Università degli Studi di Padova

Laurea: Informatica

Corso: Ingegneria del Software
Anno Accademico: 2025/26

BitByBit

Gruppo 17

Nome: BitByBit
Email: swe.bitbybit@gmail.com

Norme di progetto

Registro delle modifiche

Versione	Data	Autore	Descrizione	Verificatore
0.2.1	2025-12-02	Riccardo Manisi	Aggiunta sezioni per processo di configurazione e di gestione.	Marco Sanguin
0.1.1	2025-12-02	Riccardo Manisi	Sistemati errori lessicali	Marco Sanguin
0.1.0	2025-12-01	Riccardo Manisi	Creazione del documento. Aggiunto processo di documentazione e bozza per le sezioni successive.	Marco Sanguin

Indice

1. Introduzione

1.1. Scopo del documento

Il presente documento è volto a definire i processi del way of working per il gruppo BitByBit che si impegna a mantenerlo per tutta la durata del progetto. Per la sua redazione è stato preso come riferimento lo Standard ISO/IEC 12007:1995 che identifica 3 tipologie di processo:

1. **processi primari:** atti alla produzione di un prodotto software;
2. **processi di supporto:** supportano altri processi come parte integrante e contribuiscono al successo e alla qualità del prodotto software;
3. **processi organizzativi:** definiscono i metodi organizzativi del gruppo, per stabilire e implementare la struttura costituita dai processi di ciclo di vita.

Ogni membro del gruppo si impegna a visionare costantemente il presente documento e a rispettare rigorosamente i processi che vengono esposti di seguito, per ottenere un prodotto che sia professionale, coerente e uniforme.

1.2. Scopo del prodotto

L'azienda **Miriade srl** ha proposto lo sviluppo di un'applicazione mobile, denominata "L'app che Protegge e Trasforma" finalizzata alla prevenzione e al supporto delle vittime di violenze di genere.
Il prodotto punta a stabilirsi come uno strumento intelligente e sicuro che aiuta l'utilizzatore a riconoscere segnali di pericolo e a fornire risorse per tutelarsi da atti, che siano fisici o psicologici, che potrebbero nuocere alla sua salute. L'applicazione implementa metodologie volte alla tutela delle persone che sono già vittima di violenze, ma anche per la valutazione e l'individuazione di situazioni che potrebbero sfociare in atti di violenza.

1.3. Glossario

I documenti prodotti nei processi di ciclo di vita sono corredati da un glossario che costituisce un riferimento condiviso. Il suo scopo è ridurre le ambiguità lessicali che possono emergere a causa dei diversi livelli di competenze tecniche tra i destinatari dei documenti. Poiché le Norme di Progetto, l'Analisi dei Requisiti, il Piano di Progetto e il Piano di Qualifica sono rivolti a figure eterogenee, tra cui acquirenti, fornitori, sviluppatori, gestori e utenti del prodotto software, è necessario che ogni termine potenzialmente ambiguo, tecnico o soggetto a interpretazioni multiple sia riportato in modo chiaro e univoco.
Per favorire la reperibilità dei termini e rendere immediata la loro identificazione all'interno dei documenti, ogni voce del glossario è richiamata nel testo mediante la seguente convenzione stabilita dal gruppo BitByBit:

termine_nel_glosssario[G]

Il glossario è considerato parte integrante del processo di documentazione: viene aggiornato ogni volta che emergono nuovi termini rilevanti, che vengono introdotti nuovi concetti tecnici o che si rende necessario chiarire ambiguità riscontrate durante la redazione o la verifica dei documenti.

1.4. Riferimenti

1.4.1 Riferimenti normativi

- **Capitola d'appalto C4: L'app che Protegge e Trasforma**
  https://www.math.unipd.it/~tullio/IS-1/2025/Progetto/C4.pdf
- **Standard ISO/IEC 12207:1995**
  https://www.math.unipd.it/~tullio/IS-1/2009/Approfondimenti/ISO_12207-1995.pdf

1.4.2 Riferimenti informativi

-

2. Processi primari

3. Processi di supporto

La presente sezione espone i seguenti processi di supporto:

(a) **documentazione;**

(b) **gestione della configurazione;**

(c) **gestione della qualità;**

(d) **verifica;**

(e) **validazione.**

3.1. Documentazione

Il processo di documentazione ha l'obiettivo di definire le modalità, gli strumenti e le convenzioni che il gruppo adotta per la produzione, la gestione, la revisione e l'approvazione di tutti i documenti del progetto.

3.1.1 Obbiettivi

1. Garantire **completezza**, **chiarezza** e **coerenza** delle informazioni contenute nei documenti.
2. Assicurare la **tracciabilità** delle modifiche e delle versioni;
3. Favorire la **comprensibilità e manutenibilità** dei documenti da parte di tutti i membri del gruppo;
4. Fornire un **mezzo ufficiale di comunicazione** tra il gruppo, il committente e i revisori;
5. Rispettare la **uniformità formale e stilistica** tra i vari documenti prodotti.

L'implementazione del repository dedicato alla documentazione integra tutti gli strumenti necessari alla produzione dei documenti di ciclo di vita, garantendo che ogni nuovo membro del progetto possa svolgere le attività previste senza dipendere dal proprio ambiente di lavoro locale o incorrere in limitazioni tecniche.
È necessaria l'automazione dei processi dove è possibile, in modo da avere un sistema con pipeline per lo sviluppo che aiuti il membro del gruppo nei suoi compiti.

3.1.2 Strumenti a supporto

Per sostenere il processo di documentazione sono utilizzati i seguenti strumenti:

- **LaTeX**, per la stesura di documenti il gruppo ha scelto LaTeX, linguaggio di markup che permette di avere il controllo di tutte le parti di un documento, può essere eseguito nel repository senza la necessità di un editor di testo;
- **overleaf**, piattaforma online utilizzata in fase iniziale per l'apprendimento della sintassi e i comandi di LaTeX;
- **Google Sheets**: utilizzato per la redazione di pianificazioni, tracciamento di requisiti preliminari, matrici decisionali e supporto alla raccolta dati;
- **Google Docs**, impiegato durante riunioni o fasi preliminari per appunti condivisi e raccolta rapida di contenuti;
- **Google Presentazioni**: utilizzato per la redazione e condivisione dei diari di bordo richiesti dal corso;
- **GitHub**, piattaforma di versionamento distribuito e piattaforma di collaborazione utilizzata per la gestione centralizzata dei documenti, delle modifiche, dei branch e delle pull request;
- **GitHub Actions**, strumento di automazione per l'aggiunta del template comune a tutti i documenti e per la compilazione dei file `.tex` in `PDF`.

3.1.3 Tipologie di documenti

- **Documenti interni**, destinati alla gestione e al coordinamento interno (verbali, norme di progetto, glossario);
- **Documenti esterni**, destinati a revisori, docenti o aziende (Analisi dei Requisiti, Piano di Progetto, Piano di Qualifica);
- **Verbali interni ed esterni**, registrano rispettivamente, gli incontri tra i membri del gruppo e gli incontri tra il gruppo e gli enti esterni come l'azienda proponente o i professori.

Ogni documento, indipendentemente dalla sua categoria, deve essere redatto in conformità alle convenzioni formali definite nelle Norme di Progetto.

3.1.4 Implementazione di processo

Ogni documento, prima di essere integrato nella versione stabile del **repository**$^{\mathbf{G}}$, deve attraversare un ciclo strutturato di produzione che garantisce la qualità, la verificabilità e la tracciabilità delle modifiche. Le attività descritte in questa sezione definiscono il flusso operativo seguito dal gruppo per l'elaborazione, la revisione e la pubblicazione dei prodotti documentali.

3.1.4.1 Creazione delle issue GitHub Ogni modifica documentale, compresa anche la creazione di un documento, inizia con l'apertura di una issue su GitHub per permettere la tracciabilità nella board del **Project GitHub**$^{\mathbf{G}}$. Ogni commit deve essere atomico, cioè riferirsi ad un'unica azione in quanto migliorano la revisione, il recupero e la collaborazione.

3.1.4.2 Assegnazione della issue Nessuna issue può essere avviata senza una chiara assegnazione. L'assegnazione viene effettuata dal **responsabile**$^{\mathbf{G}}$, che determina chi, nel gruppo, è incaricato dell'esecuzione dell'attività. Questo passaggio garantisce che ogni modifica sia riconducibile a un autore e che la responsabilità operativa sia chiaramente definita.

3.1.4.3 Impostazione del documento Il membro del team può ora avviare la stesura del documento all'interno del branch dedicato, il cui nome segue la nomenclatura presente nella sezione (da aggiungere). La prima attività consiste nella creazione del file sorgente, denominato secondo le convenzioni stabilite (ad esempio nel formato `nome_documento.tex`). Una volta inserito il nuovo file nel repository remoto, il sistema provvede automaticamente ad applicare il template comune, garantendo l'integrazione delle componenti strutturali condivise.

Spetta quindi al redattore adattare il template al contenuto specifico del documento, completandone le sezioni pertinenti e verificando che l'impostazione risultante sia conforme agli standard definiti per la documentazione del progetto. Questa attività assicura uniformità, leggibilità e coerenza stilistica dell'intera produzione documentale lungo tutto il ciclo di vita del progetto.
Ogni membro del gruppo è quindi tenuto a verificare che i documenti prodotti rispettino un formato uniforme, verificabile e riconducibile agli standard scelti.
Questa attività garantisce la leggibilità, la coerenza e la mantenibilità dell'intera produzione documentale durante tutto il ciclo di vita del progetto.

Intestazione

Ogni pagina del documento include un'intestazione che riporta il titolo dello specifico documento e il nome del gruppo.

Frontespizio

Il frontespizio possiede i seguenti elementi:

- logo dell'ateneo;
- informazioni dell'ateneo;
- logo del gruppo;
- contatto ufficiale del gruppo;
- nome del documento.

Storico del documento

Nell'ultima pagina deve essere presente uno storico delle modifiche apportate al documento. Le entry della tabella devono essere in ordine cronologico in modo che la prima entry faccia riferimento alla modifica più recente. Una entry è composta da:

1. **versione**;
2. **data della modifica**;
3. **autore della modifica**;
4. **verificatore**, il nome del verificatore che ha verificato quella modifica.

Versione	Data	Autore	Descrizione	Verificatore

Indice

Ogni documento deve presentare un indice delle sezioni e sottosezioni per una visita veloce e facilitata di specifici all'interno dello stesso documento.

Struttura dei verbali

I verbali rappresentano i rendiconti di riunioni ufficiali interne, a cui partecipano solo i componenti del gruppo, ed esterne, a cui invece partecipano anche enti esterni al gruppo come le aziende fornitrici o i professori. Ogni verbale oltre alle sezioni precedentemente elencate deve presentare al suo interno i seguenti elementi nell'ordine in cui sono descritti:

1. **Informazioni generali**: contiene i dati identificativi dell'incontro quali:
   - il redattore;
   - il verificatore;
   - il responsabile;
   - l'amministratore;
   - la data in cui si è svolto l'incontro;
   - la durata;
   - l'elenco dei partecipanti e degli assenti.

   Queste informazioni sono volte a dare un contesto formale alla riunione.
2. **Ordine del giorno**: riassume i punti principali che si intendono discutere durante l'incontro, predisposti in anticipo dal responsabile o dall'amministratore. Questa sezione funge da guida per la conduzione del verbale.
3. **Discussioni**: riporta in modo sintetico ma chiaro i contenuti emersi durante il confronto tra i partecipanti. Devono essere descritte le osservazioni, le problematiche sollevate e le proposte di soluzione valutate durante l'incontro.
4. **Decisioni**: elenca tutte le decisioni approvate dal gruppo, incluse quelle riguardanti modifiche ai documenti, scadenze o variazioni organizzative. Ogni decisione deve essere formulata in modo oggettivo e deve avere un codice identificativo univoco che segue il seguente formato:

   `[VI/VE] [fase a cui appartiene] Y.Z`

   dove:
   - **VI/VE**, indica se la decisione fa riferimento ad un verbale interno o esterno, seguendo le convenzioni di scrittura stabilite.

- **fase**, indica la fase di progetto a cui si rifersice;
- **Y**, indica il numero del verbale in riferimento a quella revisione;
- **X** indica il numero della decisione interna a quel documento.

5. **To-do**: presenta, sotto forma di tabella, le attività operative pianificate a seguito della riunione. Ogni attività è identificata da:
   - una descrizione sintetica;
   - Il codice della decisione da cui è stata creata (per ogni decisione ci possono essere 0 o più attività);
   - il numero della issue GitHub di quella attività presente nel project.

Convenzioni di scrittura

Le convenzioni di scrittura stabiliscono regole comuni per la redazione di tutti i documenti del progetto.

3.1.4.4 Date

- Le **date** devono essere nel formato **AAAA-MM-GG**.

Le ore hanno 2 formati:

- **Durata**: Per espriemere una durata usare il formato **Hh Mm** (es: 2h 15m)
- **Ora di orologio**: Per indicare che un evento è avvenuto in una specifica ora, usare il formato **HH:MM** (es: 15:45)

Nomi dei file

- Tutti i file devono avere nomi in minuscolo, senza spazi, e le parole devono essere separate da un **underscore (_)**.
- Il nome di ogni verbale deve iniziare con `verbale_` , segue il tipo di verbale `interno` o `esterno` e la data in cui si è svolto. Un esempio di nominazione corretta è `verbale_interno_2025-11-05`.
- L'unica eccezione è nel caso in cui usare questo formato generi due verbali con lo stesso nome all'interno della stessa cartella. In tal caso è necessario aggiungere, dopo la data, una breve descrizione del contesto relativo al verbale, sostituendo agli spazi il simbolo `underscore (_)`. Ad esempio `verbale_2025-10-28_incontro_c8`.
- I `file PDF` mantengono lo stesso nome del file sorgente `.tex`.

Sigle

Di seguito sono presenti le sigle usate all interno dei documenti.

- **Fasi di progetto**:
  - **C**, Candidatura;
  - **RTB**, Requirements and Tecnology Baseline;
  - **PB**, Product Baseline.
- **Tipologie di documenti**:
  - **AdR**, analisi dei requisiti;
  - **NdP**, norme di progetto;
  - **PdQ**, piano di qualifica;
  - **PdP**, piano di progetto;
  - **G**, glossario;
  - **VI**, verbale interno;
  - **VE**, verbale esterno.

Nomi dei membri

- I nomi dei membri devono essere riportati nel formato: **Nome Cognome** (iniziali maiuscole, nessuna abbreviazione).

3.1.4.5 Modifiche a documenti Per eseguire delle modifiche alla documentazione di progetto, il membro del gruppo è tenuto a istanziare un nuovo branch di `feature`. La denominazione del branch di feature deve seguire la nomenclatura presente nella sezione (da aggiungere).
Ogni modifica a un documento deve essere accompagnata da un aggiornamento del numero di versione appropriato. Il membro del gruppo che si occupa delle modifiche è anche tenuto ad aggiornare lo **storico dei documenti** con le informazioni richieste.
Ogni modifica ad un documento deve essere tracciabile all'interno dello stesso tramite lo **storico del documento**, riportando:

- il numero di versione assegnato;
- la data della conclusione della modifica;
- il nome e cognome del membro del team che l'ha eseguita;
- una descrizione concisa, sisntetica e formale delle modifiche avvenute;
- il nome del verificatore che è tenuto ad eseguire la verifica.

3.1.4.6 Conclusione modifiche

Una volta completate le modifiche nel branch di `feature`, il membro incaricato procede ad aprire una **pull request**^{G} dal proprio branch di feature verso il branch `develop`. Nella fase di apertura della **pull request**^{G} devono essere compilate le informazioni richieste dal relativo template, specifico per il **repository**^{G} della documentazione. La **pull request**^{G} costituisce la richiesta formale per integrare le modifiche nella versione di sviluppo, permettendo al team di effettuare la revisione del codice e di approvarne l'unione.

3.1.4.7 Verifica

Il verificatore esamina sia la correttezza formale sia sostanziale del documento.

- Se le modifiche risultano corrette, il verificato approva la **pull request**^{G}.
- Se emergono incongruenze o problemi, il verificatore richiede modifiche aggiungendo dei commenti specifici al codice nel messaggio della **pull request**^{G} e aprendo una issue **GitHub**^{G} dedicata.

Le modifiche richieste devono essere applicate dall'autore originario della **pull request**^{G}, che procede aggiornando il branch `develop` fino alla completa risoluzione. I verificatori sono tenuti a controllare che i documenti di ciclo di vita del progetto comprendano gli elementi di struttura elencati di seguito nell' ordine in cui si presentano.

3.1.4.8 Approvazione e pull request

Una volta ottenuta l'approvazione del verificatore, la **pull request**^{G} viene sottoposta al **responsabile**^{G}, che effettua un ultima revisione formale. Solo il **responsabile**^{G} è autorizzato a procedere con il merge della PR nel branch `develop`. Il merge rappresenta l'integrazione definitiva del documento nella versione stabile del progetto.

3.1.4.9 Pubblicazione

Con la conclusione del merge, il documento aggiornato diventa parte integrante della **baseline**^{G} di progetto. Il branch `main` costituisce, infatti, il riferimento ufficiale per la versione stabile e coerente dei documenti.

3.2. Processo di gestione della configurazione

Il processo di gestione della configurazione ha l'obiettivo di applicare procedure per garantire il controllo sistematico degli artefatti prodotti dal gruppo *BitByBit* durante l'intero ciclo di vita del software.
Esso permette di tracciare le modifiche apportate agli elementi di configurazione, garantendo che ogni componente venga rilasciato in modo coerente e dopo le opportune verifiche. Inoltre, consente di registrare, classificare e monitorare le richieste di modifica, assicurando che i prodotti di lavoro risultino completi, coerenti e corretti. Infine, supporta una gestione ordinata degli elementi approvati, curandone l'archiviazione, la conservazione e la distribuzione.

3.2.1 Strumenti a supporto

Per le attività previste, il gruppo BitByBit utilizza gli stumenti elencati di seguito.

- **GitHub**: piattaforma per il versionamento, la gestione dei repository, delle **issue**^{G}, delle **pull request**^{G} e dei flussi di lavoro collaborativi mediante GitHub Projects e Teams. È inoltre il punto ufficiale di archiviazione degli artefatti approvati.
- **GitHub Actions**: utilizzate per automatizzare la compilazione dei documenti in LaTeX e garantire la validazione continua del repository.
- **Branch protection rules**: utilizzate per impedire modifiche non autorizzate sul branch principale e garantire che ogni integrazione avvenga secondo le regole di verifica previste dal gruppo.

3.2.2 Implementazione di processo

Le attività principali del processo di gestione della configurazione sono:

- **identificazione della configurazione**, viene definito uno schema uniforme di versionamento e naming, per identificare ogni componente del sistema;
- **controllo della configurazione**, metodologie per stabilire l'identificazione, l'approvazione o rifiuto, la verifica e l'eventuale rilascio delle change request;
- **registro di stato**, per predisporre registri di gestione e report di stato che mostrino lo stato e la cronologia degli elementi software;
- **valutazione della configurazione**, come determinare la completezza funzionale e fisica delle componenti software con i loro requisiti.

3.2.3 Identificazione della configurazione

Per il versionamento viene adottato il **Semantic Versioning (SemVer**^{G}**)**. Ogni documento riporta una versione nel formato:

`X.Y.Z`

dove:

- **X (major)**: incrementata solo alla pubblicazione ufficiale, corrispondente al merge sul branch `main`;
- **Y (minor)**: incrementata quando vengono effettuare modifiche sostanziali alla struttura o al contenuto del documento;
- **Z (patch)**: incrementata con correzioni minori, refusi o chiarimenti senza impatto tecnico.

Ogni documento di ciclo di vita presenta uno storico delle modifiche con associata la precisa versione. Questa scelta permette una chiara tracciabilità chiara, prevedibilità delle modifiche ed evoluzione controllata degli artefatti documentali.

3.2.3.1 Aggiunta dei tag

Al termine del processo di revisione, quando una **pull request**[G] viene approvata dai verificatori e successivamente chiusa dal **responsabile**[G], viene generato un commit di merge che rappresenta la versione consolidata delle modifiche introdotte nel branch `develop`.
Poiché questo commit costituisce il punto di riferimento ufficiale per la versione del documento o dell'artefatto modificato, il responsabile – oppure, se previsto, una GitHub Action – deve applicare un tag che identifichi formalmente la nuova versione.
Il formato per il tag sul branch `develop` è il seguente:

`sigla_del_documento-X.Y.Z`

dove `sigla_del_documento` fa riferimento alle convenzioni di scrittura presenti nel processo di documentazione (vedi sezione 3.1). Invece, `X.Y.Z` indica la versione con cui il documento entra ufficialmente nel repository, dopo la verifica e la validazione eseguite dalle figure del team adibite a questi compiti. Dopo il merge da develop a main si applica il tag della baseline alla commit risultante rispettando il formato seguente:

`sigla_della_baseline-X.Y.Z`

3.2.4 Controllo della configurazione

Il controllo della configurazione definisce le modalità con cui il gruppo gestisce, valuta e approva tutte le modifiche agli elementi configurabili del progetto.
L'intero ciclo di lavoro è organizzato secondo la pratica del feature branching, in cui per ogni modifica deve essere aperto un branch di `feature` seguendo la nomenclatura esposta in sezione (da aggiungere).

3.2.4.1 Gestione issue e pianificazione del lavoro

Ogni modifica nasce dall'apertura di una **issue GitHub**[G], che costituisce il punto di riferimento per la tracciabilità dell'attività.
Al momento della creazione, la **issue**[G] deve essere dotata di un titolo descrittivo e di una spiegazione chiara dell'obiettivo. Il responsabile assegna inoltre priorità, milestone, iterazione (che corrisponde allo sprint in cui l'attività deve essere completata) e l'assegnatario, selezionato in base al ruolo che il membro ricopre in quel momento. Label aggiuntive consentono una categorizzazione accurata del lavoro.
La pianificazione complessiva è supportata dalle **GitHub Project Board**[G], che permettono di consultare **product backlog**[G] e **sprint backlog**[G] tramite viste personalizzate, semplificando la gestione dei carichi di lavoro e l'avanzamento delle attività.

3.2.4.2 Creazione della pull request

Una volta completata l'attività descritta nella **issue GitHub**^{G}, il membro incaricato apre una **pull request**^{G} dal branch di feature al `branch develop`.
Ogni **pull request**^{G} deve utilizzare il template comune del gruppo, che contiene le sezioni necessarie per descrivere contesto, modifiche introdotte, issue collegate e controlli effettuati. La descrzione del templatepresente nel file `pull_request_template.md`, deve essere compilato e adattato ogni volta, così da velocizzare il lavoro di revisione e rendere uniformi tutte le richieste. Il membro deve indicare tramite il campo `Reviewers` il singolo membro incaricato del compito di verificare quella modifica.

3.2.4.3 Verifica e revisone della pull request

All'apertura della **pull request**^{G} viene richiesta automaticamente la verifica ai membri del team verificatori:

- se la verifica produce esito positivo, la **pull request**^{G} viene approvata dal verificatore incaricato;
- se sono presenti dei problemi o miglioramenti da applicare, il verificatore richiede delle modifiche, tramite l'opzione di **GitHub**^{G} `request changes`, aggiungendo un commento di quello che è il problema trovato e di una possibile soluzione se è presente. Successivamente la persona che ha aperto la **pull request**^{G} si impegna a eseguire le correzioni e a notificare il completamento dell'azione per avere una verifica il prima possibile.

3.2.4.4 Integrazione nella baseline

Quando la **pull request**^{G} viene approvata del **verificatore**^{G}, il responsabile esegue la revisione. Si impegan ad eseguire il merge nel branch `develop`. A questo punto, la modifica entra ufficialmente in **baseline**^{G} e diventa parte del progetto.

3.2.5 Registro dello stato della configurazione

Lo stato aggiornato degli elementi di configurazione è tracciato tramite:

- le issue aperte, in corso e chiuse;
- le milestone attive e completate;
- la cronologia delle pull request;
- le versioni ufficiali dei documenti pubblicate sul branch `main`.

GitHub Projects funge da registro organizzato delle attività, offrendo una rappresentazione visiva dello stato corrente.

3.2.6 Valutazione della configurazione

L'**attività di valutazione della configurazione** ha lo scopo di garantire che gli elementi software sviluppati siano completi, coerenti e conformi ai **requisiti approvati**.
Il gruppo BitByBit assicura tale controllo attraverso un tracciamento sistematico dei requisiti, che consente di collegare ogni modifica e ogni componente alla motivazione tecnica o funzionale che la giustifica. Durante lo sviluppo, i membri si impegnano inoltre a mantenere una chiara corrispondenza tra codice e requisiti, anche tramite annotazioni e riferimenti interni.
La valutazione viene svolta in modo continuativo, non solo nelle fasi conclusive, così da individuare rapidamente deviazioni dal design e garantire che:

- tutti i requisiti software siano effettivamente soddisfatti;
- non vengano introdotti comportamenti non richiesti;
- ogni parte del sistema risulti coerente con l'architettura definita.

4. Processi organizzativi del ciclo di vita

I processi organizzativi di ciclo di vita rappresentano l'insieme delle attività che il gruppo BitByBit ha scelto di adottare per strutturare e supportare in modo sistematico tutti i propri progetti.
Questi processi forniscono un quadro stabile entro cui i progetti vengono sviluppati e permettono di mantenere strumenti, procedure e competenze costantemente aggiornati. Allo stesso tempo, l'esperienza maturata nei progetti può portare all'introduzione di miglioramenti nei processi stessi, contribuendo così all'evoluzione dell'intera organizzazione.
Nelle presenti *Norme di Progetto*, i processi organizzativi sono suddivisi nelle seguenti categorie, che verranno approfondite nelle sezioni successive:

- **Gestione di processo**;
- **Infrastruttura**;
- **Processo di miglioramento**;
- **Processo di formazione**.

4.1. Gestione dei processi

Il processo di **gestione dei processi** definisce le attività organizzative necessarie per pianificare, coordinare e monitorare il lavoro del gruppo. Esso stabilisce i compiti da svolgere, i ruoli incaricati di eseguirli e le modalità di comunicazione interna ed esterna. L'obiettivo è garantire che le attività procedano in modo efficace e controllato.

4.1.1 Strumenti a supporto

Il gruppo *BitByBit* utilizza le funzionalità delle **GitHub Organizations** per organizzare il lavoro e gestire i ruoli (vedi Sezione - da aggiunggere-). Ogni team corrisponde a un ruolo specifico e include solo i membri incaricati in quel periodo, facilitando la rotazione programmata dei ruoli.
Sul branch principale (`main`) solo i membri del team dei Responsabili possono eseguire push o merge, mentre tutti gli altri membri propongono modifiche tramite **pull request**$^{\mathbf{G}}$.
La funzione `CODEOWNERS` assegna automaticamente ai Verificatori la responsabilità di revisione dei documenti, garantendo che tutte le modifiche rispettino il workflow e i controlli qualitativi previsti.
Questa configurazione assicura una chiara separazione dei ruoli, un flusso di modifica strutturato e un efficace supporto automatizzato alla gestione della configurazione.

4.1.2 Implementazione di processo

Le attività previste dal processo di gestione dei processi sono articolate come segue.

- **Inizializzazione e definizione degli ambiti**, il **responsabile**$^{\mathbf{G}}$ del progetto assieme ai membri del team identifica i requisiti per l'attività da svolgere.Si procede poi alla valutazione della fattibilità verificando disponibilità delle risorse, adeguatezza delle competenze e tempistiche.

- **Pianificazione**, si definisce il piano operativo del processo, includendo la descrizione delle attività, i prodotti software da generare e la stima dei tempi necessari. Il piano specifica inoltre risorse richieste, assegnazione delle responsabilità, analisi dei rischi, misure di qualità da adottare, costi stimati e infrastruttura di supporto.

- **Esecuzione e controllo**, le attività pianificate vengono attuate. Il **responsabile**$^{\mathbf{G}}$ di processo produce report periodici sullo stato di avanzamento, destinati sia all'uso interno sia all'acquirente. Il responsabile monitora l'aderenza al piano e interviene per risolvere eventuali scostamenti.

- **Revisione e valutazione**, perché un prodotto entri in **baseline**$^{\mathbf{G}}$, le modifiche apportate devono essere prima verificate da un **verificatore**$^{\mathbf{G}}$ (si veda Sezione ?) e successivamente revisionate dal **responsabile**$^{\mathbf{G}}$. Quest'ultimo accerta che i prodotti software e i piani risultino conformi ai requisiti stabiliti, valutando l'esito dei compiti completati e garantendo qualità e coerenza con gli obiettivi del processo.

- **Chiusura**, al completamento delle attività e dei prodotti previsti, si verifica che i criteri concordati con l'acquirente siano soddisfatti. Il **responsabile**$^{\mathbf{G}}$ controlla la completezza dei risultati e dei report.

4.1.3 Ruoli

Ruoli	Compiti e responsabilità
Responsabile	È la figura che nel momento governa il team. Identifica le attività che sono state eseguite nello spint terminato e quelle che non sono state ortate a termine. Durante l'evento di sprint planning stabilisce ed inisrisce nello Sprint Backlog le attività da eseguire nello sprint successivo, individuando i costi, i rischi e i membri del gruppo in base ai loro ruoli. Rappresenta il gruppo all'esterno, interfacciandosi con gli enti esterni quali i professori e l'azienda proponente.
Amministratore	È l'amministratore di sistema (Sys Admin), si occupa della gestione dell'infrastruttura utilizzata per eseguire le attività dei processi di ciclo di vita. Ha il compito di inserire nell'infrastruttura gli strumenti e le tecnologie per l'attuazione del Way of Working. Deve accertarsi che l'infrastruttura si trovi in uno stato stabile e qiuindi risolvere eventuali problematiche che possono derivarne. Deve redigere il presente documento.
Analisita	Il compito dell'analista è quello di identificare i requisiti del progetto, interpretando le necessità degli utenti finali in modo da ottenere una corretta definizione delle funzionalità richieste. Ha il compito di redigere il documento di analisi dei requisiti
Progettista	Il ruolo di progettista si occupa di tradurre i requisiti identificati dagli analisti in qualcosa di implementabile, e ne supervisiona l'implementazione da parte dei programmatori.
Programmatore	È la persona cui spetta il compito di tradurre le unità di design in codice funzionante. In concomitanza con il codice deve anche implementare i test automatici per la verifica del corretto funzionamento del codice.
Verificatore	Ha il compito di assicurarsi che le attività vengano svolte correttamente seguendo i processi di ciclo di vita. Si occupa di eseguire test approfonditi e revisioni sul software. Ha il compito di verificare le modifiche apportate ai vari documenti ed eventualemente esponendo le criticità che presentano, deve perciò avere un'approfondita conoscenza dei requisiti per ogni processo di vita.

4.1.4 Struttura teams GitHub

L'organizzazione interna del gruppo (si veda sezione 4.1) è stata implementata tramite la funzione nativa di **GitHub Teams**[G] di **GitHub**[G], che consente di suddividere i membri in sotto–gruppi corrispondenti ai ruoli previsti dal progetto (si veda Sezione 4.1.3). Ogni ruolo forma un team composto da uno a più membri, così da distribuire responsabilità e

carico di verifica.
Le figure che interessano le **pull request**^{G} sono principlamente due:

- **verificatori**^{G}, scelti tra i membri del team con responsabilità di verifica in quello sprint, devono controllare la correttezza formale delle modifiche;
- **responsabile**^{G}, una volta ottenuta l'approvazione dei verificatori, è incaricato di completare l'operazione di merge sul `branch main` nel rispetto delle regole di protezione del branch.

Questa struttura garantisce che ogni modifica venga validata da più persone e che il processo rispetti il processo di qualità (si veda sezione - da aggiungere-).

4.1.5 Coordinamento

Il coordinamento del gruppo è garantito attraverso riunioni periodiche, interne ed esterne, finalizzate a mantenere un allineamento continuo sullo stato dei lavori, sui problemi emersi e sulle decisioni organizzative che emergono in fase di lavoro.

4.1.5.1 Riunioni Le riunioni previste dal processo di gestione dei processi sono suddivise in due categorie principali:

- **Interne**: si svolgono tra i soli membri del gruppo e hanno luogo principalmente su **Discord**^{G}. Ogni **Sprint**^{G} ha una durata di due settimane e prevede tre incontri:
  - uno a fine **Sprint**^{G}, comprendente **Sprint Review**^{G}, **Sprint Retrospective**^{G} e **Sprint Planning**^{G}. In questa sede il **Responsabile** verifica lo stato delle attività concluse, analizza eventuali problemi riscontrati e pianifica il lavoro del successivo **Sprint**^{G};
  - due durante lo svolgimento dello **Sprint**^{G}, in cui venogono trattati i temi dell'ordine del giorno e ciascun membro espone le attività completate, quelle ancora in corso e le difficoltà incontrate.

  Per ogni riunione vengono assegnati specifici ruoli:
  - **Scriba**: i membri incaricati di prendere appunti durante gli incontri;
  - **Portavoce**: identificato nel **responsabile**^{G}, che modera e coordina la discussione;
  - **Redattore**: responsabile della stesura del verbale e del caricamento nel **repository**^{G}.
- **Esterne**: coinvolgono anche figure esterne quali i docenti o l'azienda proponente **Miriade Srl**. Queste riunioni vengono svolte generalmente tramite **Google Meet**^{G}. La frequenza concordata è di un incontro al mese, con la possibilità di richiederne ulteriori qualora fossero necessari chiarimenti o approfondimenti specifici.

L'esito di ogni incontro, interno o esterno, deve essere registrato mediante la redazione di un verbale, secondo le modalità previste dal processo di documentazione (vedi sezione -da aggiungere-).

4.2. Gestione dei processi