



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>	0.5
<b>Autori</b>	Davide Lorenzon Aldo Bettega Guerra Filippo Ana Maria Draghici
<b>Verificatori</b>	
<b>Uso</b>	Interno
<b>Destinatari</b>	Tutto il gruppo

Vers.	Data	Autore	Verificatore	Descrizione
0.5	2025-11-30	Ana Maria Draghici	-	Aggiunta sezione Definition of Done e Issue tracking System
0.4	2025-11-29	Guerra Filippo	Ana Maria Draghici	Aggiunta sezione ruolo-documento
0.3	2025-11-11	Ana Maria Draghici	-	Aggiornata struttura RE con le modifiche alla scaletta AdR
0.2	2025-11-11	Davide Lorenzon	-	Aggiunta struttura dei documenti (come stabilito da verbale 2025-11-10)
0.1	2025-11-	Davide Lorenzon	-	Stesura iniziale

# Indice

<b>1) Introduzione .....</b>	<b>1</b>
1.1) Scopo del documento .....	1
1.2) Scopo del prodotto .....	1
1.3) Glossario .....	1
<b>2) Riferimenti .....</b>	<b>2</b>
2.1) Riferimenti normativi .....	2
2.2) Riferimenti informativi .....	2
<b>3) Processi Primari .....</b>	<b>3</b>
<b>4) Processi di Supporto .....</b>	<b>4</b>
4.1) Documentazione .....	4
4.1.1) Strumenti di supporto .....	4
4.1.2) Identificazione dei documenti .....	4
4.1.2.1) Informazioni comuni .....	5
4.1.2.2) Verbali .....	5
4.1.2.3) Diario di Bordo .....	6
4.1.2.4) Piano di Progetto .....	6
4.1.3) Progettazione dei documenti .....	7
4.1.4) Produzione .....	7
4.1.5) Manutenzione <sub>G</sub> .....	7
<b>5) Processi Organizzativi .....</b>	<b>8</b>
5.1) Gestione del Processo .....	8
5.1.1) Ruoli di Progetto .....	8
5.2) Assegnazione Ruolo-Dокументo .....	8
5.3) Definition of Done (DoD) .....	9
5.4) Issue tracking System – Guida Operativa .....	10
5.4.1) Creazione di una nuova issue .....	10
5.4.2) Flusso operativo .....	11
<b>6) Metriche e standard per la Qualità .....</b>	<b>12</b>
<b>7) Metriche di Qualità del Processo .....</b>	<b>13</b>
<b>8) Metriche di Qualità del Prodotto .....</b>	<b>14</b>

# 1) Introduzione

## 1.1) Scopo del documento

In questo documento si tiene traccia e si definisce il way of working adottato dal gruppo.

La modalità di stesura di questo documento è incrementale, essa procede con l'avanzamento delle attività di progetto.

Il documento sarà soggetto ad aggiunte, modifiche o cancellazioni.

Esse sono causate dall'apprendimento e dalla sperimentazione da parte del gruppo delle «Best practices<sup>G</sup>»<sup>1</sup> comuni nell'ambito dell'ingegneria del software, in modo da poterle adattare alle necessità del team e del progetto.

## 1.2) Scopo del prodotto

Il prodotto è un'applicazione software per la verifica<sub>G</sub> automatizzata della conformità<sub>G</sub> alla norma EN18031, uno standard tecnico europeo per la sicurezza informatica dei dispositivi radio (Wi-Fi<sub>G</sub> , LTE<sub>G</sub> , BT<sub>G</sub> , IoT<sub>G</sub> wireless ).

L'obiettivo è guidare l'utente nella valutazione dei requisiti di cybersecurity<sub>G</sub> attraverso l'esecuzione automatizzata di decision tree<sub>G</sub> , velocizzando e standardizzando il processo di verifica<sub>G</sub> della conformità<sub>G</sub> e generando la documentazione necessaria.

## 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.

---

<sup>1</sup>Procedura di cui è dimostrata la capacità di fornire sicurezza, consistenza e qualità per il corrispondente caso d'uso.

## 2) Riferimenti

### 2.1) Riferimenti normativi

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

### 2.2) Riferimenti informativi

### **3) Processi Primari**

## 4) Processi di Supporto

### 4.1) Documentazione

Il processo di documentazione ha lo scopo di registrare le informazioni prodotte da processi primari.

Permette di tracciare più facilmente le decisioni intraprese, evitare fraintendimenti, favorire l'organizzazione del lavoro asincrono.

Le attività costituenti di questo processo sono le seguenti:

- 1) Implementazione del processo
- 2) Progettazione e sviluppo
- 3) Produzione
- 4) Manutenzione

#### 4.1.1) 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.

Ha una sintassi più intuitiva e supporta automazioni tramite template, funzioni e regole di stile riutilizzabili.

La caratteristica più utile è la preview istantanea del documento.

- **GitHub:**

Il gruppo ha deciso di usare GitHub come strumento di condivisione del lavoro e il suo sistema di issue tracking system per l'assegnazione delle attività. Viene inoltre usata una GitHub action per la compilazione automatica dei documenti. Tutta la documentazione è reperibile nel seguente [repository](#). È stato predisposto un sito web, sfruttando la funzionalità GitHub pages per facilitare la consultazione della documentazione.

#### 4.1.2) Identificazione dei documenti

È una fase di pianificazione.

Quasi ogni documento viene identificato dalle seguenti caratteristiche

- a- Titolo
- b- Scopo
- c- Destinatari
- d- Procedure e responsabilità nell'attività di documentazione.

**Input:** Quali dati servono per scriverlo

**Sviluppo:** Chi lo redige

**Verifica:** Chi lo controlla

**Modifica:** Chi può aggiornarlo

**Approvazione:** Chi lo approva

**Produzione:** Come viene prodotto

**Archiviazione e distribuzione:** Strumenti e tecniche

**Manutenzione e gestione della configurazione:** Versioni, tracciabilità

e- Pianificazione di revisioni intermedie e finali

#### 4.1.2.1) Informazioni comuni

Ogni documento ha una sezione iniziale costituita dai seguenti elementi.

**Pagina di copertina:** Costituito da nome e logo del gruppo, informazioni di contatto, titolo del documento.

Generata con apposito template.

**Tabella dello stato:** Tabella riassuntiva dello stato del documento.

Generata con apposito template.

**Registro delle modifiche:** Tabella contenente le informazioni sul verisonamento e sulla tracciabilità.

Generata con apposito template.

**Indice dei contenuti:** Indice generato automaticamente da typst

**Indice delle immagini e delle tabelle:** Questi indici non vengono inseriti in tutti i documenti, anche questi generati automaticamente.

#### 4.1.2.2) Verbali

**Titolo:** Verbale riunione *data della riunione*.

**Scopo:** Ha una finalità riassuntiva della riunione mettendo in evidenza l'agenda, le decisioni e le attività identificate (TODO).

**Destinatari:** Membri del gruppo, se il verbale è esterno anche la proponente, BlueWind.

**Procedure e responsabilità:** Per la redazione del documento è necessario un riassunto della riunione (ottenuto sia da appunti presi manualmente, sia dalla trascrizione da parte di tool AI).

La redazione è a carico del Responsabile e la verifica a carico del verificatore. L'approvazione è a carico del responsabile. I verbali sono salvati nel repository GitHub seguendo il path:*src/Fase di avanzamento/DocumentazioneEsterna o interna(in base al tipo di riunione)/VerbaliEsterni o interni/data della riunione\_Verbale-vx.y.z.typ*.

**Revisioni:** Il verbale viene revisionato e approvato quando pronto

**Struttura:**

0) Informazioni Comuni **Sezione 4.1.2.1**

1) Informazioni Generali

Data e luogo della riunione, orario di inizio e fine, partecipanti, Scriba della riunione, motivo della riunione.

- 2) Ordine del giorno
- 3) Riassunto della riunione
- 4) Decisioni
- 5) TODO

#### 4.1.2.3) Diario di Bordo

Nell'ambito del progetto di Ingegneria del Software, il Prof. Tullio Vardanega ha previsto delle attività volte ad accertare, in maniera condivisa, lo stato di progresso per ogni gruppo.

**Titolo:** Diario di bordo *data*.

**Scopo:** Permette ai gruppi di avere un feedback sulle attività svolte e di portare all'attenzione di tutti i dubbi sulle attività di processo

**Destinatari:** Prof. Tullio Vardanega, altri gruppi.

**Procedure e responsabilità:** La redazione è a carico del Responsabile e la verificaG a carico del verificatore. L'approvazione è a carico del responsabile. I diari di bordo sono salvati nella sezione src/DiariDiBordo/Data Data\_Diario\_di\_bordo.typ

**Revisioni:** Revisionato e approvato entro

**Struttura:**

- 1) Principali traguardi raggiunti
- 2) Attività da completare
- 3) Difficoltà incontrate
- 4) Principali dubbi

#### 4.1.2.4) Piano di Progetto

**Titolo:** Piano di Progetto

**Scopo:** Documento necessario ad attuare uno sviluppo plan-driven.

Le principali informazioni che contiene sono l'ambito del progetto, preventivo iniziale, risorse disponibili, analisi e piano di gestione dei rischi, suddivisione e scheduling del lavoro.

**Destinatari:** Interni, esterni ( BlueWind s.r.l. , professori )

**Procedure e responsabilità:** La redazione del documento è in carico al responsabile, ? analista per l'analisi dei rischi ?. La verificaG è a carico del verificatore. L'approvazioen è a carico del responsabile. Il piano di progetto è salvato nel seguente path src/RTB/DocumentazioneEsterna/Piano\_di\_progetto-vx.y.z.typ

**Revisioni:** Sono previste revisioni intermedie al termine di ogni periodo e un'approvazione finale per la RTB.

#### 4.1.3) Progettazione dei documenti

Ogni documento identificato deve rispettare gli standard di documentazione: formato A4, i contenuti devono essere coerenti con lo scopo del documento, definito nella **Sezione 4.1.2**, un indice dei contenuti (eccetto il diario di bordo), l'header e il footer della pagina deve contenere le seguenti informazioni:

- Sezione attuale del documento
- Nome del gruppo
- Titolo del documento
- Pagina corrente, espresse in numeri romani per la «prefazione», in numeri arabi per il corpo del documento

In caso si usino fonti non autorevoli è obbligatorio verificarne la correttezza. Esempi di fonti autorevoli: Standard ISO o di altre organizzazioni autorevoli, materiale delle lezioni

TODO: stabilire una checklist per la verifica ed eventuali strumenti di approvazione o revisione automatica o semi automatica

#### 4.1.4) Produzione

I documenti sono compilati automaticamente da una GitHub action. Sono visibili presso la repository [Documentazione](#). Per facilitare la consultazione è possibile usare l'apposito sito web: <https://grouprubberduck.github.io/Documentazione>

Versionamento e tracciabilità vengono realizzati attraverso il registro delle modifiche.

#### 4.1.5) 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 —> • VerificaG

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>- Verificag procedure secondo norme</li> </ul>	Tutto il progetto
Verificatore	<ul style="list-style-type: none"> <li>- Testing e validazioneG</li> <li>- Controllo qualità deliverable</li> <li>- ConformitàG 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-Dокументo

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
Piano di Progetto (PdP)	R/A	-	C (supporto rischi)	-	-	V
Piano di Qualificag (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 <sub>G</sub> (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
- Controllare di aver aggiornato la versione, lo stato e gli autori ( “status TAB” ), per includere le ultime modifiche
- Controllare di aver aggiunto le ultime modifiche anche sulla “tabella delle modifiche del documento”
- 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 l’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 ha ideato l'issue (solitamente l'amministratore).

**5. Verificatore**

La persona incaricata di controllare e approvare l'issue quando viene spostata nello stato di verifica.

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

Permette di classificare la issue in base al suo ambito:

- Esterno -> documentazione rivolta anche all'esterno del gruppo
- Interno -> documentazione utile solo all'interno del gruppo

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

Permette di distinguere la natura delle attività:

- Palestra -> ore formative non redicontate, attività di ricerca o di studio
- Produttivo -> ore redicontate con risultati concreti, come la scrittura di documenti da presentare
- Bug -> correzione di errori, sia su documenti che su codice o sistemi interni.

**8. Priorità : Bassa, Media, Alta**

Il gruppo concentra le proprie energie prima sulle issue ad alta priorità.

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

Serve per stimare la complessità o l'impegno richiesto.

**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, verificatore e assegnatario/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.

## 6) Metriche e standard per la Qualità

## 7) Metriche di Qualità del Processo

## 8) Metriche di Qualità del Prodotto