

## Way of working

Versione	Data	Descrizione	Autore	Verificatore	Validatore
1.0.0	15-12-2024				Annalisa Egidi
0.2.3	06-12-2024	Verifica sezioni . utilizzo template - Documentazione/ Processi Modifica, Verifica, Approvazione		Michele Nesler	
0.2.3	23-11-2024	Documentazione/ Processi - Aggiunte in sezioni Modifica, Verifica e Approvazione	Filippo Bellon		
0.2.2	19-11-2024	Documentazione/ Utilizzo template - Tolto segretario verbalizzante dai verbali	Michele Nesler		
0.2.1	18-11-2024	Aggiunta sezione Utilizzo template	Alberto Maggion		
0.2.0	18-11-2024	Verifica ultime aggiunte		Alberto Maggion	
0.1.1	13-11-2024	1.6 - Aggiunta sezione Github repo	Filippo Bellon		
0.1.0	13-11-2024	Verifica documento		Filippo Bellon	
0.0.3	12-11-2024	1.1.2 - Modifica secondo verbale 05-11-2024 decisione 3.1	Alberto Maggion		
0.0.2	23-10-2024	1.2 - 2.1 - 1.4 - Aggiunta tabella Riepilogo Modifiche	Michele Nesler		
0.0.1	22-10-2024	Prima stesura	Alberto Maggion		

### Abstract

Questo documento serve a fornire le informazioni necessarie ad ogni membro del team riguardo i metodo di lavoro adottati e tutte le procedure da applicare.

## Contents

<b>1 Documentazione .....</b>	<b>3</b>
1.1 Struttura documento .....	3
1.1.1 Titolo del documento .....	3
1.1.2 Tabella versione del documento .....	3
1.2 Versionamento dei documenti .....	3
1.2.1 Regole cambiamento versione .....	3
1.3 Nomenclatura .....	4
1.3.1 Documenti .....	4
1.3.2 Verbali .....	4
1.4 Processi .....	4
1.4.1 Modifica .....	4
1.4.2 Verifica .....	4
1.4.3 Approvazione .....	4
1.5 Verbali .....	5
1.5.1 Presenze .....	5
1.5.2 Esiti .....	5
1.5.3 Interni .....	5
1.5.4 Esterni .....	5
1.5.5 Esempio esplicativo .....	5
1.6 Utilizzo template .....	5
1.6.1 Verbali .....	5
1.6.2 Documenti .....	6
1.7 Github repository .....	6
1.7.1 Struttura .....	6
1.7.2 Caricamento documenti .....	6
1.7.3 Caricare modifiche alla Webpage .....	6

# 1 Documentazione

## 1.1 Struttura documento

Per permettere al gruppo di produrre documenti che siano coerenti tra loro a prescindere da quale elemento si occupa della stesura, è necessario che abbiamo tutti una stessa struttura. Inoltre ci sono alcune informazioni che devono essere presenti in tutti i documenti, tra cui titolo del documento e registro delle modifiche.

Per queste ragioni è stato deciso di mantenere una struttura unica per tutti i documenti, che sarà quindi composta nel seguente modo.

### 1.1.1 Titolo del documento

Breve indicazione del contenuto del documento e della data di riferimento in caso sia utile.

### 1.1.2 Tabella versione del documento

Contiene una **raccolta** storica di tutte le **versioni** del documento, tenendo traccia sia di chi si occupa delle modifiche, dei vari verificatori, del responsabile che approva e delle differenze con la versione precedente.

Il contenuto della tabella deve essere del tipo **LIFO**, ovvero deve contenere i numeri di versione dal maggiore al minore, rispecchiando le modifiche al documento in ordine cronologico inverso: dalla più recente alla più datata.

L'unico documento da considerare **valido** in fase di consultazione è quello con numero di **versione maggiore** e che non ha subito nessuna modifica dopo l'approvazione, ovvero in documento con numero di versione del tipo **x.0.0**, in caso fossero presenti diversi documenti che corrispondono a questo tipo di versione, quello da consultare sarà quello con **x** maggiore.

## 1.2 Versionamento dei documenti

Il versionamento dei documenti tiene traccia dei seguenti stati possibili della documentazione:

**documento approvato**, **documento verificato** e **documento modificato**. Per questo motivo la versione dei documenti è del tipo "**a.b.c**" dove "**a**" rappresenta l'ultima approvazione del documento, "**b**" ne rappresenta l'ultima verifica dopo un'approvazione e "**c**" rappresenta l'ultima modifica dopo una verifica o un'approvazione. In questo modo risulta molto semplice riconoscere lo stato di un documento e capire quali sono quelli da considerare validi applicando le seguenti regole:

- **Documento modificato**  
c è un qualsiasi numero diverso da 0
- **Documento Verificato**  
b è un qualsiasi numero diverso da 0 e c è 0
- **Documento Approvato**  
b e c sono 0 e a è un qualsiasi numero diverso da 0

### 1.2.1 Regole cambiamento versione

Un documento può sempre essere **modificato**, a patto di **aggiungere una riga** alla tabella "versione del documento" aggiornando coerentemente il numero di versione e che tale modifica sia supportata da un relativo ticket. **Per poter procedere con una verifica il documento deve essere stato modificato dopo l'ultima verifica o approvazione. Per poter essere approvato un documento deve essere stato verificato e non aver subito alcun tipo di modifica.**

In caso di correzioni di semplici errori ortografici o di battitura, questi non necessitano né di un apposito ticket né dell'aggiunta di tali correzioni nella tabella di versione del documento.

### 1.3 Nomenclatura

Al fine di poter tenere comodamente traccia di tutte le diverse versioni dei **documenti**, è necessario adottare una **convenzione sui nomi** da dare ad essi nel momento del salvataggio. Il nome del documento deve essere parlante nel contesto in cui esso è contenuto. Il **nome** del file può essere diviso in **due parti**, la **prima** contiene informazioni sul **contenuto** dello stesso, mentre la **seconda differisce** tra documenti e verbali.

#### 1.3.1 Documenti

Nei documenti la seconda parte è separata dalla prima dal **simbolo** ( \_ ) e contiene la **versione** nel formato a.b.c che corrisponde alla versione attuale del documento.

#### 1.3.2 Verbali

Nei verbali la seconda parte contiene la **data** in cui è avvenuto l'incontro, nel formato yyyy-mm-dd, in modo da poter essere ordinati in ordine cronologico.

Nel caso di verbali **esterni**, quando questi vengono **approvati** dal committente, deve essere inserita la dicitura ( \_approved ) dopo la data.

### 1.4 Processi

Ogni **modifica** alla documentazione deve essere accompagnata da un **ticket** che ne specifica le **motivazioni**. Inoltre ogni documento per poter essere considerato **valido** deve aver superato almeno una fase di modifica, verifica e approvazione. Nei punti seguenti verrà spiegato più nel dettaglio in cosa consistono queste operazioni.

#### 1.4.1 Modifica

Il processo di modifica include sia la **modifica/correzione** di parti del testo che la parte di creazione di **nuovi contenuti**. Nella fase di creazione dei contenuti, al fine di ottimizzare il tempo necessario alla produzione dei documenti, non è necessario preoccuparsi troppo della forma, quanto invece nell'esprimere i **contenuti in modo chiaro**. Nei passaggi successivi questi verranno corretti sia nella forma che nella sintassi.

Nella tabella delle versioni, la persona assegnata a questo compito viene indicata all'interno della colonna **Autore**.

#### 1.4.2 Verifica

Il processo di verifica consiste nella lettura del testo prodotto e nell'eventuale **correzione di errori** ortografici rilevati nonché nella modifica della struttura del documento al fine di rendere l'argomento trattato di **facile consultazione**.

Nel caso in cui certi punti dovesse risultare poco chiari, si procederà ad una o più modifiche successive al fine di esprimere tali concetti in modo migliore.

Nella tabella delle versioni, la persona assegnata a questo compito viene indicata all'interno della colonna **Verificatore**.

#### 1.4.3 Approvazione

Il processo di approvazione consiste nella **lettura** del documento prodotto e nella **verifica della conformità** con i punti che questo dovrebbe trattare.

In caso vengano rilevati errori nei concetti espressi/mancanti, il documento tornerà alla fase di modifica.

Nella tabella delle versioni, la persona assegnata a questo compito viene indicata all'interno della colonna **Validatore**.

## 1.5 Verbali

In quanto tipologia differente di documento, i verbali necessitano di alcune **informazioni aggiuntive** tra cui la data dell'evento riportato, le presenze e le decisioni prese.

### 1.5.1 Presenze

Nella **prima parte** del verbale è necessario specificare i **nominativi** delle persone presenti e degli assenti in tale data, in modo da tenere traccia delle partecipazioni. Deve anche essere specificato, in caso di verbali esterni, quali **persone** appartengono al **gruppo di lavoro** e quali all'**entità esterna** con il quale si sta discutendo.

### 1.5.2 Esiti

Ogni **decisione** presa nella riunione andrà riportata nel verbale, includendo le motivazioni e gli eventuali punti di forza e di debolezza emersi. In caso di **contrast**i che richiedono una votazione, andrà riportato l'esito di questa insieme alla tabella che specifica chi ha votato a favore, chi contro e gli astenuti.

### 1.5.3 Interni

I verbali interni andranno sempre **approvati** dal responsabile, ma tale approvazione potrà essere discussa nella riunione successiva. In caso si riscontrino **errori**, si procederà ad identificare le parti da correggere, e alla successiva correzione.

In ogni caso, un verbale è da considerare **valido** dal momento della prima approvazione.

### 1.5.4 Esterni

I verbali esterni necessiteranno anche dell'**approvazione** dell'entità esterna che ha partecipato all'incontro, dopo aver superato l'approvazione interna.

### 1.5.5 Esempio esplicativo

Dato il caso di un nuovo documento, la sua prima stesura avrà versione **0.0.1**.

Facendo finta che non abbia bisogno di ulteriori modifiche, il documento procede con una prima verifica, che va a buon fine producendo il documento con versione **0.1.0**.

Procedendo verso la fase di approvazione viene evidenziato che nel documento un argomento non è trattato in modo completo; si procede quindi con una nuova modifica, arrivando alla versione **0.1.1**. A questo punto viene verificato nuovamente, senza riscontrare errori e si procede con la versione **0.2.0**.

Un'ultima fase di approvazione viene poi effettuata e superata, producendo quindi la prima versione ufficiale del documento, ovvero la versione **1.0.0**.

## 1.6 Utilizzo template

Per impostare tutti i documenti prodotti in modo coerente tra loro e conforme alle norme del Way of Working, sono stati prodotti dei template che richiedono che siano presenti tutti gli elementi necessari per poter produrre un documento.

I template sono impostati in modo da non dover essere modificati. Per fare ciò sono affiancati da un file esterno che comprende tutte le dipendenze che il documento richiede.

### 1.6.1 Verbali

Per i verbali viene utilizzato il template "interni" o "esterni" a seconda del tipo di richiesto. I contenuti necessari sono:

1. **Titolo:** titolo del documento
2. **Membri e versione:** file in formato json contenente i valori necessari per produrre la tabella di versione del documento e quella delle presenze
3. **Data:** data in cui si è svolto il verbale utilizzando la funzione date di Typist

4. **Abstract:** breve sintesi dei punti di discussioni più importanti e delle decisioni prese
5. **Odg:** lista degli argomenti da discutere durante l'incontro
6. **Discussione:** analisi dei punti descritti nel odg contenente i punti salienti della discussione, le decisioni prese e le motivazioni a supporto di queste
7. **Argomenti rimandati:** lista degli argomenti presenti nell odg che non si è potuto concludere in modo soddisfacente, in termini di tempo o conoscenze, e tematiche nuove emerse durante l'incontro da discutere in futuro.

### 1.6.2 Documenti

Per i documenti viene utilizzato il template "documento" che richiede i seguenti elementi:

1. **Titolo:** titolo del documento
2. **Abstract:** breve sintesi del contenuto o scopo del documento
3. **Dati versione:** file in formato json che contiene i dati necessari alla produzione della tabella di versione del documento
4. **Contenuto:** contenuto del documento

## 1.7 Github repository

Qui di seguito vengono raccolte varie informazioni riguardanti l'utilizzo e l'attuale configurazione della repository di Github.

### 1.7.1 Struttura

Il repository si compone di due rami("branch"):

1. **master:** ramo che contiene tutta la **documentazione** del progetto e composto nel seguente modo:
  - **Assets:** contenente file aggiuntivi, quali immagini e loghi riguardanti il gruppo Rod2Cod ed il progetto
  - **Candidatura:** contenente i vari file riguardanti le candidature presentate e da presentare per il progetto in essere
  - **Verbali:** contenente i file redatti in seguito a riunioni interne e esterne del gruppo
  - **WOW:** contenente i file che andranno a descrivere il Way of Working adottato dal gruppo
2. **gh-pages:** ramo che contiene tutti file di configurazione necessari a Jekyll e alle Github Actions per poter impostare la Webpage della repository.

### 1.7.2 Caricamento documenti

Per il caricamento di documenti e/o modifiche a questi ultimi, come prima cosa, si effettua la **copia** della repository in **locale**(`git clone <nome_repo>`) e ci si sposta dentro la cartella così creata.

Successivamente ci si sposta nel ramo **master**(`git checkout master`), si effettuano le modifiche/aggiunte che servono e si vanno a **modificare** anche i file di configurazione di **Jekyll** e delle **Github Actions** per includere eventuali nuove cartelle e pdf all'interno della Webpage.

**IMPORTANTE:** i documenti devono essere verificati e approvati prima di essere caricati all'interno della repository di Github.

Infine, dopo aver **aggiunto** le modifiche effettuate(`git add .`) e aver fatto il **commit** di queste ultime(`git commit -m <messaggio_commit>`), si effettua un'operazione di **push**(`git push <destinazione> <master>`) in modo da caricarle anche nella **repo** presente su **Github**.

### 1.7.3 Caricare modifiche alla Webpage

Per il caricamento di modifiche e/o aggiunte alla Webpage, anche qui, come prima cosa, si effettua la **copia** della repository in **locale**(`git clone <nome_repo>`), ci si sposta dentro la cartella così creata e

nel ramo di interesse. Successivamente ci sposta nel ramo **gh-pages**(`git checkout gh-pages`) e si effettuano le modifiche/aggiunte che servono. Infine, dopo aver **aggiunto** le modifiche effettuate(`git add .`) e aver fatto il **commit** di queste ultime(`git commit -m <messaggio_commit>`), si effettua un'operazione di **push**(`git push <destinazione> <gh-pages>`) in modo da caricarle anche nella repo presente su Github.