



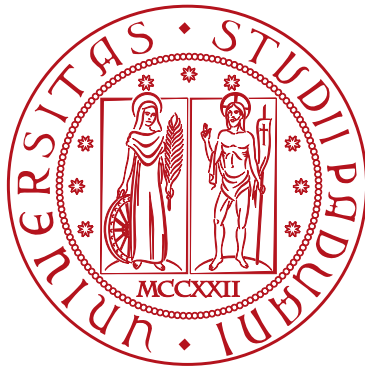
PEBKAC

Gruppo: 11

Email: pebkacswe@gmail.com

Docs: <https://pebkac-swe-group-11.github.io>

GitHub: <https://github.com/PEBKAC-SWE-Group-11>



Università degli Studi di Padova

Corso di Laurea: Informatica

Corso: Ingegneria del Software

Anno Accademico: 2024/2025

Norme di progetto

Informazioni sul documento:

Uso	Interno
Destinatari	Tullio Vardanega Riccardo Cardin Gruppo PEBKAC

Registro delle versioni

Versione	Data	Autore	Ruolo	Descrizione
0.2.0	2024/12/17	Tommaso Zocche	Verificatore	Verifica parziale
0.1.5	2024/12/17	Tommaso Zocche	Verificatore	Correzioni “strutturali” e typo
0.1.4	2024/12/5	Matteo Piron	Amministratore	§6 Metriche di qualità
0.1.3	2024/11/25	Derek Gusatto	Amministratore	§4.2 Configuration Management (struttura e scopo)
0.1.2	2024/11/22	Derek Gusatto	Amministratore	§3.2.2 Analisi dei requisiti
0.1.1	2024/11/20	Derek Gusatto	Amministratore	§3.1 Fornitura
0.1.0	2024/11/19	Alessandro Benin	Verificatore	Verifica parziale
0.0.3	2024/11/17	Derek Gusatto	Amministratore	§5.1 Gestione organizzativa
0.0.2	2024/11/16	Derek Gusatto	Amministratore	§4.1 Documentazione
0.0.1	2024/11/14	Derek Gusatto	Amministratore	§1 Introduzione, §2 Standard ISO/IEC 12207:1195

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	Standard ISO/IEC 12207:1995	8
2.1	Processi del ciclo di vita del software	8
2.1.1	Processi primari	8
2.1.2	Processi di supporto	8
2.1.3	Processi organizzativi	9
2.1.4	Ruoli	9
3	Processi Primari	10
3.1	Fornitura	10
3.1.1	Scopo	10
3.1.2	Implementazione	10
3.1.3	Gestione	10
3.1.4	Documentazione fornita	10
3.1.4.1	Piano di Progetto	11
3.1.4.2	Analisi dei requisiti	11
3.1.4.3	Piano di Qualifica	12
3.1.4.4	Glossario	12
3.1.5	Strumenti	12
3.2	Sviluppo	13
3.2.1	Scopo	13
3.2.2	Analisi dei Requisiti	13
3.2.2.1	Scopo	13
3.2.2.2	Implementazione	13
3.2.2.3	Casi d'uso	14
3.2.2.3.1	Notazione	14
3.2.2.3.2	Diagrammi UML _G	14
3.2.2.4	Requisiti	15
3.2.2.4.1	Notazione	15
3.2.2.4.2	Suddivisione	15
4	Processi di Supporto	17
4.1	Documentazione	17
4.1.1	Scopo	17
4.1.2	Documenti	17
4.1.3	Progettazione e sviluppo	17
4.1.3.1	Template	18
4.1.3.1.1	Parametri	18
4.1.3.2	Struttura del documento	18
4.1.3.3	Verbali	18

4.1.3.4	Nomenclatura	19
4.1.3.4.1	Verbali	19
4.1.3.5	Versionamento	19
4.1.3.6	Convenzioni stilistiche	19
4.1.4	Ciclo di vita dei documenti	20
4.1.5	Strumenti	21
4.2	Configuration Management	21
4.2.1	Scopo	21
4.2.2	Configuration control	21
4.2.2.1	Scopo	21
4.2.2.2	Descrizione	21
4.2.2.3	ITS _G	21
5	Processi Organizzativi	22
5.1	Gestione organizzativa	22
5.1.1	Scopo	22
5.1.2	Ruoli	22
5.1.2.1	Responsabile	22
5.1.2.2	Amministratore	22
5.1.2.3	Analista	22
5.1.2.4	Progettista	23
5.1.2.5	Programmatore	23
5.1.2.6	Verificatore	23
5.1.3	Attività	23
5.1.3.1	Pianificazione	23
5.1.3.1.1	Strumenti	23
5.1.3.2	Esecuzione	23
5.1.3.3	Revisione	24
5.1.3.3.1	Strumenti	24
5.1.3.4	Chiusura	24
5.1.3.5	Tracciamento orario	24
5.1.3.5.1	Strumenti	24
5.1.4	Comunicazione	24
5.1.4.1	Comunicazioni interne	24
5.1.4.1.1	Comunicazioni sincrone	24
5.1.4.1.2	Comunicazioni asincrone	25
5.1.4.1.3	Strumenti	25
5.1.4.2	Comunicazioni esterne	25
5.1.4.2.1	Comunicazioni sincrone	25
5.1.4.2.2	Comunicazioni asincrone	25
5.1.4.2.3	Strumenti	25
5.1.4.3	Norme comportamentali	26
6	Metriche di qualità	27
6.1	Metriche di qualità del processo	27
6.1.1	Fornitura	27
6.1.2	Sviluppo	28
6.1.3	Documentazione	28

6.1.4	Gestione delle qualità	28
6.2	Metriche per la qualità del prodotto	29
6.2.1	Funzionalità	29
6.2.2	Affidabilità	29
6.2.3	Manutenibilità	29
6.2.4	Efficienza	29

Elenco delle figure

Elenco delle tabelle

1	Documenti del ciclo di vita del prodotto SW	17
---	---	----

1 Introduzione

1.1 Scopo del documento

Il presente documento ha l'obiettivo di definire le *best practices*_G e il *way of working*_G che i componenti del team *PEBKAC* hanno l'obbligo di rispettare per l'intero svolgimento del progetto. L'intento è di garantire un metodo di lavoro omogeneo, verificabile e migliorabile nel tempo. La creazione delle norme è progressiva e incrementale nel tempo per consentire al team di apportare aggiornamenti continui in risposta alle esigenze e alle problematiche incorse durante lo sviluppo dell'intero progetto.

1.2 Scopo del prodotto

Il progetto “Vimar GENIALE” mira a sviluppare un'applicazione intelligente che supporti installatori elettrici nell'uso di dispositivi Vimar_G, facilitando l'accesso alle informazioni tecniche sui prodotti, rispondendo a domande poste in linguaggio naturale. La tecnologia alla base prevede l'uso di modelli di *LLM*_G e di tecniche *RAG*_G, con una struttura di gestione basata su *container*_G e integrata in un ambiente cloud. Il sistema include tre componenti principali: una *applicativo web responsive*_G, un *applicativo server*_G e un'*infrastruttura cloud-ready*_G.

1.3 Glossario

Per evitare ambiguità relative al linguaggio utilizzato nei documenti, viene fornito il Glossario V1.0.0, nel quale si possono trovare tutte le definizioni di termini che hanno un significato specifico che vuole essere disambiguato. Tali termini sono marcati con una G a pedice.

1.4 Riferimenti

1.4.1 Riferimenti normativi

- Regolamento del progetto didattico
<https://www.math.unipd.it/tullio/IS-1/2024/Dispense/PD1.pdf>
(Ultimo accesso 2024-11-14)
- ISO/IEC 12207:1995 Information technology - Software life cycle processes
<https://www.math.unipd.it/tullio/IS-1/2010/Approfondimenti/A03.pdf>
(Ultimo accesso 2024-11-14)

1.4.2 Riferimenti informativi

- Capitolato C2
<https://www.math.unipd.it/tullio/IS-1/2024/Dispense/PD1.pdf>
(Ultimo accesso 2024-11-14)
- Capitolato C2 - slides
<https://www.math.unipd.it/tullio/IS-1/2024/Dispense/PD1.pdf>
(Ultimo accesso 2024-11-14)

- Documentazione_G GitHub_G
<https://docs.github.com/en>
(Ultimo accesso 2024-11-14)

2 Standard ISO/IEC 12207:1995

Il gruppo ha deciso di applicare nelle proprie modalità di lavoro e quindi nel presente documento di adottare lo *standard ISO/IEC 12207:1995 Information technology - Software life cycle processes*. In questa sezione del documento si possono trovare i criteri di applicazione e i processi definiti nell'ambito di questo standard.

2.1 Processi del ciclo di vita del software

Questo documento è usato per normare il *way of working_G* del gruppo, in particolare l'organizzazione dei processi del ciclo di vita del software_G secondo lo *standard ISO/IEC 12207:1995 Information technology - Software life cycle processes*, questi sono organizzati in una organizzazione gerarchica in cui ogni processo_G è costituito da un insieme di attività, le quali possono prevedere delle procedure e un elenco di strumenti usati per lo svolgimento.

2.1.1 Processi primari

Lo standard adottato presenta cinque processi primari (*Acquisizione, Fornitura, Sviluppo, Operazione, Manutenzione*), ma all'interno del contesto del progetto universitario in atto, il gruppo non ritiene i processi di *Operazione* e *Manutenzione* pertinenti, mentre il processo di *Acquisizione* è di competenza del committente, pertanto, il gruppo decide di escluderli dalla presentazione nel documento. I processi primari presentati nel presente documento sono:

- **Fornitura:** definisce le attività del fornitore, l'organizzazione che fornisce il prodotto software all'acquirente;
- **Sviluppo:** definisce le attività dello sviluppatore, l'organizzazione che definisce e sviluppa il prodotto software.

2.1.2 Processi di supporto

I processi di supporto presentati nel presente documento sono:

- **Documentazione_G:** definisce le attività per la registrazione delle informazioni prodotte da un processo del ciclo di vita;
- **Configuration Management_G:** definisce le attività di gestione della configurazione;
- **Accertamento di qualità:** definisce le attività per assicurare in modo oggettivo che i prodotti e i processi software siano conformi ai requisiti_G specificati;
- **Verifica_G:** definisce le attività per verificare il prodotto software;
- **Validazione_G:** definisce le attività per validare il prodotto software;
- **Risoluzione dei problemi:** definisce un processo per analizzare e risolvere i problemi di qualsiasi natura o origine, sorti durante l'esecuzione di processi.

2.1.3 Processi organizzativi

I processi organizzativi presentati nel presente documento sono:

- **Gestione organizzativa:** definisce le attività dell'acquirente, l'organizzazione che acquisisce un prodotto software;
- **Infrastruttura:** definisce le attività del fornitore, l'organizzazione che fornisce il prodotto software all'acquirente;
- **Miglioramento:** definisce le attività dello sviluppatore, l'organizzazione;
- **Formazione:** definisce le attività atte a provvedere una adeguata formazione del gruppo.

2.1.4 Ruoli

I ruoli definiti all'interno di questo progetto didattico universitario sono:

- **Docente del corso:** committente_G;
- **Azienda proponente:** cliente e mentore;
- **Gruppo di lavoro:** fornitore.

3 Processi Primari

3.1 Fornitura

3.1.1 Scopo

La fornitura è il processo che descrive le attività svolte dal fornitore, coinvolge pianificazione, acquisizione e gestione delle risorse necessarie. Il processo determina le procedure e le risorse necessarie per gestire e garantire il progetto. L'obiettivo di questo processo è garantire l'efficienza_G e la conformità ai requisiti del progetto per raggiungere gli obiettivi stabiliti dal proponente.

3.1.2 Implementazione

Il processo di fornitura è composto delle seguenti fasi:

1. **Risposta alla richiesta:** il fornitore, dopo aver analizzato i requisiti di una richiesta del proponente (il Capitolato) prepara in risposta una proposta;
2. **Negoziazione:** il fornitore negozia e stipula un contratto con il proponente;
3. **Pianificazione:** il fornitore rivede i requisiti e valuta le opzioni per lo sviluppo del prodotto software in base ad un'analisi dei rischi associati alle varie opzioni per definire la struttura di un piano di gestione del progetto al fine di garantire la qualità del prodotto finale;
4. **Esecuzione e controllo:** il fornitore esegue il piano di gestione del progetto, monitorando il progresso e la qualità del prodotto per tutto il ciclo di vita del prodotto;
5. **Revisione:** il fornitore coordina le comunicazioni con il proponente e partecipa a riunioni e revisioni. Il fornitore verifica e convalida il processo per dimostrare che i prodotti e i processi soddisfano i requisiti;
6. **Consegna:** il fornitore consegna il prodotto finale, fornendo assistenza al proponente a supporto del prodotto consegnato.

3.1.3 Gestione

Al fine di identificare e comprendere i bisogni del Proponente, per poter individuare i requisiti e i vincoli del progetto, deve essere mantenuta costante comunicazione con il Proponente, mediante riunioni SAL periodiche calendarizzate, in presenza o su Microsoft Teams_G e con scambio di messaggi su Microsoft Teams_G e mail qualora fosse necessario. Il dialogo continuo permette anche una valutazione costante dell'operato del fornitore, in modo da apportare correzioni, integrazioni e miglioramenti in modo tempestivo, incrementale e costruttivo.

3.1.4 Documentazione fornita

Sono di seguito elencati i documenti che PEBKAC si impegna a consegnare ai Commitenti e al Proponente:

3.1.4.1 Piano di Progetto

Il Piano di Progetto V1.0.0, redatto dal Responsabile con l'aiuto degli Amministratori, offre una guida per la pianificazione l'esecuzione e il controllo del progetto e viene utilizzato come punto di partenza principale per il monitoraggio del progresso del progetto, la gestione dei rischi e la comunicazione tra proponente e fornitore. Il Piano di Progetto comprende:

- Calendario di Progetto;
- Stima dei costi di realizzazione;
- Rischi e relativa mitigazione;
- Pianificazione e modello di sviluppo;
- Preventivo e consuntivo;
- Retrospettiva.

3.1.4.2 Analisi dei requisiti

L'Analisi dei Requisiti V1.0.0, redatto degli Analisti, è un documento fondamentale che ha l'obiettivo principale di definire nel dettaglio le funzionalità che il prodotto deve necessariamente avere per soddisfare a pieno le richieste del Proponente. Il documento di Analisi dei Requisiti è formato da una serie di definizioni essenziali:

- **Attori:** vengono definite entità e persone che interagiscono col sistema_G;
- **Casi d'uso:** vengono descritti narrativamente degli scenari specifici che descrivono come gli attori interagiscono col sistema_G. Lo scopo dei casi d'uso è offrire una visione semplice e chiara delle azioni eseguibili all'interno del sistema e delle interazioni degli utenti con lo stesso. Per ciascun caso d'uso viene fornito un elenco delle azioni dell'attore_G per attivare il caso d'uso, facilitando la comprensione dei requisiti corrispondenti;
- **Requisiti:** vengono individuati i requisiti obbligatori, desiderabili e opzionali e la loro categorizzazione in:
 - **Requisiti funzionali:** specificano le operazioni che il sistema deve essere in grado di eseguire;
 - **Requisiti di qualità:** definiscono gli standard e gli attributi che il software deve possedere per garantire prestazioni, affidabilità, sicurezza e usabilità ottimali;
 - **Requisiti di vincolo:** definiscono vincoli e limitazioni che il sistema deve rispettare. Possono includere restrizioni tecnologiche, normative o di risorse.

3.1.4.3 Piano di Qualifica

Il Piano di Qualifica V1.0.0, redatto dall'Amministratore, descrive gli approcci e le strategie che il gruppo ha adottato per garantire la qualità del prodotto. Lo scopo di questo documento è quello di definire le modalità di verifica e validazione, oltre che gli standard e le procedure di qualità che il gruppo ha deciso di adottare per il ciclo di vita del progetto. Si compone delle sezioni riguardanti:

- **Qualità di processo:** vengono definiti standard e procedure adottate per garantire la qualità durante tutto lo sviluppo del progetto. Vengono incluse anche informazioni sulle attività di gestione della qualità, i metodi utilizzati e le misurazioni dei processi stessi;
- **Qualità di prodotto:** vengono definiti standard, specifiche e caratteristiche che il prodotto deve soddisfare per essere considerato di qualità. Vengono incluse anche metriche e criteri di valutazione utilizzati per misurare la qualità del prodotto;
- **Specifiche dei test:** vengono definite specifiche dettagliate dei test che verranno condotti durante lo sviluppo del progetto;
- **Cruscotto delle metriche:** viene fatto un resoconto delle attività di valutazione effettuate durante il progetto per tracciare l'andamento dello stesso rispetto a obiettivi e aspettative e per identificare eventuali azioni correttive necessarie.

3.1.4.4 Glossario

Il Glossario V1.0.0 serve come un catalogo completo dei termini tecnici impiegati all'interno del progetto, fornendo definizioni chiare e precise. L'obiettivo di questo documento previene fraintendimenti a favore di una comprensione condivisa della terminologia specifica, migliorando la coerenza e la qualità della documentazione_G prodotta dal gruppo.

3.1.5 Strumenti

Gli strumenti utilizzati per il processo di fornitura sono:

- Google Calendar
- Google Sheets
- Microsoft PowerPoint
- Microsoft Teams

3.2 Sviluppo

3.2.1 Scopo

Il processo di sviluppo rappresenta la serie di attività svolte dal team PEBKAC al fine di implementare il prodotto software, rispettando le scadenze e i requisiti concordati col Proponente. Il processo è suddiviso nelle seguenti attività:

- Analisi dei requisiti,
- Progettazione;
- Codifica;
- Testing;
- Integrazione software.

3.2.2 Analisi dei Requisiti

3.2.2.1 Scopo

Lo scopo dell'analisi dei requisiti è comprendere e definire in modo chiaro e completo le necessità e le aspettative del Proponente e degli utenti relativamente al prodotto software.

3.2.2.2 Implementazione

L'analisi dei requisiti, raccolta nel documento Analisi dei Requisiti V1.0.0, viene svolta secondo le seguenti fasi:

1. Studio del capitolato e delle esigenze del Proponente;
2. Individuazione dei casi d'uso e dei requisiti;
3. Confronto con il Proponente su quanto prodotto;
4. Divisione dei requisiti nelle categorie individuate e applicazione dei quanto emerso nella discussione col Proponente.

L'attività di analisi può essere svolta in modo incrementale, quindi le sue fasi possono essere svolte più volte durante lo sviluppo del progetto.

L'Analisi dei Requisiti V1.0.0 contiene:

- **Introduzione:** descrive lo scopo del documento, del prodotto e i riferimenti utilizzati;
- **Descrizione:** esplicita le funzionalità attese del prodotto;
- **Attori:** descrive gli utilizzatori del prodotto;
- **Casi d'uso:** individua le possibili interazioni tra gli attori e il sistema_G;
- **Requisiti:** elenca le caratteristiche da soddisfare;

3.2.2.3 Casi d'uso

I casi d'uso sono strutturati nel seguente modo:

- **Attore_G**: l'attore_G che intende compiere lo scopo rappresentato dal caso d'uso;
- **Precondizioni**: stato in cui il sistema_G si deve trovare prima dell'avvio della funzionalità rappresentata dal caso d'uso;
- **Postcondizioni**: stato in cui il sistema_G si troverà dopo che l'utente avrà terminato lo scopo rappresentato dal caso d'uso;
- **Scenario principale**: descrizione della funzionalità rappresentata dal caso d'uso;
- **Scenari secondari** (se necessario);
- **Estensioni** (se presenti);
- **Specializzazioni** (se presenti).

3.2.2.3.1 Notazione

i casi d'uso seguono la seguente notazione: **UC[Codice] - [Titolo]** in cui:

- **UC** sta per Use Case_G;
- **[Codice]** è l'identificativo univoco del caso d'uso. Si tratta di un numero intero progressivo assegnato in base all'ordine di descrizione, se il caso d'uso non ha padre, altrimenti se si tratta di un sottocaso d'uso si segue la notazione **[Codice_padre]-[Numero_figlio]**, ricorsivamente senza porre limite alla profondità della gerarchia;
- **[Titolo]** è il titolo del caso d'uso.

3.2.2.3.2 Diagrammi UML_G

Un diagramma dei casi d'uso è uno strumento di modellazione che rappresenta visivamente le funzionalità di un sistema e le modalità con cui gli utenti interagiscono con esso. È particolarmente utile nella progettazione di sistemi poiché offre una rappresentazione intuitiva delle dinamiche operative e delle interazioni tra attori e sistema, senza entrare nei dettagli implementativi. I componenti principali di un diagramma dei casi d'uso sono:

1. **Attori_G**: gli attori_G rappresentano entità esterne (umane o meno) che interagiscono con il sistema e sono raffigurati con un'icona stilizzata e un'etichetta identificativa. Possono essere generalizzati: un attore generico può avere attori più specifici che ne ereditano le funzionalità e aggiungono comportamenti contestuali;
2. **Casi d'uso**: un caso d'uso descrive un'operazione che un utente può compiere attraverso il sistema. Ogni caso d'uso ha un'identificazione univoca e una breve descrizione della funzione. Può includere sequenze di azioni che illustrano le possibili interazioni con il sistema ed è collegato agli attori autorizzati tramite linee continue.

Nei diagrammi in questione poi possono comparire delle relazioni:

1. **Generalizzazioni:** le generalizzazioni possono riguardare sia gli attori che i casi d'uso. Gli attori o i casi figli ereditano le funzionalità dei genitori, aggiungendo aspetti specifici. La relazione è rappresentata con una freccia continua e un triangolo vuoto bianco;
2. **Inclusioni:** si verificano quando un caso d'uso ne richiama un altro in modo obbligatorio. Questo favorisce la riduzione della duplicazione e il riutilizzo delle strutture. La relazione è indicata con una freccia tratteggiata e l'etichetta "include";
3. **Estensioni:** rappresentano relazioni condizionali in cui un caso d'uso aggiuntivo viene eseguito solo in circostanze particolari, interrompendo temporaneamente il flusso principale. La relazione è raffigurata con una freccia tratteggiata e l'etichetta "extend".

3.2.2.4 Requisiti

3.2.2.4.1 Notazione

Ogni requisito analizzato sarà identificato univocamente da una signa del tipo **R[Tipo].[Importanza].[Codice]** nella quale:

- **[R]** sta per Requisito_G;
- **[Tipo]** può essere:
 - **F** per Funzionale;
 - **Q** per Qualità;
 - **V** per Vincolo.
- **[importanza]** classifica i requisiti in:
 - **O** per Obbligatorio;
 - **D** per Desiderabile;
 - **P** per Opzionale.
- **[Codice]** identifica univocamente i requisiti per ogni tipologia. È un numero intero progressivo univoco assegnato in ordine di importanza se il requisito non ha padre, se invece si tratta di un sotto-requisito segue il formato **[Codice_padre].[Numero_figlio]** e trattandosi di una struttura ricorsiva non c'è limite alla profondità della gerarchia.

3.2.2.4.2 Suddivisione

1. **Requisiti Funzionali:** descrivono le funzionalità del sistema_G, le azioni che il sistema può compiere e le informazioni che il sistema può fornire. Seguendo la notazione sopra riportata, si possono partizionare in:
 - RF.O - Requisito Funzionale Obbligatorio;
 - RF.D - Requisito Funzionale Desiderabile;
 - RF.P - Requisito Funzionale Opzionale;

2. **Requisiti di Qualità:** descrivono come un sistema_G deve essere, o come il sistema deve essere visualizzato, per soddisfare le esigenze dell'utente. Seguendo la notazione sopra riportata, si possono partizionare in:
- RQ.O - Requisito di Qualità Obbligatorio;
 - RQ.D - Requisito di Qualità Desiderabile;
 - RQ.P - Requisito di Qualità Opzionale;
3. **Requisiti Funzionali:** descrivono i limiti e le restrizioni normative/legislative che un sistema_G deve rispettare per soddisfare le esigenze dell'utente. Seguendo la notazione sopra riportata, si possono partizionare in:
- RV.O - Requisito di Vincolo Obbligatorio;
 - RV.D - Requisito di Vincolo Desiderabile;
 - RV.P - Requisito di Vincolo Opzionale;

4 Processi di Supporto

4.1 Documentazione

4.1.1 Scopo

Il processo di documentazione procede sempre di pari passo con tutte le attività di sviluppo, con l'obiettivo di fornire tutte le informazioni necessarie, sotto forma di testo scritto facilmente consultabile, inerenti al prodotto e alle attività stesse. Oltre a svolgere un ruolo essenziale nella descrizione del prodotto per coloro che lo sviluppano, lo distribuiscono e lo utilizzano, la documentazione svolge un ruolo di storicizzazione e di supporto alla manutenzione.

4.1.2 Documenti

In questa sezione viene descritto il piano che identifica i documenti da produrre durante il ciclo di vita del prodotto software. Tutti i documenti da redigere sono presentati nella tabella che segue, vengono esclusi i documenti presentati per la candidatura per il progetto didattico, quali *Lettera di presentazione*, *Preventivo dei costi e assunzione degli impegni* e *Analisi dei capitolati*.

Nome	Scopo	Redattore	Destinatari	Consegne
Analisi dei requisiti	Definizione dei requisiti utente	Analista	Azienda proponente, Docenti	RTB _G , PB _G
Norme di progetto	Regolamento normativo del gruppo	Amministratore, Responsabile	Docenti	RTB, PB
Piano di Progetto	Definizione temporale scadenze e progressi	Responsabile	Docenti	RTB, PB
Piano di qualifica	Definizione qualità e testing	Amministratore	Docenti	RTB, PB
Verbali esterni	Tracciamento riunioni esterne	Responsabile, Amministratore	Azienda proponente, Docenti	Candidatura, RTB, PB
Verbali interni	Tracciamento riunioni interne	Responsabile, Amministratore	Docenti	Candidatura, RTB, PB

Tabella 1: Documenti del ciclo di vita del prodotto SW

4.1.3 Progettazione e sviluppo

In questa sezione vengono presentati gli standard e le regole (nello specifico di stile) a cui i membri di PEBKAC si devono attenere per la stesura dei documenti relativi al progetto.

4.1.3.1 Template

Per la stesura dei documenti il gruppo ha creato un template in formato Latex_G . Il template fornisce una struttura e un formato predefinito per semplificare la creazione di documenti, al fine di garantire coerenza, efficienza e standardizzazione della presentazione. Il template è progettato per essere facile da usare, dovendo inserire solo con piccole modifiche per rispecchiare le specificità di ciascun tipo di documento.

In particolare nel template è definite la pagina di copertina con intestazione contenente logo informazioni del gruppo e dell'Università di Padova, titolo del documento, informazioni sul documento (uso, destinatari) e un breve abstract del contenuto, oltre che altre specifiche di stile come il titolo dell'indice in italiano e il numero di pagina come **X** di Tot, dove **X** è il numero della pagina e Tot è il numero totale di pagine.

4.1.3.1.1 Parametri

Nel principale file Latex del template sono definiti una serie di comandi personalizzati per l'inserimento automatico delle informazioni come titolo, data, uso, destinatari e abstract. Sono inoltre già presenti ma commentate le voci necessarie solo per i verbali (vedi §4.1.3.3 Verbali)

4.1.3.2 Struttura del documento

Tutti i documenti prodotti da PEBKAC presentano la medesima struttura, alla quale ogni membro si deve attenere durante la procedura di stesura e modifica.

- **Pagina di copertina:** come nella sezione Template precedente;
- **Registro delle versioni:** questo registro è utilizzato per tenere traccia delle varie versioni per permettere di comprendere velocemente chi ha realizzato o modificato determinate sezioni della documentazione e quando. Il registro presenta le versioni ordinate a partire dalla versione più recente;
- **Indice:** presente per facilitare la consultazione del documento, dotato di sezioni. Il suo scopo è di facilitare e agevolare l'accesso ad un determinato contenuto all'interno nel documento;
- **Contenuto:** il contenuto vero e proprio del documento.

4.1.3.3 Verbali

I verbali differiscono dalla struttura precedentemente esposta in quanto ad essi prevedono delle sezioni aggiuntive ed obbligatorie:

- **Pagina di copertina:** nel caso di un verbale tra le informazioni sul documento compaiono anche i nominativi con i rispettivi ruoli dei membri che hanno lavorato alla loro produzione;
- **informazioni generali:** la prima sezione di un verbale deve sempre essere quella nominata "Informazioni generali" che prevede, sotto forma di elenco puntato, le seguenti informazioni:
 - Tipo di riunione,
 - Luogo in cui si è tenuta la riunione (anche se telematica),

- Data in cui si è tenuta la riunione,
- Ora di inizio della riunione,
- Ora di fine della riunione,
- Membri presenti ed eventuali altre persone alla riunione,
- Membri assenti dalla riunione;
- **Todo:** l’ultima sezione di un verbale deve sempre essere quella che elenca i task_G emersi durante la riunione da aggiungere al backlog_G. Questi vengono presentati sotto forma di tabella a due colonne:
 - **Assegnatario:** il membro a cui quel task è stato assegnato, nel caso in cui non ve ne sia uso ma il task possa essere autoassegnato da uno dei membri si scriverà “autoassegnazione” in corsivo;
 - **Task Todo:** denominazione del task.

4.1.3.4 Nomenclatura

La nomenclatura per i documenti si ottiene unendo il nome del file in *Snake_Case* quindi con le parole separate da un underscore (`_`) (`Nome_del_File`), un underscore (`_`) e la sua versione (`1.2.3`), ottenendo per esempio `Norme_di_Progetto_1.2.3.pdf`. Nel caso di documenti il cui nome contiene una data, essa si inserisce dopo il nome, ma prima della versione, sempre usando gli underscore come separatori, nella forma YYYY-MM-DD: YYYY rappresenta l’anno, MM il mese e DD il giorno, sempre scritto in due cifre.

4.1.3.4.1 Verbalì

Per quanto riguarda i verbalì, per facilitarne l’ordinamento) il loro nome è la data in cui la riunione di è tenuta nella forma YYYY-MM-DD: YYYY rappresenta l’anno, MM il mese e DD il giorno, sempre scritto in due cifre. Nel caso si tratti di un verbale esterno viene aggiunta una E, sempre separata da underscore tra la data e la versione.

4.1.3.5 Versionamento

La versione di un documento è del tipo `[x].[y].[z]`:

- **z:** è un numero intero che incrementato dal Redattore ad ogni modifica;
- **y:** è un numero intero incrementato dal Verificatore ad ogni verifica;
- **x:** è un numero intero che viene incrementato dal Responsabile dopo la sua approvazione (versione di produzione).

4.1.3.6 Convenzioni stilistiche

- **Date:** tutte le date nella documentazione prevedono il seguente formato YYYY-MM-DD, dove DD indica il giorno a due cifre, MM il mese a due cifre e YYYY l’anno a 4 cifre;
- **Elenchi:** elenchi puntati o numerati, ogni punto inizia con la lettera maiuscola e termina con “;” ad eccezione dell’ultimo che termina con “.”;

- **Menzioni:** ogni menzione ad una persona, interna o esterna, avviene nel formato Nome Cognome;
- **Riferimenti interni:** i riferimenti a sezioni interne allo stesso documento devono essere riportati seguendo la notazione §1.2 Nome sezione, dove §1.2 è il numero della sezione. Inoltre questi riferimenti devono essere opportunamente collegati tramite link al paragrafo indicato, senza alterare lo stile del testo;
- **Riferimenti esterni:** i riferimenti a sezioni di documenti esterni devono essere riportati seguendo la notazione Nome Documento (versione di riferimento), Nome sezione;
- **Link URL:** possono essere estesi o avere una visualizzazione abbreviata, ma sempre visualizzati di colore blu;
- **Caratteri maiuscoli:** devono essere utilizzati per
 - Le iniziali dei nomi;
 - Le lettere che compongono degli acronimi e le iniziali delle rispettive definizioni;
 - Le iniziali dei ruoli svolti dai componenti del gruppo;
 - Le iniziali dei ruoli definiti all'interno del progetto didattico;
 - La prima lettera di ogni elenco puntato.
- **Grassetto:** devono essere visualizzati in grassetto
 - I titoli di sezioni/sottosezioni/paragrafi di un documento;
 - Le parole che meritano enfasi;
 - Le definizioni negli elenchi puntati.
- **Caption:** ogni immagine o tabella deve avere una caption, utile a fornire una breve descrizione o spiegazione del contenuto visivo.

4.1.4 Ciclo di vita dei documenti

Ogni documento segue le fasi del seguente workflow_G:

1. **Assegnazione:** il gruppo assegna un documento a uno o più redattori, affiancati da uno o più verificatori;
2. **Branch:** si crea un branch per lo sviluppo del documento nell'apposita repository_G Docs;
3. **Template:** si copia il Template all'interno della cartella appropriata;
4. **Stesura:** si redige il documento o una sua sezione. Qualora serva un elevato parallelismo di lavoro è possibile usare Google Drive per la prima stesura e successivamente caricare il documento all'interno del branch;
5. **Commit:** si esegue la commit sul branch creato;

6. **Pull Request:** si apre una pull request dal branch appena creato verso il branch develop: se il documento non è pronto per la verifica, ma ha bisogno di ulteriori modifiche, si apre la pull request in modalità draft, per marcarla successivamente come “Ready to Review”, altrimenti in modalità normale;
7. **Verifica:** se il verificatore richiede modifiche si ripete, in ordine, dal punto 3 al punto 5;
8. **Chiusura branch:** si elimina, quando la pull request viene chiusa o risolta, il branch creato.

Per la versione finale di un documento spetta al Responsabile conferire l’approvazione definitiva, annotando opportunamente nel registro delle versioni la versione **x.0.0** e la sua approvazione finale.

4.1.5 Strumenti

- Latex
- Visual Studio Code
- GitHub

4.2 Configuration Management

4.2.1 Scopo

In questa sezione vengono presentate le attività svolte da PEBKAC per il processo di Configuration Management. Il processo in questione consiste nell’applicazione di procedure amministrative e tecniche per l’intero ciclo di vita del software, al fine di:

- Identificare, definire e stabilire una base per gli elementi software di un sistema_G;
- controllare le modifiche e le release degli elementi;
- Registrare lo stato degli elementi e delle richieste di modifica;
- Garantire la completezza, la coerenza e la correttezza degli elementi.

4.2.2 Configuration control

4.2.2.1 Scopo

Il processo di configuration management è finalizzato a garantire il controllo e la coerenza delle configurazioni del sistema, assicurando che tutte le modifiche apportate a software, artefatti e documenti siano tracciate, gestite e allineate agli obiettivi e ai requisiti del progetto.

4.2.2.2 Descrizione

In questa

4.2.2.3 ITS_G

In questa

5 Processi Organizzativi

5.1 Gestione organizzativa

5.1.1 Scopo

Lo scopo di questo processo è esporre le modalità e gli strumenti di coordinamento usati dal gruppo per la comunicazione, interna ed esterna, e normare l'assegnazione di ruoli e compiti, oltre che la gestione dei rischi.

5.1.2 Ruoli

Per ottimizzare la gestione delle attività e dei compiti da svolgere vengono definiti sei ruoli distinti, ciascuno con mansioni e responsabilità specifiche. Ogni componente del gruppo dovrà assumere ciascun ruolo per un numero di ore significativo.

5.1.2.1 Responsabile

Il responsabile è il punto di riferimento per tutto il gruppo e anche per le comunicazioni con il committente e con l'azienda proponente. Inoltre il responsabile è la figura che ha il compito di coordinare le azioni dei membri del gruppo, perciò deve avere competenze tecniche in ogni ambito del progetto. Le responsabilità di questo ruolo sono:

- Coordinamento tra gruppo ed enti esterni;
- Gestione delle comunicazioni interne;
- Pianificazione di progetto,
- Gestione dei task e delle risorse;
- Gestione dell'avanzamento del progetto;

5.1.2.2 Amministratore

L'amministrazione è la figura che definisce, gestisce e mantiene l'ambiente e l'infrastruttura necessari per lo sviluppo del progetto facendo in modo che siano affidabili e sicuri. Si occupa della gestione della configurazione, del versionamento, delle varie automazioni e della documentazione. Si occupa di:

- Selezionare e abilitare risorse informatiche a supporto del *way of working*_G;
- Gestire errori e malfunzionamenti nei meccanismi nell'infrastruttura.

5.1.2.3 Analista

La funzione dell'analista è quella di analizzare il problema per definire i requisiti del prodotto, per questo deve avere buona conoscenza del dominio del problema. L'analista raccoglie le sue produzioni nel documento Analisi dei Requisiti. Si tratta di un ruolo fondamentale all'inizio del progetto, ma la cui utilità cala nelle seguenti fasi del progetto.

5.1.2.4 Progettista

Al progettista spettano le scelte realizzative e le specifiche architetture del prodotto. Deve avere buone competenze tecniche e tecnologiche. Durante il processo di sviluppo la sua utilità è massima, ma tende a calare dalla fase di manutenzione in poi.

5.1.2.5 Programmatore

Quello del programmatore è un ruolo chiave nella fase di sviluppo. In particolare si occupa di :

- Codificare ciò che è stato definito dai progettisti;
- Implementare i test;
- Redigere il Manuale utente.

5.1.2.6 Verificatore

Ha il compito di verificare il lavoro degli altri e per questo deve avere competenze tecniche ed essere presente per l'intera durata del progetto. Questa figura deve controllare che tutto ciò che viene prodotto sia conforme alle norme e alle aspettative di qualità del gruppo.

5.1.3 Attività

Ogni membro del gruppo può proporre attività da svolgere, ma è compito del responsabile stabilire la fattibilità rispetto alle risorse usufruibili.

5.1.3.1 Pianificazione

Le attività definite devono poi essere pianificate in termini di tempo e risorse dal responsabile, stabilendo quindi:

- La tempistica prevista per il completamento dell'attività;
- Il membro che dovrà eseguire l'attività, in base a ruolo e risorse disponibili;
- Il verificatore;
- Il rischio associato.

È poi compito dell'Amministratore inserire il task associato all'interno di Trello (maggiori dettagli alla sezione § [INSERIRE]).

5.1.3.1.1 Strumenti

- Trello

5.1.3.2 Esecuzione

L'esecuzione delle attività avviene obbligatoriamente per mano dell'assegnatario, definito dal responsabile. L'esecuzione deve essere obbligatoriamente conforme alla documentazione associata precedentemente redatta. L'esecutore dovrà proporre la sua soluzione con una pull request_G.

5.1.3.3 Revisione

La revisione dell'attività è effettuata dal verificatore prima dell'effettivo inserimento delle modifiche nel repository GitHub: la pull request_G aperta dell'esecutore viene accettata o rifiutata, riportando le parti non valide ed eventuali accorgimenti possibili, a seconda dell'esito della verifica.

5.1.3.3.1 Strumenti

- GitHub

5.1.3.4 Chiusura

Solo nel caso dell'esito positivo della verifica, con l'accettazione della pull request, viene chiuso il branch di cui è stato effettuato il merge e l'attività viene segnata come completata.

5.1.3.5 Tracciamento orario

Il gruppo utilizza Google Sheets_G per avere un foglio di calcolo condiviso in cui tenere conto del tempo speso per svolgere le attività. Ogni membro è tenuto a registrare, alla fine di ogni sessione lavorativa, il numero di ore effettive di lavoro e il ruolo ricoperto.

5.1.3.5.1 Strumenti

- Google Sheets

5.1.4 Comunicazione

5.1.4.1 Comunicazioni interne

5.1.4.1.1 Comunicazioni sincrone

Le riunioni interne si svolgeranno sulla piattaforma Slack_G oppure in presenza, dovranno in ogni caso decise e organizzate alcuni giorni prima per consentire la presenza di tutti i membri. In ogni riunione il gruppo predilige un approccio libero alla discussione, incentrato sulla crescita e allo scambio di opinioni.

La gestione delle riunioni interne viene affidata al Responsabile che, coadiuvato dall'Amministratore, ha il compito di:

1. Fissare data, ora e luogo della riunione;
2. Stabilire un ordine del giorno per le riunioni;
3. Fare da moderatore durante la discussione, per garantire a tutti l'opportunità di esprimersi;
4. Se necessario, comunicare con l'esterno in base alle decisioni prese dal gruppo.

5.1.4.1.2 Comunicazioni asincrone

Per le comunicazioni asincrone il gruppo utilizzerà:

- **Slack:** in un area di lavoro sono stati creati:
 - **Canali:** uno principale con tutti i membri e altri, alla necessità, in cui i componenti possono organizzarsi e lavorare su attività collaborative;
 - **Canvas:** uno per ogni canale in cui sia necessario, per fissare messaggi importanti, documenti e risorse che richiedono facile accesso.
- **Whatsapp:** solo per comunicazioni immediate e poco formali.

5.1.4.1.3 Strumenti

- Slack
- Whatsapp

5.1.4.2 Comunicazioni esterne

5.1.4.2.1 Comunicazioni sincrone

Per quanto riguarda gli incontri con l'azienda proponente, cruciali per discutere in modo semplice e immediata di argomenti anche complessi, potranno essere in presenza (presso la sede R&D di Vimar S.p.A.) oppure da remoto sulla piattaforma Teams. Per quanto riguarda le riunioni con l'azienda proponente, anche chiamate SAL - Stato di Avanzamento Lavori.

- Il gruppo ha concordato con l'azienda un calendario di incontri bisettimanali della durata di 60 minuti fino alla prima revisione, poi settimanali della durata di 30 minuti;
- Il gruppo si impegna a presenziare in maniera assidua agli incontri, segnalando per tempo eventuali assenze o modifiche a quanto precedentemente concordato;
- Il gruppo si impegna a redigere un verbale per ogni incontro per documentarne il contenuto e farlo approvare all'azienda.

5.1.4.2.2 Comunicazioni asincrone

Per le comunicazioni asincrone con il proponente o altri soggetti esterni vengono utilizzati:

- **Teams:** per domande corte o piccoli chiarimenti si potrà usare la chat condivisa creata dall'azienda;
- **Posta elettronica:** per comunicare con soggetti esterni e con l'azienda proponente per domande articolate o modifiche agli appuntamenti fissati si utilizzerà la mail del gruppo: pebkacswe@gmail.com.

5.1.4.2.3 Strumenti

- Teams
- Google Gmail

5.1.4.3 Norme comportamentali

I membri del gruppo, per garantire il rispetto delle norme e degli altri membri del gruppo, sono obbligati a:

- Essere sempre puntuali o almeno comunicare tempestivamente al responsabile eventuali problemi;
- Partecipare attivamente alla discussione;
- Mantenere un atteggiamento rispettoso, disciplinato, aperto alle discussioni e disponibile.

Inoltre, per le riunioni SAL con l'azienda proponente i membri hanno concordato di rispettare le seguenti regole:

- Tenere i telefoni spenti o silenziosi (a meno di particolari motivi);
- Non utilizzare PC o Tablet, fatta eccezione per il responsabile o chi deve raccogliere degli appunti.

6 Metriche di qualità

6.1 Metriche di qualità del processo

6.1.1 Fornitura

- **CV (Cost Variance):** Misura la deviazione dei costi rispetto al budget, se il costo è negativo significa che si è sforato il limite del budget ($SPI > 1$: in anticipo rispetto ai tempi pianificati, $SPI < 1$: in ritardo rispetto ai tempi pianificati).
 $CV = EV - AC$.
- **PV (Planned Value):** Il valore pianificato, ovvero il costo stimato del lavoro previsto in un determinato momento del progetto.
- **EV (Earned Value):** Rappresenta il valore guadagnato, che rappresenta il costo stimato del lavoro effettivamente completato in quel momento.
- **AC (Actual Cost):** Rappresenta il costo effettivo, cioè quanto è stato realmente speso fino a quel punto.
- **CPI (Cost Performance Index):** Indica se il progetto sta spendendo meno o più del previsto ($CPI > 1$: sotto budget, $CPI < 1$: sopra budget).
La formula è $CPI = \frac{EV}{AC}$.
- **SPI (Schedule Performance Index):** Rappresenta l'efficienza temporale con cui il lavoro pianificato è stato completato rispetto a quanto programmato.
La formula è $SPI = \frac{EV}{PV}$.
- **BAC (Budget At Completion):** Rappresenta il budget totale pianificato per il completamento del progetto.
- **EAC (Estimated At Completion):** Rappresenta l'aggiornamento della stima del valore per la realizzazione del progetto, ovvero il BAC ricalcolato in base allo stato attuale del progetto. La formula è $EAC = \frac{BAC}{CPI}$.
- **VAC (Variance At Completion):** Rappresenta la differenza tra il budget previsto e quello attuale alla fine del progetto.
La formula è $VAC = BAC - EAC$.
- **ETC (Estimated To Completion):** Rappresenta la valutazione del costo supplementare richiesto per portare a termine il progetto.
La formula è $ETC = EAC - AC$.
- **SV (Schedule Variance):** Indica se le attività pianificate del progetto sono in linea, anticipate o in ritardo rispetto alla programmazione.
La formula è $SV = EV - PV$.
- **BV (Budget Variance):** Indica se, alla data attuale, le spese sostenute sono superiori o inferiori rispetto a quanto originariamente previsto nel budget.
La formula è $BV = PV - AC$.

6.1.2 Sviluppo

- **SC (Statement Coverage):** Rappresenta la percentuale di istruzioni nel codice che vengono eseguite durante i test. La formula è $SC = \frac{N \text{ Statement eseguiti}}{N \text{ Statement totali}} * 100$.

6.1.3 Documentazione

- **IG (Indice Gulpease):** Rappresenta un indicatore per analizzare la facilità di lettura di un testo scritto in italiano. L'Indice Gulpease si basa su due variabili linguistiche principali: la lunghezza delle parole e quella delle frasi. La formula per determinarlo è: $IG = 89 + \frac{300 * NF - NL}{NP}$, dove:

- **NF:** Indica il numero delle frasi.
- **NL:** Indica il numero delle lettere.
- **NP:** Indica il numero di parole.

Questo indice fornisce un punteggio che varia da 0 a 100. I possibili punteggi possono essere:

- **0-55:** Testo incomprensibile.
- **56-70:** Testo molto difficile.
- **71-80:** Testo difficile.
- **81-95:** Testo facile.
- **95-100:** Testo molto facile.

Per calcolarlo viene utilizzato un software online: https://farfalla-project.org/readability_static/

- **CO (Correttezza Ortografica):** Rappresenta il numero di errori grammaticali ed ortografici che presenta un documento.

6.1.4 Gestione delle qualità

- **MNS (Metriche Non Soddisfatte):** Rappresenta le quantità di metriche che il progetto non riesce a soddisfare o mantenere.

6.2 Metriche per la qualità del prodotto

6.2.1 Funzionalità

- **ROS (Requisiti Obbligatori Soddisfatti):** Rappresenta la percentuale di requisiti obbligatori che sono stati soddisfatti durante la creazione del prodotto.
La formula è $ROS = \frac{\text{requisiti obbligatori soddisfatti}}{\text{requisiti obbligatori totali}} * 100$.
- **RDS (Requisiti Desiderabili Soddisfatti):** Rappresenta la percentuale di requisiti desiderabili che sono stati soddisfatti durante la creazione del prodotto.
La formula è $RDS = \frac{\text{requisiti desiderabili soddisfatti}}{\text{requisiti desiderabili totali}} * 100$.
- **RPS (Requisiti Opzionali Soddisfatti):** Rappresenta la percentuale di requisiti opzionali che sono stati soddisfatti durante la creazione del prodotto.
La formula è $RPS = \frac{\text{requisiti opzionali soddisfatti}}{\text{requisiti opzionali totali}} * 100$.

6.2.2 Affidabilità

- **PTCP (Passed Test Cases Percentage):** Rappresenta la percentuale di casi di test completati con successo rispetto al numero totale di casi di test pianificati.
La formula è $PTCP = \frac{\text{test superati}}{\text{test totali}} * 100$.
- **CC (Code Coverage):** Rappresenta il numero di linee di codice Verificate con esito positivo all'interno di un processo di test.
La formula è $CC = \frac{\text{linee di codice scritte}}{\text{linee di codice totali}} * 100$.

6.2.3 Manutenibilità

- **SFIN (Structure Fan IN):** Rappresenta la quantità di moduli o componenti che interagiscono direttamente o dipendono da un modulo o una funzione specifica. Un valore elevato suggerisce che molte parti del sistema fanno affidamento su quel particolare modulo.
- **SFOUT (Structure Fan Out):** Rappresenta la quantità di connessioni o relazioni che un componente o modulo ha con altri elementi del sistema. Questa misura riflette il numero di moduli che interagiscono o su cui si basa un determinato modulo. Un fan-out elevato può segnalare che un modulo è fortemente dipendente da molti altri.

6.2.4 Efficienza

- **TDE (Tempo di Elaborazione):** Rappresenta il tempo di risposta dal momento in cui vengono inseriti dati all'interno del prodotto software al momento in cui vengono visualizzati dall'utente in questo caso l'installatore.