



Norme di Progetto

Autore Nicolò Lattanzio

Verificatore

Approvazione

Indice

1	Registro delle versioni	3
2	Informazioni introduttive	3
2.1	Scopo del documento	3
2.2	Scopo del prodotto	4
2.3	Riferimenti	4
3	Processi Primari	4
3.1	Processo di Fornitura	4
3.1.1	Documenti Principali:	5
3.1.2	Strumenti a supporto	6
3.2	Processo di Sviluppo	7
3.2.1	Analisi dei Requisiti	7
3.2.2	Nomenclatura Casi d'Uso	7
3.2.3	Nomenclatura Requisiti	8
3.2.3.1	Tipologia	8
3.2.3.2	Importanza	8
3.2.3.3	Codice	9
3.2.4	Codifica del codice sorgente	9
3.2.5	Strumenti di supporto	9
4	Processi Di Supporto	10
4.1	Documentazione	10
4.1.1	Struttura dei documenti	10
4.1.2	Tabella Decisioni e Azioni	11
4.1.3	Diari di bordo	11
4.1.4	Strumenti a supporto	11
4.2	Gestione della configurazione	12
4.2.1	Produzione e archiviazione	12
4.2.2	Versionamento della documentazione	12
4.2.3	Visualizzazione della documentazione	13
4.2.4	Strumenti a supporto	13
4.3	Garanzia della Qualità (Quality Assurance)	13
4.4	Verifica	14
4.5	Validazione	14
5	Processi Organizzativi	14
5.1	Gestione dei processi	15
5.2	Gestione dei ruoli	15
5.2.1	Responsabile	15
5.2.2	Amministratore	16
5.2.3	Analista	16
5.2.4	Progettista	16
5.2.5	Sviluppatore	17
5.2.6	Verificatore	17

5.3	Riunioni	17
5.3.1	Riunioni interne	17
5.3.2	Riunioni esterne DA RIVEDERE	18
5.4	Gestione delle comunicazioni	18
5.4.1	Comunicazioni interne	18
5.4.2	Comunicazioni esterne	18
5.5	Gestione delle infrastrutture e degli strumenti	18
5.6	Processo di miglioramento	19
5.7	Formazione	20
6	Metriche e standard per la Qualità	20
7	Metriche di Qualità del Processo	20
7.1	Processi Primari	20
7.1.1	Schedule Performance Index (SPI)	20
7.1.2	Cost Performance Index (CPI)	21
7.1.3	Requirements Stability Index (RSI)	21
7.2	Processi di Supporto	21
7.2.1	Code Coverage (Test di Unità)	21
7.2.2	Indice di Gulpease (Leggibilità Documenti)	22
8	Metriche di Qualità del Prodotto (ISO/IEC 25010:2023)	22
8.1	Adeguatezza Funzionale	22
8.1.1	Copertura Funzionale	22
8.2	Affidabilità	22
8.2.1	Mean Time To Repair (MTTR)	22
8.3	Efficienza Prestazionale	22
8.3.1	Average Response Time	23
8.4	Compatibilità	23
8.4.1	Browser Compatibility Rate	23
8.5	Usabilità	23
8.5.1	Average Error Rate (AER)	23
8.6	Sicurezza	23
8.6.1	Vulnerability Density	24
8.7	Manutenibilità	24
8.7.1	Cyclomatic Complexity (Complessità Ciclomatica)	24
8.8	Portabilità	24
8.8.1	Environment Deployment Success	24

1 Registro delle versioni

Versione	Data	Autore	Descrizione delle modifiche
0.0.8	12/01/2026	Nicolò Lattanzio	Aggiunta sezione Metriche di Qualità del processo e prodotto, e completamento sezione 3.2 Processi di Sviluppo.
0.0.7	08/01/2026	Nicolò Lattanzio	Aggiunta sotto sezione diari di bordo e sezione Gestione della configurazione
0.0.6	05/01/2026	Giulia Romanato	Aggiunta sezione processi organizzativi
0.0.5	22/12/2025	Nicolò Lattanzio	Aggiunta sezione Strumenti di supporto e sezione Produzione e archiviazione.
0.0.4	15/12/2025	Nicolò Lattanzio	Aggiunta sezione Documenti Principali, insieme ai primi documenti chiave del progetto.
0.0.3	07/12/2025	Nicolò Lattanzio	Aggiunta sottosezione 4.1.1 per la Struttura dei documenti, 4.1.2 per il loro versionamento e 4.1.3 per la tabella Decisioni e Azioni.
0.0.2	06/12/2025	Nicolò Lattanzio	Aggiunta sezione Processi Primari, con relative sottosezioni, e prima redazione della sezione Processi di Supporto.
0.0.1	30/11/2025	Nicolò Lattanzio	Creazione e prima redazione del documento.

2 Informazioni introduttive

2.1 Scopo del documento

Questo documento definisce le norme di progetto adottate dal gruppo **BYTE HOLDERS** per la gestione e lo sviluppo del progetto. Esso stabilisce le linee guida per la documentazione, la comunicazione, la gestione delle versioni e le responsabilità dei membri del team.

Per la realizzazione del documento, e' stato deciso di prendere come riferimento lo standard ISO/IEC 12207, che definisce i processi di ciclo di vita del software e fornisce una struttura per la gestione del progetto. Le tipologie di processi sono le seguenti:

- **Processi Primari** : Consiste nelle attività principali che portano alla creazione, consegna e mantenimento del prodotto finale.
- **Processi di Supporto**: Processi che forniscono supporto alle attività principali, possono essere utilizzato da qualsiasi processo e ne garantiscono la qualità e la correttezza

- **Processi Organizzativi:** Processi necessari a creare l'ambiente e le risorse necessarie affinché i processi primari possano esistere.

2.2 Scopo del prodotto

Il prodotto sviluppato dal gruppo **BYTE HOLDERS** è un'applicazione web che mira a fornire una piattaforma intuitiva per la gestione delle repository GitHub, fornendo informazioni utili in base alle varie necessità. Tra queste funzionalità vi sono:

- Analisi statica della qualità del codice
- Analisi della qualità della documentazione
- Analisi OWASP per la verifica delle vulnerabilità di sicurezza tramite l'integrazione di strumenti di terze parti
- Integrazione con API di GitHub per fornire statistiche dettagliate sulle repository
- Analisi specifica sulle pull request per valutare la qualità del codice proposto
- Proposte di remedation per migliorare la qualità del codice e della documentazione

Il prodotto segue la proposta del **capitolato C2 - Code Guardian** fornito dal proponente **Var-Group S.r.l.**

Il gruppo si è posto l'obiettivo di completare l'MVP di questo progetto entro il **20 Marzo 2026**, con un costo totale di **13.290 Euro**.

2.3 Riferimenti

- [Capitolato C2 - Code Guardian](#)
- [Documentazione del progetto](#)

3 Processi Primari

Questa sezione descrive i processi del ciclo di vita primari, definiti dallo standard ISO/IEC 12207, che sono direttamente coinvolti nella creazione, fornitura e manutenzione del prodotto software. Essi costituiscono il cuore operativo del progetto, governando le attività principali dall'acquisizione delle esigenze alla consegna.

Possiamo distinguere due macro processi:

- **Processo di Fornitura:** Si occupa della gestione delle relazioni con il committente, dalla risposta alla richiesta iniziale fino alla consegna del prodotto.
- **Processo di Sviluppo:** Copre tutte le attività ingegneristiche necessarie per trasformare i requisiti in un prodotto software funzionante, includendo analisi, progettazione, implementazione, testing e rilascio.

3.1 Processo di Fornitura

Questo processo definisce le attività che il gruppo Byte Holders, in qualità di fornitore, implementa per garantire la corretta fornitura del prodotto software a VarGroup S.r.l., committente

del progetto. Esso disciplina l'intero ciclo, dall'aggiudicazione del capitolato alla consegna finale.

Vediamo ora le attività principali coinvolte in questo processo:

1. **Risposta al Bando:** Analisi del capitolato e preparazione dell'offerta (documento *Candidatura*).
2. **Negoziazione Requisiti:** Realizzazione di una controproposta per il proponente, definendo i requisiti funzionali e non funzionali che il gruppo Byte Holders, come fornitore, prevede di riuscire a realizzare.
3. **Pianificazione della Fornitura:** Creazione del *Piano di Progetto* che dettaglia come verranno eseguiti i processi di sviluppo e gestione.
4. **Esecuzione e Controllo:** Realizzazione del progetto conforme al piano, con reporting periodico allo stakeholder.
5. **Revisione e Valutazione:** Condotta di revisioni tecniche e attività di verifica e validazione.
6. **Consegna e Completamento:** Consegna del prodotto finale insieme alla relativa documentazione.

3.1.1 Documenti Principali:

Per questo progetto saranno prodotti i seguenti documenti chiave:

Analisi Dei Requisiti

Autore: Analista

Destinatari: VarGroup S.r.l., ByteHolders, Professori Tullio Vardanega e Riccardo Cardin

Scopo:

Nell'analisi dei requisiti il gruppo Byte Holders si impegna a raccogliere, analizzare e documentare i requisiti funzionali e non funzionali del sistema richiesto dal committente VarGroup S.r.l. In particolare saranno descritti i vari casi d'uso, le specifiche di interfaccia e i vincoli tecnici.

Glossario

Autore: Amministratore

Destinatari: VarGroup S.r.l., ByteHolders, Professori Tullio Vardanega e Riccardo Cardin

Scopo:

Nel glossario il gruppo Byte Holders si impegna a definire e spiegare i termini tecnici, acronimi e abbreviazioni utilizzati nel contesto del progetto richiesto dal committente VarGroup S.r.l., al fine di garantire una comprensione comune tra tutti gli stakeholder coinvolti.

Lettera di Candidatura

Autore: Responsabile

Destinatari: VarGroup S.r.l., Professori Tullio Vardanega e Riccardo Cardin

Scopo:

Con la lettera di candidatura il gruppo Byte Holders si propone come fornitore per la realizzazione del progetto descritto nel capitolato C2 - Code Guardian, presentando le proprie competenze, esperienze e motivazioni per intraprendere questa collaborazione con il committente VarGroup S.r.l.

Piano di Progetto

Autore: Responsabile

Destinatari: VarGroup S.r.l., Professori Tullio Vardanega e Riccardo Cardin

Scopo:

Nel Piano di Progetto il gruppo Byte Holders si impegna a definire le strategie, le risorse e le tempistiche necessarie per la realizzazione del progetto richiesto dal committente VarGroup S.r.l. garantendo il rispetto degli standard qualitativi e dei vincoli contrattuali.

Piano di Qualifica

Autore: Responsabile

Destinatari: VarGroup S.r.l., Professori Tullio Vardanega e Riccardo Cardin

Scopo:

Nel Piano di Qualifica il gruppo Byte Holders si impegna a definire le strategie, i criteri e le procedure di verifica e validazione che garantiranno che il prodotto software sviluppato soddisfi i requisiti specificati dal committente VarGroup S.r.l. e rispetti gli standard di qualità concordati.

3.1.2 Strumenti a supporto

Per il processo di fornitura il gruppo **BYTE HOLDERS** ha deciso di utilizzare i seguenti strumenti, suddivisi in due parti, una per la comunicazione interna al gruppo:

- **Discord:** Per riunioni principalmente vocali tra i membri del gruppo. Grazie alla possibilità di creare canali testuali specifici per ogni attività, viene utilizzato per tenere traccia di file e link utili allo sviluppo del progetto.
- **WhatsApp:** Per comunicazioni rapide e asincrone tra i membri del gruppo, utilizzato in oltre per tenere traccia degli appuntamenti previsti e per inviare promemoria.

La seconda sezione degli strumenti a supporto riguarda la comunicazione verso l'azienda proponente:

- **Teams:** Per riunioni vocali e video con il proponente VarGroup S.r.l., grazie alla sua integrazione con Outlook si ha anche la possibilità di pianificare riunioni direttamente dal calendario.

- **Slack:** Per la comunicazione rapida con l'azienda proponente, utilizzato principalmente per concordare riunioni.

3.2 Processo di Sviluppo

Il processo di sviluppo definisce il ciclo di vita ingegneristico attraverso il quale i requisiti del committente vengono trasformati nel prodotto software finale. Questo processo organizza in fasi strutturate le attività di analisi, progettazione, implementazione e validazione, garantendo un approccio sistematico e tracciabile alla realizzazione del sistema.

- **Attività Chiave (Fasi del Ciclo di Vita):**

1. **Analisi dei Requisiti:** Raccolta, analisi, specifica e validazione dei requisiti con gli stakeholder. Prodotto: *Documento di Analisi dei Requisiti*.
2. **Progettazione dell'Architettura:** Definizione dell'architettura di sistema e software ad alto livello.
3. **Progettazione di Dettaglio:** Definizione dettagliata dei componenti e delle interfacce.
4. **Costruzione (Implementazione e Codifica):** Scrittura del codice secondo gli standard di progetto e utilizzo di un sistema di versioning (Git). Prodotto: Codice sorgente e repository.
5. **Integrazione e Testing:**
 - Testing delle unità (sviluppatori)
 - Testing d'integrazione (team di integrazione)
 - Testing di sistema (team di qualità)

- **Metodologia:** Adotteremo pratiche ibride Agile, con sprint per lo sviluppo e milestone formali per le revisioni.

3.2.1 Analisi dei Requisiti

L'analisi dei requisiti è una fase cruciale del processo di sviluppo, in cui il gruppo **BYTE HOLDERS** si impegna a comprendere e documentare le esigenze del committente VarGroup S.r.l. Tale analisi è reperibile nel documento *Analisi Dei Requisiti*, che include:

- Requisiti funzionali e non funzionali
- Casi d'uso dettagliati
- Specifiche di interfaccia
- Vincoli tecnici e di progetto

3.2.2 Nomenclatura Casi d'Uso

Per garantire l'univocità e la facile identificazione delle funzionalità offerte dal sistema, ogni caso d'uso è identificato da un codice alfanumerico strutturato secondo la seguente sintassi:

UC[CodicePadre].[CodiceFiglio]

Dove:

- **UC**: Acronimo fisso che sta per *Use Case*;
- **CodicePadre**: Numero intero progressivo che identifica il caso d'uso principale (o padre);
- **CodiceFiglio** (Opzionale): Numero intero progressivo che identifica un sotto-caso d'uso (o figlio) incluso o esteso dal padre.

Esempi:

- *UC1*: Login (Caso d'uso principale);
- *UC1.1*: Recupero Password (Sotto-caso d'uso del Login).

3.2.3 Nomenclatura Requisiti

In conformità con la necessità di tracciabilità dei requisiti (processo di sviluppo software ISO/IEC 12207), ogni requisito è classificato e identificato tramite un codice univoco che ne esplicita la tipologia e la priorità.

La sintassi adottata è la seguente:

R[Tipologia][Importanza][Codice]

Dove le singole voci assumono i seguenti significati:

3.2.3.1 Tipologia

Indica la natura del requisito:

- **F**: *Funzionale* - Descrive una funzionalità o un comportamento del sistema;
- **Q**: *Qualitativo* (o Non Funzionale) - Descrive attributi di qualità come performance, affidabilità o sicurezza;
- **V**: *Vincolo* - Descrive limitazioni tecniche, normative o progettuali imposte dal committente o dall'ambiente;
- **P**: *Prestazionale* - Sottocategoria specifica per requisiti legati alle performance (opzionale, spesso accorpato in Q).

3.2.3.2 Importanza

Definisce la priorità di implementazione del requisito:

- **O**: *Obbligatorio* (Must have) - Requisito indispensabile per il funzionamento del sistema o richiesto esplicitamente dal vincolo contrattuale;
- **D**: *Desiderabile* (Should have) - Requisito che porta valore aggiunto ma la cui assenza non compromette le funzionalità base;
- **F**: *Facoltativo* (Could have/Nice to have) - Requisito opzionale da implementare solo se le risorse e le tempistiche lo permettono.

3.2.3.3 Codice

Un numero intero progressivo univoco che identifica il requisito (spesso gerarchico).

Esempi:

- *RFO1*: Requisito Funzionale Obbligatorio numero 1;
- *RQD12*: Requisito Qualitativo Desiderabile numero 12.

3.2.4 Codifica del codice sorgente

Questa sezione ha l'obiettivo di definire le linee guida per la scrittura di codice sorgente in maniera chiara, mantenibile e coerente all'interno del progetto sviluppato dal gruppo **BYTE HOLDERS**.

Le seguenti convenzioni di codifica sono adottate per garantire uniformità e qualità del codice:

- **Lingua**: I nomi delle variabili, funzioni ed eventuali classi devono essere scritti in inglese.
- **Funzioni**: Realizzare funzioni con una singola responsabilità, evitando funzioni troppo lunghe o complesse, l'utilizzo di funzioni specifiche per ogni compito le rende riutilizzabili e facilmente testabili.
- **Commenti**: Utilizzare commenti per spiegare il funzionamento del codice, specialmente in sezioni complesse o non intuitive. I commenti devono essere chiari e concisi, evitando di commentare cose facilmente interpretabili dal codice.
- **Indentazione e Spaziatura**: Utilizzare un'indentazione coerente (ad esempio 2 o 4 spazi) per migliorare la leggibilità del codice. Mantenere una spaziatura adeguata tra le sezioni di codice per separare logicamente i blocchi.

Relativamente al posizionamento dei file all'interno della repository GitHub, il gruppo **BYTE HOLDERS** ha deciso di adottare la seguente struttura:

- **Cartelle**: Ogni macro funzionalità del progetto avrà una propria cartella dedicata, contenente tutti i file correlati a quella funzionalità. (Es. Cartelle per Frontend e backend separate)
- DA FINIRE

3.2.5 Strumenti di supporto

Per il processo di sviluppo il gruppo **BYTE HOLDERS** ha deciso di utilizzare i seguenti strumenti:

- **Visual Studio Code**: per la scrittura del codice sorgente, scelto per la sua versatilità e supporto a numerose estensioni.
- **Draw.io**: per la realizzazione dei diagrammi UML e altre rappresentazioni grafiche necessarie per la realizzazione degli use-case pubblicati nel documento di Analisi Dei Requisiti e per progettazione del software finale.

4 Processi Di Supporto

Questa sezione descrive i processi di supporto secondo lo standard ISO/IEC 12207, che forniscono l'infrastruttura tecnica e metodologica necessaria per garantire qualità e coerenza nell'esecuzione dei processi primari. Tali processi, di natura trasversale, abilitano e migliorano le attività di sviluppo, gestione e verifica attraverso strumenti, procedure e controlli dedicati.

4.1 Documentazione

La documentazione del progetto viene gestita utilizzando **Latex** e viene archiviata in un repository dedicato su **GitHub**.

Ogni documento segue una struttura standardizzata per garantire coerenza e facilità di lettura.

4.1.1 Struttura dei documenti

Tutti i documenti formali prodotti dal gruppo seguono una struttura standardizzata per garantire coerenza, professionalità e facilità di consultazione. Il template **Latex** utilizzato definisce chiaramente gli elementi che compongono ciascun documento:

- **Frontespizio:**

- Logo del gruppo Byte Holders e titolo del progetto
- Nome del documento (es. "Norme di Progetto", "Verbale Interno")
- Informazioni di identificazione: data, autore, verificatore e approvatore

- **Registro delle Versioni:**

- Tabella posizionata dopo l'indice
- Tracciamento cronologico inverso (LIFO) di tutte le modifiche
- Colonne per versione, data, autore e descrizione delle modifiche

- **Indice:**

- Elenca tutte le sezioni, sottosezioni e relative pagine
- Fornisce una mappa navigabile del documento

- **Contenuto Principale:**

- Organizzato gerarchicamente in sezioni, sottosezioni e sotto-sottosezioni

- **Stile Grafico Uniforme:**

- Palette di colori definita dal gruppo (blu-viola in diverse tonalità)
- Font sans-serif (Helvetica) per migliorare la leggibilità
- Margini standardizzati

Questa struttura si applica a tutti i tipi di documenti prodotti, tra cui: Verbali(interni ed esterni), Norme di Progetto, Analisi dei Requisiti. . .

4.1.2 Tabella Decisioni e Azioni

Ogni verbale relativo al progetto conterrà una tabella decisioni e azioni, che riassume le decisioni prese durante la riunione e le azioni assegnate ai membri del team. Anche in questo caso, per garantire la tracciabilità, ogni decisione o azione avrà un proprio codice di identificazione, composto da un prefisso che ne indica la natura (DEC per decisione, AZ per azione); seguito dalla fase del progetto in cui è stata presa (RTB o PB); e infine un numero progressivo che identifica univocamente la decisione o l'azione all'interno di quella fase.

Tutto ciò ci permette di tenere traccia delle decisioni e delle azioni in modo chiaro e organizzato, facilitando il monitoraggio del progresso del progetto e la responsabilizzazione dei membri del team.

Qui sotto è riportato un esempio di tabella decisioni e azioni:

Codice	Descrizione	Assegnatario
DEC-RTB-001	Scelta del capitolato	Tutti
DEC-PB-003	Delineamento dell'analisi dei requisiti	Tutti
AZ-RTB-001	Brainstorming Analisi dei requisiti	Tutti
AZ-PB-004	Miglioramento struttura dei documenti	Tutti

4.1.3 Diari di bordo

Nell'ambito delle lezioni di **Ingegneria del Software**, il prof. Tullio Vardanega ha organizzato, tramite il calendario delle lezioni, delle date per l'attività denominata "Diario di bordo". Questa attività prevede che per ogni gruppo ci sia un membro designato che rediga un breve resoconto dei progressi fatti fino alla data di presentazione del diario di bordo. Il diario di bordo deve includere:

- Risultati raggiunti e confronto con previsioni all'inizio dello sprint.
- Obiettivi e attività previste per il periodo successivo, fino al diario di bordo seguente.
- Difficoltà riscontrate e problemi ancora da risolvere.

4.1.4 Strumenti a supporto

Per la documentazione del progetto il gruppo **BYTE HOLDERS** utilizza i seguenti strumenti:

- **Linguaggio \LaTeX** : adottato come standard per la stesura di documentazione tecnica; garantisce un'alta qualità tipografica, uniformità stilistica e una gestione avanzata di riferimenti incrociati e bibliografia, separando il contenuto dalla formattazione.
- **Visual Studio Code**: IDE (Integrated Development Environment) utilizzato come editor principale per la scrittura del codice sorgente \LaTeX , scelto per la sua leggerezza, estensibilità e integrazione nativa con i sistemi di controllo versione.
- **LaTeX Workshop**: estensione per Visual Studio Code che automatizza il processo di compilazione e build dei documenti; fornisce funzionalità di linting (controllo sintattico),

intellisense e anteprima PDF sincronizzata (SyncTeX) per un feedback visivo immediato durante la stesura.

4.2 Gestione della configurazione

L'obiettivo di questa sezione è definire l'approccio del gruppo **BYTE HOLDERS** per la gestione della configurazione della documentazione di progetto, al fine di garantire coerenza, tracciabilità e controllo delle modifiche. In questo caso, come specificato nella sezione degli strumenti di supporto, il gruppo utilizza **GitHub** come piattaforma principale per il versionamento e la gestione della configurazione dei documenti.

In particolare è stato deciso di realizzare due repository distinte per l'RTB:

- **Repository Documentazione:** Contiene tutti i documenti formali del progetto, come Norme di Progetto, Verballi, Analisi dei Requisiti, Piano di Progetto e altri documenti chiave. Ogni documento è organizzato in cartelle specifiche per facilitarne la navigazione.
- **Repository Codice Sorgente PoC:** Contiene il codice sorgente della Proof of Concept (PoC) sviluppata durante l'RTB. Questa repository include il codice, gli script di build, i file di configurazione e la documentazione tecnica correlata alla PoC.

4.2.1 Produzione e archiviazione

Il gruppo **BYTE HOLDERS** adotta una strategia ben definita per la produzione e l'archiviazione della documentazione di progetto, al fine di garantire accessibilità, tracciabilità e sicurezza delle informazioni. La produzione di un documento segue questi passaggi:

1. **Creazione Issue e Branch dedicato:** A seguito dell'apertura di una Issue su GitHub, viene creato un branch di lavorazione dedicato che si distacca dal ramo principale (*main*). La nomenclatura del branch deve seguire rigorosamente la convenzione: `azione_nome_documento_data` (es. `aggiunta_verbale_interno_11_12_2025`).
2. **Assegnazione e Stesura:** Il membro assegnato alla Issue lavora localmente sul file `.tex`. Durante questa fase vengono effettuati commit frequenti sul branch dedicato per salvare i progressi, utilizzando Visual Studio Code e LaTeX Workshop per la scrittura e il controllo sintattico in tempo reale.
3. **Verifica (Pull Request):** Completata la stesura, viene aperta una *Pull Request* su GitHub. In questa fase, i verificatori controllano il contenuto (accuratezza e completezza) e la forma (corretta compilazione LaTeX), segnalando eventuali correzioni direttamente nel codice o nei commenti.
4. **Approvazione e Merge:** Superata la fase di verifica, il Responsabile o un membro designato approva il documento e di seguito la Pull Request ed esegue il *merge* (unione) del branch dedicato nel ramo *main*. A questo punto, il documento aggiornato è considerato ufficiale e viene compilata la versione PDF definitiva per il rilascio.

4.2.2 Versionamento della documentazione

Il versionamento dei documenti è un processo fondamentale per garantire la tracciabilità delle modifiche, la collaborazione ordinata tra i membri del team e la chiara identificazione dello stato corrente di ciascun documento.

Per questo motivo, il gruppo **Byte Holders** adotta un sistema di numerazione semantica e un

registro strutturato delle modifiche.

Per la gestione del versionamento, abbiamo deciso di usare uno schema a tre livelli, X.Y.Z, dove:

- **X (Approvazione):** Incrementato per cambiamenti significativi che richiedono l'approvazione formale. E quindi il raggiungimento di un documento stabile e definitivo.
- **Y (Major Update):** Incrementato per l'aggiunta di nuove sezioni o modifiche sostanziali che alterano la struttura complessiva del documento. Il documento non è ancora definitivo.
- **Z (Minor Update):** Incrementato per correzioni minori, come errori di battitura o aggiornamenti di dettaglio.

4.2.3 Visualizzazione della documentazione

Per facilitare la consultazione e la fruizione della documentazione di progetto, il gruppo **BYTE HOLDERS** ha implementato un sistema di visualizzazione basato su GitHub Pages. Questo sistema, inserito nella repository dedicata alla documentazione, consente di visualizzare tutta la documentazione presente nel branch main della repository, sotto forma di pdf. Il sito web è suddiviso in 4 sezioni principali:

- **Candidatura:** Contiene tutti i documenti relativi alla fase di candidatura e risposta al capitolato.
- **RTB:** Contiene tutti i documenti prodotti durante la fase di RTB.
- **PB:** Contiene tutti i documenti prodotti durante la fase di PB, alcuni di questi documenti sono versioni aggiornate di documenti già presenti durante la fase di RTB.
- **Diari di bordo:** Contiene tutti i diari di bordo prodotti durante il corso del progetto.

Oltre alle 4 sezioni elencate sono presenti dettagli relativi ai membri del gruppo e email per eventuali comunicazioni.

4.2.4 Strumenti a supporto

Per la documentazione del progetto il gruppo **BYTE HOLDERS** utilizza i seguenti strumenti:

- **GitHub:** Piattaforma per il versionamento e collaborazione, utilizzata per la gestione del codice sorgente e della documentazione..
- **GitHub Pages:** Servizio che permette di pubblicare il contenuto della repository su un sito web statico, utilizzato per visualizzare i documenti in formato PDF, come descritto nel punto 4.2.3.

4.3 Garanzia della Qualità (Quality Assurance)

Il gruppo **Byte Holders** implementa il processo di Garanzia della Qualità in conformità con il processo 6.3 dello standard ISO/IEC 12207:1995. Questo processo ha l'obiettivo di fornire un'assicurazione oggettiva che i prodotti software e i processi del ciclo di vita siano conformi ai requisiti specificati e ai piani stabiliti. Le attività includono:

- **Controllo di Processo:** Verifica periodica che le attività svolte dai membri del team siano conformi alle procedure descritte nel Piano di Progetto e nel Piano di Qualifica.
- **Verifica degli Standard:** Ogni documento e artefatto viene controllato per assicurare il rispetto delle convenzioni di formattazione, nomenclatura e versionamento.
- **Indipendenza della valutazione:** Per garantire l'oggettività, l'attività di **Quality Assurance** è svolta, se possibile, da membri non direttamente coinvolti nella creazione del documento o sezione di codice.

4.4 Verifica

In accordo con il processo 6.4 dello standard ISO/IEC 12207:1995, il processo di Verifica ha lo scopo di determinare se i prodotti di una determinata fase dello sviluppo soddisfano le condizioni imposte all'inizio di quella fase.

Le attività di verifica adottate dal gruppo **Byte Holders** comprendono:

- **Walkthrough e Inspection:** Revisioni interne approfondite sui documenti (Analisi dei Requisiti, Specifica Tecnica) e sul codice sorgente per identificare anomalie, discrepanze o errori logici prima di procedere alla fase successiva.
- **Analisi Statica:** Utilizzo di strumenti automatici per la verifica della qualità del codice (es. controllo dello stile, rilevamento di potenziali bug) senza l'esecuzione del programma.
- **Tracciabilità:** Verifica che ogni requisito software sia correttamente tracciato e coperto nelle fasi di progettazione e implementazione.

4.5 Validazione

Il processo di Validazione, conforme al processo 6.5 dello standard ISO/IEC 12207:1995, assicura che il prodotto finale soddisfi l'uso specifico previsto e le esigenze dell'utente.

Le attività di validazione includono:

- **Test di Sistema:** Esecuzione del software completo per verificare che rispetti i requisiti funzionali e non funzionali specificati nel documento di Analisi dei Requisiti.
- **Test di Accettazione (UAT):** Coinvolgimento del committente per dimostrare che il sistema è pronto per l'uso operativo. Questa fase corrisponde all'acquisizione del feedback formale da parte del proponente.
- **Validazione Documentale:** Conferma che la documentazione prodotta (Manuali Utente, Manuali Sviluppatore) sia comprensibile, completa e utilizzabile dal pubblico di destinazione.

5 Processi Organizzativi

Questa sezione descrive i processi organizzativi, definiti dallo standard ISO/IEC 12207, che regolano le modalità e gli strumenti di coordinamento utilizzati dal gruppo, ovvero come vengono gestite le persone, i ruoli, le comunicazioni e attraverso quali strumenti ciò avviene. Inoltre lo scopo di questa sezione è anche quello di stabilire procedure e metodologie che il team seguirà durante il ciclo di vita del progetto e il processo di sezione è anche quello di stabilire procedure e metodologie che il team seguirà durante il ciclo di vita del progetto e il processo di miglioramento continuo dei processi.

5.1 Gestione dei processi

Il gruppo adotta una metodologia Agile, organizzando il lavoro in Sprint della durata di due settimane. Ogni Sprint inizia con una riunione di pianificazione, in cui vengono definiti gli obiettivi e le attività da svolgere durante lo Sprint e finisce con una riunione in cui viene fatta una retrospettiva.

5.2 Gestione dei ruoli

Ogni membro del gruppo in ogni Sprint può assumere diversi ruoli, in base alle attività che dovrà svolgere durante quel periodo. L'unico ruolo unico all'interno di uno Sprint è quello di Responsabile, che viene assegnato ad un membro diverso ad ogni Sprint. Gli altri ruoli possono essere assegnati a più membri contemporaneamente, in base alle necessità del progetto. I ruoli sono ruotati in modo che per fine del progetto ogni membro abbia ricoperto tutti i ruoli almeno una volta.

5.2.1 Responsabile

Il responsabile analizzando lo stato di avanzamento del progetto ha il compito di pianificare ed individuare le attività da svolgere per raggiungere gli obiettivi di progetto e organizzare le risorse in modo da rispettare tempi e costi preventivati.

Ha il compito di verificare che tutti i componenti del gruppo portino a termine le attività a loro assegnate nei tempi previsti.

Deve analizzare i possibili rischi e gestire problemi ed imprevisti in modo da raggiungere gli obiettivi del progetto.

Si occupa dei rapporti e delle comunicazioni con l'azienda proponente.

Le attività che il responsabile svolge sono:

- **Ad inizio sprint:** definire gli obiettivi dello sprint, individuare i possibili rischi, suddividere le attività e assegnarle ai membri del gruppo, controllando che ad ogni membro sia assegnato un numero coerente di ore per ogni ruolo in base ai compiti che dovrà svolgere
- **Durante lo sprint:** dovrà assicurarsi che le attività vengano svolte nei tempi previsti, monitorare i rischi individuati, cercando di risolvere eventuali problemi che dovessero sorgere, e gestire le comunicazioni con l'azienda proponente
- **A fine sprint:** guidare l'attività di retrospettiva dello sprint, capendo le difficoltà riscontrate e cercando di trovare una possibile soluzione, inoltre verifica che gli obiettivi siano stati raggiunti
- **Redazione piano di progetto:** dovrà redigere e aggiornare il piano di progetto con gli obiettivi, le ore preventivate, i rischi attesi, le ore effettive, i rischi incontrati e le osservazioni uscite durante la retrospettiva dello sprint, inoltre si occupa della gestione dei rischi
- **Approvazione documenti:** ha il compito di approvare i documenti prodotti dal gruppo prima della loro pubblicazione OPPURE approvare i verbali e verificare il glossario e le norme di progetto

5.2.2 Amministratore

L'Amministratore si occupa di gestire e migliorare l'ambiente di lavoro, overro si occupa tenere aggiornati e migliorare l'infrastruttura tecnologica e organizzativa (gli strumenti di sviluppo, verifica e validazione, i sistemi di archiviazione e versionamento della documentazione e del codice).

Le attività che l'amministratore svolge sono:

- **Redazione norme di progetto:** si occupa di redigere e aggiornare le norme di progetto, in modo da garantire che tutti i membri del gruppo seguano le stesse regole e procedure.
- **Redazione piano di qualifica:** si occupa di redigere e aggiornare il piano di qualifica, in modo da garantire che tutti i membri del gruppo seguano le stesse regole e procedure per la verifica e validazione del prodotto
- **Redazione piano dei verbali:** si occupa di redigere i verbali delle riunioni interne ed esterne, in modo da garantire che tutte le decisioni prese durante le riunioni siano documentate e tracciabili
- **Automazione e Miglioramento dei Processi:** si occupa di automatizzare i processi, individuare punti di miglioramento nei processi e mettere in opera risorse per migliorare l'ambiente di lavoro
- **Gestione delle Infrastrutture:** si occupa di gestire l'infrastruttura e gli strumenti utilizzati, mantenere efficiente l'ambiente di sviluppo e fornire strumenti adeguati ai membri del gruppo e gestire errori e segnalazioni di malfunzionamenti con gli strumenti tecnologici

5.2.3 Analista

L'analista si occupa della raccolta e dell'analisi dei requisiti, interagendo con il proponente per chiarimenti. Le attività che l'analista svolge sono:

- **Raccolta Requisiti:** Interagire con l'azienda proponente per comprendere e documentare i requisiti funzionali e non funzionali del sistema e la loro importanza (obbligatori, desiderabili, facoltativi)
- **Redazione Analisi dei Requisiti:** redigere il documento di analisi dei requisiti, valutandone la fattibilità e le implicazione dei requisiti attraverso la stesura dei casi d'uso che li soddisfano assicurando chiarezza e completezza. Valutare la fattibilità tecnica e le implicazioni dei requisiti raccolti Collaborare con il team di verifica per assicurare che i requisiti siano testabili e verificati correttamente
- **Redazione Glossario:** redigere e aggiornare il glossario del progetto, definendo i termini tecnici e le abbreviazioni utilizzate nel contesto del progetto

5.2.4 Progettista

Il progettista, partendo dai requisiti definiti nel documento di analisi dei requisiti, definisce il design architetturale del sistema. Le attività che il progettista svolge sono:

- **Progettazione Architettuale:** definire l'architettura di alto livello del sistema, identificando i componenti principali e le loro interazioni
- **Progettazione di Dettaglio:** sviluppare specifiche dettagliate per ciascun componente, incluse le interfacce e i protocolli di comunicazione

- **Redazione Specifiche tecniche:** redigere il documento di specifiche tecniche in cui si riportano le scelte architetturali e tecnologiche (design pattern adottati, schema UML delle classi, tecnologie usate)

5.2.5 Sviluppatore

Lo sviluppatore si occupa di sviluppare il software/implementa le funzionalità del software secondo le specifiche progettuali. Le attività che lo sviluppatore svolge sono:

- **Scrittura del Codice:** scrivere codice che segui le specifiche ottenute durante la progettazione seguendo le linee guida (best practices) di codifica stabilite
- **Test delle Unità:** sviluppare e eseguire test unitari per garantire che ogni componente funzioni correttamente in isolamento (NON SONO SICURA se sia il programmatore o il verificatore)
- **Documentazione del Codice:** mantenere una documentazione aggiornata del codice, per facilitarne la comprensione e la manutenzione

5.2.6 Verificatore

Il verificatore si assicura che i documenti siano corretti, verifica il corretto funzionamento del software attraverso la scrittura di test e verifica della qualità del prodotto, assicurandosi che soddisfatti i requisiti. Le attività che il verificatore svolgere sono:

- **Test del software:** eseguire e implementare test funzionali, di integrazione e di sistema per identificare difetti e garantire la conformità ai requisiti
- **Bug reporting:** registrare e tracciare i difetti riscontrati durante i test
- **Verifica dei Documenti:** verifica i verbali delle riunioni interne ed esterne, il piano di progetto (?oppure amministratore) e il piano di qualifica

5.3 Riunioni

Le riunioni si dividono in interne ed esterne. A fine delle quali l'amministratore (scelto a giro tra i membri del gruppo che non svolgono attualmente la carica di responsabile) redige un verbale in cui vengono documentate le decisioni prese, le azioni assegnate e i punti discussi. Tale verbale viene in seguito verificato da un verificatore e approvato dal responsabile; se si tratta di un verbale esterno viene mandato all'azienda proponente per la firma e l'approvazione.

5.3.1 Riunioni interne

Le riunioni interne sono incontri tra i membri del gruppo che avvengono almeno una volta a settimana, solitamente il martedì pomeriggio. Una riunione interna nella maggior parte dei casi si svolge a distanza in modalità virtuale e si svolge nel seguente modo:

- **Apertura:** viene presentando l'ordine del giorno, deciso solitamente durante la riunione precedente e aggiornato durante la settimana in base alle problematiche e i dubbi emersi
- **Aggiornamenti sullo stato del progetto:** ogni membro del gruppo aggiorna gli altri membri riguardo il lavoro svolto, le problematiche e dubbi incontrati

- **Discussione delle problematiche:** si discutono le problematiche e i dubbi emersi e si cercano delle soluzioni condivise
- **Pianificazione delle attività future:** si pianificano le attività da svolgere fino alla prossima riunione, assegnando compiti specifici ai membri del gruppo
- **Chiusura:** si chiude la riunione, riassumendo le decisioni prese e le azioni da intraprendere

5.3.2 Riunioni esterne DA RIVEDERE

Le riunioni esterne sono incontri tra i membri del gruppo e l'azienda proponente (VarGroup S.r.l.) che avvengono su richiesta del gruppo o dell'azienda e dal 7 gennaio 2026 avvengono bisettimanalmente. Una riunione esterna si svolge solitamente in modalità virtuale tramite Microsoft Teams oppure in presenza presso la sede di VarGroup e serve per presentare il lavoro svolto, i dubbi riacontrati e ricevere feedback e consigli tecnici dall'azienda proponente.

5.4 Gestione delle comunicazioni

I canali di comunicazione usati per la comunicazione interna tra membri del gruppo e la comunicazione con l'azienda proponente sono diversi.

5.4.1 Comunicazioni interne

Per la comunicazione interna tra i membri del gruppo vengono utilizzati principalmente due canali:

- **Discord:** utilizzato per la comunicazione sincrona tra i membri del gruppo (riunioni interne) e la condivisione di materiale di riferimento
- **WhatsApp:** utilizzato per comunicazioni asincrona tra i membri del gruppo, soprattutto per questioni organizzative e logistiche (organizzazione delle riunioni, aggiornamento sullo stato del verbale, messaggi per esterni).

5.4.2 Comunicazioni esterne

Per la comunicazione con l'azienda proponente (VarGroup S.r.l.) vengono utilizzati principalmente due canali:

- **Slack:** utilizzato per la comunicazione asincrona con l'azienda proponente, per concordare riunioni, chiarire dei dubbi, scambiare materiale
- **Microsoft Teams:** utilizzato per le riunioni virtuali (comunicazione sincrona) con l'azienda proponente
- **Email:** BHO...scambio link riunioni utilizzata per inviare documenti ufficiali all'azienda proponente, come i verbali delle riunioni esterne e la documentazione di progetto

5.5 Gestione delle infrastrutture e degli strumenti

Per la gestione delle infrastrutture e degli strumenti il gruppo utilizza i seguenti strumenti:

- **GitHub:** github actions per automatizzare verbali, dashboard kanban, utilizzato per il versionamento del codice sorgente e della documentazione di progetto, tramite repository dedicati
- **Git:** utilizzato come sistema di controllo versione distribuito per gestire le modifiche al codice sorgente e alla documentazione
- **Visual Studio Code:** utilizzato come IDE principale per lo sviluppo del codice sorgente e la scrittura della documentazione in LaTeX
- **LaTeX Workshop:** estensione di Visual Studio Code utilizzata per la compilazione e il controllo sintattico dei documenti in LaTeX
- **Discord:** utilizzato come piattaforma di comunicazione interna tra i membri del gruppo
- **Slack:** utilizzato come piattaforma di comunicazione con l'azienda proponente
- **Microsoft Teams:** utilizzato per le riunioni virtuali con l'azienda proponente
- **WhatsApp:** utilizzato per la comunicazione rapida e informale tra i membri del gruppo
- **Gmail:** utilizzata per inviare documenti ufficiali all'azienda proponente
- **Google Documents:** utilizzato per la collaborazione in tempo reale su documenti condivisi tra i membri del gruppo
- **Google fogli:** utilizzato per la gestione e l'analisi dei dati condivisi tra i membri del gruppo
- **Draw.io:** utilizzato per la creazione di diagrammi e schemi UML per la documentazione di progetto
- **typescript:** utilizzato come linguaggio di programmazione principale per lo sviluppo del software
- **Node.js:** utilizzato come ambiente di esecuzione per il codice TypeScript
- **AWS:** utilizzato per l'hosting e la gestione dell'infrastruttura cloud del progetto
- **Docker:** utilizzato per la containerizzazione delle applicazioni e la gestione degli ambienti di sviluppo e produzione

5.6 Processo di miglioramento

La fase di miglioramento dei processi ha l'obiettivo di garantire l'evoluzione continua del modo di lavorare del gruppo, aumentando l'efficacia e l'efficienza delle attività svolte e riducendo il rischio di errori organizzativi e operativi. Tale processo consente di individuare criticità, valutare le soluzioni adottate e aggiornare, se necessario, le norme e le procedure di progetto.

Il gruppo segue il principio di miglioramento continuo, ovvero il processo di miglioramento è visto come un processo continuo e iterativo basato su revisioni periodiche dei processi e delle pratiche adottate. Il momento principale di revisione avviene durante il momento di retrospettiva alla fine di ogni sprint, in cui si analizzano le difficoltà incontrate, le soluzioni adottate e si identificano possibili miglioramenti.

Il processo adottato segue il ciclo di miglioramento continuo definito dal modello PDSA (Plan-Do-Study-Act) e si articola nelle seguenti 4 fasi:

- **Plan (Pianificazione):** vengono individuate le criticità, inefficienze o opportunità di miglioramento emerse durante lo svolgimento delle attività; vengono analizzate le cause e vengono identificate delle possibili azioni correttive o migliorative.
- **Do (Esecuzione):** le azioni correttive o migliorative individuate vengono implementate e messe in pratica durante lo svolgimento delle attività
- **Study (Analisi):** viene analizzato l'impatto delle azioni implementate, valutando se hanno portato ai miglioramenti attesi e se hanno risolto le criticità individuate
- **Act (Azione):** viene deciso se adottare definitivamente le azioni implementate, modificarle ulteriormente o tornare alla fase di pianificazione per individuare nuove soluzioni

5.7 Formazione

Prima formazione fornita dall'azienda proponente VarGroup S.r.l. seguita poi da uno studio personale in cui si predilige l'apprendimento pratico attraverso piccoli esempi giocattolo.

6 Metriche e standard per la Qualità

Per garantire la massima qualità del prodotto e del processo, il gruppo ha deciso di adottare lo standard **ISO/IEC 25010:2023** (System and software quality models), l'ultima iterazione della serie SQuaRE (Systems and software Quality Requirements and Evaluation).

Questo standard scompone la qualità del prodotto software in otto caratteristiche principali:

- **Adeguatezza Funzionale** (Functional Suitability);
- **Efficienza Prestazionale** (Performance Efficiency);
- **Compatibilità** (Compatibility);
- **Usabilità** (Usability);
- **Affidabilità** (Reliability);
- **Sicurezza** (Security);
- **Manutenibilità** (Maintainability);
- **Portabilità** (Portability).

Ogni metrica è identificata dal codice **M[Tipo][ID]**, dove *Tipo* è **PC** (Processo) o **PD** (Prodotto).

7 Metriche di Qualità del Processo

Le metriche di processo mirano a monitorare l'andamento delle attività, i costi e la stabilità dei requisiti durante il ciclo di vita (rif. ISO/IEC 12207).

7.1 Processi Primari

7.1.1 Schedule Performance Index (SPI)

Codice: MPC01

• **Formula:**

$$SPI = \frac{EV}{PV}$$

Dove:

- *EV* (Earned Value): Valore monetario/orario del lavoro realizzato;
- *PV* (Planned Value): Valore monetario/orario del lavoro pianificato.

• **Descrizione:** Indica l'efficienza della pianificazione temporale.

• **Soglie:**

- $SPI \geq 1$: Progetto in pari o in anticipo (Ottimo);
- $0.9 \leq SPI < 1$: Ritardo gestibile (Accettabile);
- $SPI < 0.9$: Ritardo grave (Critico).

7.1.2 Cost Performance Index (CPI)

Codice: MPC02

• **Formula:**

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

Dove *AC* (Actual Cost) è il costo reale sostenuto (ore spese).

• **Descrizione:** Indica l'efficienza dei costi rispetto al budget preventivato. Un valore < 1 indica che si stanno spendendo più risorse del previsto.

7.1.3 Requirements Stability Index (RSI)

Codice: MPC03

• **Formula:**

$$RSI = \left(1 - \frac{R_{var}}{R_{tot}} \right) \times 100$$

Dove R_{var} è il numero di requisiti aggiunti, modificati o cancellati e R_{tot} il totale iniziale.

• **Descrizione:** Misura la stabilità dei requisiti. Variazioni frequenti impattano negativamente sulla progettazione.

7.2 Processi di Supporto

7.2.1 Code Coverage (Test di Unità)

Codice: MPC04

• **Formula:**

$$Coverage = \frac{LOC_{covered}}{LOC_{total}} \times 100$$

• **Descrizione:** Percentuale di linee di codice (LOC) eseguite durante i test automatizzati.

• **Obiettivo:** $\geq 80\%$.

7.2.2 Indice di Gulpease (Leggibilità Documenti)

Codice: MPC05

- **Formula:**

$$G = 89 + \frac{300 \times Frasi - 10 \times Lettere}{Parole}$$

- **Descrizione:** Valuta la leggibilità di un testo in lingua italiana.
- **Obiettivo:** $G \geq 40$ (Leggibilità media/buona per testi tecnici).

8 Metriche di Qualità del Prodotto (ISO/IEC 25010:2023)

8.1 Adeguatezza Funzionale

Capacità del prodotto di fornire funzioni che soddisfano le esigenze dichiarate e implicite quando utilizzato in condizioni specificate.

8.1.1 Copertura Funzionale

Codice: MPD01

- **Formula:**

$$CF = \frac{Requisiti_{impl}}{Requisiti_{tot}} \times 100$$

- **Descrizione:** Percentuale di requisiti funzionali (Obbligatori/Desiderabili) effettivamente implementati e superati nei test di sistema.
- **Obiettivo:** 100% per i requisiti Obbligatori alla release finale.

8.2 Affidabilità

Grado in cui un sistema, prodotto o componente svolge le funzioni specificate sotto condizioni determinate per un periodo di tempo stabilito.

8.2.1 Mean Time To Repair (MTTR)

Codice: MPD02

- **Formula:** Media del tempo trascorso tra la rilevazione di un guasto e la sua risoluzione (fix + deploy).
- **Descrizione:** Indica la capacità del team e del sistema di ripristinare il servizio (Recoverability).

8.3 Efficienza Prestazionale

Prestazioni relative alla quantità di risorse utilizzate sotto condizioni determinate.

8.3.1 Average Response Time

Codice: MPD03

- **Formula:**

$$\frac{\sum Temp_i_{risposta}}{N_{richieste}}$$

- **Descrizione:** Tempo medio che intercorre tra l'invio della richiesta e la ricezione della risposta completa.
- **Obiettivo:** < 0.5s per API standard; < 2s per operazioni complesse.

8.4 Compatibilità

Grado in cui un prodotto, sistema o componente può scambiare informazioni con altri prodotti (Interoperabilità) e svolgere le sue funzioni condividendo lo stesso ambiente hardware/software (Coesistenza).

8.4.1 Browser Compatibility Rate

Codice: MPD04

- **Formula:**

$$BCR = \frac{Browser_{supportati}}{Browser_{testati}} \times 100$$

- **Descrizione:** Percentuale di browser (Chrome, Firefox, Safari, Edge) in cui il software non presenta anomalie grafiche o funzionali.

8.5 Usabilità

Grado in cui un prodotto o sistema può essere utilizzato da specifici utenti per raggiungere specifici obiettivi con efficacia, efficienza e soddisfazione.

8.5.1 Average Error Rate (AER)

Codice: MPD05

- **Formula:**

$$AER = \frac{N_{errori_utente}}{N_{azioni_totali}} \times 100$$

- **Descrizione:** Percentuale di errori commessi dall'utente durante l'esecuzione di un task specifico (misura l'intuitività dell'interfaccia).

8.6 Sicurezza

Grado in cui il prodotto protegge informazioni e dati in modo che persone o altri sistemi abbiano il grado di accesso ai dati appropriato al loro livello di autorizzazione.

8.6.1 Vulnerability Density

Codice: MPD06

- **Formula:**

$$VD = \frac{Vulnerabilita_rilevate}{KLOC}$$

- **Descrizione:** Numero di vulnerabilità di sicurezza (secondo OWASP Top 10) rilevate tramite analisi statica per 1000 righe di codice.
- **Obiettivo:** 0 Vulnerabilità di livello High/Critical.

8.7 Manutenibilità

Grado di efficacia ed efficienza con cui il prodotto può essere modificato dagli sviluppatori.

8.7.1 Cyclomatic Complexity (Complessità Ciclomatica)

Codice: MPD07

- **Formula:**

$$V(G) = E - N + 2P$$

- **Descrizione:** Misura la complessità logica del codice. Funzioni con $V(G) > 10$ sono considerate difficili da mantenere e testare.
- **Obiettivo:** Mantenere la complessità media < 10 .

8.8 Portabilità

Grado di efficacia ed efficienza con cui un sistema, prodotto o componente può essere trasferito da un ambiente operativo ad un altro (es. Docker deploy).

8.8.1 Environment Deployment Success

Codice: MPD08

- **Formula:**

$$EDS = \frac{Deploy_riusciti}{Deploy_tentati} \times 100$$

- **Descrizione:** Percentuale di installazioni/deploy andati a buon fine al primo tentativo su diversi ambienti (Locale, CI/CD, Produzione).