LINV Team

Way of working

Progetto di ingegneria del software
A.A 2022/2023

Informazioni

Versione	0.3
Data	05-11-2022
Responsabile	Nicola Ravagnan
Amministratore	Alessandro Baldissera
Verificatore	Matteo Cusin
Redattori	Alessandro Baldissera
	Mauro Carnuccio
	Alberto Casado Moreno
	Matteo Cusin
	Nicola Ravagnan
	Riccardo Rossi
	Alessandro Santin

1 Introduzione

1.1 Scopo del documento

Lo scopo del documento è quello di definire regole e "best practices" che ogni membro del gruppo si impegna a rispettare per raggiungere in modo efficiente ed efficace la realizzazione del prodotto finale. Vengono inoltre specificate delle convenzioni sull'uso dei vari strumenti scelti per lo sviluppo del prodotto.

1.2 Glossario

Questo documento, come tutti gli altri stilati durante la realizzazione del progetto, e corredato da un *Glossario* che si può trovare allegato alla documentazione, nella quale si definiscono tutti i termini specifici al progetto o di significato ambiguo. Quando un termine è definito nel glossario si trova una G a pedice del termine stesso.

Indice

2 Utilizzo dei repository

Innanzitutto il gruppo utilizzerà GIT$_G$ e GitHub$_G$, come sistema di versionamento$_G$ e ITS$_G$ (issue tracking system). Il gruppo si doterà di 3 repository$_G$, ognuno con il proprio utilizzo:

- **Documentazione privata**: che conterrà tutti i sorgenti LaTeX dei documenti, oltre alla documentazione interna al team;
- **Documentazione pubblica**: che conterrà tutti i documenti da condividere con il committente;
- **Software**: che conterrà tutto il software prodotto nelle varie fasi di sviluppo.

2.1 Uso dei branch

Nei vari repository utilizzeremo una versione di Gitflow semplificata su 3 livelli di branch$_G$:

- **main** dove verranno convogliate tutte le modifiche per le varie release;
- **devel** derivata da main, dove verranno convogliate tutte le attività sviluppate e confermate dai verificatori;
- per ogni attività un branch derivante da "devel" dedicato.

2.2 Convenzioni sui commit

Qualsiasi codice che viene caricato sul repository condiviso deve almeno compilare e passare una buona parte dei test scritti.

Il messaggio di commit$_G$ dovrà essere breve ma conciso, senza perdersi in dettagli, eventualmente da descrivere nella issue$_G$ relativa all'attività che si sta svolgendo o in altra sede consona.

3 Gestione delle attività

Le varie attività da svolgere durante il progetto verranno formalizzate come issue$_G$ sui relativi repository. Le issue devono necessariamente aver assegnato un label$_G$ appropriato e devono essere aggiunte al GitHub Project$_G$ con stato "To Do". Se necessario va assegnata anche una milestone per aggiungere una priorità di tempo.
Per evitare che più persone lavorino indipendentemente ad una stessa attività, quando un componente del team la prende in carico deve aggiungersi alla lista degli "Assignees" e spostare lo stato della issue a "In progress". Adottando il Gitflow descritto in precedenza nella sezione Uso dei banch, ogni issue ha un branch separato che verrà fatto convergere al branch principale tramite una "pull request"$_G$ che verrà prima verificata da chi di dovere. Quando una pull request è approvata, ed è stata fatta convergere al branch principale, in automatico cambierà il suo stato sul "Project" da "In progress" a "Done".

3.1 Assegnazione delle attività

Per assegnare un'attività si consulta la project board di GitHub e si sceglierà un'attività in base alla priorità e alle relative milestone. In particolare ogni attività verrà assegnata ad un membro del team tramite il campo "Assignees" e deve passare di stato da "To Do" a "In progress" nella project board.

Il responsabile di progetto si occupa di assegnare le priorità alle attività e quando necessario assegnare ai componenti del team delle specifiche attività urgenti.

4 Gestione delle scadenze

Le scadenze di progetto verranno gestite tramite le milestone di GitHub, alle quali verranno collegate le issue relative alle attività da svolgere entro la milestone prefissata.

Per convenzione interna: tutte le attività non completate entro il termine di una milestone verranno considerate come "attività in ritardo".

Inoltre per verificare quali attività sono in corso, in attesa o in ritardo, utilizzeremo i GitHub Projects, automatizzando lo "spostamento" (dove possibile) delle attività al variare dello stato della issue relativa.

5 Pubblicazione dei prodotti

Tutti i prodotti da rendere disponibili al proponente devono passare per una fase di review$_G$ che si attiva quando viene creata una pull request su Github. Quando un prodotto viene accettato dai revisionatori e deve essere rilasciato al proponente, deve essere revisionato nuovamente dal Responsabile che, una volta confermato che il prodotto è pronto per il rilascio, carica nella repository pubblica il prodotto[1]. In particolare al momento del rilascio bisognerà fare una pull request del branch *devel* verso *main* e il relativo tag deve essere fatto sul commit in *main*. L'apertura della pull request deve essere fatta da un verificatore quando ritiene che i prodotti siano pronti per il rilascio, inoltre è da assegnare come revisionatore il Responsabile di progetto. È consigliato che tutto il team si impegni a verificare che il prodotto da rilasciare sia allo stato dell'arte.

5.1 Pubblicazione dei documenti

Dopo l'esecuzione del processo visto prima, si passa alla pubblicazione nelle GitHub Pages dei documenti. Questo processo è di dovere del Responsabile, dopo che la sua revisione dei documenti ha dato esito positivo. Per eseguire la pubblicazione si eseguono le seguenti fasi:

1. Si effettua il download dei documenti compilati dalla pagina delle GitHub Actions della repository privata della documentazione, assicurandosi che l'esecuzione automatica sia avvenuta nel branch *main* e nel relativo commit nel quale si è confermato il documento;

2. Si inseriscono i file PDF nella repository pubblica, eventualmente creando la relativa cartella necessaria ad una buona organizzazione dei contenuti;

[1]Si può valutare un processo di rilascio automatico tramite Github Action facendo generare al Responsabile un git tag sul commit che contiene i prodotti da rilasciare.

3. Si aggiungono i riferimenti ai documenti nel file *README.md* nella sezione appropriata al documento in fase di pubblicazione[2], nel caso il documento sia provvisto di versione sarà necessario aggiornarla anche nel file di cui prima;

4. Effettuare il commit e push delle modifiche nella repositiry condivisa, una GitHub Action si occuperà di aggiornare in automatico il sito reso disponibile per consultare la documentazione.

A pubblicazione avvenuta, tutti i documenti saranno disponibili ed organizzati a questo link `https://linvteam.github.io/SmartLog/`.

[2]I link ai file devono iniziare tutti per **/SmartLog/** e poi rispecchiare la struttura delle cartelle della repository.

6 Processi di supporto

Nella seguente sezione sono descritti tutti i processi di supporto legati allo sviluppo del progetto, dalla documentazione alla verifica di tutte le componenti prodotte.

6.1 Documentazione

6.1.1 Scopo

Tutti i processi e le attività di sviluppo dovranno essere documentate al fine di tenere traccia e consultare in modo semplice e veloce ciò che è stato fatto. Questa sezione annota tutte le norme$_G$ per una corretta stesura e verifica di tutta la documentazione che verrà prodotta durante lo sviluppo del progetto.

6.1.2 Ciclo di vita dei documenti

- **Pianificazione**: il contenuto viene ideato e organizzato logicamente. Fase di fondamentale importanza per documenti ricchi di contenuti;
- **Impostazione**: viene creato lo scheletro del documento a partire da un template adatto. Il *Responsabile* quindi crea delle issue$_G$ su GitHub$_G$ che rappresentano le varie parti del documento;
- **Scrittura**: i membri identificati alla stesura del documento si auto organizzano, assegnandosi le issue relative ai contenuti che ognuno deve stilare;
- **Verifica**: ogni documento deve essere verificato da un componente (che non è il redattore del documento stesso) alla ricerca di errori sia grammaticali che di stile;
- **Approvazione**: terminata la verifica il documento passa al *Responsabile* che lo rilegge e decide se approvarlo o meno; se il risultato è positivo si passa alla pubblicazione nella repository pubblica, altrimenti si torna alle fasi precedenti per sistemare i problemi identificati.

6.1.3 Strumenti utilizzati

Per scrivere i documenti utilizzeremo il linguaggio LaTeXtramite i seguenti strumenti:

- **pdfTex** versione `3.141592653-2.6-1.40.24` dalla distribuzione **TexLive** scaricabile dal seguente link: `https://www.latex-project.org/get/#tex-distributions`;
- **Microsoft Visual Studio Code** come editor di testo (d'ora in poi VSC);
- **LaTeX Workshop** versione `v8.29.0` come plug-in di VSC come supporto alla scrittura del documento;
- **GitHub Actions** per la compilazione automatica dei documenti direttamente nella repository attraverso l'immagine **xu-cheng/latex-action@v2**.

6.1.4 Nomi dei file

Tutti i documenti devono essere contenuti nella propria cartella omonima, se il nome del documento è composto da più parole si utilizzano le abbreviazioni che si possono trovare nelle tabelle riguardanti i documenti e le revisioni. Il nome del file invece deve essere pari al nome del documento per esteso in **PascalCase**.

6.1.5 Utilizzo dei template

Tutti i documenti scritti con LaTeX devono inizialmente essere impostati tramite dei comandi. In particolare utilizzeremo due impostazioni per i nostri documenti:

- **Documento generico** e **Verbali**: utilizzeranno le impostazioni definite nel template `Generico.tex`;
- **Lettere**: utilizzeranno le impostazioni definite nel template `Lettere.tex`.

Per scrivere un nuovo documento basterà copiare il template, rinominare il file e procedere con la scrittura.

6.1.6 Comandi di comodo

Nel file `Common/Goodies.tex` è presente una raccolta di comandi personalizzati per la formattazione o per inserire particolari costrutti ripetuti nei documenti e mantenere separata la parte logica di supporto e formattazione dai contenuti. Ogni membro del gruppo è libero di inserire un nuovo comando in base alle sue necessità nel file indicato prima. Le regole per l'inserimento di un nuovo comando sono le seguenti:

- Tutti i nuovi comandi devono poter compilare basandosi solo sui pacchetti inclusi nel template, in caso contrario bisognerà attivarsi per correggere tale problema in modo da evitare possibili fallimenti durante la compilazione del codice;
- Ogni comando inserito deve essere preceduto da una descrizione dettagliata di cosa fa e come si utilizza, fornendo una lista esaustiva degli eventuali parametri che necessita;
- Si aggiunge il comando alla lista riassuntiva, all'inizio del file, con una breve descrizione del comportamento.

6.1.7 Prima pagina

Tutti i documenti devono iniziare con un'opportuna intestazione, posizionata in prima pagina, standardizzata nei comandi contenuti nel file `../Common/Intestazione.tex` da importare nel file principale del documento utilizzando il comando `\input`. Le diverse intestazioni sono documentate direttamente nel file `../Common/Intestazione.tex`. Attualmente le intestazioni disponibili sono:

- **Documento generico**: un'intestazione completa ed esaustiva per i principali documenti (ad es. ***Way of Working***, ***Piano di Progetto***, etc.);
- **Verbali**: un'intestazione adattata alla stesura di un verbale;
- **Lettere**: un'intestazione semplificata per la stesura di lettere (ad es. ***Lettera di Candidatura***, lettere di presentazione alle revisioni).

6.1.8 Indice

Tutti i documenti generici (ad eccezione di *Verbali* e *Lettere*) devono essere provvisti di indice. Dove necessario deve essere presente un indice delle tabelle (vedi Tabelle) e delle immagini (vedi Immagini).

6.1.9 Regole tipografiche

- I **titoli** delle sezioni iniziano con la lettera **maiuscola**;
- Gli **elementi degli elenchi** iniziano tutti con la lettera **maiuscola** e terminano con un ';' tranne per l'ultimo elemento che termina con un '.';
- Gli **acronimi** si scrivono utilizzando solo lettere **maiuscole**, tranne nei casi nel quale il nome preveda esplicitamente delle lettere minuscole (ad es. VoIP);
- I **nomi dei ruoli** si scrivono con l'iniziale **maiuscola** e in **italico** (ad es. *Responsabile*)
- I **nomi dei documenti** e delle revisioni si scrivono per esteso, in italico e grassetto, si scrivono con le iniziali maiuscole tranne per le proposizioni (ad es. ***Piano di Progetto***);
- I **nomi dei documenti** e delle **revisioni** che si ripetono spesso possono essere abbreviati seguendo la tabella presentata nella sezione Nomi dei documenti abbreviati, mantenendo sempre la regola del carattere italico e grassetto.

6.1.10 Stili di testo

Il testo nei documenti può essere enfatizzato utilizzando i seguenti stili:

- **Grassetto**: per parole ritenute importanti;
- **Italico**: per i nomi dei ruoli, nome del progetto, nomi propri;
- **Grassetto e italico**: per i nomi dei documenti;
- **Monospace**: per link esterni ai documenti e snippet di codice;
- **Sottolineato**: per riferimenti interni al documento.

6.1.11 Registro delle modifiche

Tutti i documenti generici (ad eccezione di *Verbali* e *Lettere*) devono essere provvisti di un *Registro delle modifiche* che contiene un riassunto delle modifiche apportate al documento in formato tabellare. La tabella non deve essere registrata nell'indice delle tabelle ma deve essere inserita nella sezione *Registro delle modifiche* subito dopo l'indice del documento. La tabella deve registrare le seguenti informazioni:

1. **Versione del file**;
2. **Data di rilascio**;
3. **Changelog**$_G$: un riassunto delle modifiche apportate.

6.1.12 Nomi dei documenti e revisioni abbreviati

Nei documenti nel quale ci sono molte ripetizioni nei nomi dei documenti si possono utilizzare le abbreviazioni che si trovano nelle seguenti tabelle:

Nome documento	Nome abbreviato
Piano di Progetto	*PdP*
Piano di Qualifica	*PdQ*
Way of Working	*WoW*
Glossario	*G*
Manuale Utente	*MU*

Tabella 1: Tabella riassuntiva delle abbreviazioni dei nomi dei documenti

Nome revisione	Nome abbreviato
Requirements and Technology Baseline	*RTB*
Product Baseline	*PB*
Customer Acceptance	*CA*

Tabella 2: Tabella riassuntiva delle abbreviazioni dei nomi delle revisioni

6.1.13 Tabelle

Tutte le tabelle che si inseriscono nel documento devono essere provviste di descrizione tramite il comando `\caption` e, quando nel documento si trova almeno una tabella, il documento deve essere provvisto di indice delle tabelle. Inoltre devono rispettare i seguenti stili di base:

- Le righe e colonne di intestazione devono avere un colore diverso dalle righe e colonne che contengono informazioni;
- Devono essere centrate sul foglio;
- Non devono essere di tipo 'flottante' in modo che la tabella venga inserita nel punto nel quale è stata inserita nel sorgente;
- La descrizione (*caption*) della tabella deve trovarsi sotto.

6.1.14 Immagini

Tutte le immagini che si inseriscono nel documento devono essere provviste di descrizione tramite il comando `\caption` e, quando nel documento è presente almeno un'immagine, il documento deve essere provvisto di indice delle immagini. Inoltre devono rispettare i seguenti stili di base:

- Le figure devono essere centrate nel foglio;
- Non devono essere di tipo 'flottante' in modo che l'immagine venga inserita nel punto nel quale è stata inserita nel sorgente;
- La descrizione (*caption*) dell'immagine deve trovarsi sotto.

6.1.15 Riferimenti cliccabili all'interno del documento

È possibile inserire dei riferimenti cliccabili all'interno del documento per una migliore navigabilità. Per fare ciò si utilizzano le `\label` come segnaposto e `\hyperref` per creare un link. I link devono seguire le seguenti regole:

- Devono essere sottolineati;
- Non devono avere un colore diverso dal resto del testo;
- Non devono usare font monospace in quanto riservato ai link a risorse esterne.

6.1.16 Glossario

Il ***Glossario*** è un documento che contiene tutte le definizioni dei termini per cui si ritiene che sia necessario specificarne o disambiguarne il significato. Tutti i membri del gruppo sono liberi di aggiungere termini e definizioni al glossario, imepegnandosi a controllare prima che il termine non sia già presente.

Nei documenti i termini definiti nel glossario vengono indicati con una *G* a pedice dopo la parola che si vuole indicare. Non è obbligatorio che tutte le parole presenti nel glossario siano segnalate ma almeno la prima occorrenza deve essere segnalata per ogni documento.

6.1.17 Verbali

I verbali sono un documento che segue una struttura semplificata rispetto ad altri documenti. In particolare, come descritto in Prima pagina e in Indice, i verbali sono provvisti di intestazione più breve e adatta al tipo di documento e sprovvisto di indice. Inoltre non è un documento versionato, quindi non necessita della tabella di versionamento. Le informazioni generali sono riportate in prima pagina, nell'intestazione e comprendono:

- Riunione interna o esterna;
- Luogo;
- Data;
- Ora di inizio;
- Ora di fine;
- *Responsabile*;
- *Scriba*;
- *Verificatore*;
- Partecipanti.

I contenuti del verbale sono così organizzati:

- **Punti di discussione**: riporta, tramite un elenco puntato, i punti di discussione, riflessione o domande da porre al gruppo e/o al proponente;
- **Decisioni**: riporta, tramite elenco puntato, le decisioni prese e le risposte alle domande poste;
- **Extra**: una sezione facoltativa in cui inserire eventuali note sull'incontro.

6.2 Verifica

6.2.1 Scopo

Tutto ciò che viene prodotto dal team, dalla documentazione al software, deve essere verificato e validato da una persona indipendente da colui che ha creato il prodotto. La verifica permette di non introdurre modifiche al prodotto che rischiano una recessione o uno stagnamento del prodotto. Il gruppo si doterà sia di **verifiche manuali**, svolte dai *Verificatori*, sia di verifiche automatiche quali **Unit Test**$_G$, **Integration Test**$_G$, e ogni altra verifica automatica considerata utile.

6.2.2 Quando un prodotto può essere verificato

I prodotti possono essere verificati quando la persona identificata ad apportarne modifiche ha terminato le issue relative all'attività e ritiene di aver effettuato un buon lavoro. In questo momento si può aprire una **pull request** dal branch della feature$_G$ al branch **'devel'**. Quando una nuova pull request viene creata un *Verificatore* la può prendere in carico assegnandosi come **Reviewers** in GitHub e procedere con la verifica.

6.2.3 Verifica dei documenti

La verifica dei documenti deve assicurarsi che:

- Tutte le issue create dal *Responsabile* associate al documento siano state completate;
- Il documento prodotto segua la struttura decisa durante la progettazione del documento;
- Tutti i contenuti siano consoni;
- Vengano rispettate le regole definite per la stesura dei documenti (vedi sezione Documentazione);
- L'artefatto generato dalla pipeline delle GitHub Actions esista e corrisponda al sorgente nel repository.

6.2.4 Come verificare del sorgente

Quando un'attività viene creata, si apre una pull request verso il branch **'devel'** e il *Verificatore* può iniziare la fase di verifica, composta dalle seguenti fasi:

1. Controllo che tutte le verifiche automatiche diano esito positivo, per fare ciò si apre la pull request su GitHub e, nella sezione **check**, i test non devono dare esito negativo;
2. Controllo manuale del sorgente nella sezione **Files changed** avviando una revisione, assicurandosi che i cambiamenti rispettino le regole definite e lascianodo dei commenti, evidenziando le righe di codice di riferimento, se qualcosa non è conforme o genera dei dubbi; nel caso un file non presenti problemi può essere indicato come **Viewed** nella revisione;
3. Infine, se le due fasi precendenti vengono completate senza riscontrare problemi da risolvere, la pull request può essere accettata e integrata.

6.3 Automazione

6.3.1 Scopo

L'automazione è una fase molto importante nella realizzazione efficiente ed efficace di un progetto. Permette di rendere automatico o semi-automatico un compito ripetitivo e/o tedioso, indifferentemente dalla difficoltà di tale compito. Inoltre permette di eliminare il fattore umano dal compito che potrebbe generare errori involontari.

6.3.2 Automazione con script

Lo 'scripting' è un'attività di scrittura di codice utilizzata per automatizzare un'attività. Normalmente si crea uno script$_G$ per creare uno strumento estremamente flessibile ed adattato allo use-case preso in considerazione, rispetto ad uno strumento già esistente. Il gruppo non mette dei vincoli sul linguaggio da utilizzare per creare lo script, tuttavia utilizzare linguaggi e strumenti di base permette a tutti di poter utilizzare lo script stesso da tutti senza nessuna difficoltà. In ogni caso, chi crea uno script deve tener presente che:

- Va documentato annotando il codice e aggiornando l'apposita sezione Script custom;
- Il codice deve essere inserito nel repository e quindi versionato;
- Lo script deve essere specifico per una precisa attività, quindi deve avere dimensioni ridotte.

6.3.3 Automazione con GitHub Actions

Le GitHub Actions$_G$ sono il sistema di automazione proprietario di GitHub. Esse permettono di eseguire script di shell o software forniti da uno store interno per eseguire attività di ogni genere in modo automatizzato ad ogni evento che accade nel repository. Ogni membro del gruppo può creare una nuova Action a patto che venga documentata in questo documento nella sezione dedicata GitHub Actions.

6.3.4 Script custom

Documentazione di uno script Per documentare un nuovo script si deve:

- Indicare su quale repository si trova;
- Indicare il nome dello script
- Descriverne il funzionamento e l'utilizzo, specificando ove necessario la lista dei parametri.

Pulizia dai file temporanei di LaTeX

- Repository: `linvteam/documentazione`
- Nome: `cleanup.ps1`

Descrizione e utilizzo Lo script si occupa di rimuovere tutti i file temporanei creati da LaTeX. Spesso è necessario rimuovere quei file per pulire la repository o per far generare il PDF da zero. Per rendere il processo automatico ed evitare di commettere errori causati dalla distrazione si è pensato di creare questo script che implementa delle funzioni di sicurezza, ad esempio ignorare la cartella `Common`.
Per utilizzarlo necessita dei seguenti parametri:

1. Directory target della pulizia o `--all` per far pulire tutte le cartelle della reposito- L'automazione nasce pry;
2. `--no-pdf` per ignorare l'eliminazione del file PDF generato da LaTeX.

6.3.5 GitHub Actions

Documentazione di uno script Per documentare un nuova Action si deve:

- Indicare su quale repository si trova;
- Indicare il nome della Action;
- Descriverne il funzionamento ed eventualmente come modificarla.

Build documenti Latex

- Repository: `linvteam/documentazione`
- Nome: `Build Latex document`

Descrizione e modifica Questa GitHub Action permette di compilare in automatico i documenti e caricare un archivio `.zip` nell'artifact repository. Per modificare questa Action si deve editare il file `.github/workflows/build.yml`, andando ad aggiungere il nome del file sorgente del documento (inteso come il 'main' del documento) e il nome del file PDF prodotto nei relativi punti indicati dai commenti che si trovano sul file stesso.

6.4 Accettazione

6.4.1 Scopo

Lo scopo della fase di accettazione è rendere consapevole il responsabile del prodotto che si sta rilasciano al *Committente*.

6.4.2 Quando si attiva l'Accettazione

L'accettazione si attiva nel momento in cui i verificatori ritengono che le attività svolte e integrate nel branch **devel** dei repository si trovano in uno stato sufficientemente maturo e consistente per essere presentate al *Committente*. Un altro momento in cui si deve attivare la fase di accettazione è allo scadere prossimo di una milestone, la quale prevede la pubblicazione del materiale prodotto.

6.4.3 Come attivare l'Accettazione

Per attivare l'Accettazione i *Verificatori* devono aprire una pull request su GitHub dal branch **devel** verso il branch **main**, impostando come Reviewers il *Responsabile*.

6.4.4 Regole di accettazione dei prodotti

Il *Responsabile* deve assicurarsi che:

- I prodotti, visti come black-box$_G$, siano consistenti con gli obbiettivi previsti;
- I prodotti rispettino i vincoli e i requisiti obbligatori imposti dal *Committente*.

Il *Responsabile*, una volta eseguite le verifiche indicate prima, dovrà:

- Create un tag su GitHub corrispondente al commit che contiene i prodotti da pubblicare;
- Copiare i prodotti nella repository pubblica (`linvteam\SmartLog`);
- Aggiornare il file `README.md` inserendo i riferimenti ai nuovi prodotti ed aggiornando le versioni corrispondenti ai prodotti precendenti pubblicati.

Svolte queste azioni, una GitHub Action si occuperà di pubblicare in automatico le GitHub Pages con i prodotti aggiornati. I prodotti pubblicati saranno disponibili alla link: `https://linvteam.github.io/SmartLog/`.

7 Processi organizzativi

Il seguente capitolo descrive tutte le modalità di comunicazione interne ed esterne, oltre a presentare le modalità di ricezione di comunicazioni automatiche generate dagli strumenti utilizzati.

7.1 Comunicazioni interne al gruppo

Il gruppo si impegna a mantenere delle comunicazioni attive tra i componenti in modalità sia sincrone che asicrone a seconda delle necessità. Inoltre attiveremo un sistema di notifiche automatiche riguardanti gli eventi importanti dello sviluppo del progetto.

7.1.1 Comunicazioni sincrone

Le comunicazioni sincrone avverranno nelle seguenti modalità:

- **Presenza** sia negli intervalli tra le lezioni che in incontri organizzati;
- **Discord**$_G$: che consente di effettuare chiamate, videochiamate e di condividere il contenuto dello schermo dei partecipanti, consentendo inoltre lo scambio di informazioni testuali e multimediali tramite chat.

Gli incontri devono essere pianificati, possibilmente di settimana in settiamana in modo tale che i vari componenti del gruppo si possano organizzare per essere presenti attivamente all'incontro. Nell'impossibilità di un componente di prendere parte all'incontro verrà comunque stilato un verbale che sintetizzi i punti di discussione e le relative decisioni prese nell'incontro. Tale verbale sarà reso disponibile il prima possibile nella cartella condivisa su Google Drive e successivamente verrà esportato in formato LaTeX e caricato nella repository della documentazione privata.

7.1.2 Comunicazioni asincrone

Le comunicazioni asincrone avvengono secondo le seguenti modalità:

- **Gruppo Telegram**$_G$ da utilizzare per cominicazioni che non richiedo risposta immediata e possono essere recepite in qualsiasi momento da tutti i membri del gruppo;
- **Chat privata Telegram** per comunicazioni dirette e private tra due componenti del gruppo.

Ad ogni modo, Telegram non deve essere inteso come canale di comunicazione formale.

7.1.3 Comunicazioni formali

Le comunicazioni tecniche e formali avvengono secondo le seguenti modalità:

- **GitHub Issues**$_G$ con i metodi descritti nella sezione Gestione delle attività;
- **GitHub Discussions**$_G$ nella quale si può discutere di tematiche rigorosamente tecniche, le quali possono anche essere rese disponibili al committente.

7.2 Comunicazioni esterne al gruppo

Il gruppo è intenzionato a mantenere comunicazioni attive e costanti con il proponente secondo i seguenti metodi.

7.2.1 Comunicazioni asincrone

Le comunicazioni asincrone avvengono secondo le seguenti modalità:

- **E-Mail** per le comunicazioni formali, come l'organizzazione di incontri o review.

7.2.2 Comunicazioni sincrone

Le comunicazioni sincrone avvengono secondo le seguenti modalità:

- **Microsoft Teams**$_G$ per incontri in videochiamata.

7.3 Notifiche automatiche

Per tenere sempre aggiornati tutti i membri del gruppo sugli eventi che accadono nelle repository del progetto, è attivo il sistema di notifiche automatico di GitHub che, attraverso un webhook$_G$, invierà un messaggio in un canale dedicato su Discord. Queste notifiche riguarderanno nuovi commit, eventi sulle issue, apertura di pull request e revisioni del codice, risultati sul sistema di GitHub Actions.

7.4 Svolgimento di incontri formali interni

Durante lo sviluppo del progetto saranno necessari degli incontri sincroni e formali nei quali discutere di diverse tematiche sia interne al gruppo che riguardanti lo sviluppo del progetto stesso. Per questo motivo gli incontri formali seguiranno le seguenti regole:

1. Il *Responsabile* trova la data migliore per il prossimo incontro chiedendo ai membri la disponibilità;
2. Una volta trovato il giorno e l'ora dell'incontro, il *Responsabile*, nella cartella condivisa su Google Drive$_G$ dedicata ai verbali, crea un Google Doc$_G$ a partire dal template e si compilano i vari campi dell'intestazione;
3. I membri del gruppo sono liberi di inserire i punti di discussione all'interno del documento;
4. Il *Responsabile* organizza i punti di discussione in base all'urgenza e al tempo a disposizione;
5. Il *Responsabile* dirige la discussione seguendo i punti inseriti precedentemente;
6. Lo *Scriba* eletto dal *Responsabile* trascrive per ogni punto di discussione una breve conclusione;
7. Lo *Scriba* ufficializza il verbale formattandolo in LaTeX e caricandolo nella repository della documentazione.

7.5 Svolgimento di incontri formali esterni

Durante lo sviluppo del progetto saranno necessari molti incontri con il proponente per discutere di diverse tematiche e fare il punto della situazione sullo sviluppo del prodotto. Gli incontri seguiranno le seguenti regole:

1. Il *Proponente* o il *Responsabile* richiedono di organizzare un incontro;
2. Una volta organizzato l'incontro, il *Responsabile*, nella cartella condivisa su Google Drive$_G$ dedicata ai verbali, crea un Google Doc$_G$ a partire dal template e si compilano i vari campi dell'intestazione;
3. I membri del gruppo sono liberi di inserire i punti di discussione all'interno del documento;
4. Il *Responsabile* organizza i punti di discussione in base all'urgenza e al tempo a disposizione;
5. Il *Responsabile* espone i punti di discussione al proponente, lasciando la parola ai membri del gruppo interessati dove necessario;
6. Lo *Scriba* eletto dal *Responsabile* trascrive per ogni punto di discussione una breve conclusione;
7. Lo *Scriba* ufficializza il verbale formattandolo in LaTeX e caricandolo nel repository della documentazione.