



Università di Padova



Corso di Ingegneria del Software A.A.:2022/2023

Norme di Progetto

Versione documento: V2.0.0

| | |
|---------------------|--------------------------|
| Uso | Interno |
| Destinatario | Il gruppo Committente |

Registro delle modifiche

| Versione | Data | Modifica | Persone | | | | | | |
|---------------------|----------------------|--|---|---------------------|---------------|------------------|----------------------|-----------------|------------------|
| 2.0.0 | 17 lug 2023 | <ul style="list-style-type: none"> Approvazione del documento | Approvazione Romano Davide | | | | | | |
| 1.1.0 | 17 lug 2023 | <ul style="list-style-type: none"> Revisione del documento | Verifica Peron Samuel | | | | | | |
| 1.0.5 | 17 lug 2023 | <ul style="list-style-type: none"> Rimosso registro modifiche dall'indice | <table border="1"> <tr> <td>Approvazione</td> <td>Romano Davide</td> </tr> <tr> <td>Redazione</td> <td>Zarantonello Giorgio</td> </tr> <tr> <td>Verifica</td> <td>Peron Samuel</td> </tr> </table> | Approvazione | Romano Davide | Redazione | Zarantonello Giorgio | Verifica | Peron Samuel |
| Approvazione | Romano Davide | | | | | | | | |
| Redazione | Zarantonello Giorgio | | | | | | | | |
| Verifica | Peron Samuel | | | | | | | | |
| 1.0.4 | 16 lug 2023 | <ul style="list-style-type: none"> Rimossa sezione duplicata | <table border="1"> <tr> <td>Approvazione</td> <td>Romano Davide</td> </tr> <tr> <td>Redazione</td> <td>Zarantonello Giorgio</td> </tr> <tr> <td>Verifica</td> <td>Peron Samuel</td> </tr> </table> | Approvazione | Romano Davide | Redazione | Zarantonello Giorgio | Verifica | Peron Samuel |
| Approvazione | Romano Davide | | | | | | | | |
| Redazione | Zarantonello Giorgio | | | | | | | | |
| Verifica | Peron Samuel | | | | | | | | |
| 1.0.3 | 15 lug 2023 | <ul style="list-style-type: none"> Aggiunta sezione documentazione WebApp | <table border="1"> <tr> <td>Approvazione</td> <td>Romano Davide</td> </tr> <tr> <td>Redazione</td> <td>Zarantonello Giorgio</td> </tr> <tr> <td>Verifica</td> <td>Peron Samuel</td> </tr> </table> | Approvazione | Romano Davide | Redazione | Zarantonello Giorgio | Verifica | Peron Samuel |
| Approvazione | Romano Davide | | | | | | | | |
| Redazione | Zarantonello Giorgio | | | | | | | | |
| Verifica | Peron Samuel | | | | | | | | |
| 1.0.2 | 4 mag 2023 | <ul style="list-style-type: none"> Aggiunta sezione Workflows | <table border="1"> <tr> <td>Approvazione</td> <td>Romano Davide</td> </tr> <tr> <td>Redazione</td> <td>Pierobon Luca</td> </tr> <tr> <td>Verifica</td> <td>Peron Samuel</td> </tr> </table> | Approvazione | Romano Davide | Redazione | Pierobon Luca | Verifica | Peron Samuel |
| Approvazione | Romano Davide | | | | | | | | |
| Redazione | Pierobon Luca | | | | | | | | |
| Verifica | Peron Samuel | | | | | | | | |
| 1.0.1 | 2 apr 2023 | <ul style="list-style-type: none"> Aggiunta specifica registro modifiche Rimosso glossario dalle sezioni dei documenti | <table border="1"> <tr> <td>Approvazione</td> <td>Pierobon Luca</td> </tr> <tr> <td>Redazione</td> <td>Bonavigo Michele</td> </tr> <tr> <td>Verifica</td> <td>Casarotto Mattia</td> </tr> </table> | Approvazione | Pierobon Luca | Redazione | Bonavigo Michele | Verifica | Casarotto Mattia |
| Approvazione | Pierobon Luca | | | | | | | | |
| Redazione | Bonavigo Michele | | | | | | | | |
| Verifica | Casarotto Mattia | | | | | | | | |
| 1.0.0 | 1 apr 2023 | <ul style="list-style-type: none"> Approvato documento versione 1.0.0 | Approvazione Pierobon Luca | | | | | | |
| 0.1.0 | 1 apr 2023 | <ul style="list-style-type: none"> Revisione del documento | Verifica Casarotto Mattia | | | | | | |

| | | | | | | | | | |
|---------------------|---|--|---|---------------------|-----------------------|------------------|---|-----------------|--------------------------------------|
| 0.0.3 | 28 mar 2023 | <ul style="list-style-type: none"> • Aggiunte appendici sugli standard | <table border="1"> <tr> <td>Approvazione</td> <td>Pierobon Luca</td> </tr> <tr> <td>Redazione</td> <td>Massarenti Alessandro</td> </tr> <tr> <td>Verifica</td> <td>Casarotto Mattia Bonavigo Michele</td> </tr> </table> | Approvazione | Pierobon Luca | Redazione | Massarenti Alessandro | Verifica | Casarotto Mattia Bonavigo Michele |
| Approvazione | Pierobon Luca | | | | | | | | |
| Redazione | Massarenti Alessandro | | | | | | | | |
| Verifica | Casarotto Mattia Bonavigo Michele | | | | | | | | |
| 0.0.2 | 2 mar 2023 | <ul style="list-style-type: none"> • Aggiunto capitolo relativo ai processi organizzativi | <table border="1"> <tr> <td>Approvazione</td> <td>Massarenti Alessandro</td> </tr> <tr> <td>Redazione</td> <td>Zarantonello Giorgio</td> </tr> <tr> <td>Verifica</td> <td>Bonavigo Michele</td> </tr> </table> | Approvazione | Massarenti Alessandro | Redazione | Zarantonello Giorgio | Verifica | Bonavigo Michele |
| Approvazione | Massarenti Alessandro | | | | | | | | |
| Redazione | Zarantonello Giorgio | | | | | | | | |
| Verifica | Bonavigo Michele | | | | | | | | |
| 0.0.1 | 20 gen 2023 | <ul style="list-style-type: none"> • Inizializzato documento tex • Aggiunto capitolo relativo ai processi di primari • Aggiunto capitolo relativo ai processi di supporto | <table border="1"> <tr> <td>Approvazione</td> <td>Bonavigo Michele</td> </tr> <tr> <td>Redazione</td> <td>Bonavigo Michele Massarenti Alessandro</td> </tr> <tr> <td>Verifica</td> <td>Pierobon Luca</td> </tr> </table> | Approvazione | Bonavigo Michele | Redazione | Bonavigo Michele Massarenti Alessandro | Verifica | Pierobon Luca |
| Approvazione | Bonavigo Michele | | | | | | | | |
| Redazione | Bonavigo Michele Massarenti Alessandro | | | | | | | | |
| Verifica | Pierobon Luca | | | | | | | | |

Indice

| | | |
|----------|---|----------|
| 1 | Introduzione | 1 |
| 1.1 | Scopo del documento | 1 |
| 1.2 | Scopo del prodotto | 1 |
| 1.3 | Maturità del documento | 1 |
| 1.4 | Riferimenti e richiami | 1 |
| 1.4.1 | Riferimenti normativi | 1 |
| 1.4.2 | Riferimenti informativi | 1 |
| 2 | Processi primari | 3 |
| 2.1 | Fornitura | 3 |
| 2.1.1 | Scopo | 3 |
| 2.1.2 | Aspettative | 3 |
| 2.1.3 | Attività | 3 |
| 2.2 | Sviluppo | 4 |
| 2.2.1 | Scopo | 4 |
| 2.2.2 | Aspettative | 4 |
| 2.2.3 | Attività | 4 |
| 2.2.4 | Analisi dei requisiti | 4 |
| 2.2.5 | Progettazione | 5 |
| 2.2.6 | Codifica | 6 |
| 3 | Processi di supporto | 7 |
| 3.1 | Documentazione | 7 |
| 3.1.1 | Scopo | 7 |
| 3.1.2 | Ciclo di vita dei documenti | 7 |
| 3.1.3 | Classificazione dei documenti | 8 |
| 3.1.4 | Nome dei file codificato | 8 |
| 3.1.5 | Sigle e abbreviazioni | 9 |
| 3.1.6 | Struttura | 9 |

| | | |
|----------|--|-----------|
| 3.1.7 | Stili testuali | 11 |
| 3.1.8 | Elementi grafici | 11 |
| 3.1.9 | Strumenti di redazione e tecnologici | 11 |
| 3.1.10 | Workflows | 11 |
| 3.2 | Gestione di configurazione | 14 |
| 3.2.1 | VCS | 14 |
| 3.2.2 | ITS | 15 |
| 3.2.3 | Strutturazione dei repository | 16 |
| 3.3 | Gestione della qualità | 16 |
| 3.3.1 | Scopo | 16 |
| 3.4 | Gestione della verifica | 17 |
| 3.4.1 | Scopo | 17 |
| 3.4.2 | Premesse fondamentali | 17 |
| 3.4.3 | Metodologie e ciclo di vita | 17 |
| 3.4.4 | Testing | 17 |
| 3.4.5 | Strumenti | 18 |
| 3.5 | Validazione | 18 |
| 3.5.1 | Scopo | 18 |
| 4 | Processi organizzativi | 19 |
| 4.1 | Gestione dell'organizzazione | 19 |
| 4.1.1 | Scopo | 19 |
| 4.1.2 | Aspettative | 19 |
| 4.1.3 | Descrizione | 20 |
| 4.2 | Ruoli | 20 |
| 4.2.1 | Responsabile di progetto | 20 |
| 4.2.2 | Amministratore di progetto | 20 |
| 4.2.3 | Analista | 21 |
| 4.2.4 | Progettista | 21 |
| 4.2.5 | Programmatore | 21 |
| 4.2.6 | Verificatore | 21 |
| 4.3 | Formazione | 21 |
| 4.3.1 | Materiale utilizzato | 22 |
| 4.4 | Procedure | 22 |
| 4.4.1 | Gestione delle comunicazioni | 22 |
| 4.4.2 | Gestione degli incontri | 22 |

| | | |
|----------|--|-----------|
| 4.5 | Gestione dei rischi | 23 |
| 4.5.1 | Codifica dei rischi | 23 |
| 4.6 | Strumenti | 23 |
| A | Standard ISO/IEC 12207:1997 | 24 |
| A.1 | Processi primari | 24 |
| A.1.1 | Acquisizione | 24 |
| A.1.2 | Fornitura | 24 |
| A.1.3 | Sviluppo | 25 |
| A.1.4 | Operation | 25 |
| A.1.5 | Manutenzione | 26 |
| A.2 | Processi di supporto | 26 |
| A.2.1 | Documentazione | 26 |
| A.2.2 | Gestione della configurazione | 26 |
| A.2.3 | Gestione e controllo della qualità | 27 |
| A.2.4 | Verifica | 27 |
| A.2.5 | Validazione | 27 |
| A.2.6 | Revisione della congiunta | 27 |
| A.2.7 | Audit | 27 |
| A.2.8 | Risoluzione dei problemi | 28 |
| A.3 | Processi organizzativi | 28 |
| A.3.1 | Infrastrutture | 28 |
| A.3.2 | Miglioramento | 28 |
| A.3.3 | Formazione | 29 |
| B | Standard di qualità ISO/IEC 9126 | 30 |
| B.1 | Modello di qualità | 30 |
| B.1.1 | Qualità esterna ed interna | 30 |
| B.1.2 | Funzionalità | 30 |
| B.1.3 | Affidabilità | 30 |
| B.1.4 | Usabilità | 31 |
| B.1.5 | Efficienza | 31 |
| B.1.6 | Manutenibilità | 31 |
| B.1.7 | Portabilità | 32 |
| B.1.8 | Qualità in uso | 32 |
| B.2 | Metriche per la qualità esterna | 32 |
| B.3 | Metriche per la qualità interna | 32 |

B.4 Metriche per la qualità in uso 32

Capitolo 1

Introduzione

1.1 Scopo del documento

Questo documento ha lo scopo di definire il *way of working* che ogni membro del gruppo *SWEasabi* si impegna a seguire durante lo svolgimento del progetto *Lumos Minima*, per garantirne la massima omogeneità e coesione.

1.2 Scopo del prodotto

Il risparmio delle risorse del Pianeta e in particolare delle fonti energetiche è entrato con forza nell'agenda politica dell'Unione Europea. Fra gli impatti più evidenti, spicca la crescita esponenziale del prezzo del gas, risorsa ancora largamente utilizzata come materia prima per la produzione di energia elettrica. Per far fronte al rincaro delle bollette energetiche, molti Comuni italiani stanno annunciando il taglio dell'illuminazione pubblica, che necessita di una quantità considerevole di energia elettrica.

Il capitolato C2, *Lumos Minima*, pone come obiettivo lo sviluppo di un sistema per l'ottimizzazione dell'illuminazione pubblica che permetta ai gestori di sfruttare la possibilità di regolare l'intensità della luce emessa dagli impianti di illuminazione.

1.3 Maturità del documento

Il presente documento è redatto con un approccio incrementale in modo tale da trattare modifiche o aggiunte in modo efficiente. Non può pertanto essere considerato definitivo nella sua attuale versione.

1.4 Riferimenti e richiami

1.4.1 Riferimenti normativi

- Capitolato d'appalto C2 - Lumos Minima:
<https://www.math.unipd.it/~tullio/IS-1/2022/Progetto/C2.pdf>

1.4.2 Riferimenti informativi

- Slide T03 del corso di Ingegneria del Software:
<https://www.math.unipd.it/~tullio/IS-1/2022/Dispense/T03.pdf>
- ISO/IEC 12207:1997:
https://www.math.unipd.it/~tullio/IS-1/2009/Approfondimenti/ISO_12207-1995.pdf

- Slide T07 del corso di Ingegneria del Software:
<https://www.math.unipd.it/~tullio/IS-1/2022/Dispense/T07.pdf>
- ISO/IEC 9126:
http://www.colonese.it/00-Manuali_Pubblicatii/07-ISO-IEC9126_v2.pdf

Capitolo 2

Processi primari

2.1 Fornitura

2.1.1 Scopo

Il processo di fornitura ha lo scopo di consegnare al richiedente un prodotto che soddisfi le richieste concordate con il fornitore. Queste richieste verranno analizzate in uno studio di fattibilità, che servirà per definire costi e tempi del progetto. Concordato ciò il proponente e il fornitore stipuleranno un contratto per la consegna del prodotto.

2.1.2 Aspettative

Durante l'intero processo di fornitura, il gruppo intende:

- Mantenere una struttura chiara dei documenti;
- Avere un costante dialogo con il proponente;
- Stabilire vincoli e tempistiche per la realizzazione del prodotto;
- Effettuare una continua verifica del lavoro.

2.1.3 Attività

Il processo di fornitura si compone delle seguenti attività:

- Inizializzazione;
- Contrattazione delle richieste;
- Pianificazione;
- Esecuzione e controllo;
- Revisione e valutazione;
- Rilascio.

2.2 Sviluppo

2.2.1 Scopo

Il processo di sviluppo ha lo scopo di descrivere i compiti e le attività che il gruppo dovrà svolgere al fine di sviluppare il software richiesto dal proponente.

2.2.2 Aspettative

Le aspettative sono le seguenti:

- fissare vincoli tecnologici;
- fissare vincoli di design;
- fissare gli obiettivi di sviluppo;
- realizzare un prodotto finale che sia conforme alle richieste del proponente.

2.2.3 Attività

L'attività del processo di sviluppo sono:

- analisi dei requisiti;
- progettazione;
- codifica.

2.2.4 Analisi dei requisiti

Durante questa attività gli analisti dovranno individuare i requisiti del capitolato. Essi saranno raccolti grazie ad una prima analisi e ad una successiva riunione con il proponente.

Tutto ciò sarà contenuto in un documento, che esporrà:

- descrizione del prodotto;
- casi d'uso;
- requisiti.

2.2.4.1 Casi d'uso

Ogni caso d'uso è strutturato come segue:

- codice identificativo;
- titolo;
- precondizioni;
- postcondizioni;
- scenario principale;
- estensioni;
- generalizzazioni.

Codice identificativo Il codice identificativo dei casi d'uso ha la seguente struttura:

$$UC[X].[Y] - [T]$$

Con:

- **X**: numero del caso d'uso;
- **Y**: numero del sottocaso del caso d'uso principale;
- **T**: titolo del caso d'uso.

2.2.4.2 Requisiti

Ogni requisito è strutturato come segue:

- codice identificativo;
- classificazione;
- descrizione;
- fonti.

Codice identificativo Il codice identificativo dei requisiti ha la seguente struttura:

$$R[X]_[Y]$$

Con:

- **X**: Tipologia: può assumere i seguenti valori:
 - **F**: requisito funzionale;
 - **Q**: requisito di qualità;
 - **V**: requisito di vincolo;
- **Y**: numero del requisito per classificazione.

Classificazione La classificazione del requisito può assumere i seguenti valori:

- obbligatorio;
- desiderabile;
- opzionale.

2.2.5 Progettazione

Durante questa attività bisogna trovare una soluzione al capitolato proposto, partendo dall'analisi dei requisiti. Infatti l'analisi dei requisiti divide il problema in requisiti, mentre la progettazione unisce le varie parti definendo le funzionalità, racchiudendo il tutto in un'unica soluzione.

Essa è formata da due parti:

- **Technology Baseline**: illustra e motiva le tecnologie scelte per il prodotto;
- **Product Baseline**: illustra la baseline architeturale del prodotto.

Technology Baseline essa dovrà contenere:

- Diagrammi UML delle classi e di attività;
- Tecnologie adottate;
- Design pattern utilizzati;
- PoC (Proof of Concept)

Product Baseline essa dovrà contenere:

- Definizione delle classi;
- Test di unità su ogni componente, per verificarne il corretto funzionamento.

2.2.6 Codifica

Durante questa attività bisogna realizzare il software vero e proprio, partendo dalla progettazione. Il codice scritto dovrà rispettare delle norme per essere facilmente leggibile e in futuro manutenibile.

2.2.6.1 Stile della codifica

- **Indentazione:** i blocchi di codice dovranno avere una indentazione di quattro spazi;
- **Parentesi:** la parentesi aperta andrà inserita nella stessa riga di quella del costrutto, la parentesi chiusa andrà inserita nella riga successiva all'ultima istruzione del blocco;
- **Metodi:** il nome dei metodi seguirà la codifica *camelCase*;
- **Classi:** il nome delle classi seguirà la codifica *PascalCase*;
- **Variabili:** il nome delle variabili seguirà la codifica *camelCase*;
- **Costanti:** il nome delle costanti sarà completamente in maiuscolo;
- **Commenti:** i commenti saranno inseriti prima dell'inizio del costrutto e dovranno essere coincisi ed esplicativi;
- **File:** il nome dei file seguirà la codifica *kebab-case*;
- **Lingua:** i commenti dovranno essere scritti in italiano, fatta eccezione per alcuni inglesismi, mentre le variabili, i metodi, le classi ed i nomi dei file potranno essere scritti sia in italiano che in inglese;

Capitolo 3

Processi di supporto

3.1 Documentazione

3.1.1 Scopo

Scopo del processo di documentazione è la raccolta di tutto quanto sia utile per rendere consistente, coeso e ripetibile il processo di produzione.

3.1.2 Ciclo di vita dei documenti

Ogni documento segue le fasi seguenti nel suo ciclo di vita. Inoltre queste fasi si ripeteranno per tutto il tempo in cui anche il prodotto software sarà vivo.

Questo ciclo di vita sarà incrementale e tracciato dalle versioni.

Prodotto software e documentazione coesistono per consentire il raggiungimento dello scopo di questo processo.

Approvazione delle modifiche A risultato delle riunioni interne il responsabile approva le modifiche e quindi anche le aggiunte che dovranno essere eseguite al documento in oggetto.

Redazione Un documento viene redatto da uno o più redattori. Il documento durante la redazione verrà incrementato di tutto quanto deciso durante le riunioni. Il documento viene considerato redatto quando se ne è completata la stesura.

Verifica Quando un documento è stato redatto viene verificato dalla figura del *verificatore*, questo ne controllerà consistenza, validità e ortografia.

Approvazione e Rilascio Quando il documento sarà stato verificato il responsabile approverà e rilascerà il documento seguendo le regole di versionamento definite.

3.1.2.1 Ciclo di vita dei verbali

I verbali seguono lo stesso ciclo di vita dei documenti, ma non avranno fasi di ripetizione, in quanto verranno redatti, verificati e rilasciati solamente una volta. Un verbale non sarà infatti un documento modificabile.

3.1.2.2 Ciclo di vita della documentazione sviluppatori microservizio WebApp

La documentazione per gli sviluppatori riguardante il microservizio della WebApp segue lo stesso ciclo di vita dei documenti, senza però fasi di ripetizione, in quanto verrà redatta, verificata e rilasciata una volta sola dopo che la WebApp sarà stata terminata e approvata.

3.1.3 Classificazione dei documenti

I documenti vengono raggruppati in due classi principali definite dallo scopo delle stesse.

3.1.3.1 Documenti ad uso interno

I documenti ad uso interno sono destinati solo all'uso da parte dei membri del gruppo

- Verbali interni;
- Norme di progetto.

3.1.3.2 Documenti ad uso esterno

- Verbali esterni;
- Analisi dei requisiti;
- Piano di progetto;
- Piano di qualifica;
- Documentazione per lo sviluppatore microservizio WebApp.

3.1.4 Nome dei file codificato

Tutti i documenti seguono le regole generali nella definizione del nome del file. Su queste regole possono esserci poi specifiche espansioni o eccezioni.

3.1.4.1 Regole generali

Notazione I nomi dei file seguono la notazione *Kebab case*.

[titolo-del-documento]

Per il VCS Per consentire il funzionamento delle pipeline di verifica¹ dei documenti, nel sistema di *VCS* i documenti in *tex* vanno nominati come segue.

d_[titolo-del-documento]

¹Vedasi sezione 3.4

3.1.4.2 Regole specifiche per i verbali

I nomi file dei verbali vengono codificati secondo la seguente struttura:

$$[TTT]_[AAMMGG]_[N]$$

Con:

- **TTT**: tipologia di verbale come da tabella 3.1 ;
- **AAMMGG**: data di svolgimento dell'incontro a cui fa riferimento il verbale con ordinazione anno mese giorno;
- **N**: numero del verbale con partenza da 1 da inserire nel caso siano stati fatti più incontri nella medesima giornata.

| Tipologia verbale | Prefisso |
|-------------------|----------|
| Verbale interno | VIN |
| Verbali esterni | VEX |
| Verbali di sprint | SPR |

Tabella 3.1: Tipologie verbali

3.1.5 Sigle e abbreviazioni

All'interno dei documenti per brevità e semplicità di lettura verranno spesso utilizzate sigle e abbreviazioni, di seguito quelle con portata generale.

| Sigla | Definizione |
|------------|-----------------------|
| AdR | Analisi dei requisiti |
| NdP | Norme di progetto |
| PdP | Piano di progetto |
| PdQ | Piano di qualifica |
| MU | Manuale Utente |
| MM | Manuale Manutentore |

3.1.6 Struttura

Ogni documento ufficiale prodotto deve seguire una struttura precisa e definita per garantire consistenza comunicativa. Su queste regole possono esserci poi specifiche espansioni o eccezioni.

3.1.6.1 Regole generali

La struttura per i documenti generali è la seguente:

| Sezione | Scopo |
|--------------------------|--|
| Frontespizio/copertina | Definisce per il lettore le informazioni base sul documento e su chi lo ha redatto |
| Registro delle modifiche | Sono presenti tutti i cambiamenti al documento |
| Tavola dei contenuti | Contiene l'indice dei contenuti presenti nel documento |
| Corpo del documento | Sono presenti tutte le informazioni comunicative del documento |

Frontespizio/copertina Nella copertina devono essere presenti:

- il logo dell'Università di Padova in alto a sinistra;
- il logo del gruppo in alto a destra;
- nome del corso, anno accademico, titolo del documento e versione al centro;
- uso e destinatario, in forma tabellare, al centro.

Una versione già strutturata del frontespizio è presente nel repository *global-assets*²

Registro delle modifiche Nella registro delle modifiche saranno presenti, in forma tabellare

- versione;
- data;
- modifica;
- persone, in formato *Cognome Nome*.

Tavola dei contenuti La tavola dei contenuti viene generata automaticamente utilizzando L^AT_EX.

Glossario Nel glossario sono presenti tutti i termini specifici attinenti al documento. Tipograficamente sono definiti utilizzando dei paragrafi.

Corpo del documento In questa sezione viene redatto utilizzando i capitoli il contenuto del documento.

3.1.6.2 Regole specifiche per i verbali

La struttura per i verbali è la seguente:

| Sezione | Scopo |
|-------------------------|--|
| Frontespizio/copertina | Definisce per il lettore le informazioni base sul documento e su chi lo ha redatto |
| Adempimenti burocratici | Definisce le informazioni generali del verbale |
| Corpo del documento | Sono presenti tutte le informazioni comunicative del documento |

Adempimenti burocratici ³ In questa sezione sono scritte in elenco puntato:

- luogo e data dell'incontro:
 - luogo;
 - data;
- presenze.

Nel corpo del documento devono essere presenti:

- ordine del giorno;
- svolgimento;
- risultati.

²<https://github.com/SWEasabi/global-assets>

³In alcuni verbali legacy viene descritto come **Informazioni generali**

3.1.7 Stili testuali

3.1.8 Elementi grafici

3.1.9 Strumenti di redazione e tecnologici

L^AT_EX Utilizzato per stendere i documenti principali poiché consente di utilizzare metodologie strutturate di versionamento⁴ e di verifica dei contenuti redatti.

GitHub Actions Per controllare la consistenza del documento è presente un'azione che controlla la compilazione del documento in L^AT_EX.

3.1.10 Workflows

3.1.10.1 Aggiunta a backlog

Ogni issue e pull request aperte in submodules che fanno uso di questo workflow vengono aggiunte al GitHub Project dell'organizzazione.

```

1 name: Add issue/PR to backlog project
2
3 on:
4   issues:
5     types:
6       - opened
7   pull_request:
8     types:
9       - opened
10
11 jobs:
12   add-to-project:
13     name: Add issue/PR to project
14     runs-on: ubuntu-latest
15     steps:
16       - uses: actions/add-to-project@v0.4.0
17         with:
18           project-url: https://github.com/orgs/SWEasabi/projects/3/
19           github-token: ${ secrets.AUTOBACKLOG }
```

3.1.10.2 Compilazione dei documenti, GitHub Page e Release

Impostazione variabili d'ambiente Per semplificare l'installazione del workflow nei diversi submodules dedicati alla documentazione è stato pensato un template che richiede modifiche solo nella sezione dedicata alla dichiarazione delle variabili d'ambiente. Queste variabili determinano come si chiameranno gli artifacts che produrrà il workflow (REPO_NAME) e il titolo che verrà mostrato nella GitHub Page (DISPLAY_NAME).

```

1 on: [push, pull_request]
2
3 env:
4   REPO_NAME: REPO_NAME
5   DISPLAY_NAME: DISPLAY_NAME
```

Esempio per il documento corrente

```

1 on: [push, pull_request]
2
3 env:
4   REPO_NAME: norme-di-progetto
```

⁴Vedasi sottosezione 3.2.1 per ulteriori informazioni.

5 `DISPLAY_NAME: Norme di progetto`

Compilazione dei documenti La compilazione avviene mediante `latexmk -lualatex` di tutti i file `.tex` che presentano nel nome il prefisso "d_". Prima della compilazione il workflow si occupa anche di prelevare tutti i documenti presenti nei sottomoduli. Infine, dopo la compilazione, genera un artifact contenente esclusivamente i file in formato `.pdf` che saranno utilizzati nei prossimi job.

```

1 jobs:
2   build_docs:
3     runs-on: ubuntu-latest
4     name: Build Documents
5     steps:
6       - name: Set up Git repository
7         uses: actions/checkout@v3
8         with:
9           submodules: recursive
10
11      - name: Build LaTeX files
12        run: |
13          docker run -i --rm -v "$(pwd)/:/work" -w /work ghcr.io/xu-cheng/texlive-full bash -c '
14            shopt -s globstar
15            set -e
16            for d_path in ./**/d_*.tex; do (
17              cd "$(dirname $d_path)"
18              latexmk -lualatex "$(basename $d_path)"
19            ) done'
20
21      - name: Generate artifact containing PDF files
22        uses: actions/upload-artifact@v3
23        with:
24          name: ${ env.REPO_NAME }
25          path: "**/d_*.pdf"

```

Creazione GitHub Page Questa sezione si occupa di generare le pagine HTML che GitHub interpreterà e renderà disponibili. Il workflow fa uso della Action "apindex" per generare le pagine a partire da un template messo a disposizione nel sottomodulo `global-assets`.

```

1
2 build_ghpages:
3   runs-on: ubuntu-latest
4   name: Build GitHub Pages
5   needs: build_docs
6   steps:
7     - name: Set up Git repository
8       uses: actions/checkout@v3
9       with:
10        token: ${ secrets.GITHUB_TOKEN }
11        submodules: recursive
12
13     - name: Download artifact ${ env.REPO_NAME }
14       uses: actions/download-artifact@v3.0.1
15       with:
16        name: ${ env.REPO_NAME }
17        path: ./${ env.REPO_NAME }
18
19     - name: Remove prefix d_ in PDF files
20       run: find ./${ env.REPO_NAME } -type f -name 'd_*.pdf' | while read f; do mv "$f" "${f/d_/}"; done
21
22     - name: Install apindex and update HTML template
23       run: |
24         git clone --depth=1 https://github.com/paperbenni/apindex.git
25         cd apindex
26         rm share/apindex/index.template.html
27         curl -o share/apindex/index.template.html https://raw.githubusercontent.com/SWEasabi/global-assets/main/html/ghpages_template.html

```

```

28     sed -i "34d; 35i ${ env.DISPLAY_NAME }}" share/apindex/index.template.html
29     cmake . -DCMAKE_INSTALL_PREFIX=/usr
30     sudo make install
31     cd ..
32     rm -rf apindex
33
34 - name: Build static page
35   run: apindex ./${ env.REPO_NAME }}
36
37 - name: Generate artifact containing Directory Listings Page
38   uses: actions/upload-artifact@v3
39   with:
40     name: ${ env.REPO_NAME }}_html
41     path: ./${ env.REPO_NAME }}

```

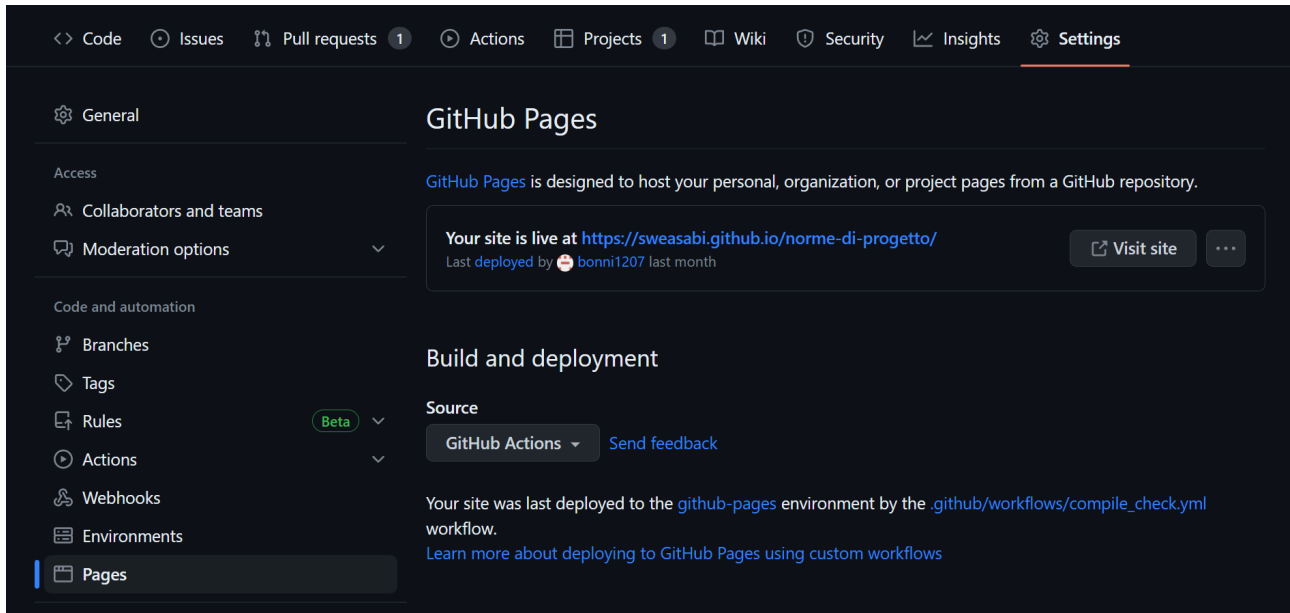
Distribuzione GitHub Page L'Action di GitHub dedicata al deploy consente la costruzione del sito web a partire dalle pagine HTML presenti nell'artifact generato nel job precedente. L'aggiornamento del sito web avviene esclusivamente al rilevamento di un tag (ad esempio il push della v1.0.0).

```

1
2 deploy_ghpages:
3   runs-on: ubuntu-latest
4   if: github.ref == 'refs/heads/main'
5   name: Deploy GitHub Pages
6   needs: [build_docs, build_ghpages]
7   environment:
8     name: github-pages
9     url: ${ steps.deployment.outputs.page_url }}
10  permissions:
11    pages: write
12    id-token: write
13  steps:
14    - name: Download artifact ${ env.REPO_NAME }}_html
15      uses: actions/download-artifact@v3
16      with:
17        name: ${ env.REPO_NAME }}_html
18        path: ./html
19
20    - name: Setup Github Pages
21      uses: actions/configure-pages@v2
22
23    - name: Generate github-pages artifact
24      uses: actions/upload-pages-artifact@v1
25      with:
26        path: ./html
27
28    - name: Deploy artifact to GitHub Pages
29      uses: actions/deploy-pages@v1

```

Nota Per consentire a GitHub di rendere disponibile il sito web è necessario modificare un'impostazione del repository. Verificare, quindi, che in **Settings**→**Pages**→**Build and deployment** sia selezionata "GitHub Actions" come Source.



The screenshot shows the GitHub Pages settings page for a repository. The left sidebar contains navigation options: General, Access, Collaborators and teams, Moderation options, Code and automation, Branches, Tags, Rules (Beta), Actions, Webhooks, Environments, and Pages. The main content area is titled 'GitHub Pages' and includes a description: 'GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository.' Below this, it states 'Your site is live at https://sweasabi.github.io/norme-di-progetto/' and shows it was last deployed by 'bonni1207' last month. There is a 'Visit site' button. The 'Build and deployment' section shows 'Source' set to 'GitHub Actions' with a 'Send feedback' link. It also mentions the site was last deployed to the 'github-pages' environment by the '.github/workflows/compile_check.yml' workflow, with a link to 'Learn more about deploying to GitHub Pages using custom workflows'.

Rilascio versione Nell'ultima sezione i file in formato `.pdf` e privi di prefisso `"d_"` vengono raccolti in un archivio `.zip` che viene allegato alla Release del repository. Il rilascio dell'archivio avviene esclusivamente al rilevamento di un tag (ad esempio il push della `v1.0.0`).

```

1
2  release:
3    runs-on: ubuntu-latest
4    if: startsWith(github.ref, 'refs/tags/')
5    name: Release
6    needs: build_docs
7    steps:
8      - name: Download artifact ${ env.REPO_NAME }
9        uses: actions/download-artifact@v3.0.1
10       with:
11         name: ${ env.REPO_NAME }
12         path: ./${ env.REPO_NAME }
13
14      - name: Remove prefix d_ in PDF files
15        run: find ./${ env.REPO_NAME } -type f -name 'd_*.pdf' | while read f; do mv "$f" "${f}/d_"/"; done
16
17      - name: Generate ZIP archive
18        run: zip -r ${ env.REPO_NAME }.zip ${ env.REPO_NAME }
19
20      - name: Release
21        uses: softprops/action-gh-release@v1
22        with:
23          files: ${ env.REPO_NAME }.zip

```

3.2 Gestione di configurazione

Per la configurazione del prodotto e per la gestione documentale vengono utilizzate varie strategie per organizzare e rendere consistente.

3.2.1 VCS

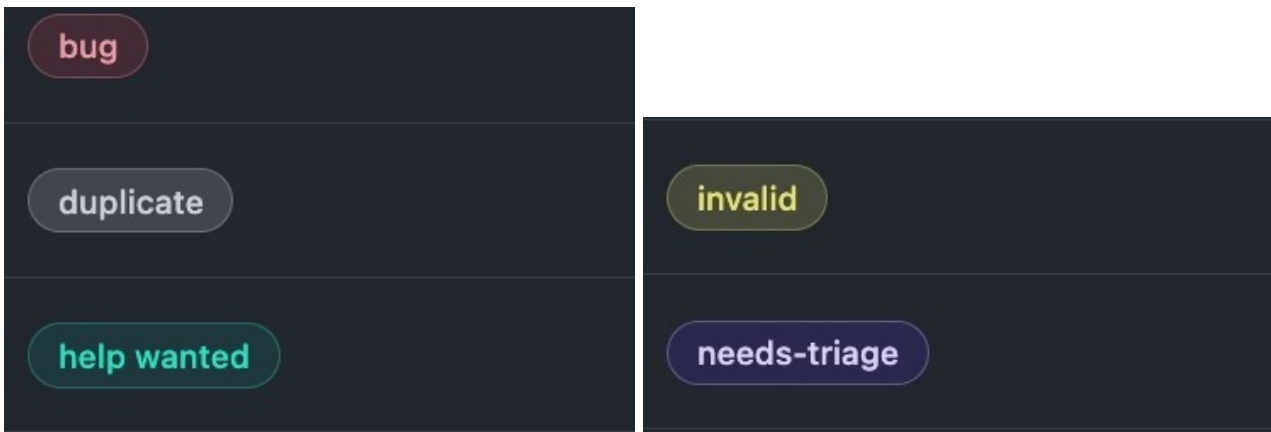
Per il versionamento e tracciamento delle modifiche apportate ai documenti e al codice steso durante il progetto viene utilizzato il *version control system* Git.

In abbinata a Git viene utilizzato GitHub per permettere facile collaborazione oltre ad accorpate in un unico luogo anche gli strumenti di ITS e CI/CD pipeline.

3.2.2 ITS

Per il tracciamento delle issue viene utilizzato GitHub.

3.2.2.1 Etichette



Per migliorare la visione e l'organizzazione delle issue vengono usate delle etichette standard per ogni repository, oltre a etichette specifiche analizzate repository per repository.

| Etichetta | Scopo |
|---------------------|---|
| bug | Segnala che la issue è relativa ad un bug |
| duplicate | Poichè le issue possono essere aperte da ognuno del gruppo è possibile che ci si creino due issue simili, questa etichetta denota la seconda delle due come duplicata |
| help wanted | Etichetta di default di <i>GitHub</i> che denota issue da gestire in più persone o che si sono fermate per via di imprevisti |
| invalid | Se quanto scritto nella issue non è attinente oppure è diventato obsoleto prima del completamento, questa issue ne denota l'invalidità |
| needs-triage | Quando una issue viene aperta per la prima volta non da un responsabile, ottiene questa etichetta |

Tabella 3.2: Tipologie di etichette

3.2.2.2 Ciclo di vita di una issue

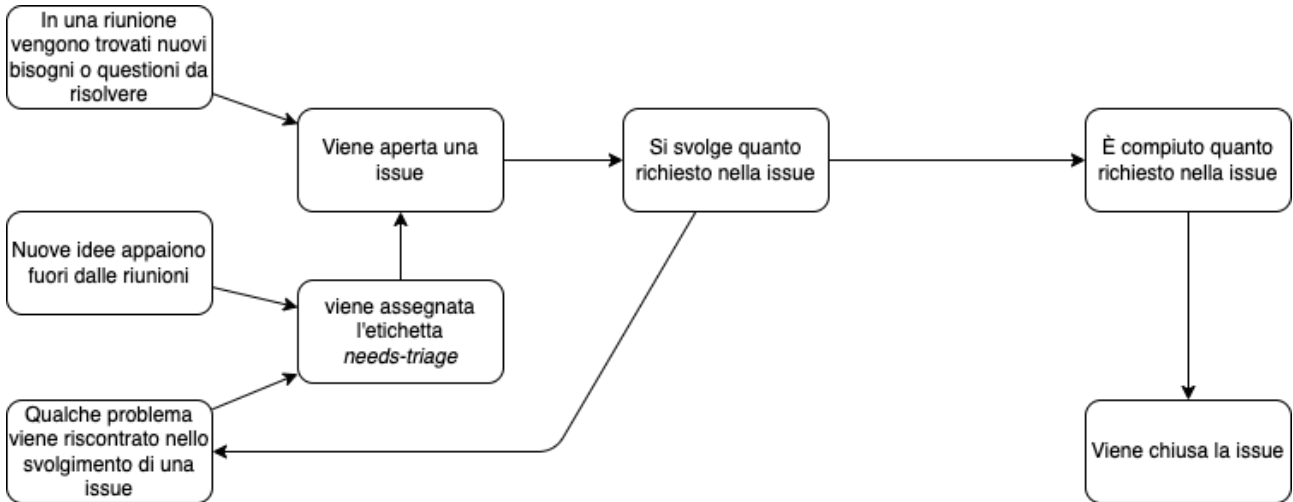
Nota per il lettore Si consiglia la visione della parola *triage* nel glossario.

Quando viene deciso o proposto un cambiamento, quando si affermano problemi tecnici o quando ci sono questioni da affrontare, viene aperta una issue.

Una issue⁵ deve contenere tutte le informazioni affinché la questione posta possa essere affrontata e risolta.

I cicli di vita delle issue sono i seguenti:

⁵Si consiglia la visione della sezione 3.2.2.3



3.2.2.3 Apertura di una issue

Una issue deve raccontare nel modo più semplice e veloce possibile una problematica oppure un bisogno.

Una issue deve contenere al suo interno la motivazione dell'apertura e il requisito che ne può determinare la chiusura.

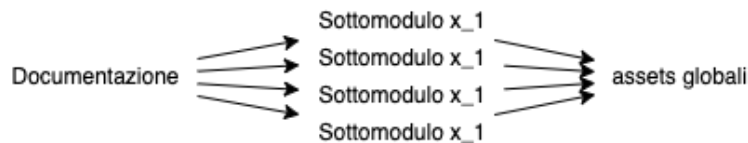
Una issue deve essere pensata come una questione risolvibile da una persona sola, se così non è⁶ la issue va segmentata in issue più piccole e gestibili da singoli in un tempo minore o uguale alla lunghezza di uno sprint.

3.2.3 Strutturazione dei repository

Nel sistema di controllo del versionamento sono presenti due tipologie di repository:

- **Repository documentale:** dedicato al sorgente in tex dei documenti;
- **Repository codice-sorgente:** dedicato al sorgente dei vari software.

Documentazione Per la documentazione si è scelto un approccio a sottomoduli. Per ognuno dei documenti viene definito un singolo repository. Per la consegna degli artefatti rilasciati al *Committente* e al *Proponente* viene però utilizzato un repository centrale unico, contenente tutti i sottomoduli documentali.



3.3 Gestione della qualità

3.3.1 Scopo

Scopo del processo di Gestione della qualità è di stabilire metriche precise per tutte le attività nell'ambito della verifica e della validazione.

In tal modo è possibile garantire il rispetto del livello di qualità fissato.

Per approfondire le metriche utilizzate si rimanda al **PdQ**

⁶A meno di casi estremi.

3.4 Gestione della verifica

3.4.1 Scopo

Scopo del processo di gestione della verifica è individuare errori all'interno del prodotto durante la sua produzione e la sua manutenzione.

Sia la documentazione⁷ che il software sono soggetti a questo processo.

3.4.2 Premesse fondamentali

La documentazione e il software seguono due processi differenti per la verifica ma con delle metodologie e degli standard simili.

In ogni caso, la verifica segue il concetto del *fail fast*, si vanno quindi ad eseguire verifiche dal più piccolo al più grande agglomerato, sia di codice che di testo.

Più in generale il processo di verifica ha due forme:

- **Analisi statica** che non richiede l'esecuzione dell'oggetto di verifica.
- **Analisi dinamica** che si può effettuare solamente durante l'esecuzione dell'oggetto di verifica.

Documentazione Poiché la documentazione è redatta tramite l'utilizzo di \LaTeX si rende necessario testarne la compilazione ad ogni nuova modifica.

3.4.3 Metodologie e ciclo di vita

Poiché per il versionamento viene utilizzato GitHub il processo di verifica è stato fatto collimare con il percorso di verifica che propone lo strumento.

Il percorso di verifica segue le seguenti fasi:

1. viene aperta una issue relativamente alla parte di oggetto di verifica che deve essere espansa;
2. viene aperto un branch relativo alla issue o al gruppo di issue se le stesse sono in relazione tra loro;
3. completato il ciclo di modifiche viene aperta una *pull request* che richiede al verificatore di analizzare le modifiche fatte;
4. il sistema automatizzato tramite *GitHubActions* effettua un'analisi statica;
5. il verificatore controlla che quanto è stato scritto corrisponda con quanto affermato durante la stesura delle issue e controlla che i test vadano a buon fine;
6. la modifica viene aggiunta all'oggetto principale e nuovamente verificata staticamente.

3.4.4 Testing

Ogni manufatto prodotto viene testato il più possibile in maniera automatica.

Verifica della compilazione Si testa che il sorgente, sia esso documentale che di software compili correttamente. In particolare a livello documentale è importante che il sorgente compili anche quando questo viene integrato con il resto del documento.

⁷Vedasi sezione 3.1 per approfondire.

Test di unità Ogni modulo aggiunto o modificato al sorgente viene testato singolarmente per pregiudicarne anomalie o errori al suo interno.

Test di integrazione Ogni modulo viene unito agli altri e se ne testa la sua integrazione con i moduli verso i quali si interfaccia.

Test di sistema Si verifica che il sistema faccia quello per cui è stato pensato e che rispetti quanto trovato nell'analisi dei requisiti.

3.4.5 Strumenti

3.4.5.1 Verifiche automatiche alla alla documentazione

Alla documentazione si applica una verifica della compilazione e poi su questo vengono effettuate delle verifiche manuali da parte di un verificatore, il quale controllerà validità di quanto redatto e con questo anche l'assenza di eventuali errori grammaticali o di sintassi che possano eventualmente essere presenti.

3.5 Validazione

3.5.1 Scopo

Scopo del processo di validazione è garantire che il prodotto sia conforme alle aspettative. La validazione garantisce che le linee guida delle **NdP** vengano rispettate.

Capitolo 4

Processi organizzativi

4.1 Gestione dell'organizzazione

4.1.1 Scopo

In questa sezione saranno illustrate le tecniche di coordinamento utilizzate dal gruppo e gli obiettivi del processo, i quali comprendono:

- Definire le responsabilità e le competenze di ciascun membro dell'organizzazione.
- Sviluppare un modello organizzativo che favorisca la collaborazione tra i membri dell'organizzazione e aumenti l'efficienza delle attività svolte.
- Pianificare il lavoro in modo da rispettare le scadenze e utilizzare al meglio le risorse disponibili.
- Monitorare costantemente l'andamento delle attività per individuare eventuali problemi e adottare le correzioni necessarie.
- Promuovere la comunicazione interna all'organizzazione e la condivisione di informazioni e conoscenze tra i membri.
- Migliorare continuamente i processi organizzativi e la performance dell'organizzazione nel lungo termine.

4.1.2 Aspettative

Illustriamo ora le seguenti attese rispetto al processo:

- Utilizzare processi efficaci per regolare le attività e renderle più efficienti dal punto di vista economico.
- Ottenere una stima accurata dei tempi, delle risorse e dei costi necessari per svolgere le attività.
- Garantire un'adeguata esecuzione e controllo delle attività svolte.
- Revisionare e valutare periodicamente le attività svolte per favorire il miglioramento continuo dei processi organizzativi e della performance dell'organizzazione.
- Favorire una cultura della collaborazione e della condivisione delle conoscenze tra i membri dell'organizzazione.

4.1.3 Descrizione

Le attività di gestione sono:

- Assegnazione dei ruoli e dei compiti ai membri del gruppo.
- Definizione degli obiettivi e dello scopo del progetto.
- Implementazione dei processi organizzativi necessari per svolgere le attività.
- Pianificazione e stima dei tempi, delle risorse e dei costi necessari per svolgere le attività, attraverso la creazione di un piano di lavoro ben definito.
- Esecuzione delle attività previste seguendo il piano di lavoro definito in precedenza.
- Controllo costante dell'andamento delle attività svolte attraverso la valutazione periodica e l'eventuale adozione di misure correttive.
- Revisione e valutazione periodica delle attività svolte per identificare, in caso, inefficienze o opportunità di miglioramento.
- Garanzia della comunicazione tra i membri del gruppo, attraverso la definizione di canali di comunicazione efficaci e la promozione di una cultura della collaborazione e della condivisione delle conoscenze.

4.2 Ruoli

4.2.1 Responsabile di progetto

Il responsabile di progetto ha il compito di coordinare e dirigere il gruppo di lavoro, assicurandosi che ogni membro abbia chiari i propri compiti e che siano rispettati i tempi e le scadenze. In particolare, i compiti assegnati al responsabile di progetto sono:

- Definizione dei requisiti del progetto
- Pianificazione e monitoraggio dell'intero progetto
- Gestione delle risorse e delle scadenze
- Comunicazione con il cliente

4.2.2 Amministratore di progetto

L'amministratore di progetto ha il compito di gestire gli aspetti amministrativi e organizzativi del progetto, fornendo supporto al responsabile di progetto e agli altri membri del gruppo. In particolare, i compiti assegnati all'amministratore di progetto sono:

- Gestione della documentazione relativa al progetto, garantendo l'archiviazione e la gestione delle informazioni in modo corretto e coerente tramite versionamento e configurazione.
- Istanziare le infrastrutture atte al supporto.
- Gestione delle problematiche relative alla gestione dei processi.

4.2.3 Analista

L'analista di progetto ha il compito di analizzare i requisiti e le esigenze del cliente e di definire le specifiche tecniche del progetto. In particolare, i compiti assegnati all'analista di progetto sono:

- Raccolta dei requisiti del cliente, tramite interviste, analisi dei documenti e discussioni con gli stakeholder del progetto.
- Definizione delle specifiche funzionali e tecniche del progetto, in accordo con il committente e gli altri membri del gruppo.
- Analisi dei rischi e delle problematiche del progetto, definendo strategie e soluzioni per mitigarli.
- Redazione della documentazione tecnica relativa al progetto, come ad esempio i diagrammi dei casi d'uso, i diagrammi delle classi.

4.2.4 Progettista

L'obiettivo principale del progettista è quello di definire e pianificare l'architettura del progetto, ne segue il suo sviluppo ma non la sua manutenzione. Le sue attività includono:

- Definizione dell'architettura del sistema e delle sue componenti;
- Definizione delle specifiche tecniche per la realizzazione del sistema;
- Verifica delle soluzioni proposte in termini di fattibilità e coerenza con i requisiti.

4.2.5 Programmatore

Il programmatore è responsabile della scrittura del codice del progetto software. Le sue attività includono:

- Implementazione delle specifiche tecniche definite dal progettista;
- Implementazione di componenti di supporto per attuare verifica e validazione;
- Collaborazione con il progettista per risolvere eventuali problematiche riscontrate nell'implementazione del sistema.

4.2.6 Verificatore

Il verificatore è incaricato lungo tutta l'attività di progetto di lavorare in stretta collaborazione con gli altri membri del team per verificare la qualità del prodotto. Le sue attività includono:

- Verificare il prodotto tramite gli strumenti e le procedure definite nelle Norme di Progetto.
- Collaborare con l'autore notificando errori presenti nel prodotto.
- Verifica dell'adeguatezza dei requisiti e delle specifiche tecniche.

4.3 Formazione

La formazione interna dei singoli membri del gruppo viene attuata tramite la pratica di auto-formazione, nella modalità di studio autonomo delle tecnologie proposte dal proponente nella fase di presentazione del capitolato e gli incontri dedicati.

4.3.1 Materiale utilizzato

Si identificano i seguenti strumenti utilizzati:

- LaTeX
- GitHub
- AngularJS
- Java Spring
- Git

4.4 Procedure

Per la sua organizzazione interna e il flusso comunicativo esterno il gruppo ha adottato le seguenti procedure:

4.4.1 Gestione delle comunicazioni

4.4.1.1 Comunicazione interna

Le comunicazioni interne del gruppo si attuano tramite i canali di comunicazione che prevedono gli applicativi Discord e Telegram. Questi applicativi permettono di creare un ambiente di lavoro virtuale a distanza.

4.4.1.2 Comunicazione esterna

Le comunicazioni verso il proponente sono gestite dal responsabile di progetto e avvengono nelle seguenti modalità:

- tramite posta elettronica, usando l'indirizzo e-mail del gruppo sweasabi@gmail.com
- tramite la piattaforma Meet, per colloqui diretti con Imola Informatica.
- tramite Telegram, utilizzando un gruppo dedicato tra "sweasabi" e Lorenzo Patera (Imola Informatica)

4.4.2 Gestione degli incontri

4.4.2.1 Incontri interni

Il responsabile di progetto pianifica un incontro interno con il gruppo. Deve essere garantita la massima trasparenza e condivisione delle informazioni sullo stato di avanzamento del progetto. La partecipazione è obbligatoria, lo scopo dell'incontro è quello di individuare gli argomenti da trattare e definire un'agenda della riunione interna.

4.4.2.2 Verbali di riunioni interne

Al termine di ogni incontro interno, viene compilato un verbale dalla figura del segretario. Viene descritto all'interno del documento un riassunto dell'incontro, approvato poi dal responsabile.

4.4.2.3 Incontri esterni

Il responsabile di progetto pianifica un incontro esterno con il proponente. Inoltre il proponente oppure il Committente possono richiedere incontri con il gruppo di progetto, a cui il responsabile risponde proponendo una data ed un orario per l'incontro.

4.4.2.4 Verbali di riunioni esterne

Al termine dell'incontro esterno, il segretario ha il compito di redigere il verbale esterno, contenente il riassunto dell'incontro, approvato poi dal responsabile.

4.5 Gestione dei rischi

È compito del responsabile identificare e rendere noti i rischi. Si adotta la seguente procedura di gestione dei rischi notificati nel piano di progetto:

1. Identificazione di nuovi rischi e controllo di quelli già noti.
2. Rimodulazione della procedura di gestione dei rischi al bisogno.

4.5.1 Codifica dei rischi

Si utilizza la seguente codifica per i rischi identificati:

- **RT:** Rischi Tecnologici.
- **RO:** Rischi Organizzativi.
- **RI:** Rischi Interpersonali.

4.6 Strumenti

Il gruppo utilizza i seguenti strumenti per la gestione del progetto:

- **Git**
- **GitHub**
- **Google Drive**
- **Telegram**
- **Discord**
- **Meet**

Capitolo A

Standard ISO/IEC 12207:1997

ISO/IEC 12207 è uno standard internazionale che definisce tutti i processi necessari per lo sviluppo e la manutenzione di sistemi software.

Per ogni processo esso stabilisce inoltre le attività da svolgere ed i risultati da produrre.

I processi sono divisi in tre categorie:

- Processi primari;
- Processi di supporto;
- Processi organizzativi.

A.1 Processi primari

A.1.1 Acquisizione

Il processo descrive le attività dell'acquirente. Esso ha inizio con la necessità di un sistema, un prodotto software o un servizio.

Questo processo si conclude con la consegna del prodotto software e si compone delle seguenti attività:

- Inizializzazione;
- Preparazione della richiesta di proposta;
- Preparazione e aggiornamento del contratto;
- Monitoraggio dei fornitori;
- Accettazione e completamento.

A.1.2 Fornitura

Il processo descrive le attività del fornitore. Esso può avere inizio in due occasioni differenti:

- Decisione di presentare una proposta in risposta alla richiesta del acquirente;
- Al momento della firma del contratto per la fornitura di un sistema, prodotto software o servizio.

Questo processo si conclude con la consegna del prodotto software e si compone delle seguenti attività:

- Inizializzazione;
- Preparazione di una risposta;
- Contratto;
- Pianificazione;
- Esecuzione e controllo;
- Revisione e valutazione;
- Consegna e completamento.

A.1.3 Sviluppo

Il processo descrive le attività dello sviluppatore. Esso norma lo sviluppo di un sistema software che risponda alle necessità del cliente.

Questo processo si conclude con la consegna del prodotto software e si compone delle seguenti attività:

- Inizializzazione;
- Analisi dei requisiti;
- Progettazione architetturale;
- Progettazione di dettaglio;
- Codifica e testing;
- Integrazione del software;
- Collaudo del software;
- Integrazione del sistema:
- Collaudo del sistema;
- Installazione.

A.1.4 Operation

Il processo si svolge in contemporanea alla fase di Manutenzione. Ha lo scopo di mantenere operativo il sistema e di fornire supporto agli utenti.

Si compone delle seguenti attività:

- Implementazione del processo;
- Operational testing;
- System operation;
- Supporto all'utente.

A.1.5 Manutenzione

Il processo descrive le attività del manutentore. Esso ha inizio quando il prodotto software viene sottoposto a modifiche del codice o della documentazione associata a causa di difetti o miglioramenti.

Questo processo si conclude con il ritiro del prodotto software.

Si compone delle seguenti attività:

- Implementazione del processo;
- Analisi del problema o della modifica;
- Implementazione della modifica;
- Revisione/accettazione della manutenzione;
- Migrazione;
- Ritiro del software.

A.2 Processi di supporto

I processi di supporto hanno lo scopo di aiutare tutti gli altri processi attivi. Essi possono essere attivati da un processo primario o da un altro processo di supporto.

A.2.1 Documentazione

Il processo ha lo scopo di registrare le informazioni prodotte da un'attività o da un processo del ciclo di vita di un prodotto software.

Si compone delle seguenti attività:

- Implementazione del processo;
- Identificazione dei documenti;
- Progettazione e sviluppo;
- Produzione;
- Manutenzione.

A.2.2 Gestione della configurazione

Il processo ha lo scopo di definire e mantenere l'integrità di tutti i componenti della configurazione e di renderli accessibili a chi ne ha diritto.

Si compone delle seguenti attività:

- Implementazione del processo;
- Identificazione della configurazione;
- Controllo della configurazione;
- Registrazione dello stato della configurazione;
- Valutazione della configurazione;
- Gestione del rilascio e distribuzione.

A.2.3 Gestione e controllo della qualità

Il processo ha lo scopo di assicurare che tutti i prodotti e i processi del ciclo di vita siano conformi con gli standard ed i requisiti definiti.

Si compone delle seguenti attività:

- Implementazione del processo;
- Accertamento del prodotto;
- Accertamento del processo;
- Accertamento dei sistemi di qualità.

A.2.4 Verifica

Il processo ha lo scopo di determinare se i prodotti software di un'attività soddisfano i requisiti o le condizioni imposte dalle attività precedenti. Il processo di Verifica dovrebbe essere integrato nei processi di Sviluppo, Fornitura e Manutenzione il prima possibile.

Si compone delle seguenti attività:

- Implementazione del processo;
- Verifica

A.2.5 Validazione

Il processo ha lo scopo di accertarsi che il prodotto software o il sistema rispettino tutti i requisiti individuati.

Si compone delle seguenti attività:

- Implementazione del processo;
- Validazione

A.2.6 Revisione della congiunta

Il processo ha lo scopo di revisionare lo stato ed i prodotti delle attività rispetto agli obiettivi definiti negli accordi, sia a livello di progetto che a livello tecnico. Tali revisioni avvengono durante l'intero ciclo di vita. Ad esse partecipano le parti interessate: solo il team se si revisiona un singolo componente o con la partecipazione del committente per revisioni sull'intero prodotto.

Si compone delle seguenti attività:

- Implementazione del processo;
- Revisione della gestione del progetto;
- Revisioni tecniche.

A.2.7 Audit

Il processo ha lo scopo di determinare se vi è conformità dei prodotti e dei processi rispetto a requisiti, pianificazione e accordi. L'attività di auditing è svolta da personale che non ha partecipato direttamente allo sviluppo dei prodotti, servizi o sistemi oggetto delle revisioni.

Si compone delle seguenti attività:

- Implementazione del processo;
- Audit.

A.2.8 Risoluzione dei problemi

Il processo ha lo scopo di analizzare e risolvere i problemi che vengono individuati in modo responsabile e documentato.

Si compone delle seguenti attività:

- Implementazione del processo;
- Risoluzione del problema.

A.3 Processi organizzativi

I processi organizzativi sono di responsabilità dell'organizzazione che li attiva. Essi hanno lo scopo di stabilire e implementare una struttura e gestire i processi del ciclo di vita e del personale. Inoltre sono utili per la gestione del continuo miglioramento dei processi e della struttura stessa.

A.3.0.1 Gestione

Il processo contiene le seguenti attività generiche per la gestione dei processi:

- Implementazione del processo;
- Pianificazione
- Esecuzione e controllo;
- Revisione e valutazione;
- Terminazione.

A.3.1 Infrastrutture

Il processo ha lo scopo di istituire e mantenere le infrastrutture necessarie a qualsiasi altro processo.

L'infrastruttura può comprendere: hardware, software, strumenti, tecniche, standard e funzionalità per sviluppo, operazioni o manutenzione.

Si compone delle seguenti attività:

- Implementazione del processo;
- Istituzione dell'infrastruttura;
- Manutenzione dell'infrastruttura.

A.3.2 Miglioramento

Il processo ha lo scopo di istituire, valutare, misurare, controllare e migliorare un processo di ciclo di vita del software. Si compone delle seguenti attività:

- Istituzione del processo;
- Valutazione del processo;
- Miglioramento del processo.

A.3.3 Formazione

Il processo ha lo scopo di fornire e mantenere personale istruito. L'acquisizione, la fornitura, lo sviluppo e molti altri processi dipendono da personale esperto e competente.

Si compone delle seguenti attività:

- Implementazione del processo;
- Sviluppo del materiale per la formazione;
- Implementazione del piano di formazione.

Capitolo B

Standard di qualità ISO/IEC 9126

ISO/IEC 9126 descrive un modello di qualità del software, definisce le caratteristiche che la determinano e propone metriche per la sua misurazione.

Questo standard si divide in quattro parti:

B.1 Modello di qualità

Questa prima parte del modello si compone a sua volta di due parti:

B.1.1 Qualità esterna ed interna

In questa sezione vengono descritte sei caratteristiche e varie sotto-caratteristiche per la qualità esterna ed interna, di seguito riportate.

B.1.2 Funzionalità

Capacità del prodotto software di fornire funzioni che soddisfino esigenze espresse ed implicite, necessarie ad operare in specifiche condizioni.

- **Adeguatezza:** capacità del prodotto software di fornire un appropriato insieme di funzioni per specifici task e obiettivi dell'utente;
- **Accuratezza:** capacità del prodotto software di fornire i risultati o effetti corretti o stabiliti con un certo grado di precisione;
- **Interoperabilità:** capacità del prodotto software di interagire con uno o più sistemi specificati;
- **Sicurezza:** la capacità del prodotto software di proteggere informazioni e dati, negandone l'accesso e la modifica a persone e sistemi non autorizzati ma permettendola a chi è abilitato;
- **Conformità:** capacità del prodotto software di aderire a standard, convenzioni e regolamenti di carattere legale o similari che siano attinenti alle funzionalità offerte.

B.1.3 Affidabilità

Capacità del prodotto software di mantenere livelli di performance specificati quando utilizzato sotto specifiche condizioni.

- **Maturità:** capacità del prodotto software di evitare che si verifichino errori o siano prodotti risultati non corretti;

- Tolleranza agli errori: capacità del prodotto software di mantenere livelli predeterminati di prestazioni anche in presenza di malfunzionamenti o usi scorretti del prodotto;
- Recuperabilità: capacità del prodotto software di ripristinare il livello appropriato di prestazioni e di recuperare le informazioni rilevanti in seguito a un malfunzionamento. A seguito di un errore, il software può risultare non accessibile per un determinato periodo di tempo. Questo arco di tempo è valutato proprio dalla caratteristica di recuperabilità;
- Aderenza: capacità del prodotto software di aderire a standard, regole e convenzioni relative all'affidabilità.

B.1.4 Usabilità

Capacità del prodotto software di essere comprensibile, apprendibile, usabile e attraente per l'utente quando utilizzato sotto condizioni specifiche.

- Comprensibilità: capacità del prodotto software di rendere l'utente in grado di capire le sue funzionalità e di come poterle utilizzare per svolgere determinati task.
- Apprendibilità: capacità del prodotto software di permettere all'utente di imparare l'applicazione;
- Operabilità: capacità del prodotto software di permettere agli utenti di utilizzarlo e controllarlo;
- Attrattività: capacità del prodotto software di risultare attraente per l'utente;
- Aderenza all'usabilità: capacità del software di aderire a standard o convenzioni relative all'usabilità.

B.1.5 Efficienza

Capacità del prodotto software di fornire prestazioni appropriate, relativamente alla quantità di risorse usate, sotto determinate condizioni.

- Comportamento rispetto al tempo: capacità del prodotto software di fornire adeguati tempi di risposta e di elaborazione sotto specifiche condizioni;
- Utilizzo delle risorse: capacità del prodotto software di utilizzare un numero e tipo di risorse appropriato quando esegue le funzionalità previste sotto determinate condizioni di utilizzo;
- Conformità: capacità del prodotto software di aderire a standard e convenzioni relative all'efficienza.

B.1.6 Manutenibilità

Capacità del prodotto software di essere modificato, includendo correzioni, miglioramenti o adattamenti.

- Analizzabilità: capacità di poter effettuare la diagnosi sul prodotto software ed individuare le cause di errori o malfunzionamenti;
- Modificabilità: capacità del prodotto software di permettere lo sviluppo di modifiche al prodotto originale (codice, progettazione, documentazione);
- Stabilità: capacità del prodotto software di evitare effetti indesiderati derivanti da modifiche del software;
- Testabilità: capacità del prodotto software di consentire la verifica e validazione del software modificato;
- Aderenza alla manutenibilità: capacità del prodotto software di aderire a standard e convenzioni relative alla manutenibilità.

B.1.7 Portabilità

Capacità del prodotto software di essere trasportato da un ambiente di lavoro ad un altro.

- **Adattabilità:** capacità del prodotto software di essere adattato a differenti ambienti senza richiedere azioni specifiche diverse da quelle previste dal software per tali attività;
- **Installabilità:** capacità del prodotto software di essere installato in uno specifico ambiente;
- **Coesistenza:** capacità del prodotto software di coesistere con altre applicazioni indipendenti in ambienti comuni e condividere risorse;
- **Sostituibilità:** capacità del prodotto software di sostituire un altro software specifico per svolgere gli stessi compiti nello stesso ambiente;
- **Aderenza alla portabilità:** capacità del prodotto software di aderire a standard e convenzioni relative alla portabilità.

B.1.8 Qualità in uso

In questa sezione vengono specificate quattro caratteristiche per la qualità in uso.

- **Efficacia:** capacità del prodotto software di permettere all'utente di raggiungere gli obiettivi specifici con accuratezza e completezza;
- **Produttività:** capacità del prodotto software di permettere all'utente di impegnare un numero definito di risorse, in relazione all'efficacia raggiunta, in uno specifico contesto d'uso;
- **Sicurezza fisica:** rappresenta la capacità del prodotto software di raggiungere un livello accettabile di rischio per i dati, le persone, il business, le proprietà o gli ambienti in uno specifico contesto di utilizzo;
- **Soddisfazione:** capacità del prodotto software di soddisfare gli utenti in uno specifico contesto di utilizzo.

B.2 Metriche per la qualità esterna

La seconda parte dello standard definisce le metriche per misurare quantitativamente la qualità esterna del prodotto software rispetto alle caratteristiche definite nella prima parte.

Con qualità esterna si intendono i comportamenti del prodotto software rilevabili dai test e dall'osservazione durante la sua esecuzione.

B.3 Metriche per la qualità interna

La terza parte dello standard definisce le metriche per misurare quantitativamente la qualità interna del prodotto software rispetto alle caratteristiche definite nella prima parte.

Le metriche interne si applicano al software non eseguibile durante le fasi di progettazione e codifica e misurano qualità intrinseche del prodotto. Esse permettono di individuare eventuali problemi che potrebbero influire sulla qualità finale del prodotto prima che sia realizzato il software eseguibile.

B.4 Metriche per la qualità in uso

La quarta parte dello standard definisce le metriche per misurare quantitativamente la qualità in uso del prodotto software rispetto alle caratteristiche definite nella prima parte. Esse misurano il grado con cui il prodotto software permette agli utenti di svolgere le proprie attività con efficacia, produttività, sicurezza e soddisfazione nel contesto operativo previsto.

La qualità in uso rappresenta la vista esterna che l'utente ha del prodotto ed è misurata in base ai risultati ottenuti dal suo utilizzo.