



Norme di Progetto

6Coders

`6Coders.unipd@gmail.com`

10 maggio 2024



Registro delle Modifiche - Changelog

Ver.	Data	Autore	Verificatore	Descrizione
2.0	10/05/2024		Marchiorato Pietro	Approvazione
1.6	02/05/2024	Lovato Yuri	Marchiorato Pietro	Modifiche alla sezione "Verifica"
1.5	12/04/2024	Lovato Yuri	Niagu Ana Maria	Modifiche alla sezione "Struttura della repository"
1.4	03/04/2024	Lovato Yuri	Vullent Vogli	Modifiche alla sezione "Piano di progetto"
1.3	28/03/2024	Chelhaoui Osama	Marchiorato Pietro	Modifiche alla sezione "Codifica"
1.2	15/03/2024	Bilinski Eleonora	Marchiorato Pietro	Modifiche alla sezione "Sviluppo"
1.1	08/03/2024	Lovato Yuri	Florian Edoardo	Sviluppata sezione "Specifica Tecnica" e "Manuale Utente"
1.0	25/02/2024		Marchiorato Pietro	Approvazione del documento
0.10	01/02/2024	Niagu Ana Maria	Bilinski Eleonora	Modifica della sottosezione Documentazione con applicazione delle nuove convenzioni stilistiche
0.9	16/01/2024	Vullnet Vogli	Florian Edoardo	Modifica della sottosezione Gestione della qualità
0.8	15/01/2024	Vullnet Vogli	Florian Edoardo	Modifica della sezione Processi organizzativi
0.7	01/01/2024	Bilinski Eleonora	Niagu Ana Maria	Conversione documenti in LaTeX
0.6	18/12/2023	Bilinski Elonora	Vogli Vullnet	Incremento alla sottosezione Verifica e Automazione
0.5	04/12/2023	Chelhaoui Osama	Lovato Yuri	Incremento alla sottosezione Gestione della configurazione e Gestione della qualità

Ver.	Data	Autore	Verificatore	Descrizione
0.4	26/11/2023	Marchiorato Pietro	Bilinski Eleonora	Incremento della sotto-sezione Fornitura Sviluppo di Processi Primari
0.3	24/11/2023	Marchiorato Pietro	Bilinski Eleonora	Incremento della sezione Processi Organizzativi
0.2	09/10/2023	Florian Edoardo	Bilinski Eleonora	Sviluppo sezione Processi primari, Processi Organizzativi e Processi di Supporto
0.1	08/11/2023	Florian Edoardo	Osama Chelhaoui	Creazione documento e sezione Introduzione

Tabella 1: Registro delle modifiche

Indice

1	Introduzione	6
1.1	Scopo del Documento	6
1.2	Scopo del Prodotto	6
1.3	Glossario	6
1.4	Riferimenti	6
1.4.1	Riferimenti normativi	6
1.4.2	Riferimenti informativi	6
2	Processi Primari	7
2.1	Fornitura	7
2.1.1	Descrizione	7
2.1.2	Scopo e aspettative	7
2.1.3	Piano di progetto	7
2.1.3.1	Scopo	7
2.1.3.2	Descrizione	7
2.1.4	Piano di qualifica	9
2.1.4.1	Scopo	9
2.1.4.2	Descrizione	9
2.1.5	Specifica Tecnica	10
2.1.5.1	Scopo	10
2.1.5.2	Descrizione	10
2.1.6	Manuale Utente	11
2.1.6.1	Scopo	11
2.1.6.2	Descrizione	11
2.1.7	Collaudo e rilascio	12
2.1.7.1	Scopo	12
2.1.7.2	Descrizione	12
2.2	Sviluppo	13
2.2.1	Descrizione	13
2.2.2	Scopo e aspettative	13
2.2.3	Attività	13
2.2.4	Analisi dei requisiti	13
2.2.4.1	Scopo	13
2.2.4.2	Casi d'uso	14
2.2.4.3	Requisiti	15
2.2.5	Progettazione	15
2.2.5.1	Scopo	15
2.2.5.2	Descrizione	15
2.2.6	Codifica	16
2.2.6.1	Scopo	16
2.2.6.2	Nome dei file	16
2.2.6.3	Commenti	16

3	Processi di Supporto	17
3.1	Documentazione	17
3.1.1	Descrizione	17
3.1.2	Scopo e aspettative	17
3.1.3	Ciclo di vita di un documento	17
3.1.4	Documenti prodotti	17
3.1.5	Template del documento	19
3.1.6	Struttura del documento	19
3.1.7	Prima pagina	19
3.1.8	Registro delle modifiche	20
3.1.9	Indice	20
3.1.10	Piè di pagina	20
3.1.11	Verbali	20
3.1.12	Nome dei file	21
3.1.13	Stile del testo	22
3.1.14	Norme tipografiche	22
3.1.15	Glossario	22
3.1.16	Sigle	22
3.1.17	Colori	23
3.1.18	Tabelle	23
3.1.19	Immagini	23
3.1.20	Preventivo	23
3.1.21	Consuntivo	24
3.2	Gestione della configurazione	24
3.2.1	Descrizione	24
3.2.2	Scopo e aspettative	24
3.2.3	Versione	25
3.2.4	Sistemi software utilizzati	25
3.2.5	Struttura repository	25
	3.2.5.1 Repository utilizzata	25
	3.2.5.2 Organizzazione della repository	25
3.3	Gestione della qualità	26
3.3.1	Scopo	26
3.3.2	Piano di qualifica	26
3.3.3	Testing	26
3.3.4	Metriche	27
3.3.5	Aspettative	27
3.4	Automazione	27
3.4.1	Descrizione	27
3.4.2	Scopo	27
3.4.3	Script	27
3.5	Verifica	28
3.5.1	Descrizione	28
3.5.2	Scopo	28
3.5.3	Indicazioni	28

3.5.4	Verifica della documentazione	28
3.5.4.1	Analisi statica documentazione	28
3.5.5	Verifica del prodotto	29
3.5.5.1	Analisi statica prodotto	29
3.5.5.2	Analisi dinamica prodotto	29
3.6	Validazione	29
3.6.1	Descrizione	29
4	Processi Organizzativi	30
4.1	Gestione di Processo	30
4.1.1	Coordinamento	30
4.1.1.1	Comunicazioni	30
4.1.2	Riunioni	31
4.2	Pianificazione	31
4.2.1	Ruoli	31
4.3	Infrastruttura	33
4.3.1	Strumenti	33
4.4	Miglioramenti	33
4.5	Formazione	33

1 Introduzione

1.1 Scopo del Documento

Con il seguente documento vengono definite tutte le norme ed i meccanismi che regolano il *Way of Working_G*. Tali direttive, pensate e sottoscritte da ogni membro del gruppo, verranno rispettate sino alla creazione del prodotto finale, al fine di lavorare in modo efficiente ed efficace.

1.2 Scopo del Prodotto

Vista la notevole diffusione, specie negli ultimi anni dell'Intelligenza Artificiale, e l'abilità che i *Large Language Model_G* hanno dimostrato nel generare codice in diversi linguaggi di programmazione, il *Proponente_G*, Zucchetti S.p.A., con il *Capitolato_G* C9 - ChatSQL, richiede di sviluppare un'applicazione che consenta, a partire dalla struttura dati di un *database_G* (*dizionario dati_G*), la generazione di un *prompt_G* in risposta a interrogazioni formulate in *linguaggio naturale_G*. Tale prompt sarà quindi sottoposto a un *LLM_G* (Large Language Model) come *ChatGPT_G*, il quale produrrà la *query_G* in linguaggio *SQL_G* corrispondente alla domanda posta dall'utente. Questa applicazione, sviluppata come *web application_G*, sarà accessibile da tutti i principali *browser_G* (Chrome, Firefox, Edge e Safari), e permetterà agli utenti di richiedere, previo caricamento di un file *JSON_G* che descriva il database in questione, la produzione di un prompt per poter poi far generare la query SQL necessaria ad effettuare l'interrogazione sulla base di dati, il tutto a partire da una richiesta inserita da un utente in linguaggio naturale.

1.3 Glossario

Il documento "[Glossario V2.0](#)" è parte della documentazione di progetto dove è possibile reperire definizioni chiare e precise dei vocaboli non comuni utilizzati nei documenti prodotti. La presenza di un determinato vocabolo in "Glossario" viene segnata con la lettera G a pedice (es. *Vocabolo_G*).

1.4 Riferimenti

1.4.1 Riferimenti normativi

- [Capitolato \[C9\] - ChatSQL \(Zucchetti SpA\)](#).
- [Glossario V2.0](#).

1.4.2 Riferimenti informativi

- [Processi di ciclo di vita del software](#).
- [Documentazione LaTeX Overleaf](#).

2 Processi Primari

2.1 Fornitura

2.1.1 Descrizione

Nella seguente sezione sono riportate tutte le regole, adottate dal gruppo 6Coders, al fine di mantenere attivi e produttivi i rapporti con il Proponente, Zucchetti S.p.A. , ed evitare incomprensioni durante la realizzazione del progetto.

2.1.2 Scopo e aspettative

L'obiettivo della procedura di fornitura è chiarire ogni dubbio riguardante l'intenzione del Proponente, perseguendo quindi la completa comprensione delle richieste al fine di soddisfarle completamente. Durante l'intero sviluppo del progetto, il gruppo 6Coders, si impegna a mantenere una comunicazione frequente con l'azienda Zucchetti S.p.A. al fine di poter:

- Definire in modo univoco i bisogni ($needs_G$) del proponente;
- Ottenere dei $feedback_G$ del lavoro svolto da parte del Proponente;
- Chiarire gli eventuali dubbi e incomprensioni che incontreremo nell'avanzamento dello sviluppo del progetto;
- Ottenere formazione sulle tecnologie proposte dal Proponente;
- Stabilire tempistiche e vincoli per la consegna del prodotto finale.

2.1.3 Piano di progetto

2.1.3.1 Scopo

Il "Piano di Progetto", noto anche come *project plan_G*, è un documento formale soggetto a revisioni e approvazioni, in cui vengono delineati gli obiettivi, le strategie e le risorse necessarie per il successo del progetto. Il "Piano di Progetto" include una dettagliata pianificazione delle attività, la distribuzione delle responsabilità tra i membri del *team_G*, la stima dei tempi e dei costi, nonché le procedure per il monitoraggio e il controllo dei progressi. Inoltre, il piano di progetto serve come punto di riferimento per la gestione dei rischi, identificando le potenziali strategie per affrontarli efficacemente.

2.1.3.2 Descrizione

Il "Piano di Progetto" è caratterizzato da:

- **Piano di gestione dei rischi:** il piano di gestione dei rischi prevede che i rischi siano identificati, valutati in base alla loro occorrenza e pericolosità ed, infine, che vengano prese delle misure di prevenzione e *contingenza_G*. I rischi vengono classificati secondo le seguenti categorie:

- Rischi legati alle persone;
 - Rischi legati all'organizzazione;
 - Rischi legati agli strumenti;
 - Rischi legati ai requisiti.
- **Modello di sviluppo:** descrive il modello di sviluppo adottato dal gruppo;
 - **Pianificazione:** l'attività di pianificazione prevede un'analisi dettagliata di tutte le attività necessarie per raggiungere gli obiettivi del progetto entro un determinato periodo di tempo, nel nostro caso i periodi hanno una durata prestabilita di due settimane. Durante questa fase, si scompongono le attività in compiti più piccoli e gestibili, si valutano le risorse necessarie per ciascuna attività, si stabiliscono le dipendenze tra di esse e si assegnano le responsabilità ai membri del team. Un aspetto chiave della pianificazione è lo sviluppo del *diagramma di Gantt_G*, che fornisce una rappresentazione grafica delle attività, delle loro dipendenze e delle stime di tempo, consentendo al team di visualizzare e gestire il programma di lavoro in modo efficace. Il piano viene periodicamente rivisto e aggiornato per riflettere i cambiamenti nel progetto, garantendo la sua adattabilità e coerenza nel corso del tempo;
 - **Preventivo:** l'attività del preventivo si svolge in parallelo all'attività di pianificazione e consiste nell'assegnare i ruoli ai membri del team e stabilire le ore di lavoro previste per ciascuna persona coinvolta nel progetto. Questa fase include anche la stima dei costi associati alle risorse umane impiegate, consentendo di pianificare e monitorare il *budget_G* del progetto. Durante l'attività del preventivo, si identificano i ruoli necessari per portare avanti il progetto e si assegnano ai membri del team in base alle loro competenze e disponibilità. Successivamente, si definiscono le ore di lavoro previste per ciascun ruolo, tenendo conto delle attività da svolgere e dei tempi stimati per completarle. Inoltre, questa attività permette di monitorare l'avanzamento dei costi del progetto, questo monitoraggio fornisce informazioni preziose sulla gestione del budget e consente di apportare eventuali correzioni o aggiustamenti nel corso del progetto;
 - **Consuntivo:** il consuntivo fornisce una valutazione dettagliata delle ore effettivamente impiegate da ciascun membro del team rispetto alle ore preventivate nel preventivo. Questo confronto consente di monitorare il peso lavorativo su ogni persona e il consumo economico del progetto. Durante l'attività di consuntivo, vengono registrate le ore di lavoro effettivamente svolte da ciascun membro del team per completare le attività assegnate. Queste ore vengono quindi confrontate con le ore preventivate nel preventivo per valutare eventuali scostamenti. Questa analisi fornisce informazioni importanti sulla produttività del team, sull'efficienza delle risorse e sulla gestione del tempo. Inoltre, il consuntivo fornisce una valutazione del consumo economico del progetto, poiché le ore effettivamente impiegate sono correlate ai costi associati alle risorse umane coinvolte. Questo permette di monitorare il budget del progetto e di identificare eventuali variazioni rispetto alle previsioni iniziali. In sintesi, il consuntivo fornisce una panoramica chiara e dettagliata delle prestazioni del team e dei costi del progetto, consentendo di prendere tempestivamente le misure correttive necessarie. In particolare si attua:

- **Analisi di gestione dei rischi:** Si valutano ed elencano i rischi incontrati nei vari periodi, descrivendo il loro impatto ed le misure di contingenza adottate;
- **Analisi di retrospettiva di periodo:** Un’analisi retrospettiva del periodo è molto importante per poter effettuare la pianificazione ed il preventivo per il periodo successivo , viene valutato lo scostamento tra quanto preventivato e pianificato, anche grazie ad un elenco della valutazione degli obbiettivi di periodo ,con i risultati ottenuti per stimare l’efficienza;
- **Prospettive future ed aree di miglioramento:** Dopo una retrospettiva di periodo si descrivono le aree che sono state evidenziate come critiche per la pianificazione futura.

2.1.4 Piano di qualifica

2.1.4.1 Scopo

Il documento di “Piano di Qualifica” definisce gli obiettivi e gli standard di qualità che il prodotto e i processi devono soddisfare durante lo sviluppo. Serve come guida dettagliata per assicurare che il prodotto finale risponda alle esigenze e alle aspettative del Proponente, garantendo la sua qualità e affidabilità. Questo documento fornisce inoltre una struttura per la pianificazione delle attività di verifica e validazione, che sono essenziali per assicurare che il prodotto soddisfi gli standard di qualità stabiliti. Infine, il “Piano di Qualifica” ha lo scopo di riferimento per valutare il successo del progetto e identificare eventuali aree di miglioramento nei processi di sviluppo e nelle pratiche di qualità.

2.1.4.2 Descrizione

Il Piano di Qualifica è composto da:

- **Qualità di Processo:** stabilisce i metodi e gli approcci utilizzati per gestire, monitorare e migliorare i processi coinvolti nello sviluppo del prodotto. Inoltre, stabilisce i criteri e le metriche di valutazione utilizzati per misurare il grado di conformità dei processi rispetto agli standard di qualità definiti;
- **Qualità di prodotto:** definisce gli standard, i requisiti e le procedure necessarie per garantire che il prodotto finale soddisfi le esigenze e le aspettative del proponente. Questa sezione del documento fornisce una panoramica dettagliata degli attributi di qualità che il prodotto deve possedere. Inoltre, stabilisce i criteri e le metriche di valutazione utilizzati per misurare il grado di conformità del prodotto rispetto agli standard di qualità definiti;
- **Testing:** delinea gli approcci, le strategie e le attività pianificate per condurre il processo di testing sul prodotto Fornisce una visione completa delle modalità con cui verranno eseguiti i test per garantire che il prodotto soddisfi gli standard di qualità stabiliti. Questa parte del documento fornisce una lista chiara e concisa degli attributi di qualità che il prodotto deve possedere, insieme a metriche e criteri di valutazione per misurarli;

- **Resoconto delle attività di verifica:** fornisce un'analisi dettagliata delle attività di verifica svolte durante il processo di sviluppo del software. Questa parte del documento raccoglie e riporta i risultati delle attività di verifica eseguite per garantire la conformità del prodotto agli standard di qualità stabiliti;
- **Valutazioni per il miglioramento:** fornisce una struttura per valutare e migliorare continuamente il processo di sviluppo e la qualità del prodotto. Essa si concentra sulle attività volte a identificare, analizzare e correggere eventuali problemi o inefficienze nel processo di sviluppo, al fine di garantire una progressiva ottimizzazione delle prestazioni e dei risultati.

2.1.5 Specifica Tecnica

2.1.5.1 Scopo

La "Specifica tecnica" è un documento di tipo formale che fornisce una descrizione dettagliata dell'architettura e dell'applicazione software. Questo documento si focalizza sui dettagli tecnici dell'applicazione, inclusi l'ambiente di sviluppo, i linguaggi di programmazione utilizzati, i moduli del sistema, le interfacce, i componenti, e i *design pattern*_G adottati. Essa ha lo scopo di:

- Comunicare chiaramente e dettagliatamente l'architettura e la progettazione del sistema a tutti gli *stakeholder*_G coinvolti;
- Guidare gli sviluppatori nelle decisioni di progettazione durante lo sviluppo del sistema;
- Fornire un punto di riferimento costante per il team di sviluppo durante il ciclo di vita del progetto;
- Essere utilizzata come base per valutare la fattibilità tecnica del progetto e identificare eventuali rischi;
- Servire come documento ufficiale che documenta l'architettura e la progettazione del sistema.

2.1.5.2 Descrizione

La Specifica tecnica è composta da:

- **Tecnologie utilizzate:** descrizione delle tecnologie utilizzate nel prodotto, in particolare:
 - **Linguaggio utilizzati:** elenco dei linguaggi di programmazione utilizzati;
 - **Framework:** elenco dei *framework*_G utilizzati per lo sviluppo del prodotto;
 - **Librerie utilizzate:** elenco delle librerie incluse per lo sviluppo del prodotto;
 - **Strumenti utilizzati:** elenco degli strumenti di gestione e di installazione;

- **Strumenti di analisi statica:** elenco delle librerie e framework utilizzati per effettuare l'*analisi statica_G*;
- **Strumenti di analisi dinamica:** elenco delle librerie e framework utilizzati per effettuare l'*analisi dinamica_G*.
- **Architettura di sistema:** descrive la struttura generale del sistema, inclusi i suoi componenti principali e le relazioni tra di essi. Questa sezione include diagrammi *UML_G* dell'architettura dell'applicazione per visualizzare in modo chiaro e schematico i componenti del sistema, le loro interazioni e le relazioni tra di essi.
- **Design pattern:** identifica i design pattern utilizzati nell'implementazione del sistema e spiega come sono applicati per risolvere problemi specifici di progettazione;
- **Requisiti funzionali:** specifica i requisiti funzionali identificati dall'analisi dei requisiti ed il loro grado di soddisfazione.

2.1.6 Manuale Utente

2.1.6.1 Scopo

Il "Manuale utente" è un documento di tipo formale che fornisce istruzioni dettagliate e informazioni pertinenti agli utenti finali su come utilizzare il prodotto software. Questo documento ha lo scopo di rendere accessibili e comprensibili le funzionalità del prodotto agli utenti, guidandoli attraverso il processo di utilizzo in modo efficace ed efficiente.

2.1.6.2 Descrizione

Il Manuale Utente include:

- **Requisiti di sistema:** vengono descritti ed elencati i requisiti:
 - **Software;**
 - **Hardware.**necessari all'utilizzo del programma, inoltre vengono descritti i *browser_G* supportati dall'applicativo;
- **Installazione e avvio:** guida per l'installazione del software, compresi i requisiti di sistema necessari e i passaggi per avviare correttamente il prodotto;
- **Istruzioni per l'utilizzo e funzionalità:** vengono descritte istruzioni dettagliate su come utilizzare ogni funzionalità del prodotto, spesso supportate da esempi pratici e illustrazioni, inoltre viene data una panoramica delle principali funzionalità offerte dal prodotto, spiegando cosa può fare e come può essere utilizzato per soddisfare le esigenze dell'utente;
- **Supporto tecnico:** viene descritto come contattare il supporto tecnico in caso di problemi o domande.

2.1.7 Collaudo e rilascio

2.1.7.1 Scopo

L'attività di *collaudo_G* e rilascio è l'ultima fase del ciclo di sviluppo di un progetto e coinvolge l'approvazione finale del prodotto o del sistema prima che venga messo a disposizione del *Committente_G*. Questa fase è critica per garantire che il risultato del lavoro soddisfi le specifiche del progetto e le aspettative del Committente. Se il collaudo non riscontra anomalie potranno essere consegnati al Committente il codice sorgente e la documentazione relativa al prodotto finale.

2.1.7.2 Descrizione

I principali *test_G* coinvolti sono:

- **Collaudo del prodotto:**
 - **Test di unità:** verificano singolarmente il funzionamento di ogni singola unità di codice;
 - **Test di accettazione:** si eseguono per assicurarsi che il prodotto soddisfi i requisiti specificati dal cliente;
 - **Test di integrazione:** vengono effettuati per verificare il corretto funzionamento delle varie parti del sistema integrate insieme;
 - **Test di sistema:** controllano le prestazioni del sistema nel suo complesso, verificandone il funzionamento globale.
- **Collaudo utente:** coinvolgimento degli utenti finali per valutare l'usabilità e la *user experience_G* del prodotto;
- **Approvazione e accettazione:** il Committente verifica se il prodotto soddisfa i requisiti concordati e se è pronto per il rilascio;
- **Rilascio del prodotto:** una volta ottenuta l'approvazione del Committente, il prodotto è ufficialmente rilasciato e reso disponibile per l'utilizzo.

Se una delle attività riscontra delle anomalie e/o errori (*bugs_G*), queste dovranno essere risolte al più presto.

2.2 Sviluppo

2.2.1 Descrizione

Lo sviluppo di un software prevede un'articolazione di processi, dalla pianificazione, passando per lo sviluppo, fino al *testing_G* e rilascio del prodotto, in maniera strutturata e pianificata. L'attività di sviluppo del software è una fase cruciale del ciclo di vita del progetto, in cui vengono implementate le funzionalità e le caratteristiche specificate nei requisiti. Questa coinvolge diverse attività che mirano a trasformare i requisiti in un prodotto software funzionante e di alta qualità.

2.2.2 Scopo e aspettative

L'obiettivo dell'attività di sviluppo è trasformare i requisiti e le specifiche del progetto. Affinché il prodotto rispetti le aspettative deve:

- Essere conforme alle richieste del Proponente;
- Garantire la qualità del prodotto;
- Rispettare tempi e costi concordati con il Proponente.

Al fine di produrre un software che rispetti le aspettative del Proponente è necessario:

- Determinare i vincoli tecnologici;
- Determinare gli obiettivi di sviluppo e *desing_G*;
- Realizzare un prodotto che superi i test di verifica e validazione.

2.2.3 Attività

Il processo di sviluppo prevede le seguenti attività:

- Analisi dei requisiti;
- Progettazione;
- Codifica.

2.2.4 Analisi dei requisiti

2.2.4.1 Scopo

L'analisi dei requisiti raccoglie e descrive in dettaglio i requisiti del sistema software che deve essere sviluppato, fornisce una visione chiara e completa delle funzionalità, dei vincoli e delle prestazioni che il sistema deve soddisfare.

Descrizione

L'analisi dei requisiti descrive:

- **Casi d'uso:** si focalizza sulla descrizione dettagliata degli scenari di utilizzo del sistema da parte degli attori coinvolti. Ogni caso d'uso rappresenta un'interazione specifica tra un *attore_G* e il sistema stesso, evidenziando i passaggi coinvolti e gli obiettivi da raggiungere. Questa sezione fornisce una panoramica chiara e concisa delle funzionalità del sistema dal punto di vista degli utenti;
- **Requisiti:** descrivono requisiti funzionali e non funzionali che il sistema deve soddisfare. In dettaglio i requisiti vengono classificati in:
 - **Requisiti funzionali:** definiscono le funzionalità specifiche del sistema e descrivono le azioni che il sistema deve essere in grado di eseguire;
 - **Requisiti di Dominio o di Vincolo:** stabiliscono le restrizioni o le condizioni che devono essere rispettate dal sistema;
 - **Requisiti di Qualità:** definiscono gli attributi di qualità del sistema, come l'affidabilità, la sicurezza, la manutenibilità e l'usabilità;
 - **Requisiti di Prestazioni:** stabiliscono le caratteristiche di prestazione che il sistema deve soddisfare, come i tempi di risposta, la capacità di elaborazione, la scalabilità e la gestione delle risorse.

Notare che la descrizione specifica cosa il sistema dovrà fare, non il come. L'insieme delle descrizioni vengono raccolti in documenti noti come documenti di specifica.

2.2.4.2 Casi d'uso

Definiscono uno scenario in cui uno o più *attori_G* interagiscono con il sistema. Per definire i casi d'uso il gruppo ha definito la seguente nomenclatura:

UC[Numero].[Sottocaso]-[Titolo]

Dove:

- **Numero:** indica il numero del caso d'uso;
- **Sottocaso:** indica il sottocaso del caso d'uso in esame;
- **Titolo:** indica un titolo che descrive il caso d'uso.

2.2.4.3 Requisiti

Per definire i requisiti il gruppo ha definito la seguente nomenclatura:

R[Tipologia]-[Codice]

Dove:

- **Tipologia:** indica la tipologia del requisito:
 - **F:** indica i requisiti funzionali;
 - **V:** indica i requisiti di dominio / vincolo;
 - **Q:** indica i requisiti di qualità;
 - **P:** indica i requisiti prestazionali.
- **Codice:** indica l'identificativo univoco in base alla tipologia.

2.2.5 Progettazione

2.2.5.1 Scopo

La progettazione si occupa di definire come soddisfare i requisiti individuati dall'Analisi dei Requisiti. Viene definita l'architettura del software, i moduli che lo compongono e come questi interagiscono tra loro, e le interfacce con l'utente.

2.2.5.2 Descrizione

La progettazione può essere suddivisa in due parti principali:

- **Progettazione logica:** Vengono identificati e delineati i moduli principali del prodotto, insieme alle loro interazioni e alle regole di *business_G* che regolano il comportamento del sistema. Motiva le scelte tecnologiche usate per la realizzazione del prodotto dimostrandole attraverso un *PoC_G*, essa contiene:
 - I *framework_G* e le librerie utilizzate;
 - Il PoC;
 - I *diagrammi UML_G*.
- **Progettazione di dettaglio:** E' il passaggio successivo alla progettazione logica nel processo di sviluppo. Essa si occupa di tradurre le astrazioni concettuali della progettazione logica in una forma più dettagliata e implementabile. Essa contiene:
 - Diagramma delle classi;
 - Tracciamento delle classi;
 - Test di unità per ogni componente.

2.2.6 Codifica

2.2.6.1 Scopo

La codifica è il processo di implementazione del software. Nella codifica avviene la traduzione delle specifiche della progettazione di dettaglio in codice sorgente.

2.2.6.2 Nome dei file

I nomi dei file devono avere le seguenti specifiche:

- Essere univoci;
- Essere descrittivi, cioè devono esplicitarne il contenuto.

2.2.6.3 Commenti

I commenti riferiti al codice devono essere chiari e concisi, se viene descritta una funzione seguire la seguente struttura:

- Breve descrizione della funzione;
- Descrivere i parametri di ingresso e di uscita;
- Descrivere le *pre – condizioni*_G e *post – condizioni*_G della funzione.

3 Processi di Supporto

3.1 Documentazione

3.1.1 Descrizione

Questa sezione fornisce le linee guida per la scrittura e la verifica dei documenti prodotti nel contesto del progetto.

3.1.2 Scopo e aspettative

Il fine di questa sezione è stabilire norme e vincoli che regolino l'intera documentazione generata durante il ciclo di vita del progetto. Questo garantirà una struttura comune e coesa per tutti i documenti. Le aspettative che il gruppo intende soddisfare sono:

- Fornire una struttura uniforme e chiara per agevolare sia la redazione che la lettura dei documenti;
- Fornire norme precise per ciascun aspetto della struttura documentale, consentendo a ciascun membro di lavorare in modo asincrono ed autonomo.

3.1.3 Ciclo di vita di un documento

Il ciclo di vita di un documento descrive le fasi dettagliate della sua realizzazione:

- **Pianificazione:** vengono analizzati i requisiti e vengono organizzate le varie parti del documento;
- **Impostazione:** viene creata la struttura del documento. In modo sincrono il "Responsabile" crea una *issue_G* per ogni sezione del documento, suddividendo logicamente il lavoro;
- **Realizzazione:** ad ogni *issue_G* sono assegnati uno o più membri del team. Il Componente, una volta selezionata la *issue_G* di suo interesse, inizia lo sviluppo della sezione corrispondente;
- **Verifica:** una volta completata, la sezione viene sottoposta a verifica da parte del verificatore, che ne controlla l'idoneità rispetto ai requisiti attesi;
- **Approvazione:** dopo la verifica dell'intero documento, il responsabile in carica decide se approvare il documento con successiva aggiunta al *repository_G* di sviluppo oppure di rifiutarlo, richiedendo ulteriori revisioni.

3.1.4 Documenti prodotti

I documenti prodotti sono di due tipi:

- **Documenti Formali:** seguono delle regole specifiche per struttura, formato e linguaggio contenuto, vengono redatti secondo standard definiti. Sono soggetti a revisione e approvazione da parte del "Responsabile" prima di essere considerati ufficiali e vincolanti. Regolano l'operato del gruppo e lo sviluppo delle attività durante il ciclo di vita del software. Le caratteristiche di questi documenti sono:
 - **Struttura standard:** i documenti formali seguono regole e standard definiti riguardo la loro struttura;
 - **Responsabilità e tracciabilità:** chiare responsabilità sono associate alla creazione, verifica, approvazione e archiviazione di questi documenti. Hanno una tracciabilità definita in modo da identificare chi ha effettuato quali modifiche e quando;
 - **Approvazione:** approvazione della versione da parte del "Responsabile".

Se un documento ha più di una versione, si considera sempre quella più recente approvata dal responsabile. I documenti formali possono essere di due tipi:

- **Interni:** utili al team di sviluppo in quanto riguardano le dinamiche interne al gruppo, ma poco utili al Committente e Proponente;
- **Esterni:** interessano direttamente il Committente e Proponente.

Sono documenti formali:

- **"Norme di progetto":** stabilisce le linee guida, le procedure, le convenzioni e le regole specifiche che il team 6Coders dovrà seguire durante lo svolgimento del progetto (Documento interno);
 - **"Glossario":** elenco di termini tecnici, acronimi, abbreviazioni e concetti specifici utilizzati nel contesto del progetto (Documento esterno);
 - **"Piano di progetto":** delinea la struttura, le attività, le risorse, le tempistiche e le strategie per la gestione e l'esecuzione del progetto (Documento esterno);
 - **"Piano di qualifica":** stabilisce le strategie, le metodologie, le attività e le risorse necessarie per garantire e controllare la qualità del progetto (Documento esterno);
 - **"Analisi dei requisiti":** vi è l'identificazione, la raccolta e analisi dei bisogni, delle funzionalità e delle caratteristiche che il prodotto deve possedere per soddisfare le esigenze del committente (Documento esterno);
 - **"Manuale utente":** fornisce istruzioni, guide e informazioni dettagliate sull'utilizzo del prodotto (Documento esterno);
 - **"Specifica tecnica":** fornisce una descrizione dettagliata dell'architettura del prodotto. Descrive in modo approfondito la struttura, i componenti, i moduli, le interazioni e le relazioni tra le varie parti del sistema (Documento esterno).
- **Documenti Informali:** sono tutti quei documenti non ancora approvati dal "Responsabile" e non soggetti a $Versione_G$ nella $repository_G$, sono meno strutturati e senza

seguire uno standard preciso, vengono utilizzati per comunicazioni interne, appunti personali, *brainstorming_G* ed idee preliminari. Questi documenti sono contenuti nel *Google Drive_G* condiviso al team.

3.1.5 Template del documento

Per standardizzare la struttura dei documenti si è deciso di utilizzare il linguaggio *LaTeX_G*. I vantaggi che questa scelta comporta sono:

- **Struttura uniforme:** facilita la stesura e l'integrazione dei contenuti;
- **Maggiore coerenza:** standardizza il formato e la presentazione dei documenti;
- **Maggiore efficienza:** riduce il tempo dedicato alla formattazione e permette di concentrarsi sui contenuti;
- **Facile collaborazione:** semplifica la condivisione e la modifica dei documenti tra i membri del gruppo.
- **Versione:** permette di tenere facilmente traccia delle modifiche e delle variazioni apportate ai documenti.

3.1.6 Struttura del documento

Ogni documento contiene le seguenti componenti:

- La prima pagina che specifica nome e data di creazione del documento;
- Un registro delle modifiche (*Changelog_G*) per il Versione e per tenere traccia delle modifiche apportate;
- Il contenuto vero e proprio del documento la quale struttura può variare in base alla tipologia del documento stesso.

3.1.7 Prima pagina

La prima pagina del documento è formata da:

- Logo del gruppo;
- Nome del documento;
- Data di creazione del documento;
- Nome del gruppo e relativa email;
- Logo dell'università di Padova e relative informazioni sul corso.

3.1.8 Registro delle modifiche

Ogni documento contiene una tabella delle modifiche apportate al documento, nella quale viene tenuta traccia dei cambiamenti significativi del documento nel suo ciclo di vita, in particolare la tabella riporta:

- Versione del documento;
- Data di Versione;
- Nominativo dell'autore;
- Nominativo del verificatore;
- Descrizione della modifica apportata al documento.

3.1.9 Indice

I documenti presentano un indice dei contenuti per facilitare la navigazione, è posizionato appena dopo il registro delle modifiche. In particolare l'indice riporta le sezioni e i sottosezioni del documento, inoltre se sono presenti tabelle e figure, queste vengono elencate in un indice a parte, il nome delle tabelle e delle figure deve essere descrittivo e chiaro. I verbali sia interni che esterni non presentano indice.

3.1.10 Piè di pagina

Il piè di pagina deve essere così composto:

- **Nome del documento:** posizionato nella porzione sinistra del piè di pagina;
- **Numero di versione corrente:** posizionato nella porzione centrale del piè di pagina cominciante per la lettera V seguita dal numero di versione corrente del documento (E.g. V 1.0);
- **Numero di pagina:** posizionato nella porzione destra del piè di pagina.

3.1.11 Verbali

Registro scritto che riporta in modo dettagliato le discussioni, le decisioni, gli accordi e le azioni intraprese durante una riunione, una conferenza o un incontro formale, un verbale è così organizzato:

- **Informazioni e partecipanti:**
 - Tipo di riunione: può essere interna o esterna;
 - Luogo;
 - Data e ora;
 - Responsabile;

- Verbalizzante;
 - Verificatore;
 - Partecipanti interni;
 - Partecipanti esterni: se il verbale è interno viene lasciato vuoto questo campo.
- **Ordine del giorno:** riporta gli argomenti trattati e discussi durante la riunione;
 - **Svolgimento della riunione,** solo per i verbali esterni: viene presentato lo svolgimento della riunione con il Proponente;
 - **Quesiti posti al proponente,** solo per i verbali esterni: tabella che riporta le domande e le relative risposte effettuate al Proponente;
 - **Todo / Ongoing:** tabella nella quale vengono descritti brevemente i *task_G* o le modifiche da apporre al progetto discusse durante la riunione, nel dettaglio:
 - Il campo *task* descrive l'azione o la modifica che verrà effettuata;
 - Il campo *assignees_G*/assegnatari sono le persone del team al quale viene assegnato lo sviluppo della task corrispondente.

3.1.12 Nome dei file

Il nome dei documenti è diviso nelle seguenti sezioni:

- Nome del gruppo;
- Titolo effettivo che rappresenta il contenuto, scritto con lettera iniziale maiuscola. Se il titolo presenta più parole non vengono separate da spazi ma vengono scritte tutte attaccate e ogni parola inizia con la lettera maiuscola;
- Ultima Versione approvata del documento (le cifre del Versione, in questo caso, vengono separate dal simbolo “-” invece che dal consueto simbolo “.”, dato che quest’ultimo è già presente per separare il nome del file dalla sua estensione).

Ogni sezione è separata dal simbolo “_”. (e.g. 6Coders_NormeDiProgetto_1-0)

Il titolo dei verbali invece è formato da:

- Una “v” iniziale la quale indica che si tratta di un verbale seguita da una sigla che può essere “INT” se il verbale è interno, oppure “EXT” se il verbale è esterno;
- La data nel formato **aaaammgg**.

Anche qui le due sezione sono separata dal simbolo “_”, inoltre non è presente il Versione (e.g. vEXT_20231026)

3.1.13 Stile del testo

Gli stili adottati dai nei documenti sono:

- Grassetto: per titoli, sottotitoli e per parole ritenute particolarmente importanti;

3.1.14 Norme tipografiche

- Le voci degli elenchi puntati o numerati iniziano sempre per la lettera maiuscola e terminano con punto e virgola, tranne l'ultimo punto che termina con punto fermo;
- Un elenco puntato utilizza il simbolo •, un successivo annidamento utilizza il simbolo -, e un altro ancora utilizza il simbolo *;
- Gli annidamenti degli elenchi numerati sono ordinati con numeri arabi divisi da un punto fermo;
- I riferimenti a sezioni, sottosezioni, ai documenti ed ai ruoli sono compresi tra virgolette basse (e.g. "Norme di Progetto").

3.1.15 Glossario

Raccoglie e definisce in modo chiaro e conciso una serie di termini, parole o acronimi utilizzati nei documenti. Questo documento fornisce definizioni specifiche e significati precisi per ciascun termine, contribuendo a standardizzare e chiarire il linguaggio utilizzato. Ogni membro del team può aggiungere un termine, con relativa definizione, all'interno del glossario. Ciascun termine presente nel "Glossario" verrà segnalato, almeno una volta, con la lettera G a pedice della parola interessata.

3.1.16 Sigle

- Sigle per i nomi dei documenti:
 - "Norme di Progetto": NdP;
 - "Glossario": G;
 - "Piano di Progetto": PdP;
 - "Piano di Qualifica": PdQ;
 - "Analisi dei Requisiti": AdR;
 - "Manuale Utente": MU;
 - "Specifica Tecnica": ST;
 - "Verbali Interni": VI;
 - "Verbali Esterni": VE.
- Sigle per le revisioni:
 - Requirements and Technology Baseline: RTB;

- Product Baseline: PB;
- Customer Acceptance: CA.

3.1.17 Colori

Si è deciso di utilizzare il colore:

- **nero**: per i link interni;
- **blu**: per i link esterni;
- **marrone**: come background delle intestazioni delle tabelle (con trasparenza 50%);
- **grigio**: come background delle righe alterne delle tabelle (con trasparenza 10%).

3.1.18 Tabelle

Le tabelle presenti all'interno dei documenti hanno il seguente formato:

- **Intestazione**: identificano i contenuti di una colonna all'interno della tabella, il font è in grassetto e il colore di background deve essere come specificato nella sezione [3.1.17](#);
- **Righe**: hanno lo sfondo di colore alternato per semplificare la lettura della tabella;
- **Didascalia**: che descrive il contenuto della tabella, ha un allineamento centrale nella pagina.

3.1.19 Immagini

Le immagini sono centrate orizzontalmente e accompagnate da una didascalia descrittiva. I diagrammi UML ed i grafici all'interno dei documenti sono inseriti come immagini.

3.1.20 Preventivo

Per ogni periodo pianificato viene indicato:

- I ruoli che opereranno;
- *Preventivo_G* orario:
 - Tabella con i ruoli per colonna che descrive per ogni membro il numero di ore produttive preventivate in un determinato ruolo;
 - Un grafico a torta che segna le percentuali di ore per ogni ruolo.
- Preventivo economico:
 - Tabella che descrive per ogni ruolo il numero di ore complessive e il costo complessivo preventivato.

3.1.21 Consuntivo

Per ogni periodo pianificato viene indicato:

- *Consuntivo_G* orario: tabella con i ruoli per colonna che descrive per ogni membro il numero di ore produttive che ha effettivamente eseguito in un determinato ruolo;
- Ore rimanenti: tabella con i ruoli per colonna che indica le ore rimanenti per ogni membro del gruppo dopo il periodo concluso;
- Consuntivo economico: tabella che descrive per ogni ruolo il numero di ore complessive eseguite e il costo complessivo;
- Conclusioni consuntivo: sezione in cui viene fatta un'analisi fra le differenze tra preventivo e consuntivo a scopo di individuare eventuali problemi nel periodo appena concluso;
- Gestione dei rischi: Tabella nella quale vengono riassunti i rischi incontrati nel periodo, con una descrizione del loro impatto ed le misure di contenenza attuate;
- Analisi retrospettiva di periodo: viene fatta un'analisi retrospettiva di periodo per valutare lo scostamento della pianificazione e preventivo con quello effettivamente raggiunto nel periodo, ciò fa riferimento anche ad:
 - Tabella di valutazione degli obiettivi: Vengono elencati tutti gli obiettivi pianificati per il periodo ed il loro stato di avanzamento;
 - Grafico temporale delle issue chiuse: Esso rappresenta graficamente l'andamento temporale, all'interno del periodo, delle issue chiuse.
- Prospettive future ed aree di miglioramento: A monte della retrospettiva di periodo vengono evidenziate le aree di miglioramento per la pianificazione futura ed obiettivi per il miglioramento.

3.2 Gestione della configurazione

3.2.1 Descrizione

La presente sezione descrive come il gruppo intende gestire ed organizzare la produzione di risorse. Il processo di gestione della configurazione ha lo scopo di organizzare la produzione di documentazione e di codice in maniera sistematica, ordinata e standardizzata.

3.2.2 Scopo e aspettative

Le aspettative che il gruppo intende soddisfare sono:

- Registrazione delle modifiche apportate alle risorse, con la possibilità di effettuare *rollback_G*;
- Uniformazione degli strumenti utilizzati;
- Gestione semplice ed efficace degli errori riscontrati.

3.2.3 Versione

Il gruppo ha deciso di applicare il seguente sistema di Versione, per poter ricostruire la storia di un determinato documento/risorsa.

$$[X].[Y]$$

[X] corrisponde alla versione di produzione (pubblicabile in repository pubblica), soggetta ad approvazione da parte del "Responsabile". [Y] corrisponde all'incremento di versione per modifiche ed integrazioni minori, sottoposte a verifica da parte dei "Verificatori".

3.2.4 Sistemi software utilizzati

Per la creazione e la gestione della repository viene utilizzata la piattaforma *GitHub_G*, che a sua volta rende possibile l'integrazione con *GIT_G* per la gestione del sistema di *versioning_G*. I task vengono gestiti, assegnati e suddivisi utilizzando le Issues e le *Pull Requests_G*.

3.2.5 Struttura repository

3.2.5.1 Repository utilizzata

La repository pubblica dove sono presenti tutti i documenti approvati è accessibile al seguente indirizzo: [6Coders documentazione](#).

3.2.5.2 Organizzazione della repository

Per l'organizzazione della repository il gruppo si è basato sul sistema di *GitHub Flow_G*, organizzato a rami e di seguito illustrato:

- Ramo *main_G* (Principale) è presente la versione [X], verificata dei documenti;
- Rami secondari utilizzati per la redazione dei vari paragrafi della documentazione che, una volta completati, vengono uniti al ramo principale dal "Verificatore" attraverso l'operazione di **merge_G**.

Nel ramo principale sono presenti tre cartelle principali:

- Candidatura;
- PB;
- RTB;

Ognuna delle quali al suo interno può contenere:

- Lettere di presentazione;
- Verbali;
- Documentazione interna;

- Documentazione esterna.

Accedendo alla repository sarà possibile visualizzare il readme che descriverà nel dettaglio la struttura della repository.

3.3 Gestione della qualità

3.3.1 Scopo

Per gestione della qualità si intende l'insieme di processi e attività, effettuate durante lo sviluppo di un progetto, per garantire che la qualità dei *Deliverable_G* sia in accordo con gli obiettivi e i requisiti prestabiliti. Gli obiettivi che il gruppo ha stabilito sono i seguenti:

- Analizzare i requisiti del Proponente, assicurandosi che siano ben definiti;
- Ideare un piano di progetto che stabilisce stime realistiche dei tempi e delle risorse;
- Realizzare il progetto seguendo il piano stabilito, in conformità con i tempi e le risorse.

3.3.2 Piano di qualifica

Per garantire che la qualità sia in accordo con i requisiti del progetto si utilizza un "Piano di Qualifica". In tale documento:

- Vengono definite le metriche di valutazione del prodotto;
- Viene monitorato il rispetto di tali metriche;
- Vengono definiti i test da svolgere per la valutazione del prodotto;
- Viene documentato l'esito ottenuto dai test;
- Viene fatta una valutazione sui problemi riscontrati e proposte relative soluzioni.

3.3.3 Testing

Nel "Piano di Qualifica V2.0" sono definiti dei test a scopo di garantire la qualità del prodotto. Le tipologie dei test da eseguire sono:

- **Test di unità:** Verificano singole unità di codice per garantire che funzionino correttamente in isolamento;
- **Test di integrazione:** Collaudano l'interazione tra diverse unità o moduli del software;
- **Test di sistema:** Valutano il comportamento del sistema nel suo complesso;
- **Test di accettazione:** Garantiscono che il software soddisfi le aspettative e i requisiti del proponente.

3.3.4 Metriche

Per definire le metriche il gruppo ha definito la seguente nomenclatura:

M[Tipologia][Numero]

Dove:

- **Tipologia** indica il tipo della metrica:
 - C: riferimento ad un processo;
 - D: riferimento ad un documento;
 - S: riferimento al software.
- **Numero** indica l'identificativo univoco incrementale in base alla tipologia.

3.3.5 Aspettative

Con una gestione della qualità adatta, il gruppo si aspetta di:

- Riuscire ad individuare gli errori e risolverli in maniera efficace e tempestiva;
- Mantenere aggiornata la documentazione;
- Migliorare la collaborazione all'interno del gruppo;
- Ottenere il prodotto finale richiesto.

3.4 Automazione

3.4.1 Descrizione

L'aspetto dell'automazione mira a ottimizzare i processi, migliorare l'efficienza di sviluppo e ridurre gli errori umani.

3.4.2 Scopo

Il principale utilizzo dell'automazione verterà sulle procedure di test e verifica del codice. I test automatizzati possono essere eseguiti in modo rapido e ripetibile, permettendo di identificare tempestivamente le eventuali problematiche o regressioni che si potrebbero riscontrare.

3.4.3 Script

Al fine di testare il codice prodotto, si realizzano test di unità e successivamente una *pipeline*_G CI/CD con Github Actions. Github Actions ha tre livelli:

- **Workflows**: un file che contiene il processo di automazione. Un workflow può gestire più jobs. In particolare il gruppo ha definito due workflow:

- **Frontend:** Che si occupa di testare il codice del frontend;
- **Backend:** Che si occupa di testare il codice del backend.
- **Jobs:** un set di istruzioni che sono eseguite sullo stesso runner. Un runner è un sistema operativo dove il job dovrebbe essere eseguito;
- **Steps:** sono una sequenza di processi che devono essere eseguiti come parte del job.

Tale pipeline successivamente può essere eseguita effettuando un *push_G* su codice presente nella repository GitHub, che gestirà il resto del lavoro.

3.5 Verifica

3.5.1 Descrizione

Il processo di verifica prevede un'attività di controllo necessaria a stabilire se un determinato prodotto o documento è stato realizzato in maniera corretta, completa ed in linea con le richieste.

3.5.2 Scopo

La verifica prevede l'analisi di ogni processo e sottoprocesso e viene effettuata ogni qualvolta si presenta un cambiamento di stato.

3.5.3 Indicazioni

Il processo di verifica va attuato seguendo procedure chiare e definite, secondo criteri validi ed affidabili.

3.5.4 Verifica della documentazione

L'attività di verifica della documentazione segue le fasi seguenti:

- Controllo della correttezza dei contenuti del documento;
- Controllo della corretta applicazione delle regole sintattiche e ortografiche;
- Controllo stile e regole tipografiche.

3.5.4.1 Analisi statica documentazione

L'analisi statica sui documenti è molto utile al fine di correggere e rimuovere errori che si verificano durante lo sviluppo del documento. Essa è stata eseguita in due modi:

- **Walkthrough:** Il Verificatore, attraverso una lettura ad ampio spettro, ricerca eventuali errori nel documento, essa non è finalizzata alla ricerca di errori specifici;
- **Inspection:** Il Verificatore va alla ricerca di specifici errori attraverso letture mirate, alla ricerca di errori specifici e che spesso sono ricorrenti nei documenti.

L'analisi statica dei documenti vede prevalere inizialmente l'attività di Walkthrough, dove si effettua una lettura ad ampio spettro, per poi passare ad l'attività di Inspection per correggere e rimuovere errori specifici, sistemici e non, nei documenti.

3.5.5 Verifica del prodotto

Al fine di minimizzare gli errori il codice viene controllato mediante attività di analisi statica e dinamica. Le operazioni di verifica del prodotto vengono eseguite manualmente dai verificatori e programmatori.

3.5.5.1 Analisi statica prodotto

Essa consiste nella revisione del codice senza che quest'ultimo venga eseguito. Ha lo scopo di verificare che il codice sia privo di difetti, che segua le regole di buona programmazione e che rispetti le caratteristiche dei pattern utilizzati per la realizzazione del progetto. Gli strumenti utilizzati per effettuare l'analisi statica del prodotto sono:

- **ESLint**: Per la parte frontend;
- **PyLint**: Per la parte backend.

3.5.5.2 Analisi dinamica prodotto

Essa consiste nella revisione del programma quando il software è in esecuzione. Ha lo scopo di testare le funzionalità del programma, confrontando i risultati ottenuti con quelli attesi. I test per l'analisi dinamica devono essere ripetibili, per cui devono essere ben descritte le precondizioni per l'esecuzione del test, i parametri di input, il risultato atteso e il risultato ottenuto. Gli strumenti e utilizzati per effettuare l'analisi dinamica sono:

- **Vue Test Utils e Jest**: Per effettuare i test di unità e di integrazione sulla parte frontend;
- **Unittest**: Per effettuare i test di unità e integrazione sulla parte backend;
- **Cypress**: Per effettuare i test di sistema sul prodotto.

3.6 Validazione

3.6.1 Descrizione

Per validazione si intende l'insieme degli incontri effettuati con il Proponente e successivamente il Committente per verificare che lo stato di avanzamento del progetto o il prodotto finale presentato sia conforme ai requisiti minimi individuati in fase di analisi e rispetti le aspettative in termini di tempo pianificato e costo preventivato.

4 Processi Organizzativi

4.1 Gestione di Processo

4.1.1 Coordinamento

4.1.1.1 Comunicazioni

I canali di comunicazione scelti dai membri del gruppo per le **comunicazioni interne** sono:

- *Discord_G*: per le comunicazioni sincrone durante gli incontri;
- *Telegram_G*: per le comunicazioni asincrone (comunicazioni urgenti, informali, sondaggi e organizzazione incontri). In particolare le comunicazioni hanno un canale dedicato diviso per i seguenti *topics_G*:
 - Generale: per le comunicazioni di carattere generale o che non hanno un riscontro per nessun altro *topics*;
 - Documentazione: per le comunicazioni relative alla documentazione;
 - Backend: per le comunicazioni relative allo sviluppo del *backend_G*;
 - Frontend: per le comunicazioni relative allo sviluppo del *frontend_G*;
 - Testing: per le comunicazioni relative ai test;
- *Google Drive_G*: per la condivisione di documenti informali e non soggetti a revisione;
- *GitHub_G*: per la condivisione di documenti formali e per la gestione del Versione del codice. Inoltre ogni membro del gruppo deve monitorare attentamente la Project Board su GitHub per seguire lo stato di avanzamento del progetto e verificare/aggiornare lo stato delle task assegnate.

Le **comunicazioni esterne** con proponente e committente avvengono:

- In modalità asincrona tramite un indirizzo e-mail dedicato:

6coders.unipd@gmail.com

Tutti i membri del gruppo hanno accesso alla casella e-mail e saranno responsabili di verificare e notificare tempestivamente la ricezione di nuovi messaggi. Tuttavia, la composizione e l'invio di nuovi messaggi saranno prerogativa del Responsabile di Progetto. Prima dell'invio, ogni messaggio sarà sottoposto a una breve verifica e approvazione da parte dell'intero gruppo, si richiede che il linguaggio utilizzato mantenga un tono formale, privilegiando chiarezza e concisione nel contenuto;

- Per gli incontri sincroni in tempo reale attraverso la piattaforma *Zoom_G*: il responsabile di progetto in carica si occupa di programmare per tempo gli incontri con i partecipanti esterni.

4.1.2 Riunioni

- Le **riunioni interne** coinvolgono, per quanto possibile, tutti i membri del gruppo e vengono svolte a necessità, solitamente ogni due settimane. Il responsabile di progetto è incaricato di organizzare tali incontri in modo efficace per massimizzare la partecipazione del gruppo;
- Le **riunioni esterne** possono coinvolgere i membri del gruppo, l'azienda proponente e i docenti (committenti); solitamente, queste riunioni vengono richieste dal gruppo quando è necessario ottenere pareri più esperti su questioni tecniche o *way of working_G* e per supervisionare il corretto avanzamento del progetto didattico. Al fine di ottimizzare il tempo a disposizione, è cruciale preparare preventivamente un elenco dei punti da discutere, il quale può essere condiviso con i partecipanti esterni prima dell'incontro. In seguito, è necessario ottenere l'approvazione da parte dei partecipanti esterni coinvolti, prima della pubblicazione dei verbali relativi.

Il responsabile attualmente in carica deve coordinare e presiedere le riunioni, assicurando un tempestivo riepilogo per i membri del team che, per circostanze eccezionali, non hanno potuto partecipare. Dopo ciascuna riunione, il relativo verbale è redatto dal verbalista designato dal responsabile, il quale cercherà idealmente di sceglierlo in modo rotativo tra i membri del gruppo.

4.2 Pianificazione

4.2.1 Ruoli

Ciascun membro del gruppo assume, a intervalli di due settimane, un ruolo assegnato in modo rotativo, rispettando le modalità e le tempistiche stabilite:

- **Responsabile:** oltre a possedere una visione completa dello stato di avanzamento del progetto, il Responsabile ha il ruolo chiave di gestire e coordinare l'intero svolgimento dei processi. È incaricato di:
 - Assegnare compiti specifici ai membri del team;
 - Monitorare il progresso;
 - stilare i preventivi e consuntivi a fine periodo;
 - Affrontare eventuali problematiche o ritardi che possano insorgere durante l'implementazione del progetto. Ha il ruolo fondamentale di mantenere la coesione e l'efficienza del gruppo.
- **Amministratore:** svolge un ruolo essenziale nel promuovere l'efficacia e l'efficienza dei processi del progetto. Questo coinvolge:
 - La stesura di documenti chiave come le Norme di Progetto, il Piano di Progetto e il Piano di Qualifica;

- La gestione attiva dell’infrastruttura e degli strumenti impiegati, contribuendo in modo significativo a garantire uno sviluppo progressivo, efficiente ed efficace del progetto;
 - automatizzazione dei processi e identificare aree di miglioramento.
- **Analista:** riveste un ruolo cruciale nelle prime fasi del progetto, responsabile di comprendere a fondo le esigenze e richieste del proponente. In particolare, l’Analista:
 - Analizza le richieste del proponente;
 - Traduce le esigenze in requisiti chiari e dettagliati;
 - Documenta in modo dettagliato e strutturato tutti i risultati ottenuti, creando la ”Analisi dei Requisiti”, questo documento serve come guida essenziale per la progettazione e lo sviluppo successivo del sistema.

La sua attività influisce direttamente sulla definizione accurata degli obiettivi del progetto;

- **Progettista:** incaricato di modellare la soluzione ai requisiti delineati nel documento ”Analisi dei Requisiti”, in particolare si impegna a:
 - Produrre una soluzione che rispetti tutti i requisiti;
 - Garantire la qualità dell’architettura del sistema, focalizzandosi su un elevato livello di manutenibilità e cercando di ridurre al minimo le dipendenze;
 - Sviluppare conoscenze tecniche più approfondite e a ricercare nuovi strumenti tecnologici rilevanti per l’applicazione specifica.
- **Programmatore:** ha la responsabilità di implementare concretamente quanto emerso nella fase di progettazione. Possiede competenze tecniche specifiche, operando con autonomia e responsabilità circoscritte, in particolare:
 - Scrive codice in conformità alle specifiche di progettazione;
 - Applica le best practices nella scrittura del codice per garantirne la qualità;
 - Favorisce la manutenibilità del codice attraverso il Versione e la documentazione;
 - Sviluppa gli strumenti per la verifica del funzionamento del codice;
 - Redige il Manuale Utente.
- **Verificatore:** garantisce la qualità del prodotto e dei processi nel contesto di un progetto attraverso le seguenti attività:
 - Svolge un controllo approfondito per assicurare che tutto ciò che viene prodotto rispetti i requisiti di qualità definiti per il progetto;
 - Assicura uniformità e coerenza in tutte le fasi del progetto;
 - Segnala eventuali errori riscontrati durante l’attività di verifica.

4.3 Infrastruttura

4.3.1 Strumenti

- **Gmail**: servizio di posta elettronica utilizzato per l’invio e la ricezione di comunicazioni ufficiali da parte del gruppo;
- **Telegram**: piattaforma utilizzata per la comunicazione asincrona del gruppo;
- **Discord**: strumento utilizzato per la comunicazione sincrona e la condivisione di risorse utili;
- **GitHub**: sistema di Versione e gestione della repository pubblica di sviluppo e documentale;
- **Zoom**: strumento utilizzato per la comunicazione sincrona con i partecipanti esterni durante le riunioni;
- **Overleaf**: strumento utilizzato per la produzione dei documenti formali rilasciati dal gruppo;
- **Suite Google (Drive, Docs, Spreadsheet)**: strumenti utilizzati per la produzione dei documenti informali interni al gruppo.

4.4 Miglioramenti

Non avendo esperienza in questo ambito i documenti prodotti saranno soggetti a continuo miglioramento, con l’obiettivo di correggere eventuali errori. Il documento Piano di Qualifica, in particolare nella sezione Valutazioni per il miglioramento, descrive i problemi principali riscontrati dal gruppo, con opportune descrizioni e soluzioni.

4.5 Formazione

Ogni componente del gruppo deve studiare e comprendere in maniera autonoma e approfondita gli strumenti e le tecnologie utilizzati per lo svolgimento del progetto, in maniera tale da garantire un livello di avanzamento organizzato, e simultaneo. Un membro del team che ha acquisito una conoscenza più approfondita di un determinato strumento o tecnologia, sarà tenuto a condividere le proprie conoscenze con gli altri membri del gruppo in modo da poter avanzare in maniera omogenea. Di seguito vengono riportate le tecnologie che vengono adottate per lo sviluppo e la gestione del progetto e che richiedono di essere studiate e approfondite:

- Gestione di progetto:
 - [Git](#);
 - [GitHub Flow](#);
 - [GitHub Actions](#).

- Documentazione:
 - [LaTeX](#).
- Sviluppo:
 - [Python](#);
 - [PyTorch](#);
 - [Flask](#);
 - [Vue.js](#);
 - [Axios](#);
 - [Bootstrap](#);
 - [Transformers di Hugging Face](#);
 - [Vue Test Utils](#);
 - [Jest](#);
 - [Unittest](#);
 - [Cypress](#);
 - [Pinia](#).