

**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>	In progress
<b>Versione</b>	0.8.0
<b>Autori</b>	Davide Lorenzon Aldo Bettega Guerra Filippo Ana Maria Draghici
<b>Verificatori</b>	Ana Maria Draghici Davide Lorenzon Aldo Bettega
<b>Uso</b>	Interno
<b>Destinatari</b>	Tutto il gruppo

Vers.	Data	Autore	Verificatore	Descrizione
0.8.0	2025-12-09	Filippo Guerra	Davide Lorenzon	Aggiunta sezione 3.1
0.7.0	2025-12-04	Aldo Bettega	Davide Lorenzon	Aggiunta sezione 4.1.2 e sezione 9. aggiornata 5.3 definition of done
0.6.1	2025-12-02	Davide Lorenzon	Davide Lorenzon	Apportate modifiche di ordine nella sottosezione documentazione
0.6.0	2025-12-02	Ana Maria Draghici	Aldo Bettega	Aggiunta sezione 4.8 «struttura specifica dei documenti» e relative sottosezioni, aggiunto sezione 5.4.3 «Versionamento»
0.5.0	2025-11-30	Ana Maria Draghici	Aldo Bettega	Aggiunta sezione 5.3 e 5.4, rispettivamente «Definition of Done» e «Issue tracking System»
0.4.0	2025-11-29	Guerra Filippo	Ana Maria Draghici	Aggiunta sezione 5.2 «ruolo-documento»
0.3.0	2025-11-11	Ana Maria Draghici	Davide Lorenzon	Aggiunta sezione 4.8.1 struttura Analisi Requisiti
0.2.0	2025-11-11	Davide Lorenzon	Ana Maria Draghici	Aggiunta struttura dei documenti (come stabilito da verbale 2025-11-10)
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
2) Riferimenti .....	2
2.1) Riferimenti normativi .....	2
2.2) Riferimenti informativi .....	2
3) Processi Primari .....	3
3.1) Processo di fornitura .....	3
3.1.1) Introduzione .....	3
3.1.2) Scopo e descrizione .....	3
3.1.3) Rapporti con la proponente .....	3
3.1.4) Documentazione prodotta .....	4
4) Processi di Supporto .....	5
4.1) Documentazione .....	5
4.1.1) Impegno del gruppo nella produzione della documentazione .....	5
4.1.2) Workflow documentale .....	5
4.1.2.1) Stati del documento .....	5
4.1.2.2) Procedura di avanzamento tra stati .....	6
4.1.3) Strumenti di supporto .....	6
4.1.4) Identificazione dei documenti .....	7
4.1.4.1) Elenco dei documenti .....	7
4.1.5) Progettazione dei documenti .....	7
4.1.6) Pubblicazione e Distribuzione della Documentazione .....	8
4.1.7) Posizione del Documento .....	8
4.1.8) Informazioni comuni .....	8
4.1.9) Documentazione prevista per la fase RTB .....	9
4.1.10) Struttura specifica dei documenti .....	9
4.1.10.1) Analisi dei Requisiti .....	10
4.1.10.1.1) Struttura principale .....	10
4.1.10.2) Piano di Progetto .....	11
4.1.10.2.1) Struttura principale .....	11
4.1.10.3) Piano di Qualifica .....	12
4.1.10.3.1) Struttura principale .....	12
4.1.10.4) Verbalì .....	13
4.1.10.4.1) Procedure e responsabilità .....	13
4.1.10.4.2) Struttura Verbale .....	13
4.1.10.5) Diario di Bordo .....	14
4.1.10.6) Norme di Progetto .....	15
4.1.10.6.1) Struttura principale .....	15

4.1.10.7) Glossario .....	15
4.1.10.8) .....	15
4.1.11) Manutenzione .....	16
5) Processi Organizzativi .....	17
5.1) Gestione del Processo .....	17
5.1.1) Ruoli di Progetto .....	17
5.2) Assegnazione Ruolo-Documento .....	17
5.3) Definition of Done (DoD) .....	18
5.4) Issue tracking System – Guida Operativa .....	19
5.4.1) Creazione di una nuova issue .....	19
5.4.2) Flusso operativo .....	21
5.4.3) Versionamento .....	21
5.4.3.1) Codice di versione .....	21
5.4.3.2) Regole di incremento .....	21
6) Metriche e standard per la Qualità .....	23
7) Metriche di Qualità del Processo .....	24
8) Metriche di Qualità del Prodotto .....	25
9) Best Practices .....	26
9.1) Formato nome dei verbali .....	26
9.2) Scrittura dei commit .....	26
9.3) Issue Template .....	26

# 1) Introduzione

## 1.1) Scopo del documento

In questo documento il gruppo definisce e mantiene 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 alla norma EN 18031, lo standard tecnico europeo che definisce i requisiti di sicurezza informatica per i dispositivi radio wireless (es. Wi-Fi, LTE, Bluetooth, IoT wireless).

L'obiettivo dell'applicazione è supportare e guidare l'utente nella valutazione dei requisiti di cybersecurity, eseguendo in autonomia i percorsi decisionali tramite decision tree (alberi di decisione). Questo approccio permette di accelerare, standardizzare e rendere più affidabile il processo di verifica della conformità, 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 un approccio ibrido alla gestione dei riferimenti al Glossario. I termini tecnici possono essere presentati secondo 2 modalità:

- **Footnote al primo utilizzo:** applicata ai concetti critici o potenzialmente ambigui, permette un accesso immediato alla definizione senza interrompere il flusso logico del testo.
- **Marcatura tramite pedice "G" (termine <sub>G</sub>):** utilizzata per termini ricorrenti o già contestualizzati, indica semplicemente la presenza del termine nel Glossario.

## 2) Riferimenti

### 2.1) Riferimenti normativi

- [Software Engineering, Ian Sommerville](#)
- [Regolamento progetto](#)
- [Capitolato d'appalto Automated EN18031 Compliance Verification di BlueWind](#)
- [Standard ISO 12207:2017](#)

### 2.2) Riferimenti informativi

- <https://docs.github.com/en/issues/tracking-your-work-with-issues/using-issues/managing-issue-types-in-an-organization>
- <https://www.agileway.it/definition-of-done-dod-cose-cosa-serve/>
- <https://www.atlassian.com/agile/project-management/definition-of-done>

## 3) Processi Primari

### 3.1) Processo di fornitura

#### 3.1.1) 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/IEC 12207:1997** (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.

#### 3.1.2) Scopo e descrizione

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

- definire i canali ufficiali di comunicazione;
- stabilire frequenza e modalità dei confronti;
- assicurare che le richieste della proponente siano comprese, tracciate e integrate nel flusso di lavoro;
- produrre la documentazione necessaria alle revisioni e alle fasi di progetto.

Il rapporto tra committente e proponente si basa su interazioni periodiche e sul rispetto dei ruoli previsti dal capitolato, con l'obiettivo di garantire conformità alle richieste espresse e mantenere una visione condivisa dello stato del progetto.

#### 3.1.3) Rapporti con la proponente

Le interazioni con la proponente avvengono attraverso modalità sia sincrone che asincrone, garantendo continuità nelle comunicazioni.

##### Comunicazione sincrona (meeting)

- Riunioni periodiche fissate in base alle necessità del progetto;
- Gli incontri vengono documentati tramite **verbale esterno**;
- Decisioni, richieste e dubbi emersi vengono convertiti in issue GitHub.

##### Comunicazione asincrona

- Email indicata da BlueWind ;
- Eventuale uso di strumenti di messaggistica (telegram);
- Domande e chiarimenti tecnici possono essere richiesti tra una riunione e l'altra.

**Documentazione e tracciamento** Ogni comunicazione rilevante deve essere documentata. In particolare:

- I meeting generano sempre un verbale esterno;

- Le decisioni operative vengono tracciate tramite issue GitHub.

**Disponibilità** Il gruppo assicura di fornire aggiornamenti regolari e materiali richiesti entro i tempi stabiliti dal capitolato, rispettando le scadenze di sprint e di milestone.

### 3.1.4) Documentazione prodotta

Il processo di fornitura genera diversi documenti necessari a garantire tracciabilità e trasparenza nel rapporto con la proponente.

La documentazione prodotta comprende:

- **Verbali esterni:** registrazione formale dei meeting con BlueWind;
- **Norme di Progetto (NdP):** descrizione dei processi adottati e organizzazione interna;
- **Piano di Progetto (PdP):** pianificazione, disponibilità e avanzamento del lavoro;
- **Analisi dei Requisiti (AdR):** recepisce e formalizza le richieste della proponente;
- **Piano di Qualifica (PdQ):** descrive verifiche e criteri di qualità per dimostrare conformità.

Di seguito un elenco dei documenti forniti, con collegamento alla versione pubblicata:

- [Analisi dei Requisiti](#)
- [Piano di Progetto](#)
- [Piano di Qualifica](#)
- [Norme di Progetto](#)
- [Verbali esterni](#)

## 4) Processi di Supporto

### 4.1) Documentazione

Il processo di documentazione ha lo scopo di registrare e rendere disponibili le informazioni prodotte dai processi primari del progetto.

Consente di tracciare con maggiore efficacia le decisioni adottate, ridurre il rischio di ambiguità, evitare fraintendimenti e facilitare l'organizzazione del **lavoro asincrono e collaborativo**.

Le principali attività che compongono questo processo sono:

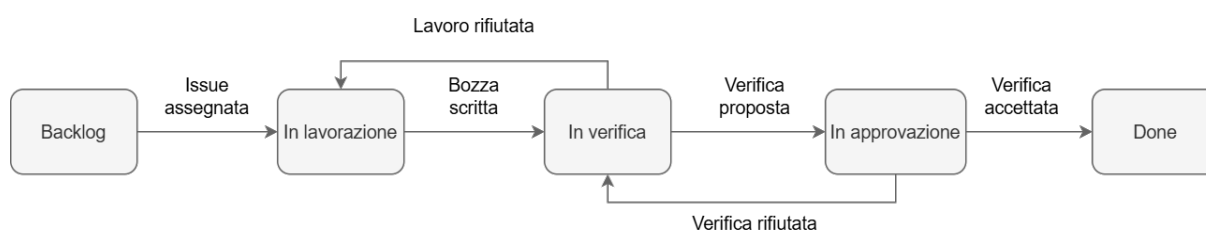
1. **Identificazione dei documenti** → individuazione dei documenti necessari e ne riferisce
2. **Progettazione dei documenti** → applicazione delle norme e del workflow stabilito;
3. **Pubblicazione e Distribuzione della Documentazione** → generazione del documento in formato destinato alla distribuzione;
4. **Manutenzione** → aggiornamento continuo del contenuto e gestione delle modifiche;
5. **Archiviazione e tracciabilità** → versionamento, conservazione e accessibilità nel tempo.

#### 4.1.1) Impegno del gruppo nella produzione della documentazione

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 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 coinvolti.

#### 4.1.2) Workflow documentale

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



##### 4.1.2.1) Stati del documento

- **Backlog**: magazzino delle attività da svolgere, ogni documento inizia in questo stato.

- **In lavorazione:** il documento è stato preso in carico da un autore.
- **In verifica:** 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.
- **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.

#### 4.1.2.2) Procedura di avanzamento tra stati

- Da **Backlog** a **In lavorazione:** un autore si assegna una issue e inizia a scrivere la bozza del documento.
- Da **In lavorazione** a **In verifica:** l'autore consegna la bozza, trasferendo la issue in revisione e assegnandola al revisore (deciso a priori) che verrà notificato automaticamente.
- Da **In verifica** a **In validazione:** il revisore ha apportato modifiche alla bozza e propone la revisione al validatore. Il revisore deve spostare la issue in validazione e assegnarla al validatore.
- Da **In validazione** a **In verifica:** il validatore rifiuta la revisione proposta allegando una lista di modifiche motivate che il revisore dovrà apportare al documento. Il validatore dovrà riassegnare la issue al revisore.
- Da **In validazione** a **Done:** il validatore accetta la revisione proposta e chiude la issue con

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

#### 4.1.3) Strumenti di supporto

**Typst:** 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 tracking**.

- Utilizzo di GitHub Actions 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:** Per velocizzare il processo, il gruppo utilizza il tool AI [TurboScribe](#) che consente di trascrivere facilmente da registrazioni audio, utile per funzionalità come il riascolto di momenti specifici della riunione tramite selezione testuale.

#### 4.1.4) Identificazione dei documenti

Si tratta di una fase 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.

##### 4.1.4.1) Elenco dei documenti

Documento 1	Analisi dei Requisiti .....	10
Documento 2	Piano di Progetto .....	11
Documento 3	Piano di Qualifica .....	12
Documento 4	Verbali .....	13
Documento 5	Diario di Bordo .....	14
Documento 6	Norme di Progetto .....	15
Documento 7	Glossario .....	15
Documento 8	.....	15

#### 4.1.5) 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 nell'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.

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.

#### 4.1.6) Pubblicazione e Distribuzione della Documentazione

I documenti vengono ricompilati automaticamente in PDF 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.

#### 4.1.7) Posizione del Documento

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 fanno supporto ad alcun processo primario, perciò sono salvati nella cartella `./<Type>src/DiariDiBordo`)

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

**Type:**

- **src** per i file in formato typst.
- **output** per i pdf.

**Milestone:**

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

**Destinatari:**

- DocumentazioneInterna per i documenti ad uso interno.
- DocumentazioneEsterna per i documenti ad uso esterno.

**Cartella del Documento:**

- VerbalInterni
- VerbalEsterni
- Per documenti complessi coincide con il nome del documento<sup>1</sup>

#### 4.1.8) 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:

---

<sup>1</sup>Per motivi di manutenibilità e facilità di aggiornamento i contenuti del file sono stati divisi in più file quando una loro sezione diventa eccessivamente corposa.

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.
5. **Indice delle immagini e delle tabelle** presente solo nei documenti che ne contengono.

#### 4.1.9) Documentazione prevista per la fase RTB

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 completa e trasparente dello stato del progetto.

In particolare, vengono prodotti:

##### **Documenti tecnici :**

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

##### **Documenti operativi e organizzativi**

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

#### 4.1.10) Struttura specifica dei documenti

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.

**4.1.10.1) Analisi dei Requisiti**

L'Analisi dei Requisiti 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 interni ed esterni al progetto (BlueWind S.r.l., docenti e gruppo interno)

**4.1.10.1.1) 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

#### 4.1.10.2) Piano di Progetto

Il Piano di Progetto definisce la pianificazione complessiva del lavoro, descrivendo l'approccio plan-driven adottato dal gruppo. Documenta obiettivi, risorse, ruoli, tempistiche e rischi, fornendo una visione strutturata dell'organizzazione del progetto. Ha inoltre lo scopo di monitorare l'avanzamento attraverso revisioni periodiche e rendicontazioni per sprint.

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

##### 4.1.10.2.1) Struttura principale

La struttura del documento comprende:

- Ambito e obiettivi del progetto;
- Preventivo iniziale e disponibilità delle risorse;
- Analisi dei rischi e piano di mitigazione;
- Distribuzione dei ruoli e pianificazione temporale;
- Sezione dedicata alle revisioni, con per ogni sprint:
  - **Retrospektiva del gruppo** → riflessioni su apprendimento, workflow ed efficacia della collaborazione
  - **Attività pianificate** → obiettivi e task previsti per il periodo di sprint
  - **Rischi e difficoltà emersi** → analisi degli impedimenti riscontrati e strategie di mitigazione
  - **Preventivo ore per ruolo** → stima dell'effort pianificato, suddiviso per responsabilità
  - **Consuntivo ore effettive** → ore realmente impiegate dal gruppo nel periodo

Documento 2: Piano di Progetto

**4.1.10.3) Piano di Qualifica**

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

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

**4.1.10.3.1) 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 3: Piano di Qualifica

#### 4.1.10.4) 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 GitHub**, che il gruppo si impegna a completare.

##### 4.1.10.4.1) Procedure e responsabilità

Il verbale deve essere un riassunto chiaro e oggettivo della riunione. Per velocizzare il processo, il gruppo utilizza il tool AI [TurboScribe](#).

##### 4.1.10.4.2) Struttura Verbale

Ogni verbale deve avere la seguente suddivisione numerata:

1. **Informazioni comuni della sezione 4.1.2.1** (standard condivisi di documento)
2. **Informazioni generali**
  - Data e luogo della riunione
  - Orario di inizio/fine
  - Partecipanti
  - Tipo(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.

**4.1.10.5) 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

#### 4.1.10.6) Norme di Progetto

Le Norme di Progetto hanno l'obiettivo di definire regole, convenzioni e standard condivisi all'interno del gruppo, al fine di garantire coerenza, qualità e uniformità nella produzione di documenti, codice e deliverable. Esse servono a stabilire procedure comuni per redazione, revisione, versionamento, gestione dei file e comunicazione interna, riducendo il rischio di errori, ambiguità o incongruenze tra i membri del team.

**Destinatari :** Tutti i membri del gruppo di progetto

##### 4.1.10.6.1) Struttura principale

Il documento delle Norme di Progetto è organizzato in sezioni facilmente consultabili riassunte in:

- **Convezioni sui documenti:** formato, intestazioni, footer, numerazione pagine, stili e template Typst da utilizzare.
- **Processo di redazione e revisione:** ruoli coinvolti, responsabilità, iter di approvazione, gestione modifiche e versionamento.
- **Standard di codifica e naming:** linee guida per nomi di file, classi, funzioni e repository, se applicabile al progetto software.
- **Procedure operative comuni:** modalità di collaborazione, uso di strumenti (GitHub, Issue Tracking, AI tools), backup e archiviazione.
- **Tracciabilità e registri:** indicazioni per mantenere il Registro delle Modifiche aggiornato e verificabile.
- **Definizione della qualità minima:** criteri generali per accettazione dei documenti e dei deliverable.

Documento 6: Norme di Progetto

#### 4.1.10.7) Glossario

Il glossario ha l'obiettivo di disambiguare le comunicazioni con l'esterno e tra i membri del gruppo.

Contiene l'elenco dei termini ritenuti non banali, le abbreviazioni o acronimi, e le loro definizioni.

il documento è sottoposto ad aggiornamento continuo.

Documento 7: Glossario

#### 4.1.10.8)

Documento 8:

#### **4.1.11) Manutenzione**

## 5) Processi Organizzativi

### 5.1) Gestione del Processo

#### 5.1.1) Ruoli di Progetto

La seguente sezione descrive le fasi della progettazione software. All'interno del team, per garantire coerenza, efficienza e qualità, ogni ruolo ha compiti specifici e interviene in momenti diversi del progetto.

Nel contesto del corso di Ingegneria del Software, tutti i membri del team devono ricoprire almeno una volta ciascun ruolo definito.

La tabella sottostante riassume in maniera chiara i compiti associati a ciascun ruolo.

#### Fasi di progetto

— ● **Analisi** → ● **Progettazione** → ● **Implementazione** → ● **Verifica**

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>- 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 procedure secondo norme</li> </ul>	Tutto il progetto
Verificatore	<ul style="list-style-type: none"> <li>- Testing e validazione</li> <li>- Controllo qualità deliverable</li> <li>- Conformità ai requisiti</li> </ul>	Tutto il progetto
Analista	<ul style="list-style-type: none"> <li>- Analisi dei requisiti</li> <li>- Definizione bisogni del sistema</li> <li>- Redazione specifiche funzionali</li> </ul>	Fase iniziale
Progettista	<ul style="list-style-type: none"> <li>- Progetta architettura sistema</li> <li>- Design e modellazione</li> <li>- Traduzione requisiti in struttura tecnica</li> </ul>	Dopo analisi
Programmatore	<ul style="list-style-type: none"> <li>- Codifica software</li> <li>- Implementazione design</li> <li>- Sviluppo funzionalità</li> </ul>	Implementazione

### 5.2) 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 = Verifica.
- A = Approvazione.
- C = Contribuente.

Ruoli:

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

Documento	RESP	AMM	ANL	PRG	PRGmm	VRF
Norme di Progetto (NdP)	R/A	R	-	-	-	V
Analisi dei Requisiti (AdR)	A	-	R	-	-	V
Piano di Progetto (PdP)	R/A	-	C (supporto rischi)	-	-	V
Piano di Qualifica (PdQ)	A	R	-	-	-	V
Design Document (DD)	A	-	C (per coerenza requisiti)	R	C	V
Manuale Utente (MU)	A	-	-	R	C	V
Verbali interni	R/A	R	-	-	-	V
Verbali esterni	R/A	R	-	-	-	V
Test Report (TR)	A	-	-	C	R	V
Documentazione tecnica interna	A	R (per strumenti e template)	R	C	-	V

### 5.3) Definition of Done (DoD)

La **Definition of Done (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 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 per la fase RTB:

- ☐ 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
- ☐ **Nei verbali:** Controllare di aver aggiornato nello status TAB:
  - stato
  - versione
  - ruoli
- ☐ Controllare di aver aggiunto le ultime modifiche anche sulla “tabella delle modifiche del documento”
- ☐ **Nei verbali:** controllare di aver aggiornato la versione nel nome del file
- ☐ **Nei verbali,** controllare che tutte le decisioni corrispondano a issue specifiche nell’issue template.
- ☐ Un documento (o una sua sezione) è considerato completato quando:
  - È stato scritto;
  - È stato verificato;
  - È stata aggiunta una riga nelle tabelle documentarie con il validatore finale.
- ☐ Quando il documento/prodotto è completato, chiudere la issue con

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

Verificare poi effettivamente la chiusura nel Projects Board.

- ☐ Quando tutti i punti sopra sono completati e tutte le issue sono spostate in “Done”:
  - Il branch develop può essere unito a main
  - Controllare l’incremento dello sprint corrispondente ( e il website)

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

## 5.4) Issue tracking System – Guida Operativa

L’**Issue Tracking System** è lo strumento utilizzato dal nostro team di sviluppo per tracciare in maniera efficiente tutte le issue da svolgere e il loro stato di completamento. Il sistema è accessibile a tutti i membri del gruppo attraverso la repository GitHub, dove è disponibile un **template di issue condiviso e centrale**, in modo da evitare incongruenze o confusione.

### 5.4.1) Creazione di una nuova issue

A seguito di verbali interni o esterni, il gruppo decide le attività su cui concentrarsi. L’**amministratore** ha il compito di creare le issue nel sistema utilizzando il **template condiviso**.

Ogni nuova issue deve includere:

**1. Assegnatario/i**

Generalmente è preferibile assegnare la issue a una sola persona. Tuttavia, per attività di formazione o esercitazioni («palestra») si possono assegnare più persone o l'intero gruppo.

**2. Descrizione**

Una spiegazione dettagliata e specifica delle azioni da svolgere.

**3. Scopo**

Indica cosa ci si aspetta di ottenere al termine dell'issue e dove andrà documentato il risultato (ad esempio, in quale documento o sezione del repository).

**4. Autore**

La persona che deve svolgere la issue.

**5. Verificatore**

La persona incaricata di verificare che la issue sia stata risolta correttamente, in base alla Definition of Done. Il verificatore, a meno di eccezioni, è diversa dall'autore.

**6. Label (ambito/destinazione)**

Questa classificazione consente di organizzare le issue in base al loro ambito o posizione all'interno del progetto:

- Analisi dei requisiti
- Piano di progetto
- Piano di qualifica
- Norme progetto
- Verbale
- Diario di bordo
- Glossario
- Generale -> attività che non rientrano nei documenti sopra: studio di materiale aziendale, lavori sul sito web, gestione repository, attività varie fuori dai documenti principali

**7. Tipo di issue (Type)**

Permette di distinguere la natura delle attività:

- Palestra -> ore formative non rendicontate, attività di ricerca o di studio
- Produttivo -> ore rendicontate con risultati concreti, come la scrittura di documenti da presentare
- Bug -> errori o malfunzionamenti nel codice;
- Correzione -> interventi su documenti o materiali già prodotti per migliorarli, aggiornarli o correggere inesattezze;

#### 8. **Priorità : Bassa, Media, Alta**

Questa suddivisione ha due scopi:

- ragionare sull'importanza della issue che si sta scrivendo
- comunicare all'assegnatario con quale tempestività dovrà svolgere la issue

#### 9. **Dimensione : ExtraSmall, Small, Medium, Large**

Serve per stimare la mole di lavoro necessaria per portare a termine quella issue.

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

Normalmente coincide con la fine dello sprint di riferimento.

### 5.4.2) **Flusso operativo**

1. L'amministratore crea una nuova issue tramite il template condiviso.
2. Si assegnano autore/i e verificatore/i.
3. Si compilano descrizione, scopo, label, type, priorità, dimensione, scadenza.
4. La issue viene inserita nello stato iniziale Backlog e segue il flusso fino a Done.

### 5.4.3) **Versionamento**

In questa sezione viene spiegata la logica di versionamento dei documenti.

#### 5.4.3.1) **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:

- **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 proposta / Approvazione generale**

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 della feature proposta / Modifica verificata**

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

#### 5.4.3.2) **Regole di incremento**

- Ogni approvazione genera un incremento della cifra di versione stabile.

- 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
- L'incremento di una cifra comporta sempre l'azzeramento delle cifre alla sua destra, mantenendo coerenza nella progressione delle versioni.

## 6) Metriche e standard per la Qualità

## 7) Metriche di Qualità del Processo

## 8) Metriche di Qualità del Prodotto

## 9) Best Practices

In questa sezione vengono riportate le best practices e pratiche standard concordate col gruppo al fine di garantire coerenza all'interno dei documenti.

### 9.1) Formato nome dei verbali

Al fine di avere ordine estetico all'interno della repo, 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

### 9.2) Scrittura dei commit

[fonte interessante](#)

I commit dovrebbero avere un tipo ed una descrizione: il tipo indica qual è l'obiettivo del commit, mentre la descrizione aiuta il lettore a comprendere meglio quali cambiamenti sono stati effettuati.

Le regole generali sono:

- iniziare il commit con tipo seguito da «:» .
- lasciare uno spazio tra tipo e descrizione.
- iniziare la descrizione con una lettera maiuscola.
- limitare la descrizione a massimo 50 caratteri.
- indicare alla fine del commit la issue a cui ci si sta riferendo con:

```
git commit -m "Commento.Issue #01"
```

Nel caso sia necessario modificare un commit (ad esempio in caso di errori) si utilizza il seguente comando

```
git commit --amend
```

N.B. : 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.

### 9.3) Issue Template