



Rod2Cod

Way of working

Il documento fornisce le **informazioni** necessarie ad ogni membro del team riguardo il **metodo** di lavoro adottato e tutte le **procedure** da applicare.

Progetto di Ingegneria del Software
A.A. 2024/2025

Informazioni

Versione	3.0
Uso	Esterno
Data	27/04/2025
Destinatari	Gruppo Rod2Cod Zucchetti Tullio Vardanega Riccardo Cardin
Responsabile	Alberto Maggion
Amministratore	Annalisa Egidi
Verificatori	Alberto Maggion Annalisa Egidi Filippo Bellon Luca Calzetta Michele Nesler
Autori	Alberto Maggion Annalisa Egidi Filippo Bellon Luca Calzetta Michele Nesler

Versione	Data	Descrizione	Autore	Verificatore	Validatore
3.0	2025-04-14				Michele Nesler
2.1	2025-04-09	- Modifica versionamento dei documenti e tracciamento issue nei verbali - Piano di Qualifica - Qualità di Processo - Piano di Qualifica - Qualità di Prodotto	Alberto Maggion, Annalisa Egidi	Filippo Bellon	
2.0.0	2025-02-24				Luca Calzetta
1.4.0	2025-02-24	Verifica ultime modifiche		Filippo Bellon	
1.3.3	2025-02-24	Aggiunta sezione Piano di Qualifica	Annalisa Egidi		
1.3.2	2025-02-24	<i>Github_G repository_G</i> - Modificare sezioni Struttura e Utilizzo (Caricamento documenti)	Filippo Bellon		
1.3.1	2025-02-23	Riorganizzazione processi in Primari, di supporto e organizzativi	Michele Nesler		
1.3.0	2025-02-21	Verifica ultime modifiche		Alberto Maggion	
1.2.1	2025-02-16	Aggiunte sezioni Piano di Progetto - Pinaficazione, Preventivo, Consuntivo, Retrospective	Luca Calzetta		
1.2.0	2025-01-31	Verifica ultime modifiche		Alberto Maggion	
1.1.1	2025-01-30	<i>Github_G repository_G</i> - Modificare	Filippo Bellon		

		sezioni Struttura e Utilizzo (Caricamento documenti)			
1.1.0	2025-01-30	Verifica ultime modifiche		Alberto Maggion	
1.0.5	2025-01-29	Modificato il modo di inserire i termini nel glossario	Michele Nesler		
1.0.4	2025-01-13	Modificato posizionamento sezione <i>Github_G</i> e aggiunta sezione Piano di Qualifica	Filippo Bellon		
1.0.3	2024-12-29	Aggiunta sezione analisi dei requisiti	Luca Calzetta		
1.0.2	2024-12-26	Documentazione - Aggiunta sezione strumento per la scrittura documenti, - Aggiunta sezione sistema di ticketing, - Aggiunta sezione modalità di riunione, - Aggiunta sezione tracciamento ore, - Aggiunta sezione glossario	Luca Calzetta		
1.0.1	2024-12-22	Piano di Progetto - Aggiunta sezione Diagrammi di Gantt	Filippo Bellon		
1.0.0	2024-12-15				Annalisa Egidi
0.2.3	2024-12-06	Verifica sezioni . utilizzo template - Documentazione/ Processi		Michele Nesler	

		Modifica, Verifica, Approvazione			
0.2.3	2024-11-23	Documentazione/ Processi - Aggiunte in sezioni Modifica, Verifica e Approvazione	Filippo Bellon		
0.2.2	2024-11-19	Documentazione/ Utilizzo template - Tolto segretario verbalizzante dai verbali	Michele Nesler		
0.2.1	2024-11-18	Aggiunta sezione Utilizzo template	Alberto Maggion		
0.2.0	2024-11-18	Verifica ultime aggiunte		Alberto Maggion	
0.1.1	2024-11-13	1.6 - Aggiunta sezione <i>Github_G</i> <i>repo_G</i>	Filippo Bellon		
0.1.0	2024-11-13	Verifica documento		Filippo Bellon	
0.0.3	2024-11-12	1.1.2 - Modifica secondo verbale 05-11-2024 decisione 3.1	Alberto Maggion		
0.0.2	2024-10-22	1.2 - 2.1 - 1.4 - Aggiunta tabella Riepilogo Modifiche	Michele Nesler		
0.0.1	2024-10-22	Prima stesura	Alberto Maggion		

Contents

1 Riferimenti normativi	7
2 Riferimenti informativi	7
3 Processi primari	7
3.1 Fornitura	7
3.1.1 Inizializzazione	7
3.1.2 Preparazione risposta e Contratto	7
3.1.3 Pianificazione	7
3.1.4 Esecuzione e Controllo	8
3.1.5 Revisione e Valutazione	8
3.1.6 Consegna e completamento	8
3.2 Sviluppo	8

3.2.1 Implementazione Processi	8
3.2.2 Analisi dei requisiti di sistema	9
3.2.3 Progettazione Architettura di sistema	9
3.2.4 Progettazione di dettaglio	9
3.2.5 Codifica e test del software	9
3.2.6 Integrazione software	9
3.2.7 Test di qualificazione del software e Integrazione di sistema	10
3.2.8 Test di qualificazione del sistema	10
3.2.9 Installazione del software	10
4 Processi di supporto	10
4.1 Documentazione	10
4.1.1 Strumento per la scrittura dei documenti	10
4.1.2 Struttura documento	10
4.1.2.1 Titolo del documento	10
4.1.2.2 Tabella versione del documento	10
4.1.3 Versionamento dei documenti	11
4.1.4 Regole cambiamento versione	11
4.1.5 Nomenclatura	11
4.1.5.1 Documenti	11
4.1.5.2 Verbali	11
4.1.6 Azioni	12
4.1.6.1 Modifica	12
4.1.6.2 Verifica	12
4.1.6.3 Approvazione	12
4.1.7 Verbali	12
4.1.7.1 Presenze	12
4.1.7.2 Esiti	12
4.1.7.3 Interni	12
4.1.7.4 Esterni	12
4.1.7.5 Esempio esplicativo	12
4.1.8 Utilizzo template	13
4.1.8.1 Verbali	13
4.1.8.2 Documenti	13
4.1.9 Glossario	13
4.1.9.1 Suddivisione delle parole	14
4.1.9.2 Aggiunta di un Termine	14
4.1.10 Piano di Progetto	14
4.1.10.1 Diagrammi di Gantt	14
4.1.10.2 Pianificazione	14
4.1.10.3 Preventivo ore e costi	14
4.1.10.4 Consuntivo delle ore e dei costi	14
4.1.10.5 Retrospective	15
4.1.11 Analisi dei Requisiti	15
4.1.11.1 Casi d'uso (UC_G)	15
4.1.11.2 Requisiti	15
4.2 Gestione della configurazione	16
4.2.1 $Github_G repository_G$	16
4.2.1.1 Struttura	16

4.2.1.2 Utilizzo	16
4.2.1.2.1 Caricamento documenti	16
4.2.1.2.2 Caricare modifiche alla Webpage	17
4.2.1.3 Sistema di ticketing	17
4.2.1.3.1 Creazione del ticket	17
4.2.1.3.2 Project board	17
4.2.1.3.2.1 Project board per i documenti	17
4.3 Assicurazione della qualità	18
4.3.1 Piano di Qualifica	18
4.3.1.1 Metriche Utilizzate	18
4.3.1.2 Struttura del documento	18
4.3.1.3 Qualità di Processo	19
4.3.1.4 Qualità di Prodotto	19
4.3.1.5 Specifica dei Test	20
4.3.1.6 Resoconto delle Attività di Verifica	20
4.3.2 Assicurazione prodotti	20
4.3.3 Assicurazione Processi	20
4.4 Risoluzione dei problemi	20
4.5 Verifica	20
4.5.1 Retrospective	21
4.5.2 Verifica documentazione	21
4.5.3 Verifica codice	21
4.6 Validazione	21
4.6.1 Validazione documentazione	21
4.6.2 Validazione Prodotto	21
4.6.3 Revisione congiunta	21
4.6.4 Audit	21
4.6.5 Risoluzione dei problemi	22
5 Processi organizzativi	22
5.1 Gestione	22
5.1.1 Inizializzazione	22
5.1.2 Pianificazione	22
5.1.3 Esecuzione e controllo	22
5.1.4 Revisione e valutazione	22
5.1.5 Chiusura	22
5.2 Riunioni	22
5.2.1 Obiettivi delle Riunioni	22
5.2.2 Pianificazione delle Riunioni	22
5.2.3 Ruoli e Responsabilità	23
5.2.4 Svolgimento delle Riunioni	23
5.2.5 Documentazione	23
5.2.6 Strumenti e Tecnologie	23
5.3 Tracciamento ore	23
5.4 Infrastrutturale	24
5.5 Miglioramento	24
5.6 Formazione	24

1 Riferimenti normativi

- Regolamento progetto didattico (<https://www.math.unipd.it/~tullio/IS-1/2024/Dispense/PD1.pdf>)

2 Riferimenti informativi

- Analisi dei requisiti v2.0
- Piano di Progetto v1.0
- Piano di Qualifica v1.0
- Candidatura v1.0
- Specifica Tecnica v1.0

3 Processi primari

3.1 Fornitura

Il processo di fornitura viene avviato dopo aver ricevuto la le **proposte** dai proponenti (presentazione dei capitolati), serve ad **analizzare** le proposte per poi decidere quale accettare ed eventualmente proporre modifiche. Successivamente vengono definite le procedure e le **risorse** necessarie per completare e assicurare il progetto, stilando il piano di progetto. Si compone delle seguenti attività:

3.1.1 Inizializzazione

1. Si analizzano i **capitolati** valutando pro e contro di ognuno, focalizzandosi sull'identificare possibili criticità.
2. Per i capitolati di maggiore interesse si organizzano degli **incontri** con gli acquirenti per approfondire le proposte ed eventualmente discutere di possibili modifiche.
3. I **risultati** delle analisi e degli incontri vengono **documentati** nel documento "Resoconto_Capitolati"

3.1.2 Preparazione risposta e Contratto

Viene scelto il **capitolato** per cui ci si vuole candidare sulla base delle considerazioni contenute nel documento "Resoconto_Capitolati" e viene preparato il **documento** "Candidatura" contenente i **preventivi** preliminari riguardanti monte ore necessario al completamento, costi di realizzazione, data di consegna. Il documento conterrà anche le motivazioni della scelta del capitolato. Il documento viene inviato al committente e viene attesa la risposta. Se negativa l'attività viene ripetuta in toto se il capitolato è ancora disponibile, altrimenti viene selezionato un capitolato differente.

3.1.3 Pianificazione

1. viene definito o selezionato un modello di **ciclo di vita** del software appropriato alla portata, alla grandezza e alla complessità del progetto. Esso viene documentato all'interno del documento "Piano di Progetto".
2. I **requisiti** dell'acquirente vengono **analizzati** in dettaglio e documentati nel documento "Analisi dei requisiti", che funge da base per la progettazione e lo sviluppo del software, come definito dal processo di documentazione.
3. Viene elaborato e **documentato** il "Piano di progetto", che definisce gli obiettivi, le scadenze, le risorse, i rischi e le strategie di gestione del progetto, come definito dal processo documentazione.
4. Viene sviluppato e **documentato** il "Piano di qualifica", che stabilisce le procedure di test, le metriche di qualità e i criteri di accettazione per garantire la conformità ai requisiti.

5. Vengono **definite** le Norme di Progetto e documentate nel documento “Way of Working”, al momento del bisogno, ovvero poco **prima** che i relativi processi vengano istanziati. Vengono poi aggiornate e migliorate durante tutta la durata del progetto in modo incrementale.

3.1.4 Esecuzione e Controllo

1. Viene **aggiornato** all’inizio e alla fine di ogni periodo il Piano di progetto riportando pianificazione, preventivo, consuntivo e retrospective.
2. Viene **sviluppato** il Software in accordo con il processo di Sviluppo.
3. Viene **Monitorato** alla fine di ogni periodo i progressi raggiunti, i costi e le difficoltà incontrate, operando tempestivamente azioni correttive quando necessarie.
4. Vengono Identificati, documentati, analizzati e elaborate modalità di risoluzione per i problemi incontrati durante l’attività di retrospective nella riunione di fine periodo.
5. Viene Comunicato lo **stato di avanzamento** al proponente durante le riunioni periodiche.

3.1.5 Revisione e Valutazione

1. Vengono svolte **riunioni** settimanali informali tra i membri del team, qualora necessario vengono pianificate altre riunioni. Queste servono ad aggiornarsi sullo stato di avanzamento, a pianificare e assegnare i prossimi compiti da svolgere. Nelle riunioni che coincidono con la fine di un periodo vengono presentati i cruscotti e si esegue l’attività di retrospective come definito nell’attività verifica documentazione del processo di verifica, poi si pianifica e si stila il preventivo riguardante il periodo successivo.
2. Ogni mese viene organizzata una **riunione** con il proponente per esporre lo stato di avanzamento, discutere eventuali problematiche e modifiche da apportare.

Qualora si presentasse la necessità vengono organizzate insieme al proponente delle riunioni aggiuntive.

1. Vengono sostenute le **revisioni** di avanzamento RTB e PB come definito nel documento “Regolamento del Progetto Didattico”.
2. I cruscotti e tutti i documenti di interesse vengono esposti al proponente tramite *GitHub*-pages.
3. Ad ogni modifica del prodotto viene eseguito il processo di verifica, prima delle revisioni di avanzamento viene eseguito il processo di validazione.

3.1.6 Consegna e completamento

Alla fine del progetto il prodotto software viene installato nella macchina del proponente, e viene fornita la relativa documentazione. Esso viene anche eseguito e testato in presenza del proponente in modo da dimostrarne il funzionamento. Se durante la consegna il proponente pone delle domande gli deve essere fornita una risposta, sul momento o in differita se necessario.

3.2 Sviluppo

Il processo di sviluppo definisce le attività per l’**analisi dei requisiti**, la **progettazione**, la **codifica**, l’**integrazione**, il **test**, l’**installazione** e l’**accettazione** relative ai prodotti software. Si compone delle seguenti attività:

3.2.1 Implementazione Processi

1. Tutti i **risultati** del processo vengono **documentati** in conformità al processo di documentazione.
2. Viene eseguito il controllo delle modifiche sui risultati del processo di sviluppo in conformità con il processo gestione della configurazione.
3. Vengono **documentati** e **risolti** i problemi nel software in conformità con il processo di risoluzione dei problemi.

4. Vengono selezionati gli standard, modelli, strumenti e linguaggi di programmazione che sono documentati e appropriati per l'esecuzione delle attività del Processo di Sviluppo e dei processi di supporto.
5. Tutto ciò che è necessario alla manutenibilità del software viene consegnato con il prodotto.

3.2.2 Analisi dei requisiti di sistema

I **requisiti** di sistema vengono **definiti** e **documentati** nel documento "Analisi dei requisiti". Ogni requisito viene classificato in base alla **tipologia** (Funzionale, di qualità o di Vincolo) e all'**importanza** (Obbligatorio, desiderabile, opzionale), gli viene assegnato un codice univoco e vengono tracciate le fonti da cui deriva.

3.2.3 Progettazione Architettura di sistema

1. Viene progettata l'**architettura** del sistema (HW SW), tracciando tutti i requisiti di sistema agli elementi individuati (HW SW) nel documento "Specifica Tecnica"
2. L'architettura e gli elementi individuati vengono valutati nel documento "Specifica Tecnica" documentando **Tracciabilità** ai requisiti del sistema e Fattibilità degli elementi software che soddisfano i requisiti assegnati.
3. Viene redatta una versione preliminare della **documentazione** per gli utenti (Manuale utente e relativo glossario)
4. Vengono definiti i requisiti di **test**
5. Vengono fatte una o più **riunioni** con il proponente per revisionare la progettazione architetturale svolta.

3.2.4 Progettazione di dettaglio

1. I requisiti che sono stati assegnati ai componenti nella progettazione architetturale devono essere tracciati rispetto alle unità che li compongono.
2. Viene fatta la progettazione dettagliata delle **interfacce** (interne e verso l'esterno) che ne consente la codifica senza altre info.
3. Viene fatta la progettazione dettagliata del **database** che ne consente la codifica senza altre info.
4. Viene aggiornata la **documentazione** utente
5. Vengono definiti i requisiti di **test** e documentati nel documento "Specifica Tecnica"
6. Vengono programmati test che stressino i componenti
7. Viene definito come tracciare i requisiti dei componenti
8. Vengono fatte una o più riunioni con il proponente per revisionare la progettazione di dettaglio svolta.

3.2.5 Codifica e test del software

1. Viene redatto il **documento** "Specifica Tecnica" documentando tutto il codice prodotto, i test effettuati e i dati utilizzati per i test.
2. Viene ultimata la documentazione utente
3. Vengono aggiornati i requisiti di **test** se necessario
4. Viene pianificata l'**integrazione** del Software

3.2.6 Integrazione software

1. Viene sviluppato un **piano di integrazione** per integrare le unità software e i componenti software nell'elemento software. Il piano deve includere i requisiti di prova, le procedure, i dati, le responsabilità e il programma. Il piano deve essere documentato nel **documento** "Specifica Tecnica".
2. I risultati dell'integrazione vengono documentati nel documento "Specifica Tecnica"

3. Bisogna garantire che l'elemento software integrato sia pronto per il test di **qualificazione** del software.
4. Vengono fatte una o più **riunioni** con il proponente per mostrare i risultati dell'integrazione software svolta.

3.2.7 Test di qualificazione del software e Integrazione di sistema

Per ciascun **requisito** di qualificazione del sistema, deve essere elaborata e documentata una serie di **prove**, casi di prova (input, output, criteri di prova) e procedure di prova per l'esecuzione di prove di qualificazione del sistema. Il committente deve garantire che il sistema integrato sia pronto per le prove di qualificazione del sistema che provano il rispetto dei requisiti. Il sistema deve essere valutato in base a:

- **testare** la copertura dei requisiti di sistema
- **adeguatezza** dei metodi di prova e degli standard utilizzati
- **conformità** ai risultati attesi
- **fattibilità** delle prove di qualificazione del sistema
- **fattibilità** del funzionamento e della manutenzione.

3.2.8 Test di qualificazione del sistema

1. Durante una riunione viene presentato l'MVP al proponente che valuta il prodotto e decide se approvarlo o meno

3.2.9 Installazione del software

1. Viene documentata la procedura di installazione del Software, tenendo conto dei requisiti di sistema e delle possibili configurazioni del prodotto.

4 Processi di supporto

4.1 Documentazione

4.1.1 Strumento per la scrittura dei documenti

Per la redazione dei documenti del progetto, utilizziamo **Typst**, una tecnologia avanzata e versatile per la scrittura e la formattazione di testi. Grazie alla sua sintassi intuitiva e alla capacità di generare documenti di alta qualità, Typst permette di combinare efficienza e personalizzazione. Questo strumento garantisce una gestione semplificata del layout e uno stile professionale, ideale per soddisfare le esigenze di documentazione tecnica e collaborativa.

4.1.2 Struttura documento

Per permettere al gruppo di produrre documenti che siano coerenti tra loro a prescindere da quale componente si occupa della stesura, è necessario che abbiamo tutti una stessa struttura. Inoltre ci sono alcune informazioni che devono essere presenti in tutti i documenti, tra cui titolo del documento e registro delle modifiche.

Per queste ragioni è stato deciso di mantenere una struttura unica per tutti i documenti, che sarà quindi composta nel seguente modo.

4.1.2.1 Titolo del documento

Breve indicazione del contenuto del documento e della data di riferimento in caso sia utile.

4.1.2.2 Tabella versione del documento

Contiene una **raccolta** storica di tutte le **versioni** del documento, tenendo quindi traccia di chi si occupa delle modifiche, dei vari verificatori, del responsabile che approva e delle differenze rispetto alla versione precedente.

Il contenuto della tabella deve essere del tipo **LIFO**, ovvero deve contenere i numeri di versione dal maggiore al minore, rispecchiando le modifiche al documento in ordine cronologico inverso: dalla più recente alla più datata.

L'unico documento da considerare **valido** in fase di consultazione è quello con numero di **versione maggiore** e che non ha subito nessuna modifica dopo l'approvazione, ovvero in documento con numero di versione del tipo **x.0.0**, in caso fossero presenti diversi documenti che corrispondono a questo tipo di versione, quello da consultare sarà quello con **x** maggiore.

4.1.3 Versionamento dei documenti

Il versionamento dei documenti tiene traccia dei seguenti stati possibili della documentazione: **documento approvato** e **documento verificato**. Per questo motivo la versione dei documenti è del tipo "**a.b**" dove "**a**" rappresenta l'ultima approvazione del documento, "**b**" ne rappresenta l'ultima verifica. In questo modo risulta molto semplice riconoscere lo stato di un documento e capire quali sono quelli da considerare validi applicando le seguenti regole:

- **Documento Verificato**
b è un qualsiasi numero diverso da 0
- **Documento Approvato**
b è 0 e a è un qualsiasi numero diverso da 0

4.1.4 Regole cambiamento versione

Un documento può sempre essere **modificato**, a patto di **aggiungere una riga** o **modificando quella già prresente** alla tabella "versione del documento" e che tale modifica sia supportata da un relativo ticket. **Per poter procedere con una verifica il documento deve essere stato modificato dopo l'ultima verifica o approvazione. Per poter essere approvato un documento deve essere stato verificato e non aver subito alcun tipo di modifica.**

In caso di correzioni di semplici errori ortografici o di battitura, questi non necessitano né di un apposito ticket né dell'aggiunta di tali correzioni nella tabella di versione del documento.

La modifica di un documento non comporta nessuno scatto di versione, a differenza di una verifica o validazione.

In caso un documento venga modificato, vanno inserite le modifiche apportate nella colonna descrizione e l'autore di tali modifiche nella colonna autore della tabella versione del documento. La data e il numero di versione vanno aggiornati solo in caso di verifica o approvazione.

4.1.5 Nomenclatura

Al fine di poter tenere comodamente traccia di tutte le diverse versioni dei **documenti**, è necessario adottare una **convenzione sui nomi** da dare ad essi nel momento del salvataggio. Il nome del documento deve essere parlante nel contesto in cui esso è contenuto. Il **nome** del file può essere diviso in **due parti**, la **prima** contiene informazioni sul **contenuto** dello stesso, mentre la **seconda differisce** tra documenti e verbali.

4.1.5.1 Documenti

Nei documenti la seconda parte è separata dalla prima dal **simbolo** (_) e contiene la **versione** nel formato a.b.c che corrisponde alla versione attuale del documento.

4.1.5.2 Verbali

Nei verbali la seconda parte contiene la **data** in cui è avvenuto l'incontro, nel formato yyyy-mm-dd, in modo da poter essere ordinati in ordine cronologico.

Nel caso di verbali **esterni**, quando questi vengono **approvati** dal committente, deve essere inserita la dicitura (_approved) dopo la data.

4.1.6 Azioni

Ogni **modifica** alla documentazione deve essere accompagnata da un **ticket** che ne specifica le **motivazioni**. Inoltre ogni documento per poter essere considerato **valido** deve aver superato almeno una azione di modifica, verifica e approvazione. Nei punti seguenti verrà spiegato più nel dettaglio in cosa consistono queste operazioni.

4.1.6.1 Modifica

L'azione di modifica include sia la **modifica/correzione** di parti del testo che la creazione di **nuovi contenuti**. Durante la creazione dei contenuti, al fine di ottimizzare il tempo necessario alla produzione dei documenti, non è necessario preoccuparsi troppo della forma, quanto invece nell'esprimere i **contenuti in modo chiaro**. Nei passaggi successivi questi verranno corretti sia nella forma che nella sintassi.

Nella tabella delle versioni, la persona assegnata a questo compito viene indicata all'interno della colonna **Autore**, mentre le modifiche apportate si trovano nella colonna **Descrizione**.

4.1.6.2 Verifica

L'azione di verifica è composta dall'attività di verifica della documentazione specificata all'interno del processo di verifica.

4.1.6.3 Approvazione

L'azione di Approvazione è composta dall'attività Approvazione della documentazione specificata all'interno del processo di Validazione.

4.1.7 Verbalì

In quanto tipologia differente di documento, i verbalì necessitano di alcune **informazioni aggiuntive** tra cui la data dell'evento riportato, le presenze e le decisioni prese.

4.1.7.1 Presenze

Nella **prima parte** del verbale è necessario specificare i **nominativi** delle persone presenti e degli assenti in tale data, in modo da tenere traccia delle partecipazioni. Deve anche essere specificato, in caso di verbalì esterni, quali **persone** appartengono al **gruppo di lavoro** e quali all'**entità esterna** con la quale si sta discutendo.

4.1.7.2 Esiti

Ogni **decisione** presa nella riunione andrà riportata nel verbale, includendo le motivazioni e gli eventuali punti di forza e di debolezza emersi. In caso di **contrastì** che richiedono una votazione, andrà riportato l'esito di questa insieme alla tabella che specifica chi ha votato a favore, chi contro e gli astenuti.

4.1.7.3 Interni

I verbalì interni andranno sempre **approvati** dal responsabile, ma tale approvazione potrà essere discussa nella riunione successiva. In caso si riscontrino **errori**, si procederà ad identificare le parti da correggere, e alla successiva correzione.

In ogni caso, un verbale è da considerare **valido** dal momento della prima approvazione.

4.1.7.4 Esterni

I verbalì esterni necessiteranno anche dell'**approvazione** dell'entità esterna che ha partecipato all'incontro, dopo aver superato l'approvazione interna.

4.1.7.5 Esempio esplicativo

Dato il caso di un nuovo documento, la sua prima stesura avrà versione **0.0.1**.

Facendo finta che non abbia bisogno di ulteriori modifiche, il documento procede con una prima

verifica, che va a buon fine producendo il documento con versione **0.1.0**.

Procedendo verso la fase di approvazione viene evidenziato che nel documento un argomento non è trattato in modo completo; si procede quindi con una nuova modifica, arrivando alla versione **0.1.1**. A questo punto viene verificato nuovamente, senza riscontrare errori e si procede con la versione **0.2.0**.

Un'ultima fase di approvazione viene poi effettuata e superata, producendo quindi la prima versione ufficiale del documento, ovvero la versione **1.0.0**.

4.1.8 Utilizzo template

Per impostare tutti i documenti prodotti in modo **coerente** tra loro e conforme alle norme del Way of Working, sono stati prodotti dei template che richiedono che siano presenti tutti gli elementi necessari per poter produrre un documento.

I template sono impostati in modo da non dover essere modificati. Per fare ciò sono affiancati da un file esterno che comprende tutte le dipendenze che il documento richiede.

4.1.8.1 Verbalì

Per i verbalì viene utilizzato il template “interni” o “esterni” a seconda del tipo di richiesto. I contenuti necessari sono:

1. **Titolo:** titolo del documento
2. **Membri e versione:** file in formato json contenente i valori necessari per produrre la tabella di versione del documento e quella delle presenze
3. **Data:** data in cui si è svolto il verbale utilizzando la funzione date di Typst
4. **Abstract:** breve sintesi dei punti di discussioni più importanti e delle decisioni prese
5. **Odg_G:** lista degli argomenti da discutere durante l'incontro
6. **Discussione:** analisi dei punti descritti nel *odg_G* contenente i punti salienti della discussione, le decisioni prese e le motivazioni a supporto di queste
7. **Argomenti rimandati:** lista degli argomenti presenti nell' *odg_G* che non si è potuto concludere in modo soddisfacente, in termini di tempo o conoscenze, e tematiche nuove emerse durante l'incontro da discutere in futuro.

4.1.8.2 Documenti

Per i documenti viene utilizzato il template “documento” che richiede i seguenti elementi:

1. **Titolo:** titolo del documento
2. **Abstract:** breve sintesi del contenuto o scopo del documento
3. **Dati versione:** file in formato json che contiene i dati necessari alla produzione della tabella di versione del documento
4. **Contenuto:** contenuto del documento

I primi 2 punti sono presenti all'interno del frontespizio del documento in questione.

4.1.9 Glossario

Il glossario è un documento centrale per il progetto, concepito per **raccogliere** e definire in modo chiaro e univoco tutti i **termini**, abbreviazioni e acronimi utilizzati nei vari documenti. Questo strumento ha l'obiettivo di garantire una comunicazione uniforme all'interno del team e una comprensione condivisa del linguaggio tecnico e specifico del progetto. La gestione del glossario richiede la collaborazione attiva di tutti i membri del gruppo. Ogni volta che un **termine**, una parola o un acronimo viene **introdotto** in un documento, il membro che lo utilizza ha la responsabilità di **aggiornare il glossario**, aggiungendo la nuova voce con una definizione chiara e precisa. Questo approccio assicura che il glossario rimanga costantemente aggiornato e rappresenti un riferimento affidabile per tutto il team. Il glossario non solo facilita il lavoro del team durante la redazione e la revisione dei documenti, ma rappresenta anche un valore aggiunto per chiunque, interno o esterno

al progetto, debba consultare la documentazione. La sua consultazione permette di evitare fraintendimenti, accelerare i processi decisionali e mantenere coerenza terminologica nei vari aspetti del progetto.

4.1.9.1 Suddivisione delle parole

La struttura è basata su un sistema **alfabetico**, dove ogni lettera rappresenta una sezione che raccoglie le voci corrispondenti. Ogni termine incluso nel glossario è accompagnato da una definizione concisa e precisa, che spiega il suo significato nel contesto del progetto. All'inizio del documento è presente una tabella delle versioni, che tiene traccia delle modifiche apportate al glossario, con indicazioni su data, descrizione degli aggiornamenti, autore, verificatore e validatore. Questo sistema assicura che il glossario sia costantemente aggiornato e validato in modo collaborativo. La struttura include un indice dettagliato che facilita la navigazione tra le sezioni. Ogni termine è descritto con il massimo livello di dettaglio necessario per garantire una comprensione condivisa.

4.1.9.2 Aggiunta di un Termine

Per aggiungere un termine al glossario, è necessario inserire una coppia di valori, nel file Glossario.typ nella struttura dizionario (dict) denominata glossario. Il primo valore è composto dall'acronimo/abbreviazione e/o il termine per esteso, se sono presenti entrambi devono essere separati dal carattere "|". La seconda deve essere una descrizione chiara, breve e dettagliata, che spiega il significato del termine e il suo utilizzo nel contesto del progetto. La funzione inDict presente nel file functions.typ si occupa di inserire la "G" a pedice dopo ogni occorrenza del Primo dei 2 valori nei vari documenti.

4.1.10 Piano di Progetto

Di seguito verranno definite le modalità di lavoro utilizzate per la stesura del Piano di Progetto

4.1.10.1 Diagrammi di Gantt

Per la realizzazione dei diagrammi di Gantt, si utilizza il sito online <https://www.onlinegantt.com/#/gantt>. Utilizzando gli appositi bottoni si vanno a creare varie *Task_G* che riguardano lo svolgimento del progetto. Sono supportate anche la creazione di Subtask e dipendenze tra diverse *Task_G*.

4.1.10.2 Pianificazione

Per ogni periodo, il Responsabile individua una serie di **obiettivi** da perseguire, tenendo conto della loro rilevanza rispetto allo stato attuale del progetto e della loro fattibilità nel tempo disponibile.

Da questi obiettivi di carattere generale vengono ricavate **attività** operative **concrete**, rappresentate all'interno di un diagramma di Gantt, organizzate in modo da evidenziare le relazioni tra di esse nell'arco del periodo.

4.1.10.3 Preventivo ore e costi

In base alle disponibilità dei singoli membri e agli elementi definiti durante la fase di pianificazione, a ciascuno viene **assegnato** un **carico** di ore lavorative, suddiviso per ruolo, con la quale è possibile ricavare una stima dei costi per il periodo. La stima verrà dunque rappresentata mediante grafici quantitativi che illustrano l'andamento delle ore e delle spese rispetto al budget complessivo, fornendo proiezioni sulla situazione prevista al termine del periodo.

4.1.10.4 Consuntivo delle ore e dei costi

Al termine del periodo, il Responsabile **verifica** il lavoro svolto confrontandolo con quanto pianificato, sia dal punto di vista contabile che rispetto agli obiettivi stabiliti. Viene effettuata un'analisi comparativa tra la stima iniziale delle ore e dei costi e i dati effettivi registrati a fine

periodo. Questo confronto permette di aggiornare lo stato attuale del budget e del monte ore, fornendo una base di riferimento per la pianificazione successiva.

4.1.10.5 Retrospective

Al termine del periodo, si svolge una **riflessione** collettiva per valutare i risultati ottenuti, verificando gli aspetti quantitativi, ovvero il livello di avanzamento degli obiettivi e delle attività rispetto alle previsioni. Questa analisi è fondamentale per assicurare che l'impiego delle risorse sia in linea con gli obiettivi, per individuare eventuali miglioramenti da applicare nei processi futuri.

4.1.11 Analisi dei Requisiti

Di seguito le istruzioni per redigere il documento.

- **Scrittura dei casi d'uso:** Ogni membro del gruppo si occupa di una parte dei casi d'uso sviluppati durante il precedente incontro con l'azienda, lavorando in modo indipendente.
- **Prima stesura:** Dopo aver sviluppato individualmente i casi d'uso (UC_G), si procede con l'unione del lavoro svolto e si realizza una prima stesura del documento che include tutti gli UC_G e i requisiti.

4.1.11.1 Casi d'uso (UC_G)

I casi d'uso servono a definire uno scenario in cui uno o più attori interagiscono con il sistema. Nel documento di analisi dei requisiti, i casi d'uso saranno identificati nel seguente modo:

- UC_G : acronimo di "Use Case".
- N°: un numero progressivo che facilita l'identificazione di un singolo caso d'uso.

Un esempio è: UC_G [Numero caso d'uso].[Sottocaso].

Per sviluppare un singolo caso d'uso è necessario che siano presenti i seguenti punti:

- **Descrizione:** Una breve spiegazione del caso d'uso.
- **Scenario:** Il contesto specifico in cui si svolge l'interazione.
- **Attore principale:** L'utente o il sistema che avvia l'interazione.
- **Precondizioni:** Le condizioni che devono essere soddisfatte prima dell'inizio del caso d'uso.
- **Postcondizioni:** Lo stato del sistema dopo che il caso d'uso è stato completato con successo.
- **Scenari alternativi:** Eventuali variazioni o eccezioni rispetto al flusso principale del caso d'uso.

4.1.11.2 Requisiti

I requisiti rappresentano le specifiche necessità, funzionalità e vincoli che il sistema deve soddisfare per rispondere alle esigenze degli utenti o, nel caso del progetto, del proponente. I requisiti verranno definiti e identificati in questo modo:

R[Tipologia]-[Importanza].[Codice]

Di cui:

- **R:** acronimo di "Requisito".
- **Tipologia:** tipologia di requisito
 - **F:** Funzionale.
 - **Q:** Qualità.
 - **V:** Vincolo.
- **Classificazione:** identifica l'importanza del requisito:
 - 1: Obbligatorio.
 - 2: Opzionale.
 - 3: Desiderabile.

- **Codice:** codice univoco del requisito il quale presenta, in alcuni casi, dei sottocasi identificati con un punto seguito dal rispettivo codice del sottocaso.

4.2 Gestione della configurazione

1. Nel branch master devono comparire solo documenti **validati** e codice **verificato**
2. Prima di ogni release del codice sul branch master viene eseguito il test statico del codice come è definito nel processo di verifica attraverso una pull-request che ne causa la notifica agli altri membri del team.

4.2.1 Github_G repository_G

Qui di seguito vengono raccolte varie informazioni riguardanti l'utilizzo e l'attuale configurazione della repository_G di Github_G, in particolare per il caricamento e l'organizzazione della documentazione.

4.2.1.1 Struttura

Il repository_G si compone di tre rami("branch"):

1. **master:** ramo che contiene tutti i **file** riguardanti il progetto ed è composto nel seguente modo:
 - **Assets:** contenente file aggiuntivi, quali immagini e loghi riguardanti il gruppo Rod2Cod ed il progetto
 - **Documentazione:** contenente tutti file riguardanti la documentazione del progetto
 - **Candidatura:** contenente i vari file riguardanti le candidature presentate e da presentare per il progetto in essere
 - **Verbali:** contenente i file redatti in seguito a riunioni del gruppo. Al suo interno due cartelle distinguono fra verbali **interni** ed **esterni**.
 - Una **cartella** per ogni tipo di **documento** presentato (es. WoW, PdP, AdR,...) con al suo interno il documento in formato **pdf**, con un nome il quale presenti alla fine un **underscore**(_) seguito dal numero di **versione** (es. PdQ_v1.0.0.pdf). In aggiunta a ciò, è presente una sottocartella **Deprecated** che conterrà i file con versioni deprecate del documento.
2. **gh-pages_G:** ramo che contiene tutti file di configurazione necessari a **Jekyll_G** e alle Github_G Actions per poter impostare la Webpage della repository_G.
3. **docs_source_file:** ramo contenente i file **sorgente** di Typst dei documenti presenti nella repository_G.

4.2.1.2 Utilizzo

4.2.1.2.1 Caricamento documenti

Per il caricamento di documenti e/o modifiche a questi ultimi, come prima cosa, si effettua la **copia** della repository_G in **locale**(`git clone <nome_repo>`) e ci si sposta dentro la cartella così creata.

Successivamente ci si sposta nel ramo **master**(`git checkout master`). Qui si effettuano le modifiche/aggiunte che servono, stando attenti a rispettare il **pattern** per l'inserimento di nuovi documenti (`nome_documento/nome_documento_versione.pdf`) e lo spostamento di quelli deprecate nella rispettiva cartella. Successivamente si vanno a **modificare** anche i file di configurazione di **Jekyll_G** e delle **Github_G Actions** per includere eventuali nuove cartelle e pdf all'interno della Webpage.

IMPORTANTE: i documenti devono essere verificati e approvati prima di essere caricati all'interno della repository_G di Github_G.

Infine, dopo aver **aggiunto** le modifiche effettuate(`git add .`) e aver fatto il **commit** di queste ultime(`git commit -m <messaggio_commit>`), si effettua un'operazione di **push**(`git push <destinazione> <master>`) in modo da caricarle anche nella **repository_G** presente su **Github_G**.

4.2.1.2.2 Caricare modifiche alla Webpage

Per il caricamento di modifiche e/o aggiunte alla Webpage, anche qui, come prima cosa, si effettua la **copia** della **repository_G** in **locale**(`git clone <nome_repo>`), ci si sposta dentro la cartella così creata e nel ramo di interesse. Successivamente ci si sposta nel ramo **gh-pages_G**(`git checkout gh-pagesG`) e si effettuano le modifiche/aggiunte che servono. Infine, dopo aver **aggiunto** le modifiche effettuate(`git add .`) e aver fatto il **commit** di queste ultime(`git commit -m <messaggio_commit>`), si effettua un'operazione di **push**(`git push <destinazione> <gh-pagesG>`) in modo da caricarle anche nella **repo_G** presente su **Github_G**.

4.2.1.3 Sistema di ticketing

Il sistema di ticketing utilizzato per la gestione del progetto è quello **integrato** in **Git_GHub_G**, una soluzione flessibile ed efficiente che consente di monitorare le attività, segnalare problemi e pianificare il lavoro in maniera collaborativa. Questo strumento, fortemente integrato con il **repository_G**, permette al team di organizzare e **tracciare** ogni aspetto dello sviluppo in modo strutturato. Alla base del sistema di ticketing di **Git_GHub_G** ci sono le **Issues**, che rappresentano il cuore della gestione delle attività. Ogni issue è caratterizzata da un titolo sintetico che descrive il problema o la richiesta, una descrizione dettagliata per fornire il contesto e le istruzioni necessarie, e un insieme di etichette che aiutano a categorizzare e identificare la tipologia dell'attività. Le issues possono essere assegnate a uno o più membri del team, identificando così i responsabili della gestione e risoluzione del compito. Inoltre, è possibile associare ogni issue a obiettivi più ampi, le milestone, che consentono di raggruppare attività correlate e monitorare i progressi generali del progetto.

4.2.1.3.1 Creazione del ticket

Il processo di utilizzo del sistema prevede la **creazione** di una nuova **issue** accedendo alla relativa sezione del **repository_G**, dove viene compilato il titolo e fornita una descrizione chiara e dettagliata dell'attività. Durante questa fase, è possibile aggiungere etichette per indicare la categoria o la priorità e assegnare la issue ai membri del team più adatti. Una volta creata, l'issue diventa il **riferimento** centrale per tutte le attività correlate. I **progressi** vengono **monitorati** attraverso aggiornamenti nei commenti, eventuali modifiche alla descrizione o aggiunte di nuove etichette, in modo da riflettere lo stato più recente del lavoro. Una volta creata la issue, si passa sulla project board, dove in automatico verrà visualizzato il ticket con tutti i dettagli spiegati precedentemente.

4.2.1.3.2 Project board

Le Project Board di **Git_GHub_G** sono uno strumento visivo potente e versatile, progettato per gestire e **organizzare il lavoro** all'interno di un **progetto**. Ispirate al metodo Kanban, queste bacheche offrono una rappresentazione chiara e immediata del flusso di lavoro, consentendo al team di monitorare lo stato di avanzamento delle attività, identificare eventuali ostacoli e garantire un'efficace collaborazione. Nelle sezioni successive vedremo come è stato deciso di suddividere le board per la gestione dei ticket.

4.2.1.3.2.1 Project board per i documenti

La project board per i documenti viene suddivisa nelle seguenti colonne:

- **Non Assegnato:** Questa colonna raccoglie i ticket che non sono stati ancora assegnati a nessun membro del team. Questi elementi rappresentano attività o problemi che devono essere valutati o presi in carico.

- **Assegnato:** I ticket in questa colonna sono stati affidati a uno o più membri del team, ma il lavoro su di essi non è ancora iniziato. Questo indica che l'attività è stata identificata e attende di essere avviata.
- **In Corso:** Qui vengono spostati i ticket sui quali si sta attivamente lavorando. È una colonna fondamentale per monitorare le attività in corso e tenere traccia dei progressi del team.
- **In Revisione:** Questa colonna contiene i ticket per i quali il lavoro è stato completato, ma che devono ancora essere revisionati. Include quindi la verifica della correttezza delle modifiche inserite, sia da un punto di vista semantico che sintattico.
- **In Approvazione:** I ticket in questa colonna sono in fase di approvazione finale. È un passaggio che coinvolge il responsabile che deve accettare o rifiutare il lavoro svolto.
- **In Attesa di Approvazione Esterna:** Questa colonna rappresenta attività che richiedono l'approvazione di soggetti esterni al team, ovvero l'azienda proponente. Indica che il lavoro è in attesa di feedback o validazione esterna.
- **Completato:** I ticket in questa colonna sono stati completati con successo. Indica che l'attività è stata svolta, revisionata e approvata, ed è ora considerata chiusa.

4.3 Assicurazione della qualità

Il piano di qualifica deve includere

- **standard** di qualità, metodologie, procedure e strumenti per lo svolgimento delle attività di assicurazione della qualità (o i loro riferimenti nella documentazione ufficiale dell'organizzazione)
- le procedure di **revisione** e coordinamento dei contratti
- **procedure** per l'identificazione, la raccolta, l'archiviazione, il mantenimento e lo smaltimento dei registri di qualità; Risorse, calendario e responsabilità per lo svolgimento delle attività di assicurazione della qualità
- attività e compiti selezionati dai **processi** di supporto

4.3.1 Piano di Qualifica

Di seguito verranno definite le modalità di lavoro utilizzate e le metodologie di sviluppo delle varie sezioni del Piano di Qualifica.

4.3.1.1 Metriche Utilizzate

Le metriche utilizzate all'interno del Piano di Qualifica, sono state selezionate dopo un'analisi delle necessità del progetto. Esse sono state ricavate da varie fonti:

- **Progetti passati**, in quanto molti progetti, avendo delle basi di lavoro simili, permettono di utilizzare metriche comuni. Inoltre, essi sono serviti anche come suggerimenti e **aiuti** per la scelta di successive metriche da integrare;
- **Documenti** resi disponibili dal professore;
- Ricerche in **internet**, in aggiunta a quanto consigliato all'interno dei documenti resi disponibili.

4.3.1.2 Struttura del documento

Il Piano di Qualifica è strutturato nelle seguenti sezioni principali:

- **Introduzione:** definisce lo scopo del documento, il glossario e i riferimenti normativi e informativi;
- **Qualità di Processo:** descrive le metriche adottate per valutare la qualità dei processi primari, di supporto e organizzativi secondo lo standard ISO/IEC/IEEE 12207:1995;

- **Qualità di Prodotto:** specifica le metriche utilizzate per valutare la qualità della documentazione e del software;
- **Specifica dei Test:** definisce i test necessari per verificare la conformità del prodotto ai requisiti;
- **Resoconto delle Attività di Verifica:** presenta i risultati delle verifiche effettuate sulla documentazione e sui processi.

4.3.1.3 Qualità di Processo

La qualità di processo si fonda sull'idea che, per ottenere un prodotto che soddisfi determinati **standard** di qualità, è necessario che i **processi** che lo vanno a creare, vengano sottoposti a **controlli** periodici con lo scopo ultimo di ottimizzarli.

Il concetto di qualità di processo viene dunque applicato all'intero spettro di **attività, pratiche e metodi** utilizzati lungo l'intero ciclo di vita del software. In sintesi, la qualità di processo si pone l'obiettivo di rendere la qualità una **parte integrante** del prodotto, garantendo che sia **costruita** nel processo stesso e non sia solo un obiettivo secondario.

Per garantire il raggiungimento degli standard qualitativi prefissati, il team ha adottato un approccio strutturato alla qualità di processo, che include:

- **Processi primari**, che comprendono la gestione delle risorse e lo sviluppo del software;
- **Processi di supporto**, focalizzati sulla verifica e gestione della qualità;
- **Processi organizzativi**, finalizzati a migliorare la gestione interna del team e delle attività.

Metriche di qualità prese in esame:

- *Earned Value_G (MPC-EV_G)*: misura il valore del lavoro completato rispetto al piano;
- *Planned Value_G (MPC-PV_G)*: rappresenta il valore pianificato di un'attività in un determinato momento;
- *Actual Cost_G (MPC-AC_G)*: indica il costo effettivo sostenuto per il lavoro svolto;
- *Cost Performance Index_G (MPC-CPI_G)*: valuta l'efficienza dei costi del progetto;
- *Estimate At Completion_G (MPC-EAC_G)*: stima il costo totale previsto del progetto;
- *Estimate To Complete_G (MPC-ETC_G)*: calcola i costi rimanenti per completare il progetto;
- *Schedule Variance_G (MPC-SV_G)*: misura la differenza tra il lavoro pianificato e quello effettivamente completato;
- *Budget Variance_G (MPC-BV_G)*: verifica eventuali scostamenti rispetto al budget iniziale;
- *Satisfied Obligatory Requirements_G (MPC-SOR_G)*: verifica che tutti i requisiti obbligatori siano soddisfatti;
- *Requirements Stability Index (MPC-RSI)*: misura la stabilità dei requisiti nel tempo.

4.3.1.4 Qualità di Prodotto

La qualità del prodotto si concentra sulla valutazione del software sviluppato, ponendo particolare attenzione a caratteristiche come usabilità, funzionalità, affidabilità, manutenibilità e, più in generale, alle prestazioni complessive del sistema.

L'obiettivo principale è garantire che il software non solo soddisfi le funzionalità richieste dal cliente e operi correttamente, ma che lo faccia rispettando specifici standard di qualità.

Gli aspetti valutati includono:

- **Documentazione:** deve essere chiara, leggibile e priva di errori;
- **Software:** deve soddisfare tutti i requisiti concordati con il proponente.

Metriche di qualità prese in esame:

- *Leggibilità (MPD-IG)*: il contenuto dei documenti deve essere comprensibile all'utente;
- *Correttezza Linguistica (MPD-EO)*: i documenti devono essere privi di errori grammaticali;

- Funzionalità (**MPD-CROP, MPD-CRD, MPD-CROB**): capacità del prodotto di soddisfare tutti i requisiti individuati nell'Analisi dei Requisiti;
- Usabilità (**MPD-SU, MPD-TA**): capacità di essere comprensibile e di facile utilizzo per l'utente;
- Efficienza (**MPD-TMR**): capacità di svolgere un compito nel minor tempo possibile;
- Affidabilità (**MPD-GE**): capacità del prodotto di gestire correttamente gli errori;
- Manutenibilità (**MPD-CCM**): capacità di permettere future correzioni e modifiche;
- Copertura dei Test (**MPD-CTS**): capacità del prodotto software di superare i test.

4.3.1.5 Specifica dei Test

Il Piano di Qualifica prevede l'esecuzione di test dettagliati per verificare il corretto funzionamento del software. I test sono suddivisi in:

- **Test di Unità**: verificano il corretto funzionamento delle singole componenti;
- **Test di Integrazione**: verificano la corretta interazione tra le diverse componenti;
- **Test di Sistema**: verificano il corretto funzionamento delle singole funzionalità;
- **Test di Accettazione**: garantiscono che il prodotto soddisfi i requisiti aziendali.

Ogni test è identificato da un codice e uno stato, che può essere **Implementato (I)** o **Non Implementato (NI)**.

4.3.1.6 Resoconto delle Attività di Verifica

Per assicurare la qualità del progetto, il team esegue controlli periodici e raccoglie dati sulle performance tramite:

- **Verifica della documentazione**, per individuare eventuali errori o incongruenze;
- **Monitoraggio dei processi**, attraverso il confronto tra obiettivi e risultati effettivi;
- **Analisi delle metriche di progetto**, per valutare l'aderenza agli standard prefissati.

4.3.2 Assicurazione prodotti

1. Le **tempistiche** verranno rispettate, documentate e devono essere coerenti con il regolamento del progetto didattico come anche il SW.

4.3.3 Assicurazione Processi

1. I **processi** necessari al completamento del progetto vengono identificati, definiti e documentati nel documento Way of Working prima di essere istanziati
2. Viene **adottato** il framework standard iso/iec 12207
3. Durante l'attività di retrospective vengono **identificate** le inefficienze e i problemi nei processi e poi proposti miglioramenti.
4. Per ogni processo identificato come automatizzabile, viene **valutata** la convenienza dell'automazione in base alla dimensione e alla complessità del progetto. Se ritenuto vantaggioso, il processo viene implementato tramite una *GitHub_G* Action o altri strumenti appropriati.

4.4 Risoluzione dei problemi

Deve essere stabilito un **processo** a circuito chiuso per la **risoluzione** dei **problemi** nei prodotti e nelle attività software, garantendo la segnalazione, l'analisi delle cause, la categorizzazione, il monitoraggio delle tendenze e la verifica delle soluzioni, con la documentazione di ogni problema rilevato.

4.5 Verifica

1. Tutti i documenti vengono **verificati** dopo ogni modifica da un componente del gruppo diverso dall'autore
2. Tutto il software prodotto deve essere **verificato** ad ogni modifica

4.5.1 Retrospective

Durante l'azione di retrospective vengono identificati, documentati, analizzati i **problemi** incontrati ed elaborate modalità di risoluzione. Ciò avviene per verificare l'efficacia dei processi adottati e al contempo per renderli flessibili abbastanza da poterli adattare ai cambiamenti possibili.

4.5.2 Verifica documentazione

La verifica della documentazione consiste nella lettura del testo prodotto e nell'eventuale **correzione di errori** ortografici rilevati nonché nella modifica della struttura del documento al fine di rendere l'argomento trattato di **facile consultazione**.

Nel caso in cui certi punti dovesse risultare poco chiari, si procederà ad una o più modifiche successive al fine di esprimere tali concetti in modo migliore.

Nella tabella delle versioni, la persona assegnata a questo compito viene indicata all'interno della colonna **Verificatore**.

4.5.3 Verifica codice

1. Test **statici**:

- prima di ogni release tutto il codice che ha subito **modifiche** viene verificata da due componenti del gruppo leggendo il codice e comprendendone il funzionamento, ricercando eventuali errori.
- **code coverage**
- **cyclomatic complexity**

2. Test **dinamici**:

- il ogni test dinamico con esito negativo va **documentato** nel documento "Specifica tecnica"
- il **codice** viene **testato** tramite test di unità nella misura definita nel documento piano di Qualifica. Ogni test di unità deve stressare il componente testando il suo funzionamento in condizioni normali, limite e in cui ci si aspetta che non funzioni.
- viene testata l'**integrazione** dei vari componenti come previsto dall'attività Integrazione Software del processo di sviluppo.

4.6 Validazione

4.6.1 Validazione documentazione

consiste nella **lettura** del documento prodotto e nella **verifica della conformità** con i punti che questo dovrebbe trattare.

In caso vengano rilevati errori nei concetti espressi/mancanti, il documento tornerà alla fase di modifica.

Nella tabella delle versioni, la persona assegnata a questo compito viene indicata all'interno della colonna **Validatore**.

4.6.2 Validazione Prodotto

Durante una riunione viene **presentato** l'MVP al proponente che valuta il prodotto e decide se approvarlo

4.6.3 Revisione congiunta

Le revisioni congiunte con il proponente sono programmate periodicamente ogni mese, l'*odg* è definito in precedenza e approvato dal proponente

4.6.4 Audit

Come definito nel "Regolamento del progetto didattico" il committente **valuta** il prodotto e i processi durante gli incontri per la presentazione dell'RTB e PB.

4.6.5 Risoluzione dei problemi

I **problemi** rilevati e le relative **azioni** di mitigazione vengono **documentati** nella sezione retrospective del documento Piano di Progetto, essi vengono anche tracciati ai rischi identificati nell'analisi dei rischi.

5 Processi organizzativi

5.1 Gestione

Il processo di gestione comprende **attività** generiche applicabili a qualsiasi processo, tra cui gestione del prodotto, del progetto e dei compiti. Si articola in cinque fasi:

1. **inizializzazione e definizione** dell'ambito
2. **pianificazione**
3. **esecuzione e controllo**
4. **revisione e valutazione**
5. **chiusura**

5.1.1 Inizializzazione

Definisce i requisiti e verifica la fattibilità del processo.

5.1.2 Pianificazione

Prevede la preparazione dei piani operativi, incluse tempistiche, risorse, assegnazione di compiti, analisi dei rischi e misure di controllo qualità.

5.1.3 Esecuzione e controllo

Implementa i piani, monitora l'andamento, gestisce problemi e aggiornamenti, e garantisce la documentazione delle modifiche.

5.1.4 Revisione e valutazione

Verifica la conformità dei prodotti e dei processi agli obiettivi e ai requisiti stabiliti.

5.1.5 Chiusura

Determina il completamento del processo e archivia i risultati e le registrazioni come previsto dal contratto.

5.2 Riunioni

5.2.1 Obiettivi delle Riunioni

Le riunioni rappresentano un momento fondamentale per il buon andamento del progetto, offrendo un'occasione per il team di ritrovarsi e fare il punto della situazione. Durante questi incontri, ciascun membro ha la possibilità di presentare al gruppo i progressi compiuti, condividendo risultati e attività completate. Le riunioni sono inoltre il contesto ideale per prendere decisioni sui vari aspetti del progetto, garantendo che ogni scelta sia presa in base ad un confronto tra tutti i membri. Infine, questi momenti consentono al team di allinearsi sugli obiettivi e sulle priorità del periodo successivo, definendo con chiarezza il percorso da seguire per raggiungere i traguardi prefissati.

5.2.2 Pianificazione delle Riunioni

È stato stabilito che le riunioni si terranno con cadenza settimanale, rappresentando un appuntamento fisso per garantire continuità e organizzazione. Il giorno designato per tali incontri è il lunedì, con orario d'inizio fissato alle ore 16:00. Tuttavia, si prevede la possibilità di apportare variazioni all'orario o alla data qualora emergano necessità specifiche, assicurando così una gestione flessibile e condivisa in funzione delle esigenze collettive.

5.2.3 Ruoli e Responsabilità

Durante le riunioni, chi attualmente veste il ruolo di responsabile, sarà considerato il moderatore della riunione, ovvero la figura incaricata di guidare la discussione e garantire il corretto svolgimento degli incontri. Dovrà inoltre. Il moderatore si occupa di presentare e affrontare, punto per punto, gli argomenti all'*ordine del giorno*_G, assicurando così un dialogo produttivo e una gestione efficace delle decisioni da prendere. Nel corso delle riunioni i ruoli potranno essere soggetti a rotazione, e sarà designato anche un segretario. Quest'ultimo avrà il compito di redigere il verbale secondo le modalità precedentemente definite, assicurando una documentazione accurata e completa delle discussioni e delle decisioni adottate. Oltre a redigere il verbale, il segretario avrà anche il compito di creare le issue relative alle decisioni prese durante la riunione. Al fine di tracciare al meglio tali issue, il segretario ne riporterà il numero all'interno del verbale, con le decisioni che hanno portato alla creazione delle stesse.

5.2.4 Svolgimento delle Riunioni

Le riunioni si svolgono secondo una procedura strutturata, finalizzata a garantire ordine e produttività. Il moderatore si occupa di introdurre e leggere, punto per punto, l'*ordine del giorno*_G (*ODG*_G). All'inizio della riunione, vengono assegnati i ruoli necessari: uno dei partecipanti sarà incaricato di tenere il diario di bordo, mentre un altro assumerà il ruolo di segretario verbalizzante, responsabile della successiva redazione del verbale. La discussione procede seguendo l'ordine degli argomenti, con il moderatore che guida il dialogo e si assicura che ogni decisione venga adeguatamente analizzata e condivisa. Ogni volta che viene presa una decisione, si dedica un momento di pausa per consentire al segretario verbalizzante di registrare correttamente quanto emerso, garantendo così una documentazione chiara e precisa dell'incontro.

5.2.5 Documentazione

Per ogni riunione sarà redatto un verbale, il cui scopo è documentare in modo dettagliato tutti i punti trattati all'*ordine del giorno*_G. Il verbale riporterà in maniera chiara gli argomenti discussi, le decisioni adottate e le relative motivazioni e le issue collegate, garantendo così una traccia completa e consultabile delle attività svolte. Questa pratica non solo consente di mantenere un registro organizzato delle decisioni prese, ma permette anche ai membri assenti di recuperare agevolmente quanto discusso e stabilito durante la riunione.

5.2.6 Strumenti e Tecnologie

Poiché la maggior parte delle riunioni si svolge in modalità online, la piattaforma utilizzata per gli incontri è Microsoft Teams. Questo strumento oltre a consentire l'organizzazione di videoconferenze condivise con tutti i membri del gruppo, offre anche funzionalità avanzate per la gestione collaborativa. Attraverso Teams è quindi possibile condividere file, utilizzare chat di gruppo e accedere a strumenti integrati per agevolare la comunicazione e il coordinamento tra i partecipanti, garantendo così un ambiente di lavoro efficiente e centralizzato.

5.3 Tracciamento ore

Il tracciamento delle ore lavorative è un elemento fondamentale per **monitorare il tempo** dedicato alle diverse attività del progetto. Per questo scopo, utilizziamo un foglio condiviso in formato Excel, che funge da registro centralizzato per la rendicontazione delle ore lavorate da ciascun membro del team. Ogni partecipante è responsabile di aggiornare il foglio con le proprie ore, indicando con precisione il numero di ore produttive dedicate. Questo approccio permette di avere una visione chiara dell'impegno complessivo del team e di analizzare la distribuzione del tempo tra le varie fasi del progetto. Il foglio Excel è condiviso attraverso Microsoft Teams, il quale permette, come definito sopra, di assicurare una gestione collaborativa e in tempo reale.

5.4 Infrastrutturale

Viene utilizzato **git** come strumento di **versionamento** e **github**_G come **repository**_G, inoltre viene utilizzato **github**_G-pages per esprimere la documentazione e **github**_G-action per l'automatizzazione dei processi.

5.5 Miglioramento

Il processo di miglioramento ha lo scopo di stabilire, valutare, misurare, controllare e migliorare i **processi** del ciclo di vita del software.

Si articola in tre fasi:

1. **Stabilizzazione** del processo: definizione, documentazione e controllo dei processi aziendali relativi al ciclo di vita del software.
2. **Valutazione** del processo: sviluppo e applicazione di procedure di valutazione, con revisioni periodiche per garantirne l'efficacia.
3. **Miglioramento** del processo: aggiornamento dei processi in base ai risultati delle valutazioni, analisi dei dati storici e tecnici per individuare punti di forza e debolezza, e raccolta di dati sui costi della qualità per ottimizzare la gestione.

5.6 Formazione

1. Ogni membro del team è **responsabile** della propria **formazione** sugli strumenti e sulle tecnologie utilizzate nel progetto.
2. In caso di difficoltà o dubbi, è incentivata la **condivisione** delle conoscenze all'interno del team.
3. I **progressi** nell'apprendimento possono essere **discussi** durante gli incontri periodici per favorire l'allineamento delle competenze.