



Team Three Technologies
t3.unipd@gmail.com

Norme di Progetto

Progetto di Ingegneria del Software

Redazione	Filippo Compagno Nenad Radulovic Francesco Balestro Mattia Oliva Medin Bianca Zaghetto Sara Gioia Fichera Andrea Masiero
Verifica	Andrea Masiero Mattia Oliva Medin Bianca Zaghetto Francesco Balestro Sara Gioia Fichera Nenad Radulovic Filippo Compagno
Approvazione	Andrea Masiero
Versione	1.0
Tipo	Interno
Destinatari	Prof. Tullio Vardanega Prof. Riccardo Cardin

Università degli Studi di Padova
Corso di Laurea in Informatica
A. A. 2025/26



Registro delle modifiche

Vers.	Data	Autore	Ruolo	Verifica	Oggetto
1.0	06/03/2026	Andrea Masiero	Responsabile	-	Approvazione e rilascio
0.11	28/01/2026	Andrea Masiero	Amministratore	Filippo Compagno	Conclusione sezione Introduzione
0.10	21/01/2026	Sara Gioia Fichera	Amministratore	Bianca Zaghetto	Sezione Codifica, conclusa sezione 2
0.9	17/01/2026	Sara Gioia Fichera	Amministratore	Bianca Zaghetto	Aggiornamento sezione metriche e conclusione
0.8	09/01/2026	Bianca Zaghetto	Amministratore	Filippo Compagno	Inizio stesura sezione metriche
0.7	22/12/2025	Mattia Oliva Medin	Amministratore	Andrea Masiero	Stesura parti su Gestione di Configurazione e Qualità
0.6	20/12/2025	Mattia Oliva Medin	Amministratore	Nenad Radulovic	Sezione Analisi dei requisiti
0.5	02/12/2025	Francesco Balestro	Amministratore	Sara Gioia Fichera	Aggiornamento e conclusione sezione documentazione
0.4	22/11/2025	Nenad Radulovic	Amministratore	Francesco Balestro	Comunicazioni asincrone e cadenza riunioni con la proponente
0.3	19/11/2025	Nenad Radulovic	Amministratore	Bianca Zaghetto	Inizio sezione documentazione, versionamento verbali
0.2	11/11/2025	Filippo Compagno	Amministratore	Mattia Oliva Medin	Pianificazione impegni, strumenti finora utilizzati



Vers.	Data	Autore	Ruolo	Verifica	Oggetto
0.1	05/11/2025	Filippo Compagno	Amministratore	Andrea Masie- ro	Impostazione documento, introduzione, inizio stesu- ra processi organizzativi



Indice

1	Introduzione	7
1.1	Scopo del documento	7
1.2	Scopo del prodotto	7
1.3	Glossario	7
1.4	Riferimenti	7
1.4.1	Riferimenti normativi	7
1.4.2	Riferimenti informativi	8
2	Processi Primari	9
2.1	Fornitura	9
2.1.1	Attività previste	9
2.1.2	Documentazione fornita	9
2.1.2.1	Analisi dei Capitolati	9
2.1.2.2	Analisi dei Requisiti	10
2.1.2.3	Dichiarazione di Impegni	10
2.1.2.4	Glossario	10
2.1.2.5	Lettera di Presentazione	10
2.1.2.6	Piano di Progetto	10
2.1.2.7	Piano di Qualifica	10
2.1.2.8	Verbali	10
2.2	Sviluppo	10
2.2.1	Attività previste	11
2.2.2	Analisi dei requisiti	12
2.2.2.1	Casi d'uso	12
2.2.2.2	Diagrammi dei casi d'uso	13
2.2.2.3	Requisiti	13
2.2.3	Codifica	14
2.2.3.1	Norme di codifica	14
3	Processi di Supporto	15
3.1	Documentazione	15
3.1.1	Attività previste	15
3.1.2	Template	15
3.1.3	Versionamento	15
3.1.4	Nomenclatura	16
3.1.5	Struttura dei documenti	16
3.1.5.1	Struttura dei Verbali	17
3.1.6	Produzione	18
3.1.7	Manutenzione	18
3.2	Gestione della Configurazione	18
3.2.1	Attività previste	18



3.2.2	Identificazione della configurazione	19
3.2.3	Controllo della configurazione	19
3.2.3.1	Progetto GitHub	19
3.2.3.2	Repository	20
3.2.3.3	Sito web	20
3.2.3.4	Termini del glossario	20
3.2.4	Registrazione dello stato della configurazione	20
3.3	Garanzia di Qualità	21
3.3.1	Attività previste	21
3.3.2	Garanzia di qualità di prodotto	21
3.3.3	Garanzia di qualità di processo	21
3.4	Verifica	21
3.4.1	Attività previste	22
3.4.2	Attività di Verifica	22
3.4.2.1	Analisi statica	22
3.4.2.2	Analisi dinamica	22
3.5	Validazione	23
3.5.1	Attività previste	24
3.5.2	Attività di Validazione	24
4	Processi Organizzativi	25
4.1	Gestione dei Processi	25
4.1.1	Attività previste	25
4.1.2	Ruoli di progetto	25
4.1.2.1	Responsabile	25
4.1.2.2	Amministratore	26
4.1.2.3	Analista	26
4.1.2.4	Progettista	27
4.1.2.5	Programmatore	27
4.1.2.6	Verificatore	27
4.1.3	Comunicazione	28
4.1.3.1	Comunicazioni Interne	28
4.1.3.2	Riunioni Interne tra i Membri del Gruppo	29
4.1.3.3	Comunicazioni Esterne	30
4.1.3.4	Riunioni Esterne con il Committente e la Proponente	30
4.1.4	Pianificazione degli Impegni	30
4.2	Infrastruttura	31
4.2.1	Attività previste	31
4.2.2	Strumenti	31
4.2.2.1	Discord	31
4.2.2.2	Git	31
4.2.2.3	Github	32
4.2.2.4	Gmail	32

4.2.2.5	Google Calendar	32
4.2.2.6	Google Docs	32
4.2.2.7	Google Meet	32
4.2.2.8	Google Sheets	32
4.2.2.9	Overleaf	32
4.2.2.10	Telegram	33
4.2.2.11	WhatsApp	33
4.3	Miglioramento	33
4.3.1	Attività previste	33
4.3.2	Implementazione del processo	33
4.3.3	Valutazione del processo	34
4.3.4	Miglioramento del processo	34
4.4	Formazione	34
4.4.1	Descrizione e scopo	34
4.4.2	Formazione individuale	34
5	Metriche di qualità	35
5.1	Scopo	35
5.2	Standard di riferimento per la qualità di prodotto	35
5.3	Nomenclatura delle metriche	36
5.4	Metriche di processo	36
5.4.1	Metriche soddisfatte	36
5.4.2	Earned Value (EV)	36
5.4.3	Planned Value (PV)	36
5.4.4	Actual Cost (AC)	37
5.4.5	Estimated to Completion (ETC)	37
5.4.6	Estimated at Completion (EAC)	37
5.4.7	Cost Variance (CV)	37
5.4.8	Schedule Variance (SV)	37
5.4.9	Cost Performance Index (CPI)	38
5.4.10	Requirements Stability Index	38
5.4.11	Indice Gulpease	38
5.4.12	Correttezza Ortografica	38
5.4.13	Metriche non soddisfatte	38
5.4.14	Test Pass Rate	39
5.4.15	Rischi non calcolati	39
5.4.16	Efficienza Oraria	39
5.5	Metriche di Prodotto	39
5.5.1	Copertura dei requisiti obbligatori	39
5.5.2	Copertura dei requisiti desiderabili	40
5.5.3	Copertura dei requisiti opzionali	40
5.5.4	Code Coverage	40
5.5.5	Branch Coverage	40



5.5.6	Statement Coverage	41
5.5.7	Error Rate	41
5.5.8	Tempo di apprendimento	41
5.5.9	Response Time	41
5.5.10	Accoppiamento tra classi (CBO)	42
5.5.11	Complessità Ciclomatica	42

1 Introduzione

1.1 Scopo del documento

Questo documento ha lo scopo di descrivere i metodi e le regole (way of working_G) che il gruppo seguirà durante tutto lo sviluppo del progetto, al fine di ottenere un risultato efficace e di evitare problemi di organizzazione. Sono presenti nel documento anche i riferimenti a tutte le tecnologie e strumenti che vengono utilizzati, insieme alle motivazioni per cui si è deciso di adottarli.

Nel realizzare il proprio way of working_G il gruppo ha preso come principale riferimento lo standard ISO/IEC 12207:1995, che individua tre categorie di processi:

- **Primari:** processi centrali del ciclo di vita del software;
- **Supporto:** processi che assistono l'esecuzione dei processi primari fornendo funzioni trasversali e qualità;
- **Organizzativi:** processi utilizzati dall'organizzazione per stabilire, supportare, migliorare i processi del ciclo di vita e le competenze.

1.2 Scopo del prodotto

Lo scopo di DIPReader è fornire un sistema software per la consultazione e la conservazione digitale_G dei documenti. L'applicazione permette all'utente di caricare localmente i pacchetti informativi (DIP_G) per esplorarne i file_G, gli allegati e i relativi metadati_G. L'obiettivo principale è garantire un accesso offline rapido e sicuro alle informazioni, assicurando allo stesso tempo l'autenticità, la leggibilità e la totale conformità ai requisiti normativi di archiviazione.

1.3 Glossario

Si rimanda al documento *Glossario* per chiarire i termini nel dominio del progetto che possono risultare ambigui o di difficile comprensione.

I vocaboli presenti nel glossario sono stati segnalati seguendo la notazione:

parola_G

1.4 Riferimenti

1.4.1 Riferimenti normativi

- Capitolato C3
 - <https://www.math.unipd.it/~tullio/IS-1/2025/Progetto/C3.pdf>
(Ultimo accesso: 2025/12/20)
- Standard ISO/IEC 12207:1995

- https://www.math.unipd.it/~tullio/IS-1/2009/Appfondimenti/ISO_12207-1995.pdf

(Ultimo accesso: 2026/01/17)

1.4.2 Riferimenti informativi

- Slide del corso: T2 - Processi di ciclo di vita del software
 - <https://www.math.unipd.it/~tullio/IS-1/2025/Dispense/T02.pdf>
(Ultimo accesso: 2025/12/22)
- Versionamento semantico
 - <https://semverdoc.org/>
(Ultimo accesso: 2025/11/19)
- *Glossario v0.5*
 - https://team-three-technologies.github.io/Documenti/1%20-%20RTB/Documenti%20interni/Glossario_v0_5.pdf
(Ultimo accesso: 2026/03/17)

2 Processi Primari

2.1 Fornitura

Il processo di Fornitura definisce le attività che il fornitore_G deve svolgere per pianificare, gestire, realizzare e consegnare il prodotto software. Il processo copre la pianificazione delle attività, il coordinamento delle risorse e il controllo dell'avanzamento e della qualità. Nell'ambito della fornitura rientrano anche le attività di consegna, supporto all'accettazione e gestione delle comunicazioni con la proponente_G e il committente_G, al fine di garantire che il prodotto software fornito sia conforme ai requisiti, ai tempi e ai vincoli concordati.

2.1.1 Attività previste

Le attività previste dal processo di Fornitura sono:

- **Inizializzazione:** avvio del processo di fornitura a seguito della richiesta del committente, con identificazione dell'ambito della fornitura e delle responsabilità del fornitore;
- **Preparazione della richiesta di proposta (gara):** preparazione della proposta di fornitura, includendo la descrizione tecnica della soluzione, il piano di lavoro, le risorse previste, i costi e i tempi;
- **Preparazione e aggiornamento del contratto:** definizione, negoziazione e aggiornamento del contratto di fornitura, specificando obblighi, criteri di accettazione, modalità di controllo e gestione delle modifiche;
- **Monitoraggio del fornitore:** monitoraggio dell'esecuzione della fornitura, con controllo dell'avanzamento, della qualità e della conformità alle condizioni contrattuali;
- **Accettazione e completamento:** supporto alle attività di accettazione del prodotto da parte dell'acquirente e chiusura formale della fornitura al completamento degli obblighi contrattuali.

2.1.2 Documentazione fornita

In seguito sono elencati i documenti che, oltre al presente, sono necessari per la consegna del progetto, fino alla revisione RTB_G.

2.1.2.1 Analisi dei Capitolati

L'Analisi dei Capitolati è un documento redatto in fase di Candidatura che contiene, per ogni capitolato_G, una breve descrizione del prodotto e dei suoi obiettivi, caratteristiche positive e negative individuate dal gruppo e una considerazione finale del suo gradimento tra i membri.

2.1.2.2 Analisi dei Requisiti

L'Analisi dei Requisiti_G è un documento che contiene l'elenco dei requisiti, funzionali e non, relativi al prodotto. Contiene anche una lista dei casi d'uso con annessi attori che sono stati utilizzati, congiuntamente alle riunioni con la proponente_G, per individuare i suddetti requisiti.

2.1.2.3 Dichiarazione di Impegni

La Dichiarazione di Impegni è un documento redatto in fase di Candidatura che contiene la stima delle ore per ruolo, il preventivo dei costi del progetto e la previsione della data di consegna.

2.1.2.4 Glossario

Il Glossario è un documento, costantemente aggiornato durante tutta la durata del progetto, che ha lo scopo di chiarire i termini nel dominio del progetto che possono risultare ambigui o di difficile comprensione.

2.1.2.5 Lettera di Presentazione

Le Lettere di Presentazione sono dei documenti che presentano il prodotto software e la relativa documentazione alle fasi di revisione del progetto. Contengono un link al sito web del gruppo, dove è possibile consultare la documentazione, un elenco di quanto è stato prodotto e/o aggiornato e un preventivo del costo aggiornato.

2.1.2.6 Piano di Progetto

Il Piano di Progetto è un documento che contiene, divise per ciascuno sprint_G, le informazioni relative alla pianificazione delle attività del gruppo come ruoli, ore impiegate e un preventivo di costo aggiornato. Contiene inoltre l'analisi dei rischi che si possono presentare durante lo svolgimento del progetto.

2.1.2.7 Piano di Qualifica

Il Piano di Qualifica è un documento che contiene una descrizione dei metodi di verifica_G e validazione_G che il gruppo utilizza per garantire la qualità del prodotto.

2.1.2.8 Verbali

I Verbali, esterni ed interni, sono dei documenti che riassumono le riunioni svolte, rispettivamente tra i membri del gruppo e congiuntamente con la proponente_G.

2.2 Sviluppo

Il processo di Sviluppo definisce le attività da svolgere per progettare, realizzare e verificare il software a partire dai requisiti assegnati. Esso comprende l'analisi dei requisiti_G, la progettazione dell'architettura e dei componenti, la codifica, l'integrazione e le attività di verifica_G previste per ciascuna fase.

Lo scopo è quello di assicurare che il software prodotto sia conforme ai requisiti, correttamente documentato e pronto per le successive fasi.

2.2.1 Attività previste

Le attività previste dal processo di Sviluppo sono:

- **Implementazione del processo:** definizione e applicazione del processo di sviluppo all'interno del progetto, adattando metodologie, strumenti e procedure;
- **Analisi dei requisiti del sistema:** analisi dei requisiti del sistema, identificando le funzioni, le prestazioni e i vincoli complessivi;
- **Progettazione dell'architettura di sistema:** definizione dell'architettura di sistema, individuando componenti, interfacce e relazioni tra di essi;
- **Analisi dei requisiti software:** analisi dei requisiti software derivati dal sistema, specificando funzionalità e vincoli tecnici del software;
- **Progettazione dell'architettura software:** progettazione dell'architettura software, definendo moduli, componenti e interazioni principali;
- **Progettazione di dettaglio del software:** progettazione di dettaglio dei componenti software, incluse strutture dati, algoritmi e interfacce interne;
- **Codifica e testing del software:** implementazione del codice sorgente e test dei singoli moduli per verificarne il corretto funzionamento;
- **Integrazione software:** integrazione dei moduli software in un sistema coerente, con verifica delle interazioni tra componenti;
- **Test di qualifica del software:** verifica del software rispetto ai requisiti specificati, con test funzionali e prestazionali;
- **Integrazione del sistema:** integrazione del software nel sistema complessivo, includendo hardware e altri componenti;
- **Test di qualifica del sistema:** verifica del sistema completo, controllando la conformità ai requisiti di sistema e alle specifiche di progetto;
- **Installazione del software:** installazione del software nell'ambiente operativo, preparazione della configurazione e della documentazione per l'uso;
- **Supporto all'approvazione del software:** supporto alle attività di collaudo e accettazione da parte dell'acquirente, fornendo assistenza tecnica e documentazione necessaria.

Viste le due baseline_G previste dal progetto, ovvero Requirements and Technology Baseline_G e Product Baseline_G, il gruppo ha diviso le attività che ha ritenuto relative ad esse in due gruppi:

RTB Analisi dei requisiti (del sistema e software) e Codifica

PB Progettazione dell'architettura (di sistema e software), Progettazione di dettaglio del software, Codifica e testing del software, Integrazione software, Test di qualifica (del software e del sistema)

2.2.2 Analisi dei requisiti

L'analisi dei requisiti_G è un'attività fondamentale del processo di sviluppo e tra le più importanti entro la prima baseline_G del progetto (**RTB_G**). Il suo scopo è quello trasformare le richieste e le esigenze espresse dalla proponente_G in requisiti chiari, completi e verificabili che coprono tutte le funzionalità e i vincoli che il progetto dovrà soddisfare.

Il risultato di questa attività è formalizzato nel documento *Analisi dei Requisiti_G*, redatto dagli Analisti, il quale diventerà un riferimento chiaro per le successive attività di progettazione, implementazione e test_G, contribuendo a ridurre ambiguità e rischi durante lo sviluppo.

2.2.2.1 Casi d'uso

I casi d'uso rappresentano degli scenari in cui il sistema viene utilizzato per soddisfare i bisogni dell'utente. Il loro valore aggiunto sta nella descrizione testuale, che rende chiari le condizioni in cui si possono realizzare e come si svolgono, senza sfociare in dettagli implementativi.

Ogni caso d'uso_G è identificato da un codice e da un titolo, secondo la seguente convenzione:

UCX.Y[.Z] - Titolo

dove:

- **UC** è un prefisso per **Use Case**;
- **X** è il **numero** del caso d'uso principale, unico per caso d'uso principale;
- **Y** è il **numero** del sotto-caso d'uso, aumentato ogni volta che si definisce un sotto-caso d'uso;
- **Z** (opzionale) è il **numero** del sotto-sotto-caso d'uso, aumentato ogni volta che si definisce un sotto-sotto-caso d'uso.

Si è deciso di utilizzare la notazione di sotto-caso d'uso quando è necessario separare un caso d'uso_G per poterlo descrivere al meglio, collegando i sotto-casi tramite inclusione al caso genitore. Inoltre si è deciso di poter utilizzare quanti livelli di sotto-casi sia necessario, con le stesse regole per la numerazione, limitandosi però nel documento a trattarli come una "sotto-sotto-sezione" per mantenere la leggibilità dell'indice.

Un caso d'uso_G è composto da:

- **Pre-condizioni:** descrizione dello stato del sistema richiesto perché si possa verificare il caso d'uso;
- **Post-condizioni:** descrizione dello stato del sistema dopo l'esecuzione del caso d'uso;

- **Attore principale:** attore che interagisce attivamente con il sistema nel caso d'uso per raggiungere il suo obiettivo;
- **Attore secondario** (Opzionale): attore che interagisce passivamente con il sistema nel caso d'uso;
- **Scenario principale:** passi di esecuzione del caso d'uso;
- **Scenari alternativi** (Opzionale): passi che interrompono la normale esecuzione del caso d'uso e si discostano da essa;
- **Inclusioni** (Opzionale): riferimento ai casi d'uso collegati tramite la relazione "include";
- **Estensioni** (Opzionale): riferimento ai casi d'uso collegati tramite la relazione "extend";
- **Generalizzazione** (Opzionale): riferimento ai caso d'uso "figli", da cui il caso d'uso eredita le funzionalità.

2.2.2.2 Diagrammi dei casi d'uso

I diagrammi dei casi d'uso, realizzati secondo la specifica UML_G, sono rappresentazione grafiche delle funzionalità del sistema. Mostrano l'interazione tra gli attori e i casi d'uso e le relazioni tra i casi d'uso stessi. Essi consentono di descrivere in modo chiaro e intuitivo le funzionalità offerte dal sistema dal punto di vista degli utenti, evidenziando i comportamenti attesi senza entrare nei dettagli implementativi.

I diagrammi dei casi d'uso supportano l'analisi dei requisiti_G facilitando la comprensione delle funzionalità, la validazione_G dei requisiti con la proponente_G e la comunicazione tra tutti gli stakeholder_G coinvolti nel progetto.

2.2.2.3 Requisiti

I requisiti, ricavati dal capitolato_G e dai casi d'uso, sono specifiche che rappresentano funzionalità e vincoli del sistema. Fungono da riferimento per tutte le attività successive, dalla progettazione alla verifica_G. Il gruppo ha adottato la seguente codifica per identificare univocamente un requisito_G e facilitare il suo tracciamento:

RX-Y-Z

dove:

- **R** è un prefisso per **Requisito**;
- **X** rappresenta il **numero** del requisito;
- **Y** può assumere i valori **F**, **Q** o **V** per rappresentare, rispettivamente, un requisito **Funzionale**, di **Qualità** o di **Vincolo**;
- **Z** può assumere i valori **Ob**, **De** o **Op** per rappresentare, rispettivamente, un requisito **Obbligatorio**, **Desiderabile** o **Opzionale**.



2.2.3 Codifica

La codifica è un'attività centrale in un progetto di questo tipo, il cui successo dipende dalla qualità del lavoro di analisi e progettazione svolto.

Deve comunque sottostare a regole stabilite in modo da produrre codice di qualità, cioè facilmente comprensibile, mantenibile, testabile e che rispetti i requisiti e le specifiche.

2.2.3.1 Norme di codifica

Le norme di codifica che il gruppo ha deciso di adottare sono le seguenti:

- **Variabili globali:** vanno evitate il più possibile;
- **Nomi:** vanno utilizzati per variabili, metodi e classi nomi significativi e chiari;
- **Commenti:** nel caso in cui il nome non risulti abbastanza esplicativo o la logica di un pezzo di codice sia troppo complessa va aggiunto un commento adeguato per spiegare il funzionamento.
- **Funzioni:** non vanno definite funzioni troppo lunghe che si occupano di operazioni diverse;
- **Indentazione:** va rispettata l'indentazione del codice per favorire la sua leggibilità.

3 Processi di Supporto

3.1 Documentazione

Il processo di Documentazione è fondamentale per supportare i processi primari, serve infatti a tenere traccia di quello che è stato prodotto e delle decisioni prese. Esso garantisce la creazione e la gestione sistematica della documentazione necessaria a descrivere il software, i processi e le responsabilità, facilitando la comunicazione e la tracciabilità lungo l'intero ciclo di vita.

3.1.1 Attività previste

Le attività previste dal processo di Documentazione sono:

- **Implementazione del processo:** definizione e attuazione delle regole, delle responsabilità e degli strumenti necessari per la produzione e la gestione della documentazione;
- **Progettazione e sviluppo:** redazione dei documenti secondo le specifiche stabilite, includendo struttura, contenuti e criteri di qualità;
- **Produzione:** creazione, revisione, approvazione e distribuzione della documentazione alle parti interessate.
- **Manutenzione:** aggiornamento e controllo continuo dei documenti per mantenerli coerenti con le modifiche del software e dei processi.

3.1.2 Template

Per standardizzare e velocizzare la produzione dei documenti sono stati prodotti vari template.

- `docsTemplate.sty`
Importa tutti i pacchetti utilizzati dai documenti, definisce i comandi per impostare le informazioni nella pagina di copertina e definisce la struttura di header e footer delle pagine;
- `docTitlepage.tex`
Definisce la struttura della pagina di copertina per i documenti;
- `memoTitlepage.tex`
Definisce la struttura della pagina di copertina per i Verballi;
- `usecase.sty`
Definisce l'ambiente `usecase` e tutti i comandi per produrre i casi d'uso nel documento Analisi dei Requisiti.

3.1.3 Versionamento

Ogni documento ha una vita e cambia nel corso di essa, per questo viene versionato secondo quanto definito dall'attività di Identificazione della configurazione (§3.2.2).

3.1.4 Nomenclatura

I documenti hanno una nomenclatura specifica, nel caso di un documento generico il nome del file_G è:

Titolo_del_Documento

quindi il titolo del documento con gli spazi sostituiti dal carattere underscore ("_").

Nel caso di un Verbale il nome del file_G è:

V[E/I]_YYYYMMDD

A questa prima parte viene aggiunta la versione del documento nel formato:

_vX_Y

Nel caso si tratti di un Verbale Esterno firmato da un referente della proponente_G si aggiunge "**_firmato**".

3.1.5 Struttura dei documenti

La **pagina di copertina** presenta una parte comune a tutti i documenti e una parte aggiuntiva nel caso si tratti di un Verbale.

Dall'alto verso il basso si trova:

- **Informazioni del gruppo:** logo del gruppo affiancato dal nome e dalla e-mail;
- **Titolo del documento:** titolo del documento o, nel caso si tratti di un Verbale:

Verbale [Esterno/Interno] Data (in formato DD/MM/YYYY)

- **Informazioni progetto:** "Progetto di Ingegneria del Software";
- **Redazione:** nomi dei membri del gruppo che hanno contribuito alla redazione del documento;
- **Verifica:** nomi dei membri del gruppo che hanno contribuito alla redazione del documento;
- **Approvazione:** nomi dei membri del gruppo che hanno approvato il documento;
- **Versione:** versione attuale del documento;
- **Tipo:** tipologia del documento, può essere Esterno o Interno;
- **Destinatari:** persone alle quali il documento è destinato;
- **Informazioni Università:** nome dell'Università, corso di Laurea e anno accademico.

Nel caso si tratti di un Verbale si trovano informazioni aggiuntive quali:

- **Luogo:** luogo dove si è svolto l'incontro verbalizzato;
- **Ora:** ora in cui si è svolto l'incontro verbalizzato;
- **Assenti:** nomi degli assenti all'incontro verbalizzato.

Successivamente si trova il **registro delle modifiche**, che ha lo scopo di mantenere traccia di tutte le modifiche al documento. La sua struttura è definita dall'attività di Registrazione dello stato di configurazione (§3.2.4).

Seguono poi l'**indice del documento**, l'**elenco delle figure** e l'**elenco delle tabelle**, generati automaticamente tramite appositi comandi LaTeX_G.

Ogni pagina, esclusa quella di copertina, deve contenere un **header** con logo e nome del gruppo e un **footer** con titolo del documento e numero di pagina.

3.1.5.1 Struttura dei Verbali

I Verbali, esterni e interni, hanno una struttura ben definita, divisa in quattro sezioni:

- **Ordine del giorno:** descrizione per punti degli argomenti di discussione della riunione;
- **Svolgimento:** riassunto della riunione;
- **Decisioni:** tabella che contiene le decisioni prese e una breve descrizione, ciascuna identificata da un codice nel formato:

TIPO_DATA.dX

dove: **TIPO** può assumere i valori **VE** o **VI** se si tratta di una Verbale Esterno o Interno, **DATA** è la data della riunione nel formato YYYYMMDD e **X** è il numero progressivo identificativo della decisione;

- **Attività:** tabella che contiene le azioni risultanti dalla riunione e dalle decisioni, con il loro assegnatario, la task e il numero della corrispondente issue su GitHub (se presente). Ciascuna è identificata da un codice nel formato:

TIPO_DATA.aX

dove: **TIPO** può assumere i valori **VE** o **VI** se si tratta di una Verbale Esterno o Interno, **DATA** è la data della riunione nel formato YYYYMMDD e **X** è il numero progressivo identificativo dell'azione.

3.1.6 Produzione

L'attività di produzione inizia il ciclo di vita di un documento.

L'assegnatario della task, deciso tramite riunione sincrona o comunicazione asincrona, crea il file_G relativo, registrata la versione "v0_1" e scrive il contenuto. Terminata la stesura il documento è sottoposto a verifica_G; nel caso di successo il verificatore_G registra il proprio nome nel registro delle modifiche e condivide il documento, altrimenti riporta una lista di errori/modifiche necessarie. Quando è necessaria l'approvazione del documento questa viene richiesta al responsabile_G.

3.1.7 Manutenzione

L'attività di manutenzione garantisce l'evoluzione di un documento in modo che questo rimanga coerente.

Il suo svolgimento è simile a quello della produzione, quando è necessaria una modifica si aumenta il numero di versione corretto, si effettuano le modifiche e si richiede la verifica_G.

3.2 Gestione della Configurazione

Il processo di Gestione della Configurazione applica procedure tecniche e amministrative lungo il ciclo di vita del software per identificare e controllare gli elementi software, gestirne le modifiche e i rilasci, tracciarne lo stato e garantirne coerenza, correttezza e integrità.

3.2.1 Attività previste

Le attività previste dal processo di Gestione della Configurazione sono:

- **Implementazione del processo:** definizione e messa in atto delle regole, delle responsabilità, delle procedure e degli strumenti necessari per la gestione della configurazione durante il ciclo di vita del software;
- **Identificazione della configurazione:** individuazione degli elementi di configurazione, assegnazione di identificatori univoci e definizione delle baseline e delle relazioni tra gli elementi;
- **Controllo della configurazione:** gestione formale delle richieste di modifica, valutazione, approvazione o rifiuto dei cambiamenti e controllo dei rilasci degli elementi di configurazione;
- **Registrazione dello stato della configurazione:** raccolta, aggiornamento e comunicazione delle informazioni sullo stato degli elementi di configurazione e delle richieste di modifica;
- **Valutazione della configurazione:** verifica che gli elementi di configurazione siano completi, corretti e coerenti con i requisiti e con le baseline approvate;

- **Gestione dei rilasci e distribuzione:** preparazione, identificazione e consegna dei rilasci del software e della documentazione associata agli utenti o ai sistemi destinatari.

3.2.2 Identificazione della configurazione

Il gruppo ha adottato per i documenti il seguente sistema di versionamento a due cifre:

X.Y

dove:

- **X** indica la versione *major*, incrementata quando viene apportata una modifica incompatibile con le versioni precedenti;
- **Y** indica la versione *minor*, incrementata quando viene apportata una modifica retrocompatibile con le versioni precedenti.

La base da cui si ha preso spunto è il versionamento_G semantico (vedi §1.4.2), scegliendo di rimuovere la terza cifra per via dell'evoluzione dei documenti. Infatti il gruppo ha deciso che la correzione di piccoli errori (*es.* tipografici) rilevati dal processo di verifica_G debba ricadere nella stessa versione, mentre per errori di dimensione maggiore (*es.* sezioni semanticamente errate) in quella immediatamente successiva, prodotta in tempo relativamente breve.

3.2.3 Controllo della configurazione

3.2.3.1 Progetto GitHub

Per tracciare le task da svolgere il gruppo ha adottato la soluzione offerta da GitHub_G tramite la creazione di un progetto. Questo permette di associare le issue, che rappresentano le attività, create all'interno di un repository_G ad una project board, dove vengono visualizzate e sono facilmente consultabili.

Ogni issue è composta principalmente da:

- un **identificativo**;
- un **nome** che ne descrive il contenuto;
- una **descrizione** (opzionale);
- un **assegnatario**;
- una **milestone** a cui è associata;
- uno **stato**.

La board divide le issue in tre sezioni in base al loro stato:

- **Todo**, dove si trovano le issue create e non ancora prese in carico;
- **In progress**, dove si trovano le issue in esecuzione dagli assegnatari
- **Done**, con le issue già completate.

3.2.3.2 Repository

Il gruppo utilizza dei repository_G GitHub_G per condividere e mantenere i prodotti dei vari processi:

- **Documenti:** contiene i file sorgenti della documentazione prodotta durante lo svolgimento del progetto, divisi in cartelle per periodo di progetto (Candidatura e RTB);
- **PoC:** contiene i file sorgenti prodotti durante la realizzazione del Proof of Concept.

3.2.3.3 Sito web

Il sito web del gruppo, hostato tramite GitHub_G Pages, permette la visualizzazione dei documenti. Grazie ad uno script eseguito tramite GitHub_G Actions, ogni volta che viene effettuato un push sul branch *main* vengono prodotti i file_G *.pdf* a partire dai sorgenti *.tex*. Lo script inoltre raggruppa i documenti seguendo la stessa struttura del repository_G.

3.2.3.4 Termini del glossario

Il gruppo ha sviluppato uno script per automatizzare la segnalazione dei termini presenti nel Glossario all'interno degli altri documenti. Sempre tramite GitHub_G Actions lo script, prima della compilazione dei *.tex* in PDF, estrae i termini dal Glossario, che ha struttura precisa, li ricerca e marca ogni occorrenza secondo lo standard definito.

L'aver reso questa azione automatica accelera il processo di scrittura dei documenti, lasciando tempo per concentrarsi sui contenuti.

3.2.4 Registrazione dello stato della configurazione

Allo scopo di registrare e mantenere lo stato di configurazione dei documenti e la sua storia, il gruppo ha incluso in ognuno di essi un **registro delle modifiche**, ovvero una tabella che contiene, per ogni riga:

- **Versione:** identificativo numerico della modifica;
- **Data:** data della modifica;
- **Autore:** autore della modifica;
- **Ruolo:** ruolo dell'autore della modifica;
- **Verifica:** responsabile della verifica;
- **Oggetto:** oggetto della modifica.

Il registro è ordinato in ordine cronologico inverso, quindi in cima si trovano le modifiche più recenti.

3.3 Garanzia di Qualità

Il processo di Garanzia di Qualità da prova che i processi di ciclo di vita e i prodotti software sono conformi ai requisiti specificati e agli standard stabiliti, in altre parole che siano efficaci e efficienti.

La qualità viene invece assicurata dall'implementazione dei processi di Verifica_G e Validazione_G.

3.3.1 Attività previste

Le attività previste dal processo di Garanzia di Qualità sono:

- **Implementazione del processo:** definizione e applicazione delle attività, delle responsabilità e delle procedure necessarie per attuare il processo di assicurazione della qualità;
- **Garanzia di qualità di prodotto:** verifica che i prodotti software e la documentazione soddisfino i requisiti specificati, gli standard applicabili e i criteri di qualità definiti;
- **Garanzia di qualità di processo:** verifica che i processi utilizzati per sviluppare e mantenere il software siano correttamente definiti, seguiti e coerenti con quanto pianificato;
- **Garanzia di qualità dei sistemi di qualità:** valutazione dell'adeguatezza e dell'efficacia del sistema di gestione della qualità dell'organizzazione nel supportare i processi e i prodotti software.

3.3.2 Garanzia di qualità di prodotto

Allo scopo di assicurare la qualità del prodotto il gruppo ha scelto diverse metriche, i cui valori desiderabili e il loro andamento durante il corso del progetto sono riportati nel *Piano di Qualifica* (§4). Queste metriche sono descritte nella **sezione 5**, in particolare le metriche dedicate alla qualità di prodotto si trovano nella **sottosezione 5.5**.

3.3.3 Garanzia di qualità di processo

Allo scopo di assicurare la qualità dei processi il gruppo ha scelto diverse metriche, i cui valori desiderabili e il loro andamento misurati durante il corso del progetto sono riportati nel *Piano di Qualifica* (§4). Queste metriche sono descritte nella **sezione 5**, in particolare le metriche dedicate alla qualità di processo si trovano nella **sottosezione 5.4**.

3.4 Verifica

Il processo di Verifica_G ha come obiettivo quello di verificare che quanto è stato prodotto, in tutte le fasi del ciclo di vita, sia conforme ai requisiti, alle specifiche e agli standard definiti, garantendo che i risultati ottenuti corrispondano a quanto pianificato.

3.4.1 Attività previste

Le attività previste dal processo di Verifica_G sono:

- **Implementazione del processo:** definizione e applicazione delle procedure, dei criteri e delle responsabilità necessarie per eseguire le attività di verifica lungo il ciclo di vita del software;
- **Verifica:** accertamento che i prodotti intermedi e finali del software siano conformi alle specifiche, ai requisiti e agli standard definiti.

3.4.2 Attività di Verifica

3.4.2.1 Analisi statica

L'Analisi statica è un tipo di verifica_G che non prevede l'esecuzione dell'oggetto in esame, per questo è applicabile ad ogni prodotto di processo, dalla documentazione al software.

- **Walkthrough**
Nel walkthrough il presupposto è quello di non conoscere dove si possano trovare gli errori. Per questo si prevede una lettura per intero e senza presupposti della parte oggetto di verifica. L'esame avviene simulando possibili esecuzioni del codice oppure analizzando ogni sezione del documento.
- **Inspection**
Nell'inspection si cercano i difetti nei punti in cui si pensa che questi potrebbero trovarsi. Si prepara una checklist con quello che va controllato e poi la si segue esaminando l'oggetto.
Sono state preparate due checklist, consultabili nel *Piano di Qualifica* (§3.5): una valida per ogni documento e una specifica per alcune parti dell'*Analisi dei Requisiti*.

3.4.2.2 Analisi dinamica

L'Analisi dinamica prevede l'esecuzione dell'oggetto sottoposto a verifica_G, per questo è applicabile solamente ai prodotti software. Vengono pianificati, implementati e eseguiti diversi tipi di test_G allo scopo di scovare i difetti nella parti di codice in esame. I test_G devono essere **ripetibili** e **automatizzabili**.

Le tipologie di test_G sono:

- **Test di unità**
I test di unità verificano la correttezza delle unità software e dei moduli a loro associati. Ne esistono di due tipi: quelli **funzionali**, che hanno lo scopo di assicurare la corrispondenza tra le classi di input (valori illegali, valori legali di limite e valori legali) e gli output, e quelli **strutturali**, che invece verificano la correttezza della logica interna dell'unità.
- **Test di integrazione**
I test di integrazione vengono eseguiti per verificare che le componenti architettoniche

lavorino correttamente insieme.

L'integrazione può essere perseguita secondo due modalità: **bottom-up**, che prioritizza la realizzazione delle componenti con minori dipendenze e maggiore utilità, e **top-down**, che invece prioritizza le componenti con maggiori dipendenze e maggior valore aggiunto esterno.

- **Test di sistema**

I test di sistema verificano che i requisiti del sistema siano rispettati, quindi che il prodotto software sia corretto rispetto alle aspettative.

- **Test di regressione**

I test di regressione accertano che i cambiamenti apportati a delle unità non danneggino il resto del sistema.

Si realizzano tramite la ripetizione di test di unità, di integrazione e di sistema.

- **Test di accettazione**

Un test di accettazione accerta che i requisiti utente siano soddisfatti alla presenza del committente e della proponente.

I test_G sono identificati da un codice univoco nel formato:

TXY

dove:

- **T** è un prefisso per **Test**;
- **X** è il tipo del test e può assumere i valori **U**, **I**, **S** o **A** per rappresentare, rispettivamente, un test di **Unità**, di **Integrazione**, di **Sistema** o di **Accettazione**;
- **Y** è un numero progressivo che identifica il test, univoco per tipo.

Ad ogni test_G è inoltre associato uno stato, che può essere:

- **NI**, per **Non Implementato**;
- **I**, per **Implementato**;
- **S**, per **Superato**.

Queste informazioni sono disponibili nel *Piano di Qualifica* (§3).

3.5 Validazione

Il processo di Validazione_G ha come obiettivo quello di accertare che il prodotto software finale soddisfi i bisogni reali degli utenti e gli scopi per cui è stato sviluppato, verificando che il sistema risulti adeguato al contesto operativo previsto e che risponda efficacemente agli usi attesi.

3.5.1 Attività previste

Le attività previste dal processo di Validazione_G sono:

- **Implementazione del processo:** definizione e applicazione delle procedure, dei criteri e delle responsabilità necessarie per eseguire le attività di validazione;
- **Validazione:** conferma che il prodotto software soddisfi i bisogni dell'utente e l'uso previsto nell'ambiente operativo

3.5.2 Attività di Validazione

Allo scopo di eseguire l'attività di validazione_G il gruppo si prepara ad un test_G di accettazione. Per arrivare preparati e garantirne il successo il gruppo si impegna a svolgere la verifica_G in modo puntuale e al meglio delle proprie possibilità, rispettando tutte le regole e gli standard definiti.

4 Processi Organizzativi

4.1 Gestione dei Processi

Il processo di Gestione dei processi ha l'obiettivo di definire metodologie adeguate per coordinare in modo efficiente le risorse umane coinvolte nel progetto. Attraverso l'analisi sistematica delle attività e dei flussi di lavoro è possibile individuare eventuali criticità e ottimizzare l'organizzazione complessiva. Un aspetto fondamentale è rappresentato dalla documentazione accurata di ogni fase, che consente di mantenere traccia delle decisioni prese e dei risultati ottenuti. In questo modo si favorisce un miglioramento continuo dei processi e si assicura una maggiore qualità del prodotto finale.

4.1.1 Attività previste

Le attività previste dal processo di Gestione dei Processi sono:

- **Avvio e definizione dell'ambito:** identificazione delle richieste di modifica, analisi preliminare e definizione di cosa deve essere mantenuto o modificato, con la determinazione dei confini e degli obiettivi dell'intervento;
- **Pianificazione:** definizione delle attività, delle risorse, dei tempi, dei criteri di accettazione e delle modalità di gestione delle modifiche e dei rischi per eseguire la manutenzione;
- **Esecuzione e controllo:** realizzazione delle modifiche, test e verifica delle soluzioni implementate, monitoraggio dello stato di avanzamento e gestione delle non conformità;
- **Revisione e valutazione:** analisi dei risultati ottenuti, valutazione dell'efficacia delle modifiche, verifica della conformità ai requisiti e alle specifiche, e raccolta di eventuali lezioni apprese;
- **Chiusura:** completamento delle attività, aggiornamento della documentazione e delle baseline, rilascio delle versioni aggiornate e formalizzazione della chiusura dell'intervento di manutenzione.

4.1.2 Ruoli di progetto

Durante l'intera durata del progetto, ogni membro del gruppo si vedrà assegnato almeno una volta ciascuno dei sei ruoli. Ogni ruolo ha una funzione ben definita riguardante specifiche attività ed è necessario per il raggiungimento degli obiettivi finali. In base alle necessità dello sprint_G verranno assegnati i ruoli, ruotandoli in maniera adeguata tra i componenti del gruppo. Di seguito si trova una breve descrizione di ogni ruolo e delle attività che svolgerà.

4.1.2.1 Responsabile

Il Responsabile_G è la figura incaricata di dirigere e coordinare il lavoro del gruppo, garantendo che le attività vengano svolte in modo coerente con gli obiettivi del progetto. Ha il compito

di organizzare le diverse fasi operative, monitorarne l'avanzamento e assicurare il rispetto delle scadenze pianificate. Inoltre, svolge un ruolo di riferimento per la comunicazione tra il team e la proponente_G, facilitando lo scambio di informazioni e la risoluzione di eventuali problematiche. Infine è tenuto a valutare rischi e criticità, prendendo decisioni volte a mantenere il progetto in linea con le aspettative iniziali.

I suoi compiti sono:

- Comunicare con la proponente;
- Stesura continua del Piano di Progetto;
- Stesura dei Verbali interni ed esterni;
- Creazione dei diari di bordo;
- Organizzare ulteriori riunioni interne in caso di necessità del gruppo.

4.1.2.2 Amministratore

L'Amministratore_G è responsabile_G della gestione operativa degli strumenti e delle risorse necessarie allo svolgimento del progetto. Si occupa della configurazione e della manutenzione degli ambienti di lavoro, assicurandone il corretto funzionamento nel tempo. Tra le sue mansioni rientrano anche il controllo delle scadenze, l'organizzazione della documentazione tecnica e il supporto al gruppo nell'utilizzo delle tecnologie adottate. Il suo contributo è fondamentale per garantire un'infrastruttura stabile ed efficiente, indispensabile per il regolare avanzamento delle attività progettuali.

I suoi compiti sono:

- Configurare le repository Github;
- Assegnare eventuali issue ai membri del gruppo;
- Gestire tutti gli strumenti per la comunicazione e collaborazione del gruppo;
- Stesura continua delle Norme di Progetto;
- Stesura continua del Piano di Qualifica.

4.1.2.3 Analista

L'Analista_G è la figura incaricata di individuare e definire in modo chiaro e completo i requisiti del progetto. Ha il compito di raccogliere le informazioni necessarie, analizzarle e trasformarle in specifiche funzionali e casi d'uso comprensibili per il team di sviluppo. Collabora attivamente con la proponente_G per chiarire dubbi e approfondire le necessità espresse, assicurandosi che quanto progettato sia coerente con le aspettative iniziali. Il suo contributo è fondamentale per garantire che il prodotto finale soddisfi pienamente gli obiettivi prefissati.

I suoi compiti sono:

- Studiare i requisiti necessari per il corretto sviluppo del progetto;
- Discutere con la proponente riguardante i requisiti richiesti;
- Stesura dell'Analisi dei Requisiti.

4.1.2.4 Progettista

Il Progettista_G è responsabile_G della definizione dell'architettura del sistema software e della sua organizzazione strutturale. A partire dai requisiti elaborati dagli Analisti, individua le soluzioni tecniche più adeguate e ne definisce le modalità di realizzazione. Si occupa di stabilire le scelte progettuali fondamentali e di delineare l'impianto complessivo del prodotto, fornendo linee guida chiare per le fasi successive di sviluppo.

I suoi compiti sono:

- Scegliere le migliori tecnologie da utilizzare, valutando vantaggi e svantaggi delle diverse opzioni;
- Creare un'architettura generale mostrando le tecnologie da usare e le varie interazioni che devono esserci;
- Controllare che lo sviluppo sia conforme all'architettura progettata.

4.1.2.5 Programmatore

Il Programmatore_G ha il compito di realizzare concretamente il sistema software sulla base delle indicazioni fornite dai Progettisti. Si occupa di sviluppare le diverse componenti applicative, implementando le funzionalità richieste e integrandole tra loro in modo coerente. Durante lo svolgimento delle proprie attività deve attenersi alle specifiche tecniche e agli standard definiti, assicurando che il codice prodotto rispetti i criteri di qualità stabiliti. Il suo ruolo è essenziale per trasformare il progetto teorico in un prodotto effettivamente funzionante.

I suoi compiti sono:

- Scrivere codice coerente con le specifiche individuate;
- Risolvere eventuali bug o problemi del codice;
- Creare dei test per accertarsi del corretto funzionamento del software sviluppato.

4.1.2.6 Verificatore

Il Verificatore_G è la figura preposta al controllo della qualità del lavoro svolto all'interno del progetto. Ha il compito di esaminare le attività realizzate dagli altri membri del gruppo, accertandosi che siano conformi alle norme e alle procedure stabilite. Attraverso revisioni, test_G e controlli mirati, garantisce che i risultati ottenuti rispettino i requisiti richiesti e che eventuali errori o incongruenze vengano individuati e corretti tempestivamente.

I suoi compiti sono:

- Controllare la documentazione prodotta dai vari membri del gruppo;
- Verificare il codice prodotto individuandone eventuali errori e possibili miglioramenti.

4.1.3 Comunicazione

In questo capitolo vengono descritte tutte le modalità di comunicazione, sia sincrone che asincrone, che il gruppo adotterà per mantenere il contatto tra i propri membri e con i soggetti esterni.

4.1.3.1 Comunicazioni Interne

Per la comunicazione tra i membri del gruppo si è deciso di adottare le seguenti modalità sincrone:

- **Incontri in presenza:** verranno organizzati, in base alle disponibilità dei membri, degli incontri in presenza con tutti membri del gruppo;
- **Discord:** è la principale piattaforma che verrà usata per le riunioni da remoto. Consente di effettuare sia chiamate vocali che videochiamate di gruppo, permettendo di avere anche più canali vocali attivi contemporaneamente con la possibilità di cambiare velocemente il canale in cui si è presenti. Questo permette al gruppo di dividersi eventualmente in gruppi più piccoli in caso di necessità. Offre inoltre una chat per la condivisione di messaggi testuali o multimediali.

In supporto alle comunicazioni sincrone appena descritte, verranno utilizzate se seguenti piattaforme per le comunicazioni asincrone:

- **WhatsApp:** è stato creato un gruppo Whatsapp dedicato al progetto, nel quale sono presenti tutti i membri del gruppo. Si è deciso all'unanimità di utilizzare questo strumento rispetto ad altri vista la sua semplicità, versatilità e la maggiore familiarità che tutti i componenti del gruppo hanno con esso. Lo scopo di questo gruppo è quello di contenere le discussioni del gruppo che non richiedono di avere una tracciabilità;
- **Telegram:** è stato creato un gruppo Telegram all'interno del quale sono stati creati una serie di canali (chat), ognuno dei quali è dedicato a un argomento specifico. Questo strumento ha lo scopo di aiutare i membri del gruppo nei loro compiti in quanto facilita la ricerca delle discussioni e del materiale condiviso in passato. Nel gruppo verranno inoltrare le decisioni prese, le attività in corso e la suddivisione corrente dei ruoli e degli incarichi (ovviamente tutte queste informazioni sono presenti anche nei verbali degli incontri, il gruppo serve a darne un accesso più centralizzato, sintetizzato e a portata di mano). Sono presenti i seguenti canali:
 - **Generale;**
 - **Todo in corso:** contiene allo stato attuale gli incarichi che stanno svolgendo i membri;

- **Registrazioni:** contiene le registrazioni audio delle riunioni interne;
- **Riunioni:** contiene gli appunti fatti durante le riunioni;
- **Verbali:** contiene le copie dei verbali;
- **Link utili:** contiene link a risorse che alcuni membri hanno ritenuto utili nello svolgimento delle loro attività;
- **Back-end:** chat dedicata a discussioni riguardanti lo sviluppo back-end del progetto;
- **Front-end:** chat dedicata a discussioni riguardanti lo sviluppo front-end del progetto;
- **Diari di Bordo:** contiene una copia delle presentazioni dei diari di bordo e eventuali domande che possono venire inserite per il prossimo diario di bordo;
- **Dubbi:** chat in cui condividono dubbi che potrebbero impattare in modo significativo lo svolgimento del progetto. Generalmente sono domande da proporre al committente o alla proponente.

Per quanto venga fortemente incoraggiato il loro utilizzo massivo, è opportuno ribadire che lo scopo di questi strumenti è di supplemento agli incontri sincroni, non possono quindi essere utilizzato come rimpiazzo di questi ultimi. La programmazione delle attività, la suddivisione degli incarichi, la prenotazione degli incontri e tutte le altre attività di coordinamento e organizzazione del progetto rimarranno argomento delle riunioni sincrone.

4.1.3.2 Riunioni Interne tra i Membri del Gruppo

Le riunioni in remoto verranno preferite a quelle in presenza in quanto la loro organizzazione è molto più semplice e si integra più facilmente con le necessità (personali e organizzative) dei componenti del gruppo. A tal proposito è stata tenuta in forte considerazione la non banale distanza geografica tra i partecipanti al progetto.

Verranno organizzate:

- Una **riunione di aggiornamento** di cadenza bisettimanale;
- Ulteriori **riunioni di supplemento** qualora si incontrassero difficoltà o una mole di lavoro che necessiti di un maggiore sforzo cooperativo.

In entrambi i casi il giorno e l'ora degli incontri verranno decisi basandosi sul resoconto della **disponibilità fornita** dai membri; al termine di ogni incontro di aggiornamento verrà programmato quello successivo. Ovviamente, gli incontri bisettimanali di aggiornamento verranno programmati per non essere troppo vicini gli uni agli altri. Il gruppo si impegna a mantenere con rigore lo svolgimento, quantomeno, bisettimanale di questi incontri. Nel caso in cui si verificano imprevisti che impediscano uno svolgimento dell'incontro tale che esso porti a un risultato utile, l'incontro verrà rinviato alla prima data libera.

Per aiutare a migliorare l'accuratezza dei verbali interni si è deciso di registrare (in formato audio) le riunioni interne. Le registrazioni verranno inviate nell'apposito canale del gruppo Telegram_G.

4.1.3.3 Comunicazioni Esterne

Le comunicazioni con attori esterni avverranno nelle seguenti modalità:

- Sincrone:
 - **Google Meet**: riunioni da remoto in videochiamata.
- Asincrone:
 - **Email**: scambio di email con il committente e con la proponente tramite la email unica del gruppo;
 - **Google Sheets**: per gestire più velocemente le domande che possono sorgere si è deciso di usare un file condiviso su Google Sheets. Il gruppo inserirà in una tabella, presente nel documento, la descrizione del quesito indicandone anche la priorità. La proponente poi inserirà, sulla stessa riga la sua risposta. Questo metodo permette inoltre di avere uno storico di tutte le domande che sono state poste e le relative risposte.

4.1.3.4 Riunioni Esterne con il Committente e la Proponente

È stato deciso, in accordo con la proponente_G, che verranno svolte delle riunioni di aggiornamento ogni due settimane, salvo esigenze particolari di una delle due parti. Si è poi indentificato il giovedì come giorno più adatto in cui fissare le riunioni, in quanto è il giorno della settimana che meno interferisce con gli impegni a cadenza regolare dei membri del gruppo.

4.1.4 Pianificazione degli Impegni

Per riuscire a facilitare la pianificazione degli impegni e tenere traccia delle disponibilità dei membri del gruppo si è deciso di usare le funzionalità offerte da **Google Calendar**. Verrà usato il calendario incorporato con la email del gruppo, nel quale ogni membro indicherà i momenti della settimana in cui non potrà essere reperibile per eventuali impegni. Per fare ciò occorrerà che ogni componente crei un'attività nel giorno e ora in cui non sarà disponibile, inserisca come titolo dell'attività il suo nome e le dia il colore dedicato.

Invece, gli impegni del gruppo come riunioni, scadenze, diari di bordo ecc. verranno contrassegnati sia aggiungendo un'attività al calendario con la rispettiva ora, sia inserendo in quel giorno un "Evento" in modo che quando si apre il calendario saltino più facilmente all'occhio i giorni in cui sono programmate attività collettive.

Le attività del calendario avranno un colore specifico in base alla loro tipologia. Segue la legenda dei colori:

- **Blu (o azzurro)**: indica un momento della giornata in cui un membro del gruppo non è disponibile;
- **Rosso**: indica Diari di Bordo, consegne, e incontri con il committente;

- **Giallo**: indica incontri con la proponente;
- **Viola**: indica le milestone del progetto;
- **Verde**: indica le riunioni interne del gruppo.

4.2 Infrastruttura

Il processo di Infrastruttura è necessario per la corretta creazione ed il corretto mantenimento dei componenti necessari per permetterli tutti i processi.

4.2.1 Attività previste

Le attività previste dal processo di Infrastruttura sono:

- **Implementazione del processo**: definizione e attuazione delle procedure, delle responsabilità e degli strumenti necessari per gestire l'infrastruttura di supporto ai processi del ciclo di vita;
- **Creazione dell'infrastruttura**: progettazione, acquisizione e messa a disposizione delle risorse, degli strumenti, degli ambienti e dei servizi necessari per svolgere i processi;
- **Manutenzione dell'infrastruttura**: gestione e aggiornamento continuo delle risorse e degli strumenti, garantendo la loro disponibilità, adeguatezza e funzionamento nel tempo.

4.2.2 Strumenti

Di seguito vengono elencati i vari strumenti utilizzati per lo svolgimento del progetto didattico.

4.2.2.1 Discord

<https://discord.com/>

Piattaforma di messaggistica istantanea, di distribuzione digitale e di voice over IP, utilizzata per la comunicazione all'interno server Discord_G. Gli utenti comunicano con chiamate vocali, videochiamate, messaggi di testo, media e file_G in chat private o come membri di un server Discord_G. Quest'ultimi sono una raccolta di canali di tipo vocale e/o testuale.

4.2.2.2 Git

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

Applicazione software per il controllo di versione distribuito utilizzabile da interfaccia a riga di comando. Viene utilizzato il versionamento_G delle repository_G dedicate al codice e alla documentazione del progetto.



4.2.2.3 Github

<https://github.com/>

È un servizio di hosting per progetti software che implementa lo strumento di controllo versione distribuito Git.

4.2.2.4 Gmail

<https://workspace.google.com/products/gmail/>

È un servizio gratuito di posta elettronica fornito da Google.

4.2.2.5 Google Calendar

<https://workspace.google.com/products/calendar/>

Sistema di calendari fornito da Google. Consente la gestione di un calendario condiviso, l'annotazione di attività ed eventi e la visione simultanea di tutti i calendari a cui si partecipa.

4.2.2.6 Google Docs

<https://workspace.google.com/products/docs/>

Strumento fornito da Google che permette la creazione e la gestione condivisa di file_G di testo online.

4.2.2.7 Google Meet

<https://workspace.google.com/products/meet/>

Strumento fornito da Google che permette di comunicare tramite videochiamate.

4.2.2.8 Google Sheets

<https://workspace.google.com/intl/it/products/sheets/>

Strumento fornito da Google che permette la creazione e la gestione condivisa di fogli di calcolo online.

4.2.2.9 Overleaf

<https://www.overleaf.com/about/features-overview>

Strumento che permette la creazione e gestione di file_G LaTeX_G.

4.2.2.10 Telegram

<https://telegram.org/>

Telegram_G è una piattaforma di messaggistica istantanea e broadcasting basata su cloud. Caratteristiche di Telegram_G sono la possibilità di scambiare messaggi di testo tra due utenti o tra gruppi, effettuare chiamate vocali e videochiamate cifrate point-to-point, scambiare messaggi vocali, immagini, video, file_G di qualsiasi tipo. Permette inoltre la suddivisione dei gruppi in canali distinti.

4.2.2.11 WhatsApp

<https://www.whatsapp.com/>

Piattaforma di messaggistica istantanea.

4.3 Miglioramento

Il processo di Miglioramento ha lo scopo di definire un insieme di attività finalizzate all'analisi e all'ottimizzazione dei processi adottati durante lo sviluppo di un prodotto software. Tale processo si basa su un approccio sistematico che prevede la definizione, il monitoraggio e la revisione continua delle metodologie utilizzate, con l'obiettivo di incrementarne qualità, efficienza_G ed efficacia_G. Attraverso un'attenta valutazione delle prestazioni e l'applicazione di azioni correttive mirate, è possibile garantire un progressivo miglioramento dell'intero ciclo di sviluppo.

4.3.1 Attività previste

Le attività previste dal processo di Miglioramento sono:

- **Implementazione del processo:** definizione e documentazione del processo di miglioramento, compresi obiettivi, metodi, ruoli e criteri di successo;
- **Valutazione del processo:** analisi e misurazione delle prestazioni dei processi esistenti per individuare punti di forza, criticità e opportunità di miglioramento;
- **Miglioramento del processo:** definizione e implementazione delle azioni correttive e migliorative, con monitoraggio dei risultati e aggiornamento continuo del processo.

4.3.2 Implementazione del processo

Per poter avviare correttamente il processo di miglioramento è fondamentale definire in modo chiaro e strutturato i processi organizzativi che regolano lo sviluppo del progetto. Tali processi devono essere formalizzati e descritti attraverso un'adeguata documentazione, in modo da renderli comprensibili, ripetibili e verificabili. La fase di implementazione costituisce quindi il punto di partenza per tutte le attività successive e ha l'obiettivo di stabilire una base solida su cui fondare le future operazioni di valutazione e ottimizzazione.

4.3.3 Valutazione del processo

Una volta implementati i vari processi, diventa necessario verificarne il corretto funzionamento nel tempo. La fase di valutazione ha lo scopo di analizzare le attività svolte e di misurarne le prestazioni attraverso indicatori e metriche specifiche. Questo permette di individuare eventuali inefficienze o problematiche e di comprendere se i processi adottati risultino adeguati rispetto agli obiettivi prefissati. Le informazioni raccolte durante questa fase costituiscono un elemento fondamentale per orientare le successive azioni di miglioramento.

4.3.4 Miglioramento del processo

Sulla base dei risultati ottenuti dalle attività di misurazione e valutazione, è possibile individuare gli aspetti dei processi che necessitano di essere ottimizzati. In questa fase vengono quindi definite e applicate opportune azioni correttive volte a risolvere le criticità emerse e a incrementare l'efficacia complessiva delle metodologie utilizzate. Ogni modifica introdotta deve essere opportunamente documentata e monitorata, in modo da verificarne l'impatto e garantire un'evoluzione continua e controllata dei processi nel tempo.

4.4 Formazione

4.4.1 Descrizione e scopo

Il processo di Formazione ha l'obiettivo di garantire che tutti i componenti del gruppo acquisiscano le competenze necessarie per svolgere in modo efficace le attività previste dal progetto. Tale processo prende avvio dall'analisi delle necessità formative emerse in relazione ai requisiti stabiliti dal proponente, per poi individuare gli strumenti e le risorse più idonee a soddisfarle. Mediante un percorso di apprendimento continuo, il gruppo può ampliare progressivamente il proprio bagaglio di conoscenze tecniche e metodologiche, migliorando la padronanza delle tecnologie impiegate. La formazione rappresenta quindi un elemento essenziale per la buona riuscita del progetto, oltre a costituire un'importante occasione di crescita personale e professionale per ciascun partecipante.

4.4.2 Formazione individuale

Nel corso dello sviluppo del progetto, ogni membro del gruppo è responsabile del proprio percorso di apprendimento e dell'approfondimento delle competenze richieste. L'acquisizione delle conoscenze necessarie avviene sia attraverso lo studio autonomo sia mediante il confronto e la collaborazione con gli altri componenti del gruppo. La condivisione delle esperienze e delle informazioni consente infatti di ridurre eventuali lacune e di favorire una crescita uniforme delle capacità del gruppo. Questo modello di formazione individuale e collaborativa permette di raggiungere un livello di preparazione adeguato alle esigenze del progetto e contribuisce al miglioramento complessivo delle prestazioni del team.

5 Metriche di qualità

5.1 Scopo

Questa sezione ha lo scopo di definire e analizzare le metriche utilizzate per il monitoraggio della qualità del prodotto software e dell'efficienza_G dei processi di sviluppo. L'attività di misurazione non è fine a se stessa, ma rappresenta uno strumento essenziale di controllo di gestione: attraverso la raccolta e la valutazione costante di questi indicatori, il gruppo intende verificare oggettivamente il raggiungimento degli obiettivi di qualità prefissati. Questo approccio permette di identificare tempestivamente eventuali criticità e attuare le dovute correzioni, così da garantire un miglioramento continuo.

5.2 Standard di riferimento per la qualità di prodotto

Per garantire rigore metodologico e completezza, il gruppo ha deciso di seguire le linee guida dettate dallo standard ISO/IEC 9126. Di conseguenza, le metriche relative alla qualità del prodotto sono state definite e classificate coprendo le sei caratteristiche fondamentali previste dalla normativa:

- **Funzionalità:** rappresenta la capacità del software di fornire le funzioni necessarie per operare in un determinato contesto. Questa caratteristica valuta attributi quali l'adeguatezza, l'accuratezza dei risultati, l'interoperabilità con altri sistemi e la sicurezza dei dati;
- **Affidabilità:** indica la capacità del prodotto di mantenere un determinato livello di prestazione quando utilizzato in condizioni specificate. Essa misura la maturità del software nell'evitare malfunzionamenti, la tolleranza ai guasti e la capacità di recupero dati in caso di errore;
- **Usabilità:** definisce la capacità del software di essere compreso, appreso e utilizzato, risultando al contempo attraente per l'utente. Include la valutazione della comprensibilità delle funzioni, la facilità di apprendimento (learnability) e l'operabilità del sistema;
- **Efficienza:** misura la capacità del software di realizzare le funzioni richieste nel minor tempo possibile e ottimizzando l'uso delle risorse (sia hardware che software). Si focalizza sul comportamento rispetto al tempo (tempi di risposta ed elaborazione) e sull'utilizzo delle risorse;
- **Manutenibilità:** rappresenta la capacità del prodotto di essere modificato per correzioni, miglioramenti o adattamenti a cambiamenti nei requisiti o nell'ambiente. Valuta quanto il software sia analizzabile per diagnosi di errori, modificabile e testabile dopo le variazioni;
- **Portabilità:** indica la capacità del software di essere trasportato da un ambiente operativo all'altro. Questa caratteristica copre l'adattabilità a differenti ambienti, la facilità di installazione e la capacità di coesistere con altre applicazioni.

5.3 Nomenclatura delle metriche

Per l'identificazione delle metriche è stata adottata la seguente nomenclatura univoca:

$$M[PC/PD][X]$$

Dove i vari segmenti rappresentano:

- **M**: prefisso che sta per **Metrica**;
- **PC**: acronimo per **Processo**, indica metriche relative all'andamento del progetto;
- **PD**: acronimo per **Prodotto**, indica metriche relative alla qualità del software e dei documenti;
- **X**: numero identificativo crescente che parte da 01, le due tipologie (Processo e Prodotto) hanno conteggi separati.

5.4 Metriche di processo

5.4.1 Metriche soddisfatte

- **Notazione**: MPC01
- **Descrizione**: Percentuale delle metriche che soddisfano il valore di accettazione rispetto al numero totale di metriche definite
- **Formula**:

$$\% \text{Metriche Soddisfatte} = \frac{N_{\text{soddisfatte}}}{N_{\text{totali}}} \cdot 100$$

5.4.2 Earned Value (EV)

- **Notazione**: MPC02
- **Descrizione**: Valore (in euro) del lavoro effettivamente completato rispetto al valore pianificato alla data corrente
- **Formula**:

$$EV = \text{Budget previsto} \times \% \text{Lavoro completato}$$

5.4.3 Planned Value (PV)

- **Notazione**: MPC03
- **Descrizione**: Costo stimato per il lavoro da completare entro la data corrente secondo il piano originale
- **Formula**:

$$PV = BAC \cdot \% \text{Lavoro Pianificato}$$

(Nota: *BAC = Budget At Completion, cioè il budget totale preventivato*)

5.4.4 Actual Cost (AC)

- **Notazione:** MPC04
- **Descrizione:** Costo effettivo sostenuto alla data corrente per il lavoro svolto (somma delle ore rendicontate e spese).
- **Formula:**

$$AC = \sum \text{Costi sostenuti}$$

5.4.5 Estimated to Completion (ETC)

- **Notazione:** MPC05
- **Descrizione:** Budget stimato necessario per completare il progetto
- **Formula:**

$$ETC = BAC - AC$$

(Nota: *BAC = Budget At Completion, cioè il budget totale preventivato*)

5.4.6 Estimated at Completion (EAC)

- **Notazione:** MPC06
- **Descrizione:** Stima del costo totale del progetto alla fine, basata sul rendimento attuale e sui costi già sostenuti
- **Formula:**

$$EAC = AC + ETC$$

5.4.7 Cost Variance (CV)

- **Notazione:** MPC07
- **Descrizione:** Differenza in percentuale tra il valore guadagnato (EV) e il costo effettivamente sostenuto (AC). Un valore negativo indica che il progetto è fuori budget
- **Formula:**

$$CV = (EV - AC) \cdot 100$$

5.4.8 Schedule Variance (SV)

- **Notazione:** MPC08
- **Descrizione:** Misura se il progetto sta progredendo velocemente come pianificato. Un valore negativo indica un ritardo sui tempi
- **Formula:**

$$SV = \frac{EV - PV}{PV} \cdot 100$$

5.4.9 Cost Performance Index (CPI)

- **Notazione:** MPC09
- **Descrizione:** Misura l'efficienza dei costi, calcolando il rapporto tra il valore del lavoro svolto e il costo reale sostenuto
- **Formula:**

$$CPI = \frac{EV}{AC}$$

5.4.10 Requirements Stability Index

- **Notazione:** MPC10
- **Descrizione:** Variazione dei requisiti durante la pianificazione
- **Formula:**

$$RSI = \left(1 - \frac{N_{aggiunti} + N_{rimossi} + N_{modificati}}{N_{totali}} \right) \cdot 100$$

5.4.11 Indice Gulpease

- **Notazione:** MPC11
- **Descrizione:** Indice che valuta il grado di leggibilità di un testo in lingua italiana. Un valore basso indica bassa leggibilità, un valore alto (es. > 40 per testi tecnici) indica buona leggibilità.
- **Formula:**

$$IG = 89 + \frac{300 \cdot N_{frasi} - 10 \cdot N_{lettere}}{N_{parole}}$$

5.4.12 Correttezza Ortografica

- **Notazione:** MPC12
- **Descrizione:** Rileva la presenza di errori ortografici o grammaticali all'interno del testo.

5.4.13 Metriche non soddisfatte

- **Notazione:** MPC13
- **Descrizione:** Indica il numero totale di metriche che, alla data corrente, non raggiungono la soglia minima di accettazione definita.
- **Formula:**

$$M_{ns} = \sum \text{Metriche sotto soglia}$$

5.4.14 Test Pass Rate

- **Notazione:** MPC14
- **Descrizione:** Misura la percentuale di test superati con successo rispetto al numero totale di test eseguiti.
- **Formula:**

$$TPR = \frac{N_{\text{test_passati}}}{N_{\text{test_totali}}} \cdot 100$$

5.4.15 Rischi non calcolati

- **Notazione:** MPC15
- **Descrizione:** Metrica che indica il numero di rischi che si sono verificati ma che non erano stati preventivati nell'Analisi dei Rischi.
- **Formula:**

$$R_{nc} = \sum \text{Rischi imprevisi verificatisi}$$

5.4.16 Efficienza Oraria

- **Notazione:** MPC16
- **Descrizione:** Tiene traccia del rapporto tra le Ore di Orologio (totali) rispetto alle Ore Produttive (impiegate in attività che portano direttamente al raggiungimento di obiettivi).
- **Formula:**

$$EO = \frac{OR}{OP} \cdot 100$$

(Dove *OR* = Ore Orologio, *OP* = Ore Produttive)

5.5 Metriche di Prodotto

5.5.1 Copertura dei requisiti obbligatori

- **Notazione:** MPD01
- **Descrizione:** Indica la percentuale di requisiti obbligatori effettivamente implementati rispetto al totale dei requisiti obbligatori definiti.
- **Formula:**

$$RO = \frac{N_{\text{ReqObb_implementati}}}{N_{\text{ReqObb_totali}}} \cdot 100$$

5.5.2 Copertura dei requisiti desiderabili

- **Notazione:** MPD02
- **Descrizione:** Indica la percentuale di requisiti desiderabili effettivamente implementati rispetto al totale dei requisiti desiderabili definiti.
- **Formula:**

$$RD = \frac{N_{\text{ReqDes_implementati}}}{N_{\text{ReqDes_totali}}} \cdot 100$$

5.5.3 Copertura dei requisiti opzionali

- **Notazione:** MPD03
- **Descrizione:** Indica la percentuale di requisiti opzionali effettivamente implementati rispetto al totale dei requisiti opzionali definiti.
- **Formula:**

$$ROp = \frac{N_{\text{ReqOp_implementati}}}{N_{\text{ReqOp_totali}}} \cdot 100$$

5.5.4 Code Coverage

- **Notazione:** MPD04
- **Descrizione:** Misura la percentuale di linee di codice sorgente effettivamente eseguite durante i test rispetto al numero totale di linee di codice.
- **Formula:**

$$CC = \frac{N_{\text{Linee_testate}}}{N_{\text{Linee_totali}}} \cdot 100$$

5.5.5 Branch Coverage

- **Notazione:** MPD05
- **Descrizione:** Misura la percentuale di rami (flussi decisionali, *es.* `if/else`) percorsi durante i test rispetto al totale dei rami esistenti nel codice.
- **Formula:**

$$BC = \frac{N_{\text{Rami_testati}}}{N_{\text{Rami_totali}}} \cdot 100$$

5.5.6 Statement Coverage

- **Notazione:** MPD06
- **Descrizione:** Misura la percentuale di singole istruzioni del codice eseguite almeno una volta durante i test rispetto al totale delle istruzioni.
- **Formula:**

$$SC = \frac{N_{\text{Istruzioni_testate}}}{N_{\text{Istruzioni_totali}}} \cdot 100$$

5.5.7 Error Rate

- **Notazione:** MPD07
- **Descrizione:** Valuta la facilità d'uso del sistema misurando la frequenza con cui l'utente commette errori durante un'interazione o un task specifico.
- **Formula:**

$$ER = \frac{N_{\text{errori}}}{N_{\text{interazioni}}}$$

5.5.8 Tempo di apprendimento

- **Notazione:** MPD08
- **Descrizione:** Misura il tempo medio necessario affinché un utente riesca a comprendere e utilizzare correttamente una specifica funzionalità del sistema.
- **Formula:**

$$T_{\text{apprendimento}} = \text{Tempo impiegato per raggiungere la competenza}$$

5.5.9 Response Time

- **Notazione:** MPD09
- **Descrizione:** Misura il tempo che intercorre tra l'invio di un comando (o richiesta) da parte dell'utente e la visualizzazione della risposta (o output) da parte del sistema.
- **Formula:**

$$T_{\text{response}} = T_{\text{risposta}} - T_{\text{richiesta}}$$

5.5.10 Accoppiamento tra classi (CBO)

- **Notazione:** MPD10
- **Descrizione:** Misura il grado di dipendenza tra le classi (Coupling Between Objects). Un valore alto indica che una classe è fortemente dipendente da altre, rendendo la manutenzione più difficile.

- **Formula:**

$$CBO = \text{Numero di classi a cui una classe è accoppiata}$$

5.5.11 Complessità Ciclomatica

- **Notazione:** MPD11
- **Descrizione:** Misura la complessità strutturale del codice calcolando il numero di cammini linearmente indipendenti attraverso il grafo di controllo del flusso.

- **Formula:**

$$CC = E - N + 2P$$

(Dove E = Numero di archi, N = Numero di nodi, P = Componenti connesse)