq 8$

**Instability Index<sub>G</sub>**

- **Codice:** MPD13
- **Descrizione:**  
Misura la resilienza di un modulo al cambiamento.
- **Formula:**

$$\text{Instability index}_G = \frac{C_e}{C_e + C_a}$$

**C<sub>e</sub>:**

Numero di classi esterne da cui il modulo dipende

**C<sub>a</sub>:**

Numero di classi esterne che dipendono dal modulo

- **Interpretazione:**
  - Valore accettabile:  $I \geq 0.7$   
∨  
 $I \leq 0,30$
  - Valore ottimo:  $I \geq 0.85$   
∨  
 $I \leq 0,15$

**Coefficient of coupling<sub>G</sub>**

- **Codice:** MPD14
- **Descrizione:**  
Misura il coupling tra le componenti del sistema.
- **Formula:**

$$\text{Coefficient of coupling}_G = \frac{\text{Numero di dipendenze}}{\text{Numero di componenti}}$$

- **Interpretazione:**
  - Valore accettabile:  $\leq 0.4$
  - Valore ottimo:  $\leq 0.2$

### Code Smell

- **Codice:** MPD15
- **Descrizione:**  
Misura il rapporto tra codice smell e KLOC.
- **Formula:**  
$$\text{Code smell} = \frac{\text{Numero di codice smell}}{\text{KLOC}}$$
- **Interpretazione:**
  - Valore accettabile:  $\leq 10$
  - Valore ottimo:  $\leq 5$

### Code Coverage

- **Codice:** MPD16
- **Descrizione:**  
Rappresenta la percentuale di codice coperto da test automatizzati.
- **Formula:**  
$$\text{Code Coverage} = \frac{\text{Linee di codice testate}}{\text{Linee di codice totali}}$$
- **Interpretazione:**
  - Valore accettabile:  $\geq 80\%$
  - Valore ottimo:  $\geq 90\%$