



# Error\_\_418

[GitHub/Error-418-SWE](#)

[error418swe@gmail.com](mailto:error418swe@gmail.com)

---

## Norme di Progetto

Norme, processi e disciplina

---

### Informazioni

---

|                     |  |
|---------------------|--|
| <b>Versione</b>     | 1.0.0  |
| <b>Uso</b>          | Interno  |
| <b>Stato</b>        | Approvato  |
| <b>Responsabile</b> | Carraro Riccardo   |
| <b>Redattori</b>    | Banzato Alessio<br>Gardin Giovanni                       |
| <b>Verificatore</b> | Todesco Mattia   |
| <b>Destinatari</b>  | Gruppo Error__418<br>Vardanega Tullio<br>Cardin Riccardo |

### Registro delle modifiche

| Ver.  | Data              | PR | Titolo   | Redattore           | Verificatore        |
|-------|-------------------|----|--|---------------------|---------------------|
| 1.0.0 | 14-12-2023        | 92 | DOC-141 Redazione sezione Configuration Management process | Carraro<br>Riccardo | Todesco<br>Mattia   |
| 1.0.0 | 13-12-2023        | 92 | DOC-141 Redazione sezione Configuration Management process | Carraro<br>Riccardo | Gardin<br>Giovanni  |
| 1.0.0 | 10-12-2023        | 81 | DOC-140 Redazione paragrafo Decision management process    | Carraro<br>Riccardo | Todesco<br>Mattia   |
| 1.0.0 | 10-12-2023        | 81 | DOC-140 Redazione paragrafo Decision management process    | Carraro<br>Riccardo | Banzato<br>Alessio  |
| 1.0.0 | 09-12-2023        | 76 | DOC-139 Redazione Project assessment and control process   | Alessio<br>Banzato  | Riccardo<br>Carraro |
| 1.0.0 | 09-12-2023        | 76 | DOC-139 Redazione Project assessment and control process   | Alessio<br>Banzato  | Mattia<br>Todesco   |
| 1.0.0 | 09-12-2023        | 75 | Redazione Organizational/Project Enabling processes        | Riccardo<br>Carraro | Giovanni<br>Gardin  |
| 1.0.0 | 08-12-2023        | 75 | Redazione Organizational/Project Enabling processes        | Riccardo<br>Carraro | Giovanni<br>Gardin  |
| 1.0.0 | 03-12-2023<br>Sun | 60 | DOC-145 Human Resources management process                 | Giovanni<br>Gardin  | Alessio<br>Banzato  |
| 1.0.0 | 30-11-2023<br>Thu | 55 | DOC-109 Rimossa variabile legacy                           | Giovanni<br>Gardin  | Alessio<br>Banzato  |
| 1.0.0 | 30-11-2023<br>Thu | 49 | DOC-109 Scrivere la sezione Processi di accordo/Fornitura  | Alessio<br>Banzato  | Giovanni<br>Gardin  |
| 1.0.0 | 28-11-2023<br>Tue | 49 | DOC-109 Scrivere la sezione Processi di accordo/Fornitura  | Alessio<br>Banzato  | Giovanni<br>Gardin  |
| 1.0.0 | 30-11-2023<br>Thu | 48 | DOC-44 Processi di gestione dell'infrastruttura            | Giovanni<br>Gardin  | Alessio<br>Banzato  |
| 1.0.0 | 28-11-2023<br>Tue | 48 | DOC-44 Processi di gestione dell'infrastruttura            | Giovanni<br>Gardin  | Alessio<br>Banzato  |
| 0.0.1 |                   | 23 | Scrittura paragrafo sulla documentazione                   | Antonio<br>Oseliero |                     |
| 0.0.1 |                   | 23 | Scrivere il paragrafo sulla documentazione                 | Antonio<br>Oseliero |                     |
| 0.0.1 |                   | 7  | Pubblicazione Norme di Progetto                            | Rosario<br>Zaccone  |                     |

# Indice dei contenuti

|   |           |
|---|-----------|
| <b>1 Introduzione .....</b>                                   | <b>1</b>  |
| 1.1 Scopo del documento .....                                 | 1         |
| 1.2 Scopo del progetto .....                                  | 1         |
| 1.3 Glossario .....   | 1         |
| 1.4 Riferimenti .....   | 1         |
| 1.4.1 Riferimenti normativi .....                             | 1         |
| 1.4.2 Riferimenti informativi .....                           | 1         |
| <b>2 Processi di accordo .....</b>                            | <b>2</b>  |
| 2.1 Processo di fornitura .....                               | 2         |
| 2.1.1 Scopo e descrizione .....                               | 2         |
| 2.1.2 Rapporti con il proponente .....                        | 2         |
| 2.1.3 Documentazione prodotta .....                           | 2         |
| 2.1.4 Strumenti utilizzati .....                              | 4         |
| <b>3 Processi di supporto .....</b>                           | <b>4</b>  |
| 3.1 Documentazione .....                                      | 4         |
| 3.1.1 Scopo e descrizione .....                               | 4         |
| 3.1.2 Aspettative .....                                       | 4         |
| 3.1.3 Typst .....   | 4         |
| 3.1.4 Ciclo di vita .....                                     | 4         |
| 3.1.5 Template .....  | 5         |
| 3.1.6 Struttura dei documenti .....                           | 5         |
| 3.1.7 Convenzioni stilistiche .....                           | 6         |
| 3.1.8 Elenchi puntati .....                                   | 7         |
| <b>4 Processi di ciclo di vita .....</b>                      | <b>7</b>  |
| 4.1 Processi organizzativi abilitanti .....                   | 7         |
| 4.1.1 Processo di gestione dei modelli di ciclo di vita ..... | 7         |
| 4.1.2 Processi di gestione dell'infrastruttura .....          | 11        |
| 4.1.3 Processo di gestione delle Risorse Umane .....          | 13        |
| <b>5 Processi di gestione tecnica .....</b>                   | <b>14</b> |
| 5.1 Processo di pianificazione di progetto .....              | 14        |
| 5.1.1 Scopo .....   | 14        |
| 5.1.2 Attività .....  | 14        |
| 5.1.3 Piano di Progetto .....                                 | 16        |
| 5.2 Processo di valutazione e controllo di progetto .....     | 16        |
| 5.2.1 Scopo .....   | 16        |
| 5.2.2 Attività .....  | 17        |
| 5.3 Processo di gestione delle Decisioni .....                | 18        |
| 5.3.1 Scopo .....   | 18        |
| 5.4 Processo di gestione della configurazione .....           | 20        |
| 5.4.1 Scopo .....   | 20        |

---

|                           |    |
|---------------------------|----|
| 5.4.2 Versionamento ..... | 20 |
| 5.4.3 Tecnologie .....    | 21 |

# 1 Introduzione

## 1.1 Scopo del documento

Questo documento contiene le regole del *way of working* che disciplinano le attività di ogni membro del gruppo *Error\_418*. Queste regole mirano a garantire coerenza, uniformità e efficacia nel processo collaborativo, promuovendo un ambiente di lavoro strutturato e efficiente. L'approccio adottato per la redazione di questo documento è di natura incrementale. Ciò significa che il testo è soggetto ad aggiornamenti e revisioni continue al fine di integrare progressivamente le nuove norme, le *best practices* e i cambiamenti che emergono nel corso dello sviluppo del progetto. Questa flessibilità consente al gruppo di adattarsi prontamente alle dinamiche di lavoro e alle esigenze specifiche del contesto, garantendo un documento sempre allineato alle necessità attuali del team.

## 1.2 Scopo del progetto

Il capitolato C5, denominato *WMS3: Warehouse Management 3D* e aggiudicato al team, ha come obiettivo la realizzazione di un sistema di gestione di magazzino in tre dimensioni. L'applicazione sviluppata consentirà all'utente di accedere a un ambiente virtuale tridimensionale, dove potrà navigare all'interno di un magazzino e visualizzare gli oggetti presenti nelle scaffalature. L'utente avrà la possibilità di cercare specifici prodotti all'interno del magazzino, sfruttando la visualizzazione 3D per individuare rapidamente la posizione degli articoli desiderati, potrà modificare l'assetto del magazzino e inviare una notifica verso l'esterno in caso ci sia il bisogno di prelevare un articolo.

## 1.3 Glossario

Al fine di eliminare le ambiguità durante la consultazione dei documenti, verrà introdotto un Glossario, che rappresenta una raccolta delle definizioni dei termini più rilevanti, fornendo chiarimenti sui significati particolari associati a ciascun termine.

## 1.4 Riferimenti

### 1.4.1 Riferimenti normativi

- Capitolato d'appalto C5: *WMS3*

<https://www.math.unipd.it/~tullio/IS-1/2023/Progetto/C5.pdf>

### 1.4.2 Riferimenti informativi

- Documentazione Typst

<https://typst.app/docs/>

- Documentazione Git

<https://git-scm.com/docs>

- Documentazione Jira

<https://confluence.atlassian.com/jira>

- Documentazione Three.js

<https://threejs.org/docs/>

## 2 Processi di accordo

### 2.1 Processo di fornitura

#### 2.1.1 Scopo e descrizione

Il processo di fornitura ha come obiettivo l'offerta di un prodotto o servizio che soddisfi i requisiti concordati con proponente e committente. Tra quest'ultimi e il fornitore deve essere stabilito un accordo all'interno del quale vengono definiti i requisiti, le tempistiche e i costi da rispettare. Prima di allora, il fornitore avrà effettuato un'attenta analisi del progetto proposto e dei rischi annessi alla sua realizzazione, con relative linee guida per mitigarli.

#### 2.1.2 Rapporti con il proponente

Il dialogo tra il gruppo *Error\_418* e il proponente dovrà essere attivo e frequente fino al termine del progetto didattico, in modo che si riescano a raccogliere più feedback possibili riguardo la correttezza del lavoro svolto. Questa comunicazione avverrà in due modalità:

- 1) scritta, asincrona, utilizzata per:
  - comunicazioni di breve durata;
  - condivisione di verbali e materiali informativi;
  - coordinamento.
- 2) orale, sincrona, durante i quali si affronteranno:
  - feedback sul lavoro prodotto;
  - chiarimenti sul capitolato;
  - chiarimenti riguardo casi d'uso e requisiti.

I meeting avranno cadenza variabile, e saranno fissati al termine di altri incontri oppure via e-mail. Il contenuto di ogni incontro sarà raccolto all'interno del relativo verbale. Ognuno di questi verbali sarà validato dal proponente tramite l'apposizione di una firma, e sarà liberamente consultabile all'interno del repository GitHub del gruppo dedicato ai documenti (<https://github.com/Error-418-SWE/Documenti/tree/main>), al percorso `NomeMilestone/Documentazione esterna/Verbali`, dove `NomeMilestone` è uno tra:

- 1 - Candidatura;
- 2 - RTB;
- 3 - PB.

#### 2.1.3 Documentazione prodotta

In questa sezione viene descritta la documentazione prodotta dal gruppo nel processo di fornitura, la quale sarà resa disponibile al proponente, *Sanmarco Informatica*, e ai committenti, i professori Tullio Vardanega e Riccardo Cardin.

##### 2.1.3.1 Valutazione dei capitoli

Documento nel quale il gruppo ha analizzato le diverse proposte di progetto rese disponibili dai vari proponenti. Per ogni capitolato vengono presentati tre paragrafi:

- 1) **Descrizione:** vengono indicati i nominativi di proponente e committente, e viene presentato l'obiettivo del progetto;
- 2) **Dominio tecnologico:** vengono elencate le tecnologie consigliate dal proponente del capitolato;

3) **Considerazioni:** il gruppo dà la propria valutazione sul capitolato.

#### 2.1.3.2 Analisi dei rischi

Nel documento di Analisi dei rischi vengono presentati i rischi a cui il gruppo potrebbe essere esposto durante il periodo in cui lavora al progetto. Ogni rischio viene classificato secondo tre parametri:

- **Impatto:** esprime l'effetto generato dall'evento;
- **Probabilità:** esprime la probabilità del verificarsi del rischio;
- **Conseguenze:** effetti collaterali a breve o medio termine che il rischio può comportare.

Ad ogni rischio sono inoltre associate delle buone pratiche da seguire per mitigarlo.

#### 2.1.3.3 Preventivo dei costi

Nel Preventivo dei costi viene esposta una tabella che presenta una previsione riguardo il numero di ore di lavoro totali, per membro e per ruolo e viene fornito un calcolo del costo totale del progetto. Prima della tabella vengono spiegate le motivazioni che hanno portato alla suddivisione oraria individuata, effettuando una tripartizione del periodo di lavoro complessivo e analizzando ogni ruolo presente all'interno del team.

#### 2.1.3.4 Lettera di presentazione

Breve documento dove il gruppo si presenta e dichiara il suo impegno nello svolgimento del capitolato scelto. Viene dato un riferimento al repository dove si potranno trovare i documenti necessari alla candidatura e vengono dichiarati il costo della realizzazione del prodotto e la data di consegna prevista.

#### 2.1.3.5 Analisi dei Requisiti

In questo documento vengono raccolti tutti gli use case e requisiti individuati dal gruppo con il supporto del proponente. Ogni use case e requisito è identificato da un codice, così da essere facilmente individuabile e tracciabile. All'inizio del documento sono inoltre descritti i criteri di qualità che il gruppo ha seguito durante la redazione.

#### 2.1.3.6 Piano di Progetto

Documento che governa la pianificazione dell'avanzamento del progetto, determinando task e obiettivi da raggiungere e analizzando il lavoro svolto. È articolato in cinque sezioni:

- Rischi e loro mitigazione;
- Divisione temporale di sviluppo;
- Preventivo dei costi di realizzazione;
- Pianificazione del lavoro;
- Consuntivo del progetto.

#### 2.1.3.7 Piano di Qualifica

Nel Piano di Qualifica vengono fissati obiettivi di qualità e vengono descritti i processi necessari per conseguirli con relative procedure di controllo.

#### 2.1.3.8 Glossario

Nel Glossario vengono elencati e definiti in modo preciso tutti i termini rilevanti utilizzati all'interno del progetto. È un documento estremamente importante per evitare situazioni di ambiguità, e garantire così una corretta comprensione della documentazione da parte di tutte le parti coinvolte.

### 2.1.4 Strumenti utilizzati

In questa sezione sono indicati gli strumenti utilizzati dal gruppo nel processo di fornitura.

- **Zoom:** applicazione per videoconferenze;
- **Google slides:** servizio cloud offerto da Google, utilizzato dal gruppo per le presentazioni del diario di bordo, ovvero l'attività in cui il gruppo aggiorna il committente riguardo l'andamento del lavoro;
- **Jira:** Issue Tracking System utilizzato per la pianificazione del lavoro.

## 3 Processi di supporto

### 3.1 Documentazione

#### 3.1.1 Scopo e descrizione

Il processo di documentazione del lavoro ha molteplici obiettivi, il primo è quello di tenere traccia di ciò che è stato svolto e delle decisioni prese nell'ambito del progetto, il secondo è quello di normare le attività in modo che i componenti del gruppo possano lavorare in maniera uniforme ed ordinata. La documentazione è quindi un processo di supporto all'attività di gestione di progetto, essendo un mezzo che, tramite versionamento e integrazione continua, norma attività di codifica, progettazione, analisi ed integrazione.

Questa sezione descrive tutte le norme riguardanti la stesura, il mantenimento e l'aggiornamento della documentazione prodotta da Error\_418 per ogni fase del ciclo di vita del software.

#### 3.1.2 Aspettative

- Definire una serie di regole per la stesura di documenti o per metodi di lavoro in modo che ogni attività svolta da un membro del gruppo sia valutabile secondo criteri di correttezza decisi a priori.
- Documentare come svolgere un lavoro in modo che un membro del gruppo possa lavorare in maniera uniforme rispetto al resto del gruppo.

#### 3.1.3 Typst

Il team Error 418 ha deciso di utilizzare Typst come linguaggio per redarre la documentazione. Questo è un linguaggio di markup che permette molte delle funzionalità offerte da strumenti come LaTeX, richiedendo però molto meno sforzo durante la procedura di creazione e di compilazione del file.

Questo permette di evitare l'apprendimento della sintassi LaTeX che può risultare ostica ai novizi e consentendo al team di concentrarsi esclusivamente sul corpo del file evitando perdite di tempo e frustrazione data da errori che non si capisce come risolvere o che richiederebbero molto tempo per essere risolti.

#### 3.1.4 Ciclo di vita

La creazione e il versionamento di un documento è stato automatizzato quanto più possibile in modo da ridurre l'impatto dell'errore umano.

Le modalità che permettono di creare un documento sono:

- 1) creare un nuovo branch a partire dal branch che raccoglie i sorgenti (src);
- 2) creare una nuova cartella rinominata con lo stesso nome del sorgente che si andrà a creare;



- 3) creare un nuovo file .typ a partire da uno dei template che sono stati creati appositamente dal team (per scopo e struttura dei template controllare la sezione Sezione 3.1.5);
- 4) scrivere il corpo del documento modificando eventualmente alcuni specifici campi del template (titolo, data, ecc..);
- 5) accertarsi che il documento rispetti tutte le norme definite nelle “Norme di progetto”;
- 6) salvare e creare la pull request per integrare il sorgente all’interno del branch src;
- 7) scegliere un nome della pull request aderente alle norme definite nelle “Norme di progetto”;
- 8) una volta creata la pull request avvierà una serie di action che provvederanno a verificare la correttezza la presenza o meno di alcuni campi del documento, creare o aggiornare il changelog, compilare il documento e integrare il risultante file .pdf all’interno del branch main;

### 3.1.5 Template

Il team ha creato dei template per la creazione di documenti, questo facilita il lavoro di compilazione e verifica da parte delle GitHub.

Possono essere trovate nella root della repository github nel branch src.

I template sono:

- template.typ: per la creazione di documenti interni o esterni;
- quickstart.typ: per la creazione di verbali interni o esterni.

### 3.1.6 Struttura dei documenti

Esistono due tipi di documenti fondamentali: “documento” e “verbali” e hanno alcune differenze a livello di struttura.

Ogni documento è formato da un file Typst e da un file log.csv creato automaticamente dalle github action (e che si occupano anche di aggiornarlo). Questi due file vengono automaticamente compilati insieme e il .pdf risultante viene inserito all’interno del branch main.

#### 3.1.6.1 Verbali

Un verbale è la documentazione di un meeting svolto tra i membri del gruppo (verbale interno) o con un referente esterno (verbale esterno). Documentare le riunioni permette di tenere traccia delle decisioni prese dal gruppo in un dato momento e permette a chi non riesce a presenziare di tenersi al passo con il gruppo in maniera asincrona.

I verbali devono essere rinominati nel seguente modo “yy-mm-dd.typ” e nello stesso modo deve essere chiamata la sottocartella in cui è inserito.

##### 3.1.6.1.1 Intestazione dei verbali

Ogni verbale, interno o esterno, deve presentare:

- intestazione:
  - 1) logo del gruppo (logo.png disponibile nella root del branch src);
  - 2) nome del gruppo: Error 418;
  - 3) titolo del documento: standardizzato nella forma “Verbale”+“interno/esterno”+“data del verbale nella forma dd/mm/yy”;
  - 4) eventuale sottotitolo: specifica se una riunione ha avuto uno specifico argomento di discussione, è opzionale;

- 5) referente: referente dell'azienda proponente che ha partecipato alla riunione in caso di verbale esterno;
- 6) redattori e destinatari;
- 7) partecipanti: chi ha partecipato alla riunione e il ruolo che ha ricoperto;
- 8) ora di inizio meeting, ora di fine meeting e durata del meeting (calcolata automaticamente).

#### **3.1.6.1.2 Corpo del verbale**

L'effettivo contenuto del documento dove vengono riportati i dettagli gli argomenti trattati nel meeting, il corpo è così organizzato:

- section "Ordine del giorno": riporta gli argomenti da trattare nel meeting;
- subsection: ogni subsection riporta l'effettivo resoconto dell'incontro con eventuali dubbi e soluzioni a cui si è arrivati durante il meeting;
- section "Azioni da intraprendere": riporta le operazioni che andranno svolte a seguito del meeting per risolvere problemi emersi durante il meeting o per continuare il lavoro.
- firme esterne: per verbali esterni, firma del referente esterno che approva il resoconto stilato del gruppo.

#### **3.1.6.2 Documenti**

Documenti prodotto dal team Error 418:

- Norme di Progetto;
- Piano di Progetto;
- Piano di Qualifica;
- Glossario
- Analisi dei Requisiti
- Specifica Architettuale;
- Manuale Utente.

##### **3.1.6.2.1 Intestazione dei documenti**

Ogni documento deve presentare:

- intestazione:
  - 1) logo del gruppo (logo.png disponibile nella root del branch src);
  - 2) nome del gruppo: Error 418;
  - 3) titolo del documento
  - 4) redattori e destinatari;
  - 5) tabella con i componenti del gruppo e i ruoli che ricoprono;
  - 6) changelog;
  - 7) indice.

##### **3.1.6.2.2 Corpo del documento**

Il contenuto del documento è suddiviso in capitoli, ognuno dei quali è composto da più sezioni.

#### **3.1.7 Convenzioni stilistiche**

##### **3.1.7.1 Sintattiche/Grammaticali**

Per la scrittura dei documenti è richiesto uno stile di scrittura impersonale (preferendo per esempio frasi come "il gruppo ha fatto" anziché "abbiamo fatto"). Si richiede inoltre di spezzare i paragrafi

troppo lunghi con un “a capo” in modo da dividere il testo in aree logiche di più facile comprensione e lettura.

### 3.1.8 Elenchi puntati

Viene inserito “;” alla fine delle voci dell’elenco tranne l’ultima che termina con “.”. Ogni voce dell’elenco inizia con una lettera minuscola.

## 4 Processi di ciclo di vita

### 4.1 Processi organizzativi abilitanti

I processi organizzativi abilitanti garantiscono la capacità dell’organizzazione di acquisire e fornire prodotti o servizi attraverso l’avvio, il supporto e il controllo di progetti. Questi processi forniscono le risorse e l’infrastruttura necessarie a supportare i progetti e il conseguimento degli obiettivi dell’organizzazione e degli accordi fra parti. Non sono da intendersi come un insieme esaustivo di processi aziendali atti alla gestione strategica dell’organizzazione.

I processi organizzativi abilitanti sono i seguenti:

- 1) processo di gestione dei modelli di ciclo di vita;
- 2) processo di gestione dell’infrastruttura;
- 3) processo di gestione delle risorse umane;

#### 4.1.1 Processo di gestione dei modelli di ciclo di vita

*Conformant to outcomes to ISO/IEC/IEEE 12207:2017 clause 6.2.1*

##### 4.1.1.1 Scopo

Lo scopo del processo di gestione del modello di ciclo di vita è definire, mantenere e garantire regole, processi, modelli e procedure di ciclo di vita.

Questo processo fornisce politiche, processi, modelli e procedure del ciclo di vita coerenti con gli obiettivi dell’organizzazione, che sono definiti, adattati, migliorati e mantenuti per supportare le esigenze individuali dei progetti all’interno del contesto dell’organizzazione, e che sono in grado di essere applicati mediante metodi e strumenti efficaci e comprovati.

Il gruppo adotta il modello PDCA (Plan-Do-Check-Act) per la gestione del ciclo di vita del software. Questo modello prevede quattro fasi:

- 1) pianificazione (Plan): definizione degli obiettivi e dei processi necessari per raggiungerli;
- 2) esecuzione (Do): attuazione del piano;
- 3) verifica (Check): monitoraggio e misurazione dei processi e dei prodotti rispetto agli obiettivi e ai requisiti, e reportistica dei risultati;
- 4) attuazione (Act): azioni per migliorare le prestazioni, se necessario, in base ai risultati della verifica.

##### 4.1.1.2 Organizzazione del processo

###### 4.1.1.2.1 Pianificazione

La pianificazione del processo è compito del Responsabile, il quale al termine dello sprint precedente, in base alle attività svolte e ai risultati ottenuti, pianifica le attività da svolgere nello sprint successivo.

La pianificazione è un'attività iterativa, che viene svolta all'inizio dello sprint. Il Responsabile, in relazione al progresso svolto, può decidere di modificare la pianificazione iniziale, aggiungendo o rimuovendo attività.

Questo processo permette di individuare le attività da svolgere, le risorse necessarie e le tempistiche di svolgimento, mediante il sistema di ticketing offerto da Jira.

Il risultato di questo processo è visibile all'interno del Piano di Progetto.

#### 4.1.1.2.2 Esecuzione

Identifica il processo di sviluppo del prodotto, dove quanto pianificato viene concretamente svolto.

Il processo di esecuzione è composto da due attività principali:

- 1) sviluppo del prodotto;
- 2) sviluppo della documentazione.

Durante questo processo, ogni ruolo svolge le attività assegnate seguendo quanto stabilito nella pianificazione, e sarà compito del Responsabile verificare che le attività siano svolte correttamente e nei tempi previsti. Ogni membro avrà la possibilità di segnalare eventuali criticità, avendo a disposizione una board apposita sulla piattaforma Miro.

Questo permette di avere un resoconto pronto a fine sprint in merito al processo di avanzamento, individuando:

- “Keep doing”: attività che hanno dato buoni risultati e che quindi vanno mantenute;
- “Things to change”: attività che hanno dato risultati non soddisfacenti e che quindi vanno modificate.

La progressione del lavoro è visibile mediante:

- retrospettiva;
- grafici di burndown (Jira);
- board di avanzamento (Miro).

I prodotti di questo processo, permettono dunque di procedere con la verifica e lo stabilimento delle contromisure e dei miglioramenti necessari.

#### 4.1.1.2.3 Verifica

Al termine di ogni sprint, il gruppo procederà con il meeting di retrospettiva, durante il quale verranno analizzati i risultati ottenuti e le attività svolte, basandosi sui prodotti della fase di esecuzione:

- 1) grafici di burndown:
  - permettono di avere una visione rapida di quanto del lavoro pianificato è stato portato a termine: si tratta di una metrica puramente quantitativa, che tiene conto di del numero di ticket chiusi e del numero di ticket aperti;
- 2) board di avanzamento:
  - “Keep doing”: attività che hanno dato buoni risultati e che quindi vanno mantenute: questo permette al gruppo di individuare il *modus operandi* più efficace ed efficiente per svolgere le attività;

- “Things to change”: attività che hanno dato risultati non soddisfacenti e che quindi vanno modificate: si tratta dell’aspetto più delicato da considerare, in quanto permette di individuare le criticità e le inefficienze del processo di sviluppo, e di conseguenza di apportare le modifiche necessarie per migliorare il processo stesso.

Questa analisi individua i miglioramenti da apportare al processo di sviluppo, stabilendo le contromisure necessarie per migliorare il processo stesso.

#### **4.1.1.2.4 Attuazione**

L’attuazione è l’ultima fase del processo di gestione del ciclo di vita, e consiste nell’attuare le contromisure stabilite durante la fase di verifica.

L’obiettivo è sopperire alle mancanze e alle inefficienze del processo di sviluppo, in modo da migliorare la qualità del prodotto e la produttività del gruppo.

Diventa compito del Responsabile stabilire le attività necessarie per attuare le contromisure, e di conseguenza di pianificare le attività da svolgere nello sprint successivo.

Il risultato di questo processo è visibile all’interno del Piano di Progetto.

#### **4.1.1.3 Ruoli**

##### **4.1.1.3.1 Responsabile**

Il Responsabile è la figura chiave che guida il progetto, assumendo il ruolo di referente principale per il gruppo e per gli stakeholders.

Le responsabilità del Responsabile includono:

- 1) coordinamento: ha il compito di supervisionare i membri del gruppo, assicurandosi che le attività siano svolte seguendo le norme identificate in questo documento;
- 2) pianificazione: stabilisce le attività da svolgere, le relative scadenze e priorità, sancendo l’inizio e la fine di ogni sprint;
- 3) monitoraggio e gestione dei costi: tiene sotto controllo l’andamento del progetto, stima i costi e gestisce l’analisi dei rischi, garantendo che il progetto rimanga entro il budget previsto;
- 4) norme di progetto: si occupa della stesura e dell’aggiornamento delle norme di progetto, che devono essere rispettate da tutti i membri del gruppo;
- 5) relazioni esterne: gestisce tutte le interazioni con il Proponente e i Committenti assicurando una comunicazione fluida e efficace;

##### **4.1.1.3.2 Amministratore**

L’Amministratore è la figura che gestisce l’ambiente di lavoro e gli strumenti utilizzati dal gruppo per tutta la durata del progetto. Ha il compito di assicurare che gli strumenti proposti ai membri del gruppo siano efficienti e favoriscano la qualità del lavoro. Monitora, assieme al Responsabile, il

rispetto delle regole stabilite in questo documento e verifica che i servizi a disposizione del gruppo siano adeguati alle attività pianificate, promuovendo la produttività.

Le responsabilità dell'Amministratore includono:

- 1) redazione dei verbali: l'Amministratore è responsabile della stesura dei verbali relativi ai meeting interni ed esterni;
- 2) gestione delle risorse: si occupa dell'amministrazione delle risorse, delle infrastrutture e dei servizi necessari per l'esecuzione dei processi di supporto;
- 3) automatizzazione dei processi: determina gli strumenti necessari per automatizzare i processi, come la compilazione dei file sorgenti e il sistema di versionamento automatico;
- 4) risoluzione dei problemi: affronta e risolve i problemi legati alla gestione dei processi.

#### **4.1.1.3.3 Analista**

L'Analista individua i bisogni del Proponente e li trasforma in requisiti che saranno l'input delle attività successive. Il suo lavoro si svolge intensamente nel periodo di avvio del progetto, e si conclude con la stesura dell'Analisi dei Requisiti.

Il suo compito è di rilevanza in quanto un'incompleta o superficiale analisi può impattare in modo sensibile sulle attività successive, causando ritardi e costi aggiuntivi, andando ad inficiare sulla qualità e completezza del prodotto finale.

Le responsabilità dell'Analista includono:

- 1) documento di Analisi dei Requisiti: l'Analista è incaricato di scrivere questo documento cruciale, che dettaglia i requisiti specifici del progetto.
- 2) interazione con i clienti: l'Analista lavora a stretto contatto con il Proponente o i Committenti per capire e studiare i loro bisogni;
- 3) classificazione dei requisiti: individua i requisiti e li classifica come funzionali e non funzionali, e come obbligatori, desiderabili o opzionali;
- 4) definizione del problema e degli obiettivi: l'Analista esamina la situazione attuale, identifica il problema e stabilisce gli obiettivi da raggiungere;
- 5) trasformazione dei bisogni in requisiti: durante l'Analisi dei Requisiti, l'Analista converte i bisogni dei clienti in requisiti specifici per la soluzione.

#### **4.1.1.3.4 Progettista**

Il Progettista ha il compito di sviluppare una soluzione che soddisfi le esigenze identificate dall'Analista, rispettando i vincoli individuati. Il Progettista trasforma i requisiti in un'architettura che modella il problema.

Le responsabilità del Progettista includono:

- 1) sviluppo di un'architettura robusta e resistente ai malfunzionamenti;

- 2) realizzazione di soluzioni affidabili, efficienti, sostenibili e conformi ai requisiti;
- 3) decomposizione del sistema in componenti e organizzazione delle loro interazioni, ruoli e responsabilità;
- 4) garanzia di un basso grado di accoppiamento nell'architettura.

#### **4.1.1.3.5 Programmatore**

Il Programmatore è la figura più numerosa all'intero del gruppo con precise competenze tecniche.

Le responsabilità del Programmatore includono:

- 1) scrittura del codice sorgente, perseguendo mantenibilità e conformità a quanto stabilito dall'architettura definita dalla progettazione;
- 2) creazione di test specifici per la verifica e la validazione del codice.

#### **4.1.1.3.6 Verificatore**

Il Verificatore controlla il lavoro svolto dagli altri membri del gruppo, assicurandosi che le norme di progetto e le aspettative siano rispettate.

Le responsabilità del Verificatore includono:

- 1) verifica della qualità e della conformità della documentazione prodotta;
- 2) approvazione della documentazione sottoposta a verifica;

### **4.1.2 Processi di gestione dell'infrastruttura**

#### **4.1.2.1 Scopo**

Lo scopo del processo di gestione dell'infrastruttura è fornire l'infrastruttura e i servizi a supporto dell'organizzazione per il conseguimento degli obiettivi di progetto nel corso dell'intero ciclo di vita.

Questo processo definisce, fornisce e regola i servizi, gli strumenti e le tecnologie di comunicazione e condivisione delle informazioni a supporto degli scopi dell'organizzazione.

#### **4.1.2.2 Requisiti**

L'infrastruttura è costituita dai servizi, dagli strumenti e dalle tecnologie di comunicazione e condivisione delle informazioni adottate a supporto degli scopi dell'organizzazione. L'infrastruttura risponde alle necessità di comunicazione interna ed esterna.

I principi della comunicazione che ispirano i processi di gestione dell'infrastruttura sono:

- 1) Comunicazione aperta: le comunicazioni avvengono in modalità pubblica, ovvero tutti i membri possono partecipare (compatibilmente con i loro impegni di progetto e sempre nel rispetto delle rispettive responsabilità). I membri del gruppo hanno accesso e possono liberamente consultare i messaggi, le eventuali registrazioni e i verbali;

- 2) Comunicazione onesta: quanto comunicato rappresenta sempre il reale stato del progetto. Nessun membro, in nessun caso, deve nascondere le criticità incontrate;
- 3) Comunicazione proattiva: comunicare con cognizione di causa, offrendo spunti concreti di discussione. Ogni comunicazione deve poter essere processata dagli interessati nel minor tempo possibile.
- 4) Comunicazione frequente: la frequenza della comunicazione permette di prendere decisione in modo tempestivo e informato.

I requisiti dei processi di gestione dell'infrastruttura sono:

- 1) Semplicità di adozione di strumenti e servizi;
- 2) Accesso rapido alle informazioni, anche in mobilità;
- 3) Non ridondanza tra strumenti e servizi adottati.

#### 4.1.2.3 Infrastruttura di comunicazione

##### 4.1.2.3.1 Comunicazione interna

**Discord** rappresenta il canale primario di comunicazione interna. È una piattaforma di comunicazione che fornisce:

- 1) Un servizio di messaggistica istantanea che permette la suddivisione delle conversazioni in canali tematici;
- 2) Un servizio di videochiamate usato per le comunicazioni sincrone interne in modalità remota.

Le comunicazioni testuali tramite messaggio istantaneo sono organizzate per argomento. Discord adotta il termine “canale” per designare tale suddivisione tematica. I canali attualmente in uso sono:

- Canale generale (**#random**): usato per le comunicazioni informali;
- Canale meeting (**#meeting**): usato per l'organizzazione dei meeting interni e la condivisione degli ordini del giorno;
- Canale di riferimento (**#riferimenti**): usato come bacheca per raccogliere ed organizzare, in un unico luogo, le risorse a supporto degli scopi dell'organizzazione;
- Altri canali tematici: le comunicazioni relative ad uno specifico prodotto dei processi dell'organizzazione avvengono in un canale dedicato.

La suddivisione delle comunicazioni interne in canali ha lo scopo di ridurre le distrazioni, facilitare l'accesso alle informazioni e semplificare la comunicazione interna.

Le comunicazioni sincrone in videochiamata avvengono nei cosiddetti “canali vocali”. Vengono forniti quattro canali vocali generici ad accesso libero. I membri dell'organizzazione hanno la facoltà di incontrarsi in qualsiasi momento in videochiamate interne. I canali vocali non sono organizzati tematicamente perché offrono la persistenza.

**Miro** è un servizio di collaborazione per team basato su *whiteboard*. Offre la possibilità di creare board multimediali e permette la collaborazione asincrona. È utilizzato per:

- 1) Raccogliere i feedback interni da discutere durante i meeting di retrospettiva;
- 2) Supportare gli incontri interni di *brainstorming*;
- 3) Supportare i meeting con gli interlocutori esterni.

Oltre a Discord e Miro, l'organizzazione comunica anche tramite **Jira** (ITS) e **GitHub** (VCS). L'uso di questi strumenti è discusso in dettaglio nelle sezioni apposite.



#### 4.1.2.3.2 Comunicazione esterna

Le modalità e la frequenza delle comunicazioni esterne sono da stabilirsi con i diretti interessati, secondo necessità e disponibilità degli interlocutori. Le comunicazioni esterne avvengono su due canali primari: Gmail e Zoom.

**Gmail** è il servizio di posta elettronica di Google. L'indirizzo di posta elettronica dell'organizzazione è:

`error418swe@gmail.com`

Viene utilizzato per tutte le comunicazioni da e verso gli interlocutori esterni. Tutti i membri dell'organizzazione possono accedere in qualsiasi momento alla casella di posta elettronica. Inoltre, tutte le conversazioni vengono inoltrate automaticamente agli indirizzi e-mail istituzionali di ciascun membro.

L'indirizzo è reso noto nel frontespizio di ogni documento prodotto dall'organizzazione.

**Zoom** è un servizio di teleconferenza. A meno di accordi specifici tra le parti, l'organizzazione utilizza Zoom per effettuare videochiamate con gli interlocutori esterni.

#### 4.1.2.4 Mantenimento dell'infrastruttura

Sono compiti dell'amministratore il mantenimento dell'infrastruttura, l'aggiornamento delle norme e dei processi e l'identificazione di nuovi servizi a supporto delle attività dell'organizzazione.

L'organizzazione adotta nuovi servizi, strumenti e tecnologie di comunicazione avendo cura di non introdurre ridondanza. L'organizzazione si dota di un insieme di strumenti e servizi minimo e sufficiente a coprire tutti i requisiti di comunicazione.

### 4.1.3 Processo di gestione delle Risorse Umane

La natura didattica del progetto riduce le prerogative del processo di gestione delle Risorse umane. Per questa ragione, l'organizzazione dichiara la *conformance to outcomes to ISO/IEC/IEEE 12207:2017 clause 6.2.4*.

#### 4.1.3.1 Scopo

Questo processo fornisce all'organizzazione le risorse umane necessarie all'esecuzione dei processi di ciclo di vita al fine di realizzare gli obiettivi di progetto, dell'organizzazione e degli stakeholder.

#### 4.1.3.2 Attività e compiti

##### 4.1.3.2.1 Identificare le competenze dei membri

L'organizzazione sottopone, ad ogni nuovo membro, un form conoscitivo atto a identificare le competenze pregresse. Il form è realizzato con **Google Forms** e include domande su:

- 1) Strumenti di collaborazione;
- 2) Linguaggi di programmazione;
- 3) Tecnologie;
- 4) Strumenti di automazione;
- 5) Strumenti di controllo di versione.

Le risposte sono non vincolanti e non influiscono in alcun modo sulla rotazione dei ruoli, sui compiti assegnati o sull'organizzazione interna.

#### 4.1.3.2.2 Identificare le competenze richieste

Le competenze richieste sono identificate tramite:

- 1) Analisi dei Capitolati;
- 2) Studio del dominio di progetto;
- 3) Incontri con i Proponenti e successivi colloqui.

#### 4.1.3.2.3 Sviluppare le competenze

Lo sviluppo di nuove competenze riguarda i membri, e non i ruoli. Per questa ragione, i processi di sviluppo di competenze sono universali e condivisi.

L'organizzazione si adopera per sviluppare le competenze dei membri mediante:

- 1) Attività di *peer-tutoring* in concomitanza delle rotazioni di ruolo;
- 2) Pubblicazione interna di tutorial tecnici scritti (eventualmente accompagnati da brevi video, se utili a migliorare la comprensione degli argomenti trattati);
- 3) Attività di *tutoring* interno su richiesta, sincrono, in base alla necessità;
- 4) Attività di *mentoring* esterno su richiesta, in base alla necessità e alla disponibilità dell'interlocutore esterno;
- 5) Condivisione delle best practice in sessione collettiva.

Le sessioni di tutoring sono “a sportello” ed è responsabilità dei singoli membri richiederne l'attivazione. Il responsabile, identificati i temi di maggior interesse, può espressamente richiedere che un ruolo copra le esigenze di tutoring interno tramite le modalità sopra indicate.

#### 4.1.3.2.4 Acquisire e fornire competenze

I membri dell'organizzazione sono prestabiliti. Qualora le competenze interne all'organizzazione siano deficitarie, è richiesta l'attivazione delle attività descritte in Sezione 4.1.3.2.3.

Non sono previste variazioni della composizione dell'organizzazione, se non in via straordinaria e comunque discussa preventivamente con il Committente.

## 5 Processi di gestione tecnica

### 5.1 Processo di pianificazione di progetto

*Conformant to outcomes to ISO/IEC/IEEE 12207:2017 clause 6.3.1*

#### 5.1.1 Scopo

Il processo di pianificazione di progetto ha come scopo la produzione e coordinazione di un piano efficace ed applicabile per garantire una corretta gestione del lavoro. La pianificazione viene inserita in un documento denominato Piano di Progetto (Sezione 5.1.3).

#### 5.1.2 Attività

Nel processo di pianificazione sono presenti due attività principali:

- 1) Definizione del progetto;
- 2) Pianificazione del progetto e della gestione tecnica.

### 5.1.2.1 Definizione del progetto

In questa attività il gruppo deve definire tutto ciò che caratterizza il progetto, ovvero i suoi obiettivi e vincoli, sia di carattere funzionale che tecnologico. Durante la lavorazione del progetto verranno prodotti diversi output, che possono essere suddivisi nelle due macro categorie di: documentazione, codice. Entrambi questi prodotti dovranno essere realizzati rispettando determinate regole e processi, ed è quindi necessario che il gruppo definisca in questa attività uno o più cicli di vita da seguire.

### 5.1.2.2 Pianificazione del progetto e della gestione tecnica

È l'attività principale del processo, nella quale viene definita nel concreto la pianificazione.

#### 5.1.2.2.1 Suddivisione temporale

Il gruppo ha individuato tre periodi di lavoro principali:

- Raccolta e analisi dei requisiti: vengono delineati i requisiti che il prodotto finale dovrà rispettare tramite un continuo rapporto con l'azienda proponente;
- Sviluppo della Requirements and Technology Baseline (RTB): si studiano le tecnologie da utilizzare e si applicano le conoscenze acquisite per realizzare un PoC (Proof of Concept), ovvero un prodotto software che permetta di dimostrare la padronanza delle tecnologie selezionate ai fini dello sviluppo del progetto;
- Periodo di sviluppo del Minimum Viable Product (MVP): viene progettato e implementato un prodotto software che rispetti almeno i requisiti minimi di accettazione, e che offra tutte le funzionalità richieste.

Ognuno di questi viene suddiviso a sua volta in periodi della durata di una settimana denominati *sprint*. Al termine di ogni *sprint* viene effettuato un incontro interno di retrospettiva, nel quale si analizza criticamente la settimana appena conclusa, mostrandone aspetti positivi, aspetti da migliorare e fissando obiettivi che verranno poi riportati nell'Issue Tracking System sotto forma di task. Questi andranno a comporre il *backlog* dello *sprint* successivo, e il loro progressivo completamento andrà a produrre un *burndown-chart*, utilizzato dal gruppo come strumento che rappresenti in modo oggettivo l'andamento del lavoro.

#### 5.1.2.2.2 Definizione di ruoli, responsabilità e costi

Al fine di migliorare l'assegnazione del lavoro vengono definiti sei ruoli, ognuno dei quali con precise responsabilità da rispettare. Ogni membro del gruppo dovrà assumere ognuno di questi ruoli all'interno del periodo di lavoro al progetto. L'assegnazione dei ruoli avviene con frequenza bisettimanale. Di seguito viene riportata la descrizione di ogni ruolo con i relativi compiti:

- 1) **Responsabile:** è presente durante l'intero progetto, in particolare si occupa di:
  - coordinare il gruppo;
  - verificare che il lavoro proceda secondo le tempistiche e i costi stabiliti;
  - rappresentare il gruppo nei rapporti con il committente;
  - gestire la pianificazione di ogni *sprint*.
- 2) **Amministratore:** ruolo presente durante tutto il progetto. Ha il compito di:
  - predisporre e controllare il corretto utilizzo delle procedure e degli strumenti adottati;
  - implementare e mantenere gli automatismi in modo da migliorare l'efficienza del gruppo.

- 3) **Analista:** è presente principalmente nei primi due periodi del progetto. Si occupa di redigere il documento Analisi dei requisiti, nel quale:
  - definisce i casi d'uso;
  - raccoglie e classifica i requisiti.
- 4) **Progettista:** ruolo presente principalmente negli ultimi due periodi, nei quali:
  - delinea l'architettura del prodotto;
  - definisce le linee guida implementative valutando le scelte più efficienti e sostenibili.
- 5) **Programmatore:** è attivo negli ultimi due periodi del progetto, ed è il responsabile della traduzione in codice eseguibile dell'architettura definita dal progettista;
- 6) **Verificatore:** è presente durante l'intero progetto, e si occupa di controllare che il lavoro prodotto dal gruppo rispetti gli standard qualitativi adottati.

Ad ogni ruolo è inoltre associato un costo orario, tramite cui il gruppo calcola il preventivo totale del progetto e quello di ogni *sprint* seguito dal relativo consuntivo.

### 5.1.3 Piano di Progetto

Il Piano di Progetto è il documento principale che viene redatto durante questo processo, e ha il compito di governare la pianificazione e l'avanzamento del progetto. In questo documento vengono determinati i task e gli obiettivi da raggiungere, fornendo anche un'analisi critica del lavoro svolto fino a quel momento. Sono presenti cinque sezioni:

- 1) Rischi e loro mitigazione;
- 2) Divisione temporale di sviluppo;
- 3) Preventivo dei costi di realizzazione;
- 4) Pianificazione del lavoro;
- 5) Consuntivo del progetto.

La redazione del documento va di pari passo con l'avanzamento del progetto, in modo tale da essere sempre aggiornato alla situazione corrente del lavoro.

## 5.2 Processo di valutazione e controllo di progetto

*Conformant to outcomes to ISO/IEC/IEEE 12207:2017 clause 6.3.2*

### 5.2.1 Scopo

Il processo di valutazione e controllo ha lo scopo di determinare lo stato del progetto tramite la verifica dell'allineamento del lavoro con i piani definiti. Tale allineamento viene dimostrato osservando i seguenti parametri:

- tempistiche;
- costi;
- obiettivi raggiunti.

Questo viene fatto all'interno di revisioni periodiche chiamate retrospettive, e viene messo per iscritto all'interno dei verbali di tali riunioni e nei paragrafi di consuntivo del Piano di Progetto.

## 5.2.2 Attività

### 5.2.2.1 Piano di valutazione e controllo del progetto

La valutazione e il controllo del progetto avvengono a cadenza settimanale, in corrispondenza della riunione di retrospettiva. Durante questa riunione si valuta ciò che è stato svolto durante il periodo di lavoro (*sprint*) che si sta concludendo, se ne identificano i problemi e si decidono soluzioni di controllo che vadano a risolvere o arginare i problemi individuati.

Oltre a queste revisioni periodiche interne, sono presenti attività nelle quali il gruppo espone la propria situazione al Proponente, tramite riunione su Zoom, e Committente, tramite le attività denominate “diario di bordo”.

Inoltre, è necessario che il gruppo svolga delle revisioni tecniche con il Committente per avere una valutazione esterna. Queste due revisioni sono chiamate Requirements and Technology Baseline (o RTB) e Product Baseline (o PB).

### 5.2.2.2 Valutazione

Durante l'attività di valutazione il gruppo deve analizzare la situazione del progetto, e per fare ciò deve adottare degli strumenti che rappresentino tale situazione nel modo più oggettivo possibile. Il momento in cui si effettua questo compito è la retrospettiva settimanale. Questa si svolge tramite meeting interno su Discord con il supporto di una board Miro. Il meeting è suddiviso concettualmente in tre parti, rappresentate all'interno della board da tre riquadri:

- Keep doing: raccoglie tutti gli aspetti positivi (e di conseguenza da mantenere) dello sprint;
- Things to change: raccoglie tutte le problematiche incontrate durante lo sprint;
- To do e Improvements: raccoglie tutte le attività da svolgere nel prossimo sprint, alcune delle quali direttamente collegate agli elementi appartenenti a “Things to change”.

Tramite Miro il gruppo riesce ad avere una panoramica della situazione del lavoro, che viene poi completata da Jira, l'Issue Tracking System adottato. Al suo interno il gruppo ha definito le due milestone esterne (RTB e PB), nelle quali ha creato delle Epic che rappresentano ciò che è necessario produrre per quella milestone, e che raccolgono tutti i task necessari alla produzione di tali prodotti (documenti o software).

Milestone e Epic vengono accompagnate all'interno di Jira da una barra di completamento che rappresenta in verde il lavoro svolto, in blu il lavoro in fase di svolgimento e lascia vuota la parte dedicata alle attività definite ma non ancora in svolgimento. Queste barre contribuiscono a fornire una rappresentazione oggettiva della situazione del progetto.

Oltre a queste, Jira offre la funzionalità di visualizzazione di un burndown-chart, ovvero un grafico che rappresenta l'andamento del lavoro all'interno di uno sprint in due possibili modalità:

- quantità di story points completati;
- quantità di issue completate.

Entrambe le rappresentazioni pongono nell'asse  $x$  del grafico il tempo, indicato in giorni. Questi grafici contengono inoltre una retta rappresentante l'andamento ideale del lavoro, grazie alla quale risulta più semplice verificare l'efficienza del gruppo.

Al termine dell'incontro di retrospettiva viene redatto il paragrafo di consuntivo dello sprint nel Piano di Progetto, nel quale, oltre a fare un resoconto dello sprint, si analizza il suo l'aspetto economico: in base ai ruoli impegnati e al monte ore produttivo svolto, si calcola il costo effettivo del periodo concluso, aggiornando conseguentemente il costo preventivato e il documento Piano di Progetto. Questo indicatore contribuisce ad avere un resoconto completo del progetto, e permette al gruppo di comprendere meglio come sta lavorando e se sta gestendo correttamente le risorse a sua disposizione.

#### **5.2.2.3 Controllo**

Nell'attività di controllo si intraprendono azioni correttive nei confronti dei problemi individuati. Questi problemi possono essere di duplice natura:

- 1) mancato raggiungimento degli obiettivi prefissati;
- 2) miglioramenti e accortezze da adottare.

##### **5.2.2.3.1 Mancato raggiungimento degli obiettivi prefissati**

È necessario che alla chiusura dello sprint le attività ancora in fase di svolgimento vengano riportate nello sprint successivo, insieme a tutte quelle attività pianificate ma non ancora iniziate che sono considerate importanti. Tutte le attività che non vengono considerate importanti, ad esempio attività di cui si è rivalutato il grado di priorità, vengono riportate nel backlog. Una situazione di mancato raggiungimento degli obiettivi può essere sinonimo anche da una pianificazione errata e troppo ottimista, ed è quindi necessario che essa sia rivista e migliorata.

##### **5.2.2.3.2 Miglioramenti e accortezze da adottare**

Le soluzioni correttive vengono decise dal gruppo tramite la visualizzazione e l'analisi della board Miro durante la retrospettiva. Nella board infatti, come esposto nella Sezione 5.2.2.2, alcune delle task raccolte rispondono direttamente ai problemi individuati nella parte di Things to change.

## **5.3 Processo di gestione delle Decisioni**

*Conformant to outcomes to ISO/IEC/IEEE 12207:2017 clause 6.3.3*

### **5.3.1 Scopo**

Lo scopo del processo di gestione delle decisioni è fornire un quadro strutturato per identificare, categorizzare e valutare le decisioni che si intendono intraprendere, selezionando la migliore in termini di benefici attesi, costi e rischi associati.

#### **5.3.1.1 Presentazione delle decisioni**

Ogni decisione viene presentata identificandone:

- tipologia;
- obiettivo;
- soluzione proposta;
- vantaggi;
- svantaggi;
- impatto in termini di tempi e costi.

Il processo di decision making viene prevalentemente svolto nel meeting di retrospettiva, in modo da non contrastare la pianificazione dello sprint in corso ed evitare un eccessivo numero di meeting interni che potrebbero comportare difficoltà organizzative e un rallentamento dell'avanzamento. Solo decisioni critiche riguardo cambiamenti sostanziali o nuove direzioni di lavoro possono far scaturire meeting interni mirati.

Il resoconto di quanto deciso sarà visibile all'interno del verbale redatto a fine meeting.

#### 5.3.1.2 Tipologie delle Decisioni

Le decisioni possono riguardare diversi aspetti del capitolato, e la loro categorizzazione è utile per individuare la migliore strategia di gestione e i ruoli coinvolti.

Le decisioni vengono dunque così classificate:

- 1) Decisioni organizzative: sono relative al modo di lavorare, cioè a come vengono gestiti i processi di avanzamento del progetto:
  - documento soggetto a modifiche: Norme di progetto;
  - ruoli responsabili dell'aggiornamento: Responsabile;
  - strategia di decision-making: collaborativa.
- 2) Decisioni sui requisiti: sono relative ai requisiti del prodotto software. Possono riguardare aspetti funzionali e non funzionali:
  - documento soggetto a modifiche: Analisi dei requisiti;
  - ruoli responsabili dell'aggiornamento: Analisti;
  - strategia di decision-making: collaborativa.
- 3) Decisioni di implementazione: sono decisioni relative alla stesura del codice:
  - documento soggetto a modifiche: Norme di progetto, documenti tecnici (diagramma delle classi,...);
  - ruoli responsabili dell'aggiornamento: Responsabile, Progettisti, Programmatori;
  - strategia di decision-making: expertise decision-making che prevede la consultazione di figure dotate di maggiore esperienza come Proponente o Committente.
- 4) Decisioni architetturali: sono decisioni relative ai pattern e alle architetture riguardanti il software.
  - documento soggetto a modifiche: Norme di progetto, documenti tecnici (diagramma delle classi,...);
  - ruoli responsabili dell'aggiornamento: Responsabile, Progettisti;
  - strategia di decision-making: expertise decision-making che prevede la consultazione di figure dotate di maggiore esperienza come Proponente o Committente.
- 5) Decisioni sulla qualità: sono decisioni relative ai controlli di qualità.
  - documento soggetto a modifiche: Piano di qualifica;
  - ruoli responsabili dell'aggiornamento: Responsabile, Verificatori;
  - strategia di decision-making: collaborativa.

#### 5.3.1.3 Strategie di decision-making utilizzate

- 1) Strategia collaborativa: prevede che la decisione venga presentata e votata in modo collaborativo, coinvolgendo tutti i membri del gruppo e mediante una votazione con un sistema maggioritario,

in modo che il risultato rappresenti la volontà della maggioranza.

Strategia utilizzata per le decisioni:

- organizzative;
- dei requisiti;
- qualitative.

- 2) Strategia expertise decision-making: prevede che la decisione venga presentata e analizzata mediante la consultazione di figure esterne più esperte, individuabili nel Proponente o nel Commit-  
tente, la cui esperienza risulta determinante.

Questa strategia viene utilizzata per le decisioni:

- implementative;
- architetturali.

## 5.4 Processo di gestione della configurazione

*Conformant to outcomes to ISO/IEC/IEEE 12207:2017 clause 6.3.5*

### 5.4.1 Scopo

Lo scopo della Gestione della Configurazione è gestire e controllare gli elementi del sistema e le configurazioni durante il ciclo di vita. Lo scopo principale è tracciare e coordinare le procedure necessarie alla modifica della documentazione e del codice sorgente.

### 5.4.2 Versionamento

#### 5.4.2.1 Descrizione

Il versionamento è un processo che permette di tenere traccia delle modifiche effettuate su un prodotto software o documentale. Per ogni modifica viene creata una nuova versione del prodotto, che viene identificata da un numero di versione. Il numero di versione è composto da tre cifre separate da un punto, e segue la convenzione seguente:

X.Y.Z

dove:

- 1) X: indica il numero di versione principale, aggiornato al cambiamento della struttura del documento. Riguarda dunque cambiamenti di organizzazione del documento, dei suoi paragrafi e della presentazione delle informazioni, nonché cambiamento dei parametri necessari nel template dei documenti;
- 2) Y: indica il numero di versione secondaria, aggiornato all'aggiunta o alla rimozione di paragrafi;
- 3) Z: indica il numero di versione di revisione e correzione, aggiornato a seguito di cambiamenti minimi o correzioni ortografiche.

L'aggiornamento di una delle cifre del numero di versione azzerà le cifre di rilevanza inferiore.

- 1) Questo schema descrive il versionamento dei documenti. I documenti prodotti dal gruppo sono descritti nel dettaglio nella sezione TODO
- 2) Un normale numero di versione deve avere la forma X.Y.Z, dove X, Y e Z sono interi non negativi.
- 3) Numeri di versione con X pari a 0 indicano documenti in lavorazione, da non considerarsi pronti al rilascio.



- 4) Dopo il rilascio, il contenuto della versione non deve essere modificato. Qualsiasi modifica successiva al rilascio deve causare un cambio nel numero di versione.

#### 5.4.2.2 Tracciamento modifiche

Il tracciamento delle modifiche avviene per mezzo di automazioni che permettono di identificare:

- versione del documento modificato;
- data di modifica (gg-mm-aaaa, ddd);
- numero della pull request di riferimento;
- titolo della pull request di riferimento;
- autore della modifica;
- revisore incaricato.

Tali informazioni sono salvate in un file CSV, unico per ogni documento. Questo file, denominato `log.csv`, è salvato nella cartella dedicata al documento a cui si riferisce, e viene generato automaticamente da una GitHub Action invocata all'apertura, riapertura, sincronizzazione e chiusura di una pull request. Maggiori dettagli al paragrafo dedicato TODO.

Ogni documento, nella sezione direttamente sottostante all'indice, mostrerà in formato tabellare le informazioni relative al tracciamento delle modifiche, leggendo le informazioni dal file `log.csv`.

| Ver.  | Data               | PR | Titolo  | Redattore           | Verificatore      |
|-------|--------------------|----|---|---------------------|-------------------|
| 1.0.0 | 11-12-2023,<br>Mon | 90 | DOC-123 Redazione paragrafo<br>{nome_paragrafo} | Riccardo<br>Carraro | Mattia<br>Todesco |

Figura 1: Esempio tracciamento modifiche

### 5.4.3 Tecnologie

#### 5.4.3.1 Typst

Il gruppo utilizza Typst come strumento di scrittura e compilazione dei documenti.

##### 5.4.3.1.1 Template

Al fine di dare una struttura comune ai documenti si è creato un file `Template.typ` parametrizzato, sfruttando la possibilità di produrre un file PDF compilando insieme più file Typst. Questo file contiene le impostazioni di base per la creazione di un documento:

- `title`: titolo del documento;
- `subTitle`: sottotitolo del documento;
- `docType`: tipologia del documento (Verbale, Documento);
- `date`: data di creazione del documento;
- `externalParticipants`: partecipanti esterni al gruppo;
- `authors`: autori del documento;
- `reviewers`: revisori del documento;
- `missingMembers`: membri assenti durante i meeting;
- `location`: luogo di incontro;
- `timeStart`: ora di inizio incontro;
- `timeEnd`: ora di fine incontro;

- **showLog**: flag che indica se mostrare il tracciamento delle modifiche;
- **showIndex**: flag che indica se mostrare l'indice;
- **isExternalUse**: flag che indica se il documento è per uso esterno.

Al momento della creazione di un nuovo documento sarà sufficiente importare il modello e specificare i parametri sopra elencati.

#### 5.4.3.1.2 QuickStart

Al fine di semplificare la procedura di creazione di un documento, è stato condiviso un documento di testo denominato `quickstart.txt` che contiene la configurazione base per la stesura dei documenti.

#### 5.4.3.2 Astro

#### 5.4.3.3 GitHub

Lo strumento di versionamento scelto dal gruppo è GitHub. Il gruppo Error\_418 ha creato un'organizzazione omonima su GitHub in modo da gestire e separare il lavoro in più repository pensate per scopi e contenuti diversi:

- **Documenti**: repository contenente la documentazione prodotta;
- **PoC**: repository contenente i Proof of Concept prodotti.

Documenti è la repository dedicata alla documentazione prodotta, la quale possiede due branch principali:

- **main**: contiene i file pdf dei documenti prodotti solamente in seguito ad un processo di review con esito positivo;
- **src**: contiene i file sorgenti dei documenti prodotti, file di configurazione e di supporto.

Documenti è organizzata in modo da suddividere la documentazione necessaria alle revisioni esterne che il gruppo dovrà affrontare:

- **RTB**: contiene i file necessari alla Requirements and Technology Baseline;
  - **Documentazione esterna**: contiene i documenti ad uso esterno;
    - **Verbali**: contiene i verbali delle riunioni esterne;
    - **doc\_file\_name** /
      - `doc_file_name.typ`: file sorgente del documento;
      - `log.csv`: registro delle modifiche associato al documento;
  - **Documentazione interna**: contiene i documenti ad uso interno;
    - **Verbali**: contiene i verbali delle riunioni interne;
    - **doc\_file\_name** /
      - `doc_file_name.typ`: file sorgente del documento;
      - `log.csv`: registro delle modifiche associato al documento;
- **PB**: contiene i file necessari alla Product Baseline.

##### 5.4.3.3.1 Pull request (PR)

Al fine di garantire uno svolgimento delle attività in parallelo, la strategia utilizzata dal gruppo durante lo sviluppo è il *featuring branching*, ove per ogni modifica che si intende apporre al documento è necessario sollevare un pull request. I branch vengono creati a partire da task assegnate su Jira, e sono nominati nel formato DOC-XX, dove XX è il numero della task su Jira.

La procedura di modifica o di creazione di un file prevede:

- aggiornamento della task su Jira nello stato di “In progress”;
- creazione di un branch dedicato alla modifica;
- modifica/creazione del documento;
- creazione di una pull request;
  - verificare la conformità del titolo della PR nel formato DOC-XX: titolo;
  - designato almeno un verificatore;
  - il documento compila correttamente in locale;
- revisione della PR da parte del verificatore;
  - eventuale richiesta di modifiche;
  - apportazione delle modifiche;
- approvazione della PR da parte del verificatore;
- merge della PR.

Il processo di verifica del documento è accompagnato dall'esecuzione di GitHub Actions che si occupano di automatizzare l'aggiornamento del file log.csv con i dati relativi alla modifica apportata, e la compilazione e pubblicazione del documento nel ramo main della repository. In questo modo si assicura che ogni documento presente nel ramo main sia prima stato sottoposto ad un processo di verifica.

Solo a seguito del merge della PR, la task collegata presente in Jira può essere definita conclusa.

#### **5.4.3.3.2 Automazioni**

L'intero processo di versionamento è accompagnato da una serie di automazioni, che sollevano i componenti del gruppo dall'onere manuale di attività come la compilazione dei documenti, l'aggiornamento del registro delle modifiche (file log.csv) e la pubblicazione dei documenti dopo la verifica.

#### **Workflow delle automazioni:**

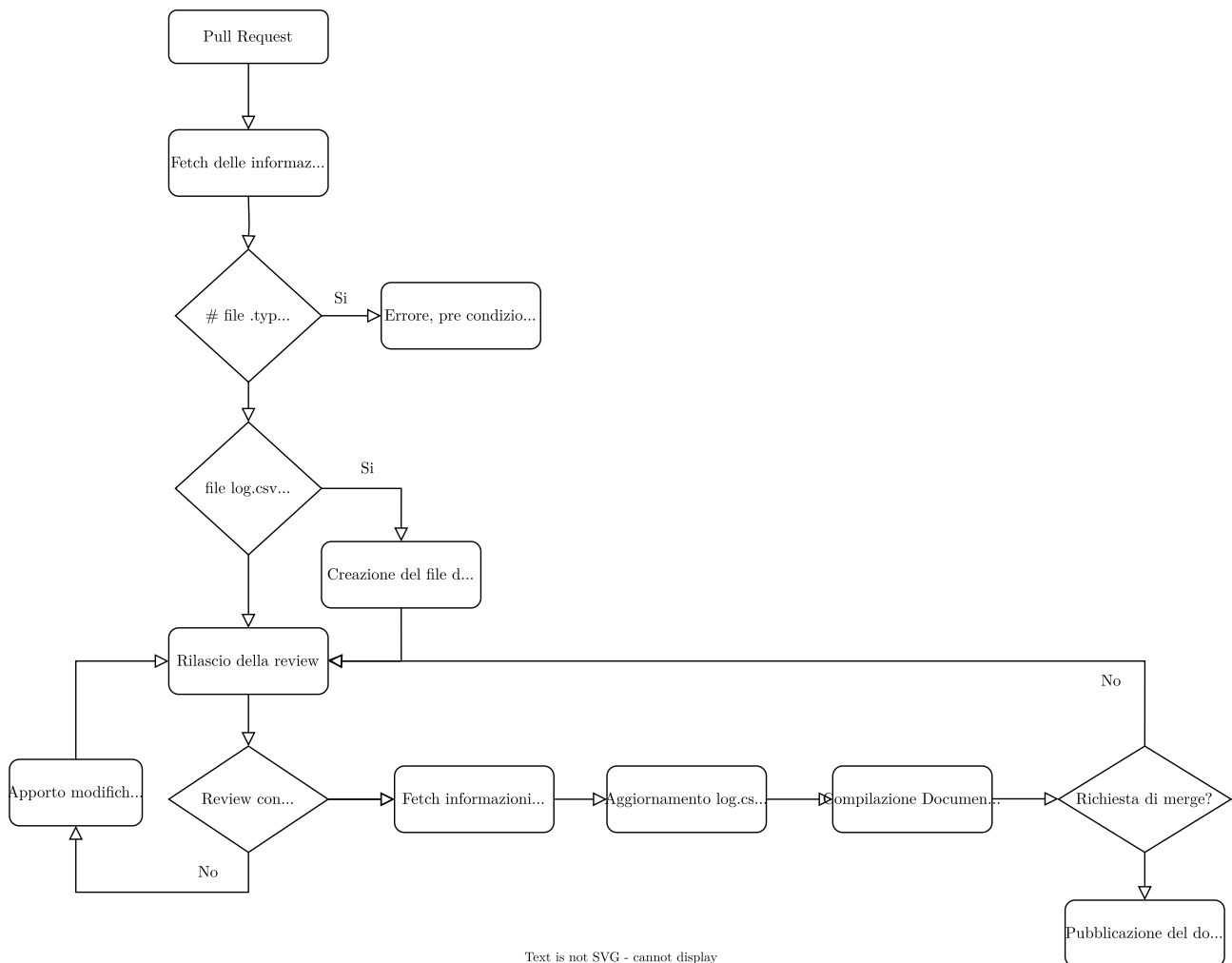


Figura 2: Workflow delle automazioni

Alla creazione della pull request si avvia il workflow per la compilazione e la registrazione delle modifiche avvenute. Prima di procedere sarà necessario inserire informazioni essenziali ai fini di chiarezza e tracciabilità nel messaggio di pull request, quali:

- titolo conforme, contenente il nome della task di riferimento su Jira legata alla pull request, nel formato *DOC-XX titolo*;
- link alla task di riferimento su Jira;
- almeno un verificatore;
- eventuali note aggiuntive.

Il workflow è composto dai seguenti passaggi:

- 1) Fetch delle informazioni dei file modificati: vengono recuperate le informazioni relative ai file modificati nella pull request, quali:
  - nome del file;
  - path del file.
- 2) controllo del numero di file modificati: se il numero di file modificati è maggiore di 1, il workflow termina con un errore;
- 3) controllo dell'esistenza del file log.csv: se il file non esiste, viene creato (sinonimo di creazione del documento);

- 4) rilascio della review: il verificatore si occupa di controllare il documento e rilasciare la review, segnando i cambiamenti richiesti;
- 5) apportate le modifiche, si richiede nuovamente la review per verificare che i cambiamenti apportati siano corretti;
- 6) nel momento in cui la review termina con esito positivo, si procede al recupero della versione corrente del documento:
  - se non esiste il corrispettivo pdf nel branch main, allora il documento non era mai stato pubblicato, pertanto la sua versione di partenza sarà fissata a 1.0.0;
  - se esiste il corrispettivo pdf nel branch main, essendo la versione contenuta nel nome del file, si procede al recupero della versione corrente del documento, modificando la versione X.Y.Z in base all'analisi del documento mediante uno script python;
- 7) aggiornamento del file log.csv: il file di log viene aggiornato con le informazioni relative alla modifica effettuata: questo passaggio, avvenendo solamente a seguito di review positiva, permette di garantire che vengano segnate solamente le modifiche che hanno superato il processo di verifica;
- 8) compilazione del documento: aggiornato il file log.csv e recuperato il numero di versione, il documento è pronto per essere compilato, mostrando numero di versione e registro delle modifiche aggiornati;
- 9) pubblicazione del documento: terminati i workflow precedenti, se si avvia la procedura di merge a seguito del processo di verifica, il documento pdf generato dalla compilazione viene pubblicato nel ramo main della repository;
- 10) merge non confermato: qualora a termine della compilazione del documento non venisse confermato il merge da parte del verificatore, significa che è stato individuato un ulteriore errore o correzione da dover apportare al documento prima della sua pubblicazione sul ramo `main` del repository. In questa circostanza sarà dunque necessario rilasciare un'ulteriore review. L'esecuzione riprende dal punto 4.

L'azione manuale si riduce solamente al rilascio di review e conferma di merge, mentre tutte le altre attività vengono automatizzate. All'approvazione della pull request, e alla conseguente chiusura del branch, un'ulteriore automazione integrata su Jira, permette di aggiornare in automatico lo stato della task collegata alla pull request, portandola allo stato di "Completata".

#### 5.4.3.4 Jira

Jira è lo strumento centrale per la gestione e la tracciabilità delle task assegnate ai membri del gruppo. L'integrazione con GitHub permette a Jira di lavorare e apportare modifiche direttamente alla repository del gruppo, permettendo la creazione, gestione e chiusura di branch e al conseguente aggiornamento dello stato delle task. Ogni task è identificata da un codice univoco incrementale nel formato DOC-XX, che permette di identificarla. Ogni task è caratterizzata da:

- codice identificativo DOC-XX generato automaticamente da Jira;
- titolo;
- descrizione (opzionale);
- stato ("Da completare", "In corso", "Completata");
- assegnatario;
- story points (stima del carico di lavoro);

- epic stories (milestone) di riferimento.

Nel processo di versionamento e di tracciamento delle modifiche, Jira ricopre un ruolo fondamentale, grazie anche alla sua integrazione con GitHub: nel momento in cui si intende avviare una task, è necessario seguire i seguenti passaggi:

- aprire la task su Jira;
- selezionare l'opzione di creare un branch dedicato alla task (integrazione con GitHub);
- selezionare la repository e il branch da cui creare il nuovo branch;

A questo punto, la task si aggiornerà nello stato “In corso”. Terminata la task, effettuata la pull request, superato il processo di verifica e chiuso il branch, Jira provvederà ad aggiornare lo stato della task in “Completata”.

#### **5.4.3.4.1 Backlog**

Ogni task da svolgere è segnata all'interno del backlog. Durante la pianificazione dello sprint, si definisce lo sprint backlog, il sottoinsieme di attività provenienti dal backlog che si intendono portare a termine entro la conclusione dello sprint.

A differenza dello sprint backlog definito durante la pianificazione, il backlog viene espanso man mano che si riscontrano nuove task necessarie o a seguito di decisioni prese durante le riunioni interne o esterne.

#### **5.4.3.4.2 Boards**

Le boards permettono, similmente allo sprint backlog, di avere una visione d'insieme delle attività da svolgere, ma con un approccio più visuale e intuitivo. Le task sono organizzate in tre colonne, rappresentanti lo stato:

- da completare: non ancora avviate, ovvero non esiste il branch dedicato;
- in corso: in fase di svolgimento, ovvero branch dedicato alla task creato;
- completata: task conclusa, ovvero branch dedicato chiuso a seguito di merge sul ramo principale.

#### **5.4.3.4.3 Timeline**

La timeline permette di avere una visione delle attività incentrata sulle tempistiche e le relazioni tra le task. La timeline permette di mostrare il grafico di Gantt delle attività evidenziando i rapporti di dipendenza tra le task e stabilendo le scadenze necessarie per il loro svolgimento.

#### **5.4.3.4.4 Grafici**

Jira offre la possibilità di produrre grafici e report relativi all'avanzamento e alla tracciabilità delle task. Tali strumenti permettono di avere delle metriche di valutazione dell'andamento del progetto e di individuare eventuali criticità. Il gruppo utilizza come metrica principale il burndown chart, che permette di avere una visione dell'avanzamento delle attività in base al tempo, basato sugli story points di ogni attività.