



7zpus.swe@gmail.com

## Norme di Progetto

---

### Indice

<b>1</b>	<b>Introduzione</b>	<b>6</b>
1.1	Scopo . . . . .	6
1.2	Glossario . . . . .	6
1.3	Riferimenti . . . . .	6
1.3.1	Riferimenti Normativi . . . . .	6
1.3.2	Riferimenti Informativi . . . . .	6
<b>2</b>	<b>Processi Primari</b>	<b>7</b>
2.1	Processo di Fornitura . . . . .	7
2.1.1	Attività di processo . . . . .	7
2.1.1.1	Avvio . . . . .	7
2.1.1.2	Preparazione della proposta di fornitura . . . . .	7
2.1.1.3	Accordo . . . . .	7
2.1.1.4	Pianificazione . . . . .	8
2.1.1.5	Esecuzione e controllo . . . . .	8
2.1.1.6	Verifica e validazione . . . . .	8
2.1.1.7	Consegna e terminazione . . . . .	8
2.1.1.8	Accordi con l'azienda proponente . . . . .	8
2.1.2	Documentazione fornita . . . . .	9
2.1.2.1	Analisi dei requisiti . . . . .	9
2.1.2.2	Glossario . . . . .	9
2.1.2.3	Piano di progetto . . . . .	9
2.1.2.4	Piano di qualifica . . . . .	10
2.1.2.5	Lettera di presentazione . . . . .	10
2.1.3	Strumenti . . . . .	10
2.2	Processo di sviluppo . . . . .	10
2.2.1	Attività di processo . . . . .	10
2.3	Processo operativo . . . . .	10
2.3.1	Pianificazione operativa . . . . .	11

2.3.2	Gestione dei rischi operativi . . . . .	11
2.4	Processo di manutenzione . . . . .	12
2.4.1	Manutenzione correttiva . . . . .	12
2.4.2	Manutenzione adattiva . . . . .	12
2.4.3	Manutenzione preventiva . . . . .	12
2.4.4	Identificazione della necessità di manutenzione . . . . .	13
2.4.5	Verifica e validazione . . . . .	13
2.4.6	Metriche . . . . .	13
<b>3</b>	<b>Processi di Supporto</b>	<b>13</b>
3.1	Processo di documentazione . . . . .	13
3.1.1	Attività: Pianificazione della documentazione . . . . .	13
3.1.1.1	Procedure di Pianificazione . . . . .	14
3.1.1.2	Strumenti di Pianificazione . . . . .	14
3.1.2	Attività: Produzione della documentazione . . . . .	14
3.1.2.1	Procedure di Produzione . . . . .	15
3.1.2.1.1	Denominazione e datazione . . . . .	17
3.1.2.2	Strumenti di Produzione . . . . .	17
3.1.3	Attività: Revisione e Approvazione . . . . .	18
3.1.3.1	Procedure di Revisione e Approvazione . . . . .	18
3.1.3.2	Strumenti di Revisione e Approvazione . . . . .	19
3.2	Processo di Gestione della Configurazione . . . . .	19
3.2.1	Attività: Identificazione della Configurazione . . . . .	19
3.2.1.1	Procedure: Identificazione degli SCIs . . . . .	19
3.2.1.1.1	Issue e SubIssue . . . . .	20
3.2.1.2	Strumenti di Identificazione . . . . .	21
3.2.2	Attività: Versionamento e Identificazione . . . . .	21
3.2.2.1	Procedure: Standard per le Branch . . . . .	21
3.2.2.2	Strumenti di Versionamento . . . . .	21
3.2.3	Attività: Controllo delle Configurazioni . . . . .	21
3.2.3.1	Procedure: Smart Commit . . . . .	21
3.2.3.2	Strumenti . . . . .	21
3.2.4	Registrazione delle Configurazioni . . . . .	22
3.2.4.1	Procedure: Registrazione delle Modifiche . . . . .	22
3.3	Processo di garanzia della qualità . . . . .	23
3.4	Processo di verifica . . . . .	23
3.5	Processo di validazione . . . . .	23
3.6	Processo di revisione congiunta . . . . .	23
3.7	Processo di risoluzione dei problemi . . . . .	23
3.8	Gestione della qualità . . . . .	23
<b>4</b>	<b>Processi Organizzativi</b>	<b>23</b>
4.1	Gestione . . . . .	23
4.2	Processo di Infrastruttura . . . . .	23

4.2.1	Attività: Implementazione . . . . .	23
4.2.1.1	Procedure di Implementazione . . . . .	23
4.2.1.2	Strumenti di Implementazione . . . . .	24
4.2.2	Attività: Creazione . . . . .	24
4.2.2.1	Procedure di Creazione . . . . .	24
4.2.2.2	Strumenti di Creazione . . . . .	24
4.2.2.3	Jira . . . . .	24
4.2.2.4	GitHub . . . . .	24
4.2.3	Attività: Manutenzione . . . . .	25
4.2.3.1	Procedure di Manutenzione . . . . .	25
4.2.3.2	Strumenti di Manutenzione . . . . .	25
4.2.3.3	Criteri di Scelta degli Strumenti . . . . .	26
<b>5</b>	<b>Knowledge Base</b>	<b>26</b>
5.0.0.1	GitHub . . . . .	26
5.0.0.1.1	GitHub Action . . . . .	27
5.0.0.2	Jira . . . . .	27
5.0.0.2.1	Automation . . . . .	27
5.0.0.2.2	Smart Dashboard . . . . .	27
5.0.0.2.3	Jira Smart Commit . . . . .	28
5.0.0.3	VSCoDe . . . . .	28
5.0.0.3.1	Atlassian: Jira, Rovo Dev, Bitbucket . . . . .	29
5.0.0.3.2	GitHub Pull Requests and Issues . . . . .	29
5.0.0.4	Google Presentazioni . . . . .	29
5.0.0.5	Google Drive . . . . .	29
5.0.0.6	Google Mail . . . . .	29
5.0.0.7	Discord . . . . .	29
5.0.0.8	Whatsapp . . . . .	30
5.1	Processo di Miglioramento . . . . .	30
5.1.1	Attività: Analisi degli Incidenti . . . . .	30
5.1.1.1	Procedure di Analisi degli Incidenti . . . . .	30
5.1.2	Attività: Monitoraggio degli Strumenti . . . . .	30
5.1.2.1	Procedure di Monitoraggio . . . . .	31
5.1.2.2	Strumenti di Monitoraggio . . . . .	31
5.1.3	Attività: Modifiche all'Infrastruttura . . . . .	31
5.1.3.1	Procedure di Modifica . . . . .	31
5.1.3.2	Strumenti di Modifica . . . . .	32
5.1.4	Frequenza dei Cicli di Miglioramento . . . . .	32
5.1.5	Metriche di Miglioramento . . . . .	32
5.2	Processo di Formazione . . . . .	32
5.2.1	Attività: Pianificazione della Formazione . . . . .	33
5.2.1.1	Procedure di Pianificazione . . . . .	33
5.2.2	Attività: Erogazione della Formazione . . . . .	33
5.2.2.1	Procedure di Erogazione . . . . .	33

5.2.3	Attività: Valutazione della Formazione . . . . .	33
5.2.3.1	Procedure di Valutazione . . . . .	34
5.2.3.2	Strumenti di Valutazione . . . . .	34
<b>6</b>	<b>Metriche della qualità</b>	<b>34</b>
6.1	Attività per la qualità . . . . .	34
6.2	Procedure per la qualità . . . . .	35
6.2.1	Qualità di Processo . . . . .	35
6.2.1.1	Processo di fornitura . . . . .	35
6.2.1.2	Processo di Sviluppo . . . . .	36
6.2.1.3	Processo di Documentazione . . . . .	37
6.2.1.4	Processo di Configurazione . . . . .	37
6.2.1.5	Processo di Verifica . . . . .	37
6.2.1.6	Processi Organizzativi) . . . . .	38
6.2.2	Qualità di Prodotto . . . . .	38
6.2.2.1	Funzionalità . . . . .	38
6.2.2.2	Affidabilità . . . . .	39
6.2.2.3	Usabilità . . . . .	39
6.2.2.4	Efficienza . . . . .	40
6.2.2.5	Manutenibilità . . . . .	40
6.3	Strumenti . . . . .	40

## Elenco delle figure

1	Flusso del processo di documentazione . . . . .	15
2	Creazione del branch di lavoro tramite estensione Jira in VSCode . . . . .	16
3	Creazione della PR verso l'issue branch <i>in_lavorazione</i> . . . . .	17
4	Commit della PR con Smart Commit verso le due issue . . . . .	18
5	Modello a V . . . . .	35

## Elenco delle tabelle

## Tabella di Versionamento

<b>Versione</b>	<b>Data</b>	<b>Autore</b>	<b>Verificatore</b>	<b>Descrizione</b>
0.8.1	19/1/2026	Soligo Lorenzo	Laoud Zakaria	Fine scrittura paragrafi 4.2, 4.3, 4.4
0.8	11/1/2026	Soligo Lorenzo	Laoud Zakaria	Stesura paragrafi 4.2, 4.3, 4.4
0.7	15/12/2025	Rocco Matteo A.	Georgescu Diana	Stesura paragrafo 5
0.6	10/12/2025	Georgescu Diana	Soligo Lorenzo	Stesura sottosezioni 2.3, 2.4
0.5	1/12/2025	Soligo Lorenzo	Rocco Matteo A.	Aggiornamento nuovo standard per gestione branch, convenzioni sui nomi, date e versioni. Sezioni 3.1.2, 3.1.3, 3.2.1.1.1, 3.2.4, 4.2.3.2.1
0.4.1	28/11/2025	Soligo Lorenzo	Fattoni Antonio	Correzione nella Procedura di Revisione Paragrafo 3.1.3.1 e aggiornamento immagini
0.4	26/11/2025	Soligo Lorenzo	Fattoni Antonio	Ristrutturazione completa Processi (ISO 12207: Attività-Procedure-Strumenti)
0.3	25/11/2025	Soligo Lorenzo	Fattoni Antonio	Creazione e stesura sezioni Processi di Infrastruttura e sottosezioni 4.2.1-4.3. Nuova struttura Paragrafi e sottoParagrafi
0.2	22/11/2025	Soligo Lorenzo	Fattoni Antonio	Creazione e stesura sezioni Documentazione e sottosezioni 3.1-3.1.5
0.1	16/11/2025	Rocco Matteo A.	Soligo Lorenzo	Creazione e stesura sezioni Introduzione e Processo di fornitura

# 1 Introduzione

## 1.1 Scopo

Questo documento ha l'obiettivo di definire e normare il *Way of Working<sub>G</sub>*, ovvero le regole di lavoro che ogni membro del gruppo deve rispettare durante lo svolgimento delle *attività di progetto<sub>G</sub>* volte allo sviluppo dell'applicativo software **DIPReader<sub>G</sub>**, proposto dall'azienda *Sanmarco Informatica<sub>G</sub>*. A ciascun membro è richiesto di seguirle integralmente per poter lavorare in maniera efficace, efficiente e omogenea. Data la natura incrementale della redazione del documento, il *responsabile di progetto<sub>G</sub>* ha il compito di mantenere aggiornate le presenti norme e gli eventuali riferimenti ad altri documenti in esse contenuti.

## 1.2 Glossario

Ogni termine tecnico o con un significato particolare nell'ambito dell'*Ingegneria del Software<sub>G</sub>*, utilizzato nella documentazione di progetto, è definito nell'apposito documento [Glossario 1.0](#) (ultimo accesso: 17/11/2025).

## 1.3 Riferimenti

Il gruppo ha redatto il presente documento in conformità con lo *standard<sub>G</sub>* ISO/IEC 12207:1995, integrandolo occasionalmente con approfondimenti tratti dalla sua versione più recente, ISO/IEC/IEEE 12207:2017, per includere dettagli aggiuntivi sugli approcci *agili<sub>G</sub>* e *iterativi<sub>G</sub>* che caratterizzano lo *sviluppo software<sub>G</sub>* moderno.

### 1.3.1 Riferimenti Normativi

- [Standard ISO/IEC 12207:1995](#) (ultimo accesso: 17/11/2025)
- [Standard ISO/IEC/IEEE 12207:2017](#)
- [Standard ISO/IEC/IEEE 24765:2017](#)
- [Capitolato C3: DIPReader](#) (ultimo accesso: 13/11/2025)
- [Regolamento di Progetto Didattico a.a. 2025/2026](#) (ultimo accesso: 17/11/2025)

### 1.3.2 Riferimenti Informativi

- Dispense del corso di Ingegneria del Software 2025/2026:
  - <https://www.math.unipd.it/tullio/IS-1/2025/Dispense/T01.pdf> (ultimo accesso: 17/11/2025)
  - <https://www.math.unipd.it/tullio/IS-1/2025/Dispense/T02.pdf> (ultimo accesso: 17/11/2025)
  - <https://www.math.unipd.it/tullio/IS-1/2025/Dispense/T03.pdf> (ultimo accesso: 17/11/2025)
  - <https://www.math.unipd.it/tullio/IS-1/2025/Dispense/T04.pdf> (ultimo accesso: 17/11/2025)

- <https://www.math.unipd.it/tullio/IS-1/2025/Dispense/T05.pdf> (ultimo accesso: 17/11/2025)
- <https://www.math.unipd.it/tullio/IS-1/2025/Dispense/T06.pdf> (ultimo accesso: 17/11/2025)
- <https://www.math.unipd.it/tullio/IS-1/2025/Dispense/T07.pdf> (ultimo accesso: 17/11/2025)
- <https://www.math.unipd.it/tullio/IS-1/2025/Dispense/T08.pdf> (ultimo accesso: 17/11/2025)
- <https://www.math.unipd.it/tullio/IS-1/2025/Dispense/T09.pdf> (ultimo accesso: 17/11/2025)
- <https://www.math.unipd.it/tullio/IS-1/2025/Dispense/T10.pdf> (ultimo accesso: 17/11/2025)
- <https://www.math.unipd.it/tullio/IS-1/2025/Dispense/T11.pdf> (ultimo accesso: 17/11/2025)
- [Linee Guida Sviluppo Sicuro AGID \(Agenzia per l'Italia Digitale\)](#)
- [Linee Guida sulla formazione, gestione e conservazione dei documenti informatici AGID](#)
- [Documentazione L<sup>A</sup>T<sub>E</sub>X by Lorenzo Pantieri](#) (ultimo accesso: 17/11/2025)
- [Documentazione Jira](#)

## 2 Processi Primari

### 2.1 Processo di Fornitura

Il  $\text{processo}_G$  di fornitura contiene le attività e i compiti svolti dal  $\text{fornitore}_G$ . Per implementare correttamente il processo il gruppo si impegna a svolgere le seguenti attività.

#### 2.1.1 Attività di processo

##### 2.1.1.1 Avvio

Il fornitore analizza i  $\text{requisiti}_G$  necessari alla proposta di fornitura, tenendo in considerazione eventuali vincoli organizzativi e normativi.

##### 2.1.1.2 Preparazione della proposta di fornitura

Il fornitore prepara la proposta di fornitura in risposta alle richieste del committente e definisce i termini in cui si articola la proposta.

##### 2.1.1.3 Accordo

Proponente e fornitore entrano nella fase di definizione dell'accordo di fornitura del prodotto software, prevedendo possibilità di negoziazione della fornitura da parte del fornitore.

#### 2.1.1.4 Pianificazione

Il fornitore rielabora l'analisi dei requisiti fondamentali per definire il *framework<sub>G</sub>* entro il quale il prodotto verrà sviluppato e gestito, in modo tale da garantire un processo di qualità durante lo sviluppo. Si impegna inoltre a definire il modello del ciclo di vita del prodotto adatto alla complessità del progetto e ai relativi rischi che potrebbero insorgere. Tutte queste decisioni convergono nel Piano di Progetto.

#### 2.1.1.5 Esecuzione e controllo

Il fornitore si impegna a sviluppare il prodotto secondo il Piano di Progetto, avendo cura di controllare che i processi siano stati eseguiti correttamente.

#### 2.1.1.6 Verifica e validazione

Il fornitore stabilisce con la proponente le modalità di rendicontazione dello stato di avanzamento del prodotto e rende disponibili i documenti che dimostrino la verifica e validazione dei processi secondo i requisiti precedentemente individuati.

#### 2.1.1.7 Consegna e terminazione

Il fornitore consegna il prodotto finale al proponente e ne espone le funzionalità.

#### 2.1.1.8 Accordi con l'azienda proponente

I capitoli presentati dalle proponenti vengono analizzati e viene redatto il documento di [Analisi dei capitoli](#), nel quale sono delineati i bisogni e i principali vincoli a cui attenersi per la fornitura del prodotto finale. Il fornitore espone ai committenti di fornitura, ovvero i Professori Vardanega Tullio e Cardin Riccardo, la Lettera di Presentazione della proposta di fornitura che descrive il preventivo di costi, cronogramma di sviluppo, suddivisione del lavoro e i ruoli coinvolti. La *proponente<sub>G</sub>*, in qualità di *stakeholder<sub>G</sub>*, esercita il diritto di ricevere la rendicontazione professionale e approfondita del lavoro svolto dal gruppo fornitore, perciò si instaura un accordo per delineare le modalità di comunicazione e il contenuto di tale rendicontazione. È previsto l'aggiornamento costante e tempestivo della proponente per quanto riguarda la pianificazione degli obiettivi e delle tempistiche di sviluppo individuate dal fornitore. Ogni qualvolta vi siano modifiche di notevole interesse esterno dal gruppo fornitore verranno comunicate all'azienda proponente attraverso appositi canali di comunicazione sincrona o asincrona. Il fornitore e la proponente hanno accordato lo svolgimento di un incontro di verifica dello stato di avanzamento lavori (*SAL<sub>G</sub>*) in modalità sincrona ogni due settimane, in cui discutere l'andamento del lavoro e chiarire eventuali dubbi da parte del fornitore o segnalazioni di difformità dai requisiti iniziali della proponente. È inoltre sempre disponibile la comunicazione via email per questioni minori e di facile risoluzione. La consegna del prodotto è suddivisa in due *milestone<sub>G</sub>* principali: *RTB<sub>G</sub>* (Requirements and Technology Baseline) e *PB<sub>G</sub>* (Product Baseline).



## 2.1.2 Documentazione fornita

### 2.1.2.1 Analisi dei requisiti

Nel documento di [Analisi dei requisiti](#) (ultimo accesso: 17/11/2025) sono riportati i bisogni e i vincoli a cui attenersi per la realizzazione del prodotto finale. L'obiettivo è definire in maniera non ambigua i *casi d'uso*<sub>G</sub> (*Use Cases*) e i requisiti (*Requirements*) del software. Il documento è diviso nelle seguenti sezioni:

1. Introduzione
2. Descrizione
3. Definizione dei casi d'uso
4. Definizione dei requisiti

### 2.1.2.2 Glossario

Il Glossario è il documento che raccoglie ogni termine di carattere tecnico, nomenclature e acronimi con particolare significato nell'ambito dell'Ingegneria del Software utilizzato nella documentazione di progetto. La definizione dei termini di glossario è coadiuvata dal contenuto dello standard ISO/IEC/IEEE 24765/2017.

### 2.1.2.3 Piano di progetto

Il [Piano di progetto v1.0](#) (ultimo accesso: 17/11/2025) è il documento che espone all'esterno il lavoro di sviluppo svolto seguendo le procedure delineate all'interno di questo documento. Fornisce una guida dettagliata alla pianificazione, esecuzione e consuntivo delle attività completate in ciascuna *sprint*<sub>G</sub>. Il documento è diviso nelle seguenti sezioni:

1. Introduzione
2. Analisi dei rischi e mitigazione
3. Modello di sviluppo
4. Pianificazione dei costi e suddivisione ruoli
5. Preventivo di periodo
6. Consuntivo di periodo
7. Retrospettiva

#### 2.1.2.4 Piano di qualifica

Il piano di qualifica descrive gli obiettivi di qualità dei processi che il fornitore si impegna a soddisfare per consegnare un prodotto finale di qualità. Le metriche di valutazione vengono determinate dall'analisi dei requisiti e dalle indicazioni date dalla proponente, suddivise in base all'applicazione sui processi o sul prodotto. Le metriche stabilite vengono poi misurate attraverso opportuni test e verifiche, di cui vengono riportate le specifiche. Il documento include una sezione di rendicontazione per la valutazione dei processi e la valutazione del prodotto, in cui riportare l'attinenza alle metriche ottenuta rispetto agli obiettivi e di conseguenza valutare azioni correttive in caso si verifichino eventuali problemi (*cruscotto di qualità<sub>G</sub>*). Il documento è diviso nelle seguenti sezioni:

1. Qualità dei processi
2. Qualità del prodotto
3. Specifiche di test e verifica
4. Cruscotto di qualità

#### 2.1.2.5 Lettera di presentazione

La lettera di presentazione è il documento necessario alla candidatura per la milestone di revisione di avanzamento *RTB<sub>G</sub>* (*Requirements and Technology Baseline*). Essa contiene le informazioni sul repository di progetto, il puntatore al *Proof of Concept<sub>G</sub>*, il consuntivo di spesa e preventivo a finire del progetto.

#### 2.1.3 Strumenti

- *GitHub<sub>G</sub>* per la gestione della documentazione di progetto e mezzo comunicativo nella fase di fornitura
- *Jira<sub>G</sub>* per la suddivisione e il monitoraggio delle attività di progetto
- Discord per la comunicazione sincrona tra i membri del gruppo
- Gmail per la comunicazione asincrona con l'azienda proponente
- VSCode con estensione con IDE di riferimento.

### 2.2 Processo di sviluppo

#### 2.2.1 Attività di processo

#### 2.3 Processo operativo

Il processo operativo comprende quell'insieme di attività trasversali che sono necessarie a garantire il corretto coordinamento tra i membri del gruppo e il raggiungimento degli obiettivi del progetto. Tale processo assicura comunicazioni efficaci, nonché una

distribuzione ottimale dei compiti, al fine di mantenere alta la qualità del software e rispettare le tempistiche stabilite. Il processo operativo si integra naturalmente con tutti gli altri processi del progetto.

### **2.3.1 Pianificazione operativa**

La pianificazione operativa rappresenta l'organizzazione delle attività quotidiane e settimanali del gruppo. Il Responsabile, all'inizio di ogni sprint, coordina la distribuzione dei compiti tra i membri.

Procedure di pianificazione operativa:

1. Svolgimento di una riunione di pianificazione all'inizio di ogni sprint;
2. Discussione degli obiettivi dello sprint e le attività necessarie per raggiungerli;
3. Assegnamento delle task ai vari membri;
4. Scelta delle scadenze intermedie.

### **2.3.2 Gestione dei rischi operativi**

La gestione dei rischi operativi mira a identificare in maniera proattiva le potenziali problematiche che potrebbero minare il normale svolgimento delle attività e a pianificare le dovute azioni di mitigazione. I principali rischi operativi identificati sono:

- Sforamento dei costi preventivati;
- Calo di produttività del team;
- Mancata comunicazione e collaborazione tra i membri del team;
- Mancata comunicazione con l'azienda proponente;
- Problemi tecnici con gli strumenti di sviluppo;
- Mancato rispetto delle norme e documenti di progetto interni.

Le strategie adottate per ogni tipologia di rischio sono rispettivamente:

- Monitoraggio costante dell'allocazione delle ore rispetto alla pianificazione iniziale, svolgimento di stand-up meetings periodici e previsione margini temporali per imprevisti;
- Pianificazione anticipata delle attività più critiche prima del periodo di calo, e le restanti tenendo conto del periodo di ridotta attività;
- Adozione di canali di comunicazione chiari e regolari e una routine di aggiornamenti pianificati per garantire che tutti i membri del team siano allineati sugli obiettivi e le responsabilità;

- Scelta di un calendario di incontri regolari con l'azienda proponente per garantire un flusso costante di comunicazione e feedback;
- Impostazione di sessioni di formazione iniziali con il supporto occasionale dell'azienda proponente per familiarizzare con gli strumenti e le tecnologie;
- Ruolo attivo di amministratori e tester per garantire il rispetto delle norme e dei documenti di progetto interni.

## **2.4 Processo di manutenzione**

Il processo di manutenzione definisce le modalità con cui il gruppo gestisce le modifiche, le correzioni e gli aggiornamenti del software e della corrispondente documentazione durante tutto il ciclo di vita del progetto. Questo processo garantisce che il prodotto rimanga conforme ai requisiti. Eventuali difetti devono essere corretti tempestivamente mentre, ove sorgesse la necessità di apportare migliorie, queste ultime vengano fornite in modo efficace ed efficiente.

### **2.4.1 Manutenzione correttiva**

Si tratta di interventi finalizzati alla correzione di difetti o malfunzionamenti trovati nel software o nella documentazione. Sono inclusi:

- Correzioni di bug;
- Risoluzione di errori di varia natura nella documentazione;
- Correzione di qualsiasi comportamento non conforme ai requisiti.

### **2.4.2 Manutenzione adattiva**

Sono le modifiche necessarie per adattare il prodotto a cambiamenti in itinere nell'ambiente operativo o nei requisiti. Sono inclusi:

- Adattamento a nuove tecnologie;
- Modifiche conseguenti a variazioni nei requisiti.

### **2.4.3 Manutenzione preventiva**

Si riferisce alle attività proattive per ridurre la probabilità di problemi futuri. Sono inclusi:

- Aggiornamenti di sicurezza;
- Miglioramento della gestione degli errori.

#### 2.4.4 Identificazione della necessità di manutenzione

La necessità di manutenzione può emergere da diverse fonti, come segnalazioni interne quali le attività di verifica e validazione, lo sviluppo del codice e le retrospettive; o segnalazioni esterne quali feedback della proponente.

#### 2.4.5 Verifica e validazione

La verifica va effettuata rigorosamente prima dell'integrazione. Il verificatore assegnato deve esaminare il codice modificato verificando la conformità agli standard e l'assenza di introduzione di problematiche note. Deve inoltre eseguire i test, verificare che non siano presenti regressioni e controllare che la documentazione sia stata aggiornata coerentemente.

La validazione riguarda modifiche significative che possono impattare i requisiti o l'architettura. Tali modifiche vengono presentate illustrando il problema e la soluzione implementata e si accoglie il feedback della proponente.

#### 2.4.6 Metriche

Per valutare l'efficacia della manutenzione, il gruppo monitora le seguenti metriche:

- Tempo medio di risoluzione: tempo medio tra l'identificazione di un problema e la sua risoluzione;
- Percentuale di regressioni: numero di nuovi difetti introdotti da attività di manutenzione rispetto al totale delle modifiche;
- Distribuzione per tipologia: suddivisione degli interventi di manutenzione per categoria.

### 3 Processi di Supporto

I processi di supporto sono volti a garantire l'efficacia e l'efficienza dei processi primari.

#### 3.1 Processo di documentazione

Il processo di documentazione è parte integrante del Progetto in quanto permette il tracciamento delle decisioni prese, delle attività svolte e dei risultati ottenuti. Tutto ciò al fine di favorire il lavoro asincrono tra membri del gruppo e promuovere il principio *Agile<sub>G</sub>* di continuo miglioramento e adattamento tramite *feedback<sub>G</sub>*.

##### 3.1.1 Attività: Pianificazione della documentazione

La pianificazione della documentazione avviene contestualmente alla pianificazione delle attività di progetto.

Durante la pianificazione di ogni *sprint<sub>G</sub>*, il *responsabile di progetto<sub>G</sub>* assegna le attività di documentazione ai membri del gruppo, tenendo conto delle competenze e della

disponibilità di ciascuno. Le scadenze per la consegna dei documenti sono stabilite in modo da garantire che la documentazione sia sempre aggiornata e disponibile per la consultazione da parte del gruppo e di eventuali attori esterni (Azienda *proponente<sub>G</sub>*, *committente<sub>G</sub>*).

Per una più efficiente scrittura dei documenti, soprattutto di tutti quei documenti periodici (Verbal Interni, Verbal Esterni, Diario di Bordo) sono presenti modelli standard approvati in [/assets](#). L'aggiornamento di tali standard deve essere argomento di Verbal Interni e risultato di una discussione e successiva decisione presa in tale sede.

### 3.1.1.1 Procedure di Pianificazione

I seguenti passaggi guidano il Team nella pianificazione delle attività di documentazione:

1. Durante la pianificazione di ogni *sprint<sub>G</sub>*, il *responsabile<sub>G</sub>* identifica le necessità di documentazione in base agli obiettivi dello sprint e alle attività previste.
2. Il responsabile assegna le attività di documentazione ai membri del gruppo, tenendo conto delle competenze e della disponibilità di ciascuno, nonché della necessità di ruotare i ruoli, per dare la possibilità a tutti i membri di acquisire esperienza in diverse aree.
3. Vengono create le *issue<sub>G</sub>* in Jira per ogni attività di documentazione, specificando i dettagli del compito, le scadenze e i verificatori per ogni attività. Specifiche in 3.2.1.1.

### 3.1.1.2 Strumenti di Pianificazione

- *Jira<sub>G</sub>* per la gestione delle attività di progetto. In particolare con la *board<sub>G</sub> Scrum<sub>G</sub>* che viene aggiornata in automatico con i commit effettuati sui Work Item e può essere personalizzata con la creazione di *Sprint<sub>G</sub>*.
- *DashBoard/Cruscotto<sub>G</sub>* di Jira per il monitoraggio delle attività assegnate per ogni membro del gruppo.

### 3.1.2 Attività: Produzione della documentazione

La produzione della documentazione, assegnata durante la pianificazione, è visibile all'assegnatario come *Work item<sub>G</sub>* grazie all'estensione Jira di VSCode 5.0.0.3. Grazie a quest'ultima è possibile creare direttamente il Branch di lavoro che si baserà sulla feature branch principale. Una volta completata la stesura, seguendo i modelli standard sopracitati, l'autore del documento crea una (PR) *Pull Request<sub>G</sub>* verso la feature branch principale, assegnando come revisore il membro del gruppo designato, diverso da se. A questo punto:

- Se il revisore **approva la PR**, questo branch viene automaticamente eliminato, il work item viene marcato come completato in Jira e l'assegnatario può proseguire con gli altri compiti a lui assegnati.

- Se il revisore richiede modifiche **la PR viene rifiutata** e l'assegnatario deve procedere con le modifiche richieste. Una volta completate, l'assegnatario notifica il revisore che procederà con una nuova revisione. Questo ciclo si ripete fino a quando la PR non viene approvata.

L'integrazione con Jira permette di controllare lo stato di avanzamento dei Work Item, la rendicontazione delle ore lavorate e la gestione delle scadenze. Risulta quindi **obbligatorio** l'utilizzo di Smart Commit per tutti i commit, compresi quelli di Pull Request. Più in 5.0.0.2.3.

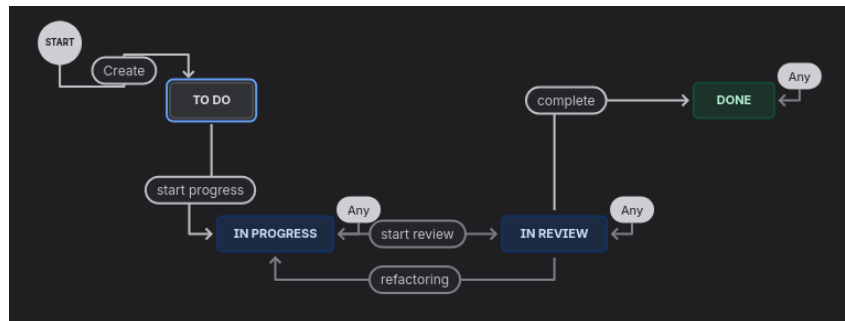


Figura 1: Flusso del processo di documentazione

### 3.1.2.1 Procedure di Produzione

I seguenti passaggi guidano il Team nella produzione di documenti:

1. Consultando l'estensione "[Atlassian: Jira, Rojo Dev, Bitbucket](#)" il membro del gruppo potrà avere accesso al Work Item assegnatogli e cliccando su "Start Work" potrà **creare il branch** di lavoro secondo le convenzioni stabilite (3.2.2.1). Nell'immagine sottostante sono indicati visivamente i passi.

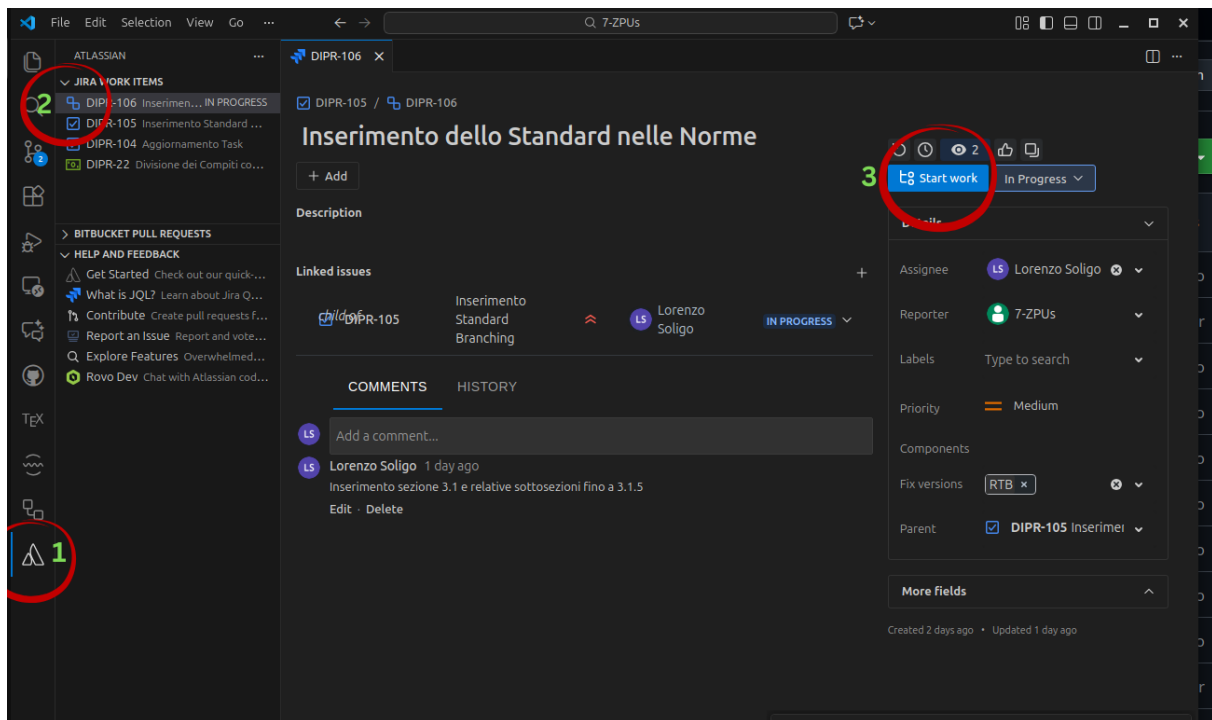


Figura 2: Creazione del branch di lavoro tramite estensione Jira in VSCode

2. Quando è necessario **effettuare un commit** è obbligatorio utilizzare lo Smart Commit. Più in 5.0.0.2.
3. Una volta completata la task, è necessario **creare la PR** verso la branch "*in\_lavorazione*" di riferimento mettendo come revisore il membro del gruppo designato. Per esempio, una volta completata la stesura di un verbale viene creata una PR verso *verbali\_in\_lavorazione* e se questa viene approvata, il documento sarà pronto per la revisione finale del responsabile che si occuperà del merge al *main* in sede di milestone.



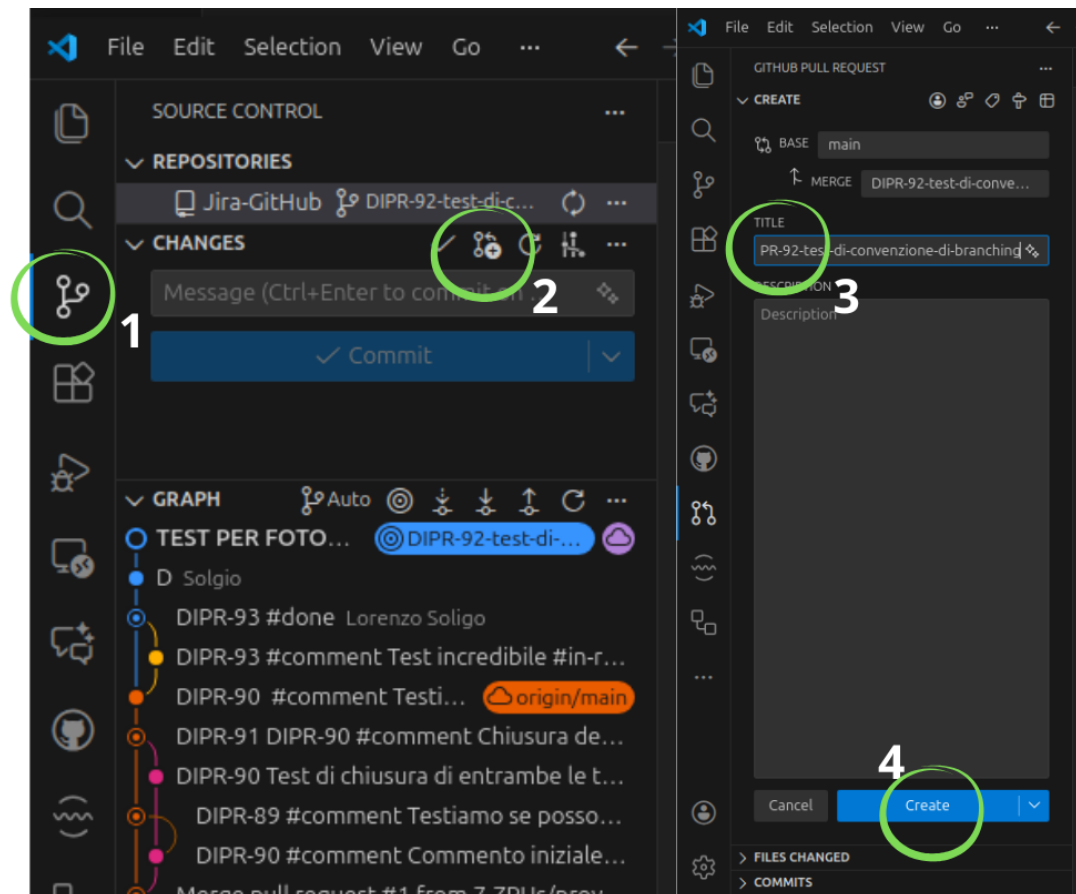


Figura 3: Creazione della PR verso l'issue branch *in\_lavorazione*

#### 3.1.2.1.1 Denominazione e datazione

Per una corretta archiviazione e reperibilità dei documenti, è necessario seguire le seguenti convenzioni per la datazione e denominazione.

- Tutti i file dei documenti devono seguire la regola del **Pascal Case**, ovvero devono essere scritti senza spazi e con la prima lettera di ogni parola in maiuscola.
- All'interno dei documenti, la data deve essere riportata nel formato **DD-MM-YYYY**, giorno-mese-anno.

La denominazione dei file sul sito necessita di seguire invece la datazione usata per il nome dei file su GitHub ovvero **YYYY-MM-DD**, con l'aggiunta del numero di versione alla fine del nome. Per le specifiche di versionamento, si rimanda alla sezione 3.2.4.1.

#### 3.1.2.2 Strumenti di Produzione

- *VSCode<sub>G</sub>* come IDE principale per la stesura dei documenti in *LaTeX<sub>G</sub>*.
- *Estensione Jira per VSCode<sub>G</sub>* per la gestione dei Work Item assegnati e la creazione automatica delle branch di lavoro.

- **GitHub<sub>G</sub>** per la gestione delle versioni della documentazione di progetto.

### 3.1.3 Attività: Revisione e Approvazione

Ogni documento redatto viene sottoposto a un processo di revisione interna che ne accerta la correttezza contenutistica, formale, e stilistica. La revisione viene effettuata da un membro del gruppo diverso dall'autore del documento seguendo la procedura definita a seguire

#### 3.1.3.1 Procedure di Revisione e Approvazione

1. Una volta ricevuta la notifica della PR da revisionare, il revisore dovrà controllare l'aderenza ai modelli approvati, la correttezza formale e sostanziale del documento. Per velocizzare, oltre alla lettura attenta, si consiglia l'uso di LLM, in particolare per l'analisi grammaticale e stilistica.
2. Completata la revisione, il revisore può:
  - **Approvare la PR**, notificando all'autore l'approvazione. È necessario che nel testo del commit del merge siano chiuse tramite Smart Commit entrambe le issue correlate.
  - **Richiedere modifiche**, fornendo un feedback dettagliato all'autore, chiudendo la PR che sarà riaperta dall'autore una volta implementate le modifiche.

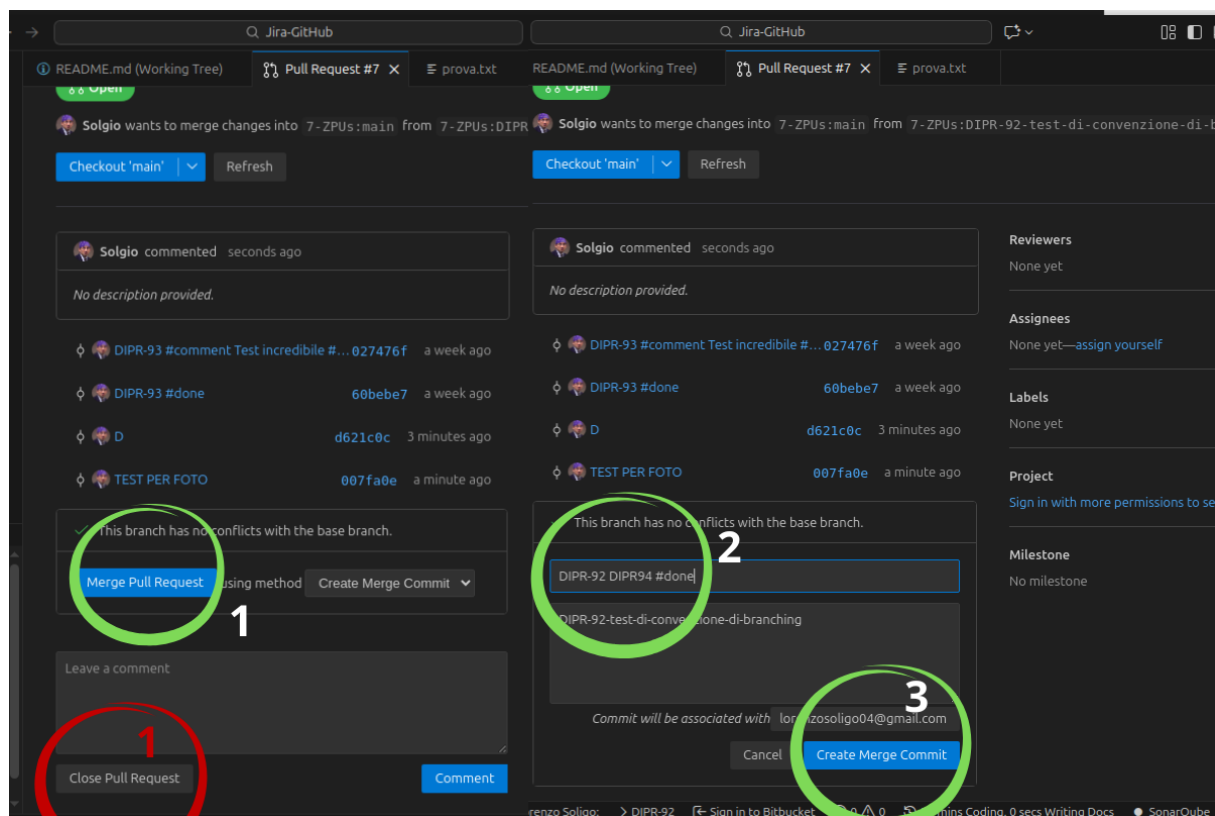


Figura 4: Commit della PR con Smart Commit verso le due issue

Per quanto riguarda l'approvazione finale del documento, questa spetta al *responsabile<sub>G</sub>*, il quale effettua il merge da "*in\_lavorazione*" nel ramo principale *main*, base per la versione ufficiale di rilascio corrispondente alla *milestone<sub>G</sub>*. Questo passaggio dovrebbe risultare puramente formale. Non di meno è il garante delle qualità del documento quindi deve impiegare il proprio tempo, il minimo possibile, per rileggere e confermare i contenuti.

### 3.1.3.2 Strumenti di Revisione e Approvazione

Per la gestione della documentazione di progetto il gruppo utilizza i seguenti strumenti:

- *GitHub<sub>G</sub>*, in particolare integrato in VSCode per la gestione delle versioni e delle pull request, comodamente nell'ambiente di sviluppo di VSCode 5.0.0.3. Per una stesura efficiente dei documenti il Team si è dotato di modelli predefiniti (Decisione del [2025-11-07](#)).
- *Jira<sub>G</sub>*: strumento di gestione delle attività di progetto, utilizzato per tracciare le attività di documentazione e assegnarle ai membri del gruppo.
- *LLM<sub>G</sub>* per il supporto alla revisione formale e stilistica dei documenti.

## 3.2 Processo di Gestione della Configurazione

Il processo di gestione della configurazione ha lo scopo di identificare, definire e controllare gli elementi della configurazione software (*SCIs<sub>G</sub>*) durante tutto il ciclo di vita del progetto, garantendo la tracciabilità delle modifiche e l'integrità dei rilasci. Gli SCIs in poche parole sono tutti gli artefatti prodotti e gestiti durante il progetto.

### 3.2.1 Attività: Identificazione della Configurazione

Questa attività prevede la definizione degli elementi della configurazione (SCIs) e la loro identificazione univoca.

#### 3.2.1.1 Procedure: Identificazione degli SCIs

Gli SCIs sono sempre associati ad un Work Item di Jira, che ne garantisce la tracciabilità e la gestione delle modifiche. In particolare il processo di creazione deve seguire i seguenti passaggi:

1. Creazione di una issue in Jira per ogni nuovo artefatto da produrre (documento, componente software, etc).
2. Identificazione dell'ambito di appartenenza (Epic), della funzionalità (Feature) o di una specifica attività (Task) già presenti nel sistema o da aggiungere se non compatibile.
3. Assegnazione della issue al membro del gruppo responsabile della sua supervisione.

4. Aggiunta di label specifici per facilitare la ricerca e la categorizzazione degli SCIs, per esempio *DOCS*, *Formazione*, *Code* etc.
5. Aggiunta di Linked Issues per collegare SCIs correlati o dipendenti tra loro, con relazioni di *child/parent of* o *blocked by* per esempio.
6. Aggiunta di eventuali allegati.
7. Definizione delle scadenze e del *Time Estimate<sub>G</sub>* per la gestione del carico di lavoro.

È quindi necessaria una specificazione sulla struttura di un Work Item. Ogni Work Item deve contenere deve consistere in una fase produttiva e una fase di revisione, per garantire la qualità del prodotto finale.

A tal fine si adottano le seguenti convenzioni:

- Ogni Work Item deve presentare una sotto issue che rappresenta la fase di produzione.
- La sotto issue di produzione deve essere collegata alla issue principale tramite la relazione *child of*.
- La sotto issue di produzione deve essere assegnata al membro del gruppo responsabile della stesura o sviluppo dell'artefatto, mentre la issue principale deve essere assegnata al membro responsabile della supervisione.

Questo approccio divide chiaramente le responsabilità e le attività, mantenendo la correlazione tra produzione e supervisione e inoltre facilita il monitoraggio dello stato di avanzamento, la gestione delle revisioni e conteggio di *Ore produttive<sub>G</sub>* consumate. Di più nello specifico in ??.

#### 3.2.1.1.1 Issue e SubIssue

In generale, come già detto, ogni Work Item deve essere composto da una issue principale e una sotto issue di produzione. Nel caso però di Riunioni e Diari di Bordo, intesi come eventi ma anche come tipi di work item Jira, la struttura cambia:

- **Riunioni:** La Riunione rappresenta l'evento, simile al concetto di Epic o Feature, mentre le issue rappresentano i verbali interni ed esterni associati alla riunione stessa.
- **Diari di Bordo:** Similarmente, il Diario di Bordo rappresenta l'evento periodico, mentre la issue associata riguarda la preparazione delle slide.

Per entrambi, la creazione della issue di produzione avviene automaticamente una volta che l'evento viene assegnato al membro del team che si occuperà della verifica del materiale prodotto.

### 3.2.1.2 Strumenti di Identificazione

- **Jira:** Strumento di gestione delle attività di progetto.

## 3.2.2 Attività: Versionamento e Identificazione

Questa attività definisce le regole per l'identificazione univoca degli artefatti e la gestione delle ramificazioni nel repository.

### 3.2.2.1 Procedure: Standard per le Branch

Per garantire una gestione ordinata e coerente del codice e della documentazione, il Team adotta il seguente standard per la denominazione dei branch in GitHub:

- La creazione del branch deve avvenire preferibilmente in modo automatico tramite l'integrazione Jira-VSCode 5.0.0.3.
- Il formato obbligatorio è:

DIPR-<numero issue>-<descrizione-breve>

- È necessario scegliere una descrizione breve che identifichi chiaramente il contenuto della modifica.

### 3.2.2.2 Strumenti di Versionamento

- **GitHub:** Repository remoto per la memorizzazione delle versioni.
- **VSCode (Estensione Jira):** Per l'automazione della nomenclatura delle branch.

## 3.2.3 Attività: Controllo delle Configurazioni

Questa attività regola il modo in cui le modifiche di avanzamento vengono registrate e tracciate rispetto ai task pianificati.

### 3.2.3.1 Procedure: Smart Commit

La necessità di tracciamento delle attività richiede l'adozione capillare degli Smart Commit per collegare i commit GitHub alle issue Jira. Questo permette di aggiornare automaticamente lo stato delle task e rendicontare il tempo. Risulta quindi uno strumento rapido ed efficace per mantenere allineati i progressi del lavoro con la pianificazione stabilita. Maggiori dettagli sugli Smart Commit sono disponibili 5.0.0.2.3.

### 3.2.3.2 Strumenti

- **Jira Automation:** Interpreta gli Smart Commit per aggiornare i Work Item.
- **Git/GitHub:** Motore di versionamento sottostante.

### 3.2.4 Registrazione delle Configurazioni

Questa attività prevede la documentazione e la registrazione delle modifiche apportate agli SCIs.

#### 3.2.4.1 Procedure: Registrazione delle Modifiche

Per ogni modifica apportata a uno SCI, è necessario documentare le seguenti informazioni, all'interno della sezione denominata "Tabella di Versionamento", situata all'inizio di ogni documento.

Al suo interno vi sono:

- Numero di versione
- Data della modifica (Di più in 3.1.2.1.1)
- Autore della modifica
- Verificatore della modifica
- Descrizione della modifica, con riferimento preciso, al paragrafo o sezione interessata.

Per la numerazione delle versioni si adotta lo standard di versionamento **MAJOR.MINOR.PATCH**, nel quale:

- **MAJOR** viene incrementato per modifiche sostanziali che introducono cambiamenti significativi o incompatibili con le versioni precedenti. Viene modificato in vista delle milestone.
- **MINOR** viene incrementato per l'aggiunta di nuove funzionalità o miglioramenti che non compromettono la compatibilità con le versioni precedenti.
- **PATCH** viene incrementato per correzioni di bug, miglioramenti minori o modifiche che non influenzano le funzionalità principali del documento.

### 3.3 Processo di garanzia della qualità

### 3.4 Processo di verifica

### 3.5 Processo di validazione

### 3.6 Processo di revisione congiunta

### 3.7 Processo di risoluzione dei problemi

### 3.8 Gestione della qualità

## 4 Processi Organizzativi

### 4.1 Gestione

### 4.2 Processo di Infrastruttura

Il processo di Infrastruttura fornisce il supporto tecnico necessario per lo sviluppo e la gestione del progetto. I ruoli coinvolti in questo processo sono:

Ruolo	Specifica
Amministratore	Ruolo incaricato di gestire le risorse e i processi di infrastruttura.
Responsabile	Ruolo incaricato di coordinare le attività di infrastruttura e di controllare che questa faciliti e assista il team.
Verificatori	A tutti i membri è richiesto di porre attenzione nell'utilizzo dei tool e delle procedure al fine di poter risolvere eventuali malfunzionamenti e per contribuire al miglioramento continuo

#### 4.2.1 Attività: Implementazione

L'attività di Implementazione comprende la selezione, l'installazione e il testing degli strumenti e delle tecnologie necessarie per supportare le attività di progetto. È la prima fase del processo di infrastruttura e getta le basi per le successive attività di creazione e manutenzione.

##### 4.2.1.1 Procedure di Implementazione

Le seguenti procedure guidano il Team nella scelta degli strumenti di infrastruttura, garantendo un uso coerente al contesto di sviluppo ed efficiente delle risorse disponibili.

1. **Analisi delle necessità:** il Responsabile e Amministratore analizzano le necessità del prodotto finale e del team di sviluppo.
2. **Valutazione delle opzioni:** vengono valutate diverse soluzioni tecnologiche in base a criteri quali costo, scalabilità, facilità d'uso e supporto.

3. **Selezione degli strumenti:** viene effettuata la scelta degli strumenti più adatti per soddisfare i requisiti identificati.

#### 4.2.1.2 Strumenti di Implementazione

Non sono previsti strumenti specifici per questa fase, in quanto la scelta degli strumenti avviene in base alle necessità e viene effettuata in base agli standard di mercato e alle best practice del settore. Per la gestione della task di implementazione si rimanda a 5.0.0.2.

#### 4.2.2 Attività: Creazione

L'attività di Creazione riguarda lo sviluppo e l'integrazione degli strumenti e delle tecnologie selezionate durante la fase di Implementazione. Questa fase include la configurazione degli ambienti di sviluppo, la creazione di procedure operative e la documentazione degli strumenti utilizzati.

##### 4.2.2.1 Procedure di Creazione

Le seguenti procedure guidano il Team nell'utilizzo degli strumenti di infrastruttura, garantendo un uso coerente ed efficiente delle risorse disponibili.

1. **Integrazione:** gli strumenti selezionati vengono integrati nell'ambiente di sviluppo esistente.
2. **Documentazione:** viene mantenuta una documentazione aggiornata sugli strumenti utilizzati, le loro configurazioni e le procedure operative. Così facendo tutto il team può accedere facilmente alle informazioni necessarie per utilizzare gli strumenti in modo efficace.

##### 4.2.2.2 Strumenti di Creazione

Gli strumenti adottati per il testing delle tecnologie e la documentazione sono i seguenti.

##### 4.2.2.3 Jira

Come descritto in 3.1.1, è necessaria la creazione di una issue concordata in fase di pianificazione per ogni nuovo strumento da adottare. In questo modo si garantisce la tracciabilità della decisione e la documentazione delle motivazioni alla base della scelta.

##### 4.2.2.4 GitHub

Strumento per l'hosting utilizzato per il testing delle nuove tecnologie. Ogni nuova tecnologia deve essere testata nell'apposita repository o branch, per evitare di compromettere l'infrastruttura esistente. Per gli standard di versionamento e gestione delle branch si rimanda a 3.2.2.1.



### 4.2.3 Attività: Manutenzione

L'attività di Manutenzione riguarda l'aggiornamento, il monitoraggio e la risoluzione di eventuali problemi legati agli strumenti e alle tecnologie utilizzati nel progetto.

#### 4.2.3.1 Procedure di Manutenzione

Le seguenti procedure guidano il Team nella manutenzione degli strumenti di infrastruttura, garantendo la loro efficienza e affidabilità nel tempo.

1. **Monitoraggio continuo:** viene effettuato un monitoraggio costante delle prestazioni degli strumenti per identificare eventuali problemi o inefficienze.
2. **Aggiornamenti regolari:** gli strumenti vengono aggiornati regolarmente per garantire la sicurezza e l'efficienza.
3. **Supporto tecnico:** viene fornito supporto tecnico ai membri del team per l'utilizzo degli strumenti di infrastruttura.
4. **Disaster Recovery:** viene mantenuto un'analisi dei rischi per i processi di progetto.

In vista dell'attività di produzione di codice, l'infrastruttura verrà aggiornata e ampliata per poter rimanere al passo con le nuove necessità. Sono in particolare previste le seguenti aggiunte:

- **Tracciamento dei requisiti:** Deploy di uno strumento automatico per il tracciamento dei requisiti che ammetta codice solo se i requisiti sono stati tracciati correttamente.  
Tale codice è in fase di scrittura e verrà integrato non appena testato.
- **Integrazioni di Analisi Statica e Dinamica:** Deploy di strumenti di analisi statica e dinamica del codice per garantire la qualità del software prodotto.
- **CI/CD:** Implementazione di pipeline di integrazione continua e distribuzione continua per automatizzare il processo di build, test e rilascio del software.

#### 4.2.3.2 Strumenti di Manutenzione

Gli strumenti adottati per il monitoraggio e la manutenzione dell'infrastruttura sono i seguenti.

- **Jira:** Tracciamento delle attività di manutenzione e monitoraggio degli interventi effettuati.
- **GitHub:** Gestione dei branch di hotfix e documentazione degli update di sicurezza e stabilità.

### 4.2.3.3 Criteri di Scelta degli Strumenti

La scelta degli strumenti di infrastruttura segue i seguenti criteri misurabili:

- **Costo:** Valutare il rapporto costo-beneficio e la sostenibilità economica a lungo termine.
- **Scalabilità:** Verificare la capacità dello strumento di adattarsi alle crescenti esigenze del progetto.
- **Facilità d'uso:** Assicurare che lo strumento sia intuitivo e richieda un tempo di apprendimento minimo.
- **Supporto e Comunità:** Valutare la disponibilità di documentazione, supporto tecnico e comunità attiva.
- **Integrazione:** Verificare la compatibilità con gli strumenti già in uso nel progetto.
- **Sicurezza:** Assicurare che lo strumento rispetti gli standard di sicurezza e privacy richiesti dal progetto.

## 5 Knowledge Base

In questa sezione sono contenute le informazioni principali sugli strumenti implementati dal gruppo, completi di istruzioni per le singole procedure e best practice adottate.

### 5.0.0.1 GitHub

Piattaforma di hosting per il versionamento e la gestione dei contenuti di progetto. Il Team deve sfruttare appieno le potenzialità di GitHub, in particolare per l'integrazione con Jira. Vengono quindi descritti gli Smart Commit, lo standard per la scrittura di commit e la gestione dello stato di vita degli work item.

L'*Enterprise 7-ZPUs* su GitHub è accessibile a tutti i membri del gruppo e viene utilizzata per la gestione del codice sorgente, della documentazione e degli artefatti di progetto. È organizzata in repository specifiche, Docs per la documentazione e PoC per il Proof of Concept.

La Repository è strutturata come segue:

- **1\_Candidatura:** Directory contenente tutta la documentazione prodotta per la candidatura al Progetto di Ingegneria del Software.
- **2\_RTB:** Directory contenente tutta la documentazione prodotta prima del raggiungimento della milestone di RTB.
  - Verbali
    - \* Interni
    - \* Esterni
  - **gh-pages:** Directory contenente la pagina GitHub Pages del progetto.

#### **5.0.0.1.1 GitHub Action**

Le GitHub Action sono utilizzate per automatizzare i flussi di lavoro all'interno del repository. Il Team ha implementato diverse azioni per migliorare l'efficienza del processo di sviluppo, tra cui:

- Test di Complessità del Testo LaTeX ad ogni commit (Indice Gulpease)
- Aggiornamento della pagina GitHub Pages ad ogni rilascio della pagina di Glossario

Tali file sono presenti nella cartella `.github/workflows` della repository.

#### **5.0.0.2 Jira**

Strumento di gestione delle attività di progetto adottato per potenzialità e flessibilità del sistema. Permette di tracciare le attività di progetto, assegnarle ai membri del gruppo, monitorare lo stato di avanzamento e gestire le scadenze.

##### **5.0.0.2.1 Automation**

Il Team ha deciso di adottare alcune automazioni per facilitare la gestione delle attività di progetto. Le automazioni attualmente implementate sono:

- Make child work items inherit labels from parent work items
- When a commit is made → then move issue to in progress
- When all child work items are completed → then close parent
- When all sub-tasks are done → move parent to done
- When Item In Progress → Parent In Progress
- Diario Assegnato → Creo Preparazione Slide
- Riunione Assegnata → Creo Scrittura Verbale

##### **5.0.0.2.2 Smart Dashboard**

Il Team utilizza una dashboard personalizzata per monitorare lo stato di avanzamento delle attività di progetto. La dashboard include grafici e report che forniscono una panoramica delle attività in corso, delle scadenze e delle performance del team. La dashboard è accessibile a tutti i membri del gruppo ([Dashboard](#)) e viene aggiornata automaticamente in base alle modifiche apportate alle attività di progetto. È stata scelta tra gli strumenti principali per la gestione delle attività di progetto e per la tracciabilità delle ore lavorate.

Una dashboard ad hoc è stata creata per l'azienda proponente, in modo da poter monitorare l'andamento del progetto in tempo reale, la quale l'ha accolta positivamente.

### 5.0.0.2.3 Jira Smart Commit

Gli Smart Commit sono utilizzati per collegare i commit GitHub alle issue Jira. Questo permette di aggiornare automaticamente lo stato delle task e rendicontare il tempo.

La sintassi obbligatoria è:

DIPR-<numero issue> #time <nd nh nm> #<stato> #comment <Descrizione>

#### Regole di applicazione:

1. **Time Tracking:** L'inserimento del tempo (#time) deve seguire il formato *nd nh nm* (giorni, ore, minuti).
2. **Transizioni di Stato:** Utilizzare i tag #start-progress, #start-review, #complete per avanzare il workflow su Jira.
3. **Pull Request:** Nelle PR è vietato inserire il tag #time per evitare la doppia contabilizzazione delle ore lavorative.
4. **Issue Multiple:** È possibile agire su più issue in un singolo commit separandole con spazi (utile per chiudere issue di sviluppo e verifica contemporaneamente).

Importante precisazione: affinché i commit vengano riconosciuti da Jira, è necessario che la **email** con cui si effettua il commit sia la **stessa associata all'account Jira**.

### 5.0.0.3 VSCode

Ambiente di sviluppo integrato (IDE) utilizzato per la scrittura del codice e la gestione della documentazione di progetto. Grazie all'estensione Atlassian per **Jira** e **GitHub**, il Team può integrare direttamente le funzionalità di gestione delle attività e del versionamento all'interno dell'IDE, migliorando l'efficienza del flusso di lavoro e uniformando le pratiche di sviluppo.

Si presuppone che tutti i membri del gruppo si adattino allo standard comune. Le estensioni attualmente utilizzate sono:

- **Atlassian: Jira, Rojo Dev, Bitbucket**
- **GitHub Pull Requests and Issues**
- **GitHub Actions**
- **GitHub Codespaces**
- **LaTeX Workshop**

Altre estensioni come GitHub Copilot possono essere utilizzate a discrezione del membro del gruppo, al fine di velocizzare, per esempio, il processo di Documentazione 3.1.

#### **5.0.0.3.1 Atlassian: Jira, Rovo Dev, Bitbucket**

Estensione per l'integrazione di Jira e GitHub in VSCode. Permette di visualizzare e gestire le issue Jira direttamente dall'IDE, creare branch di lavoro basati sulle issue e monitorare lo stato di avanzamento delle attività.

#### **5.0.0.3.2 GitHub Pull Requests and Issues**

Estensione per la gestione delle pull request e delle issue GitHub. Permette di creare, revisionare e gestire le pull request direttamente dall'IDE, migliorando l'efficienza del processo di revisione del codice.

#### **5.0.0.4 Google Presentazioni**

Strumento utilizzato per la creazione dei Diari di Bordo. La scelta di questo strumento è dovuta alla sua facilità d'uso e alla possibilità di collaborare in tempo reale tra i membri del gruppo. Così è possibile tracciare le problematiche e i dubbi emersi durante lo svolgimento delle attività di progetto. Per i Diari di Bordo è stato impostato un template presente nel Google Drive di gruppo.

#### **5.0.0.5 Google Drive**

Strumento di archiviazione e condivisione dei file con l'azienda proponente e per la condivisione rapida dei Diari di Bordo.

Questo strumento è inoltre utilizzato come mezzo di condivisione dei file con **Sanmarco Informatica**, quali materiali formativi, materiale d'esempio e normative aziendali. Sempre tramite Google Drive vengono condivisi i verbali esterni con l'azienda proponente, in modo da poter essere approvati e archiviati in modo sicuro.

#### **5.0.0.6 Google Mail**

Strumento di comunicazione formale con l'azienda proponente e per la gestione delle comunicazioni ufficiali del gruppo.

Tutti i membri del gruppo devono utilizzare l'account di posta elettronica ufficiale del gruppo per le comunicazioni formali, garantendo così la tracciabilità e la professionalità nelle interazioni esterne.

Le comunicazioni esterne sono gestite dal Responsabile salvo diverse indicazioni, il quale deve ricordare che parla a nome di tutto il gruppo.

#### **5.0.0.7 Discord**

Strumento di comunicazione principale del gruppo, utilizzato per la coordinazione delle attività, la condivisione di informazioni e la collaborazione tra i membri del team.

L'accesso al server Discord del gruppo è obbligatorio per tutti i membri, in quanto rappresenta il canale ufficiale di comunicazione e supporto.

#### 5.0.0.8 Whatsapp

Strumento di comunicazione informale e sincrono tra i membri del gruppo, utilizzato per la coordinazione rapida delle attività e la condivisione di informazioni urgenti.

Si raccomanda di essere particolarmente responsivi su questo canale, in quanto spesso viene utilizzato per comunicazioni che richiedono una risposta tempestiva.

## 5.1 Processo di Miglioramento

Il processo di miglioramento continuo mira a identificare e implementare modifiche ai processi e alle pratiche di progetto per aumentare l'efficienza e la qualità del lavoro svolto.

### 5.1.1 Attività: Analisi degli Incidenti

L'attività di Analisi degli Incidenti riguarda l'identificazione, documentazione e risoluzione dei problemi riscontrati durante l'utilizzo degli strumenti e dei processi di progetto.

#### 5.1.1.1 Procedure di Analisi degli Incidenti

Le seguenti procedure guidano il Team nella gestione degli incidenti e nella formulazione di soluzioni:

1. **Segnalazione:** Qualsiasi membro del team può segnalare un incidente tramite Jira creando un'issue con label "incident".
2. **Analisi della causa radice:** Il Responsabile coordina l'analisi per identificare le cause sottostanti.
3. **Implementazione della soluzione:** Una volta identificata la causa, viene pianificata e implementata una soluzione.
4. **Verifica della risoluzione:** Viene verificato che la soluzione risolva effettivamente l'incidente.
5. **Archiviazione e apprendimento:** L'incidente e la soluzione vengono archiviati per consultazione futura e per evitare ricorrenze. Tali informazioni vengono tracciate nei Verbalì Interni e nei Piano di Progetto.

### 5.1.2 Attività: Monitoraggio degli Strumenti

Il monitoraggio degli strumenti è un'attività fondamentale per garantire che tutti i membri del gruppo utilizzino gli strumenti in modo efficace e coerente. Questo include il controllo dell'utilizzo dei tool di comunicazione, collaborazione e gestione del progetto, nonché l'identificazione di eventuali problemi o inefficienze.

### 5.1.2.1 Procedure di Monitoraggio

Le seguenti procedure guidano il Team nel monitoraggio degli strumenti di progetto, garantendo un uso coerente ed efficiente delle risorse disponibili.

1. **Raccolta feedback:** viene raccolto il feedback dai membri del team sull'efficacia degli strumenti utilizzati.
2. **Analisi delle prestazioni:** vengono analizzate le prestazioni degli strumenti per identificare eventuali problemi o inefficienze. A questo scopo si possono utilizzare metriche specifiche per lo strumento e la task presa in considerazione.

### 5.1.2.2 Strumenti di Monitoraggio

Gli strumenti adottati per il monitoraggio degli strumenti di progetto sono i seguenti.

- **Jira:** Utilizzato per tracciare le attività di monitoraggio e raccogliere feedback dai membri del team.
- **Discord:** Utilizzato per comunicare con i membri del team e raccogliere feedback sugli strumenti utilizzati.
- **Whatsapp:** Utilizzato per comunicazioni rapide e raccolta di feedback urgenti.

### 5.1.3 Attività: Modifiche all'Infrastruttura

L'attività di Modifiche all'Infrastruttura riguarda l'implementazione di modifiche agli strumenti e alle tecnologie utilizzate nel progetto, al fine di migliorarne l'efficienza e l'affidabilità.

#### 5.1.3.1 Procedure di Modifica

Le seguenti procedure guidano il Team nell'implementazione delle modifiche all'infrastruttura di progetto, garantendo un uso coerente ed efficiente delle risorse disponibili.

1. **Pianificazione delle modifiche:** viene pianificata l'implementazione delle modifiche, definendo gli obiettivi, le risorse necessarie e i tempi di esecuzione.
2. **Implementazione delle modifiche:** le modifiche vengono implementate secondo la pianificazione stabilita e seguendo le stesse norme descritte in 4.2.1 e 4.2.2.
3. **Verifica delle modifiche:** viene effettuata una verifica delle modifiche per garantire che siano state implementate correttamente e che abbiano raggiunto gli obiettivi prefissati.

### 5.1.3.2 Strumenti di Modifica

Gli strumenti adottati per l'implementazione delle modifiche all'infrastruttura di progetto sono i seguenti.

- **Jira:** Utilizzato per tracciare le attività di modifica e monitorare lo stato di avanzamento.
- **GitHub:** Utilizzato per gestire i branch o repository di testing degli strumenti e per la documentazione di questi ultimi.

### 5.1.4 Frequenza dei Cicli di Miglioramento

I cicli di miglioramento continuo seguono una cadenza regolare:

- **Review settimanale:** Durante le riunioni di team, vengono discussi brevemente i feedback e i problemi riscontrati.
- **Review mensile:** Una sessione dedicata mira a identificare modelli nei feedback e proporre miglioramenti significativi.
- **Review trimestrale:** Una valutazione più approfondita della qualità complessiva dei processi e dell'infrastruttura.

### 5.1.5 Metriche di Miglioramento

Il progresso dei miglioramenti implementati viene misurato attraverso le seguenti metriche:

- **Tempo di Risoluzione degli Incidenti:** Media del tempo impiegato per risolvere i problemi segnalati (target: < 48 ore per problemi critici).
- **Satisfaction Index:** Feedback qualitativo raccolto dai membri del team sulla soddisfazione riguardo gli strumenti e i processi.
- **Adoption Rate:** Percentuale di utilizzo effettivo dei nuovi strumenti o processi implementati.
- **Numero di Incidenti Ricorrenti:** Monitoraggio dei problemi che si ripetono, indicatore di efficacia della risoluzione.

## 5.2 Processo di Formazione

Il processo di formazione mira a garantire che tutti i membri del gruppo abbiano le competenze e le conoscenze necessarie per svolgere efficacemente le loro attività di progetto.



### 5.2.1 Attività: Pianificazione della Formazione

La pianificazione della formazione è un'attività fondamentale per garantire un'efficace trasmissione delle conoscenze. La pianificazione di queste attività segue la pianificazione generale del progetto, in modo da integrare la formazione con le altre attività di progetto.

#### 5.2.1.1 Procedure di Pianificazione

Le seguenti procedure guidano il Team nella pianificazione delle attività di formazione, garantendo un uso coerente ed efficiente delle risorse disponibili.

1. **Identificazione delle necessità formative:** vengono identificate le necessità formative dei membri del team, in base alle competenze richieste per le attività di progetto.
2. **Pianificazione delle attività formative:** vengono pianificate le attività formative, definendo gli obiettivi, i contenuti, le risorse necessarie e i tempi di erogazione.
3. **Allocazione delle risorse:** vengono allocate le risorse umane e strumentali necessarie per garantire l'efficacia della formazione.

### 5.2.2 Attività: Erogazione della Formazione

L'erogazione della formazione si implementa attraverso sessioni di formazione. In particolare, sono identificabili due tipologie di sessioni:

#### 5.2.2.1 Procedure di Erogazione

Le seguenti procedure guidano il Team nell'erogazione delle attività di formazione e la condivisione delle conoscenze al fine di migliorare le competenze dell'intero gruppo.

- **Formazione Autonoma:** Sessioni di formazione condotte dai membri del gruppo con competenze specifiche su determinati argomenti e strumenti in modo individuale.
- **Confronto:** Sessioni di formazione condotte da tutti i membri coinvolti su un argomento specifico, con l'obiettivo di condividere conoscenze, best practice e formulare soluzioni comuni.
- **Archiviazione:** I materiali e le note della formazione vengono salvati nelle Norme di Progetto per rendere disponibili a tutti i materiali di Formazione.

### 5.2.3 Attività: Valutazione della Formazione

L'attività di Valutazione della Formazione serve a verificare l'efficacia delle sessioni di formazione e l'acquisizione delle competenze da parte dei partecipanti.

### 5.2.3.1 Procedure di Valutazione

Le seguenti procedure guidano il Team nella valutazione dell'efficacia della formazione:

1. **Feedback immediato:** Al termine di ogni sessione di formazione, viene raccolto feedback qualitativo dai partecipanti sulla chiarezza e l'utilità dei contenuti.
2. **Verifica pratica:** Nei giorni successivi alla formazione, vengono assegnati esercizi pratici o task che richiedono l'applicazione delle conoscenze apprese.
3. **Valutazione del risultato:** Le performance nei task assegnati vengono valutate per determinare il livello di acquisizione delle competenze.
4. **Seguito:** Nel caso in cui i risultati della valutazione indichino lacune, viene organizzata formazione integrativa.

### 5.2.3.2 Strumenti di Valutazione

Gli strumenti utilizzati per la valutazione della formazione includono:

- **Jira:** Per tracciare i task di verifica assegnati post-formazione.
- **Discord:** Per la discussione e il confronto tra i membri coinvolti nella formazione.
- **VSCode:** IDE principale del gruppo che grazie alle integrazioni con Jira e GitHub permette una gestione semplice della procedura di Verifica pratica.

## 6 Metriche della qualità

### 6.1 Attività per la qualità

Lo standard ISO/IEC 9126-1:1999 individua tre categorie fondamentali di requisiti per la qualità che concorrono alla creazione del prodotto e interconnesse tra loro secondo il Modello a V<sub>(Figura 5)</sub> (*V-Model<sub>G</sub>*):

- **Bisogni utente:** sono i requisiti che il prodotto deve soddisfare per incontrare i bisogni dell'utente. L'output corrispondente è la qualità in uso, verificata da apposite metriche esterne.
- **Requisiti di qualità esterni:** corrispondono ai requisiti che riguardano le caratteristiche proprie del prodotto da realizzare. Il loro soddisfacimento definisce la qualità esterna del prodotto, validata anch'essa da opportune metriche esterne.
- **Requisiti di qualità interni:** sono i requisiti legati strettamente ai processi di sviluppo software che quando rispettati risultano in codice di qualità, processi efficaci e organizzazione efficiente.

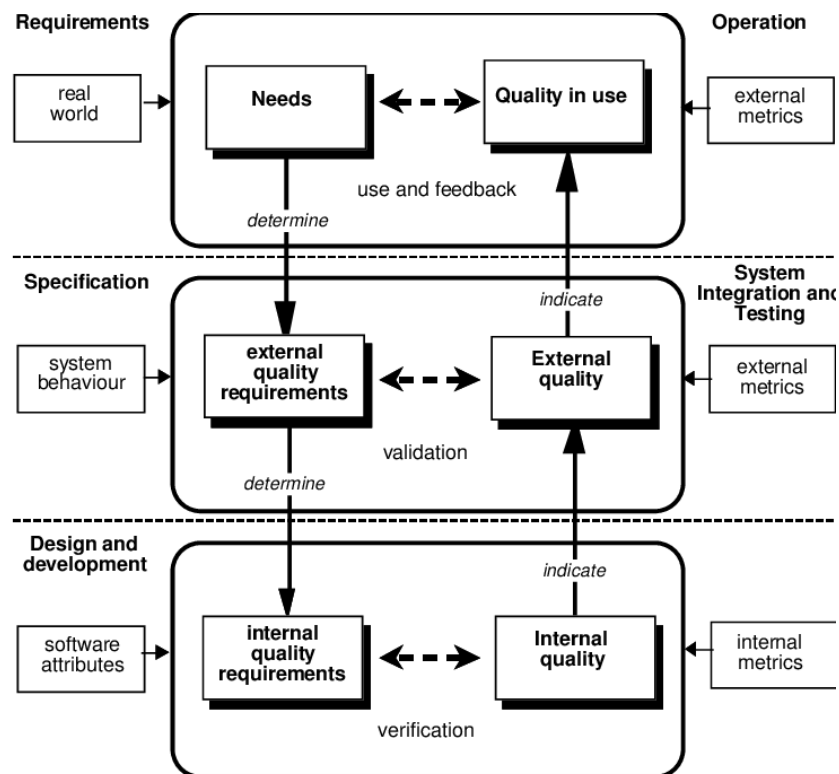


Figura 5: Modello a V

## 6.2 Procedure per la qualità

Di seguito vengono indicate le metriche utilizzate per valutare la qualità del prodotto e dei processi di sviluppo.

### 6.2.1 Qualità di Processo

In questa sezione vengono definite le metriche utilizzate per valutare l'efficienza e l'efficacia dei processi del ciclo di vita del software, in conformità con lo standard ISO/IEC 12207.

#### 6.2.1.1 Processo di fornitura

##### MPC-1 - Earned Value (EV)

**Descrizione:** Valore del lavoro effettivamente completato in un determinato momento, espresso in termini di budget approvato.

**Formula di calcolo:**

$$EV = \sum_i (\% \text{Completamento}_i \times \text{Budget}_i)$$

**MPC-2 - Planned Value (PV)**

**Descrizione:** Valore del lavoro che si era pianificato di completare entro una certa data.

**Formula di calcolo:**

$$PV = \sum_i (\%Pianificato_i \times Budget_i)$$

**MPC-3 - Actual Cost (AC)**

**Descrizione:** Costo realmente sostenuto per il lavoro svolto fino alla data corrente.

**Formula di calcolo:**

$$AC = \sum \text{Costi Sostenuti}$$

**MPC-4 - Cost Performance Index (CPI)**

**Descrizione:** Indice di efficienza dei costi. Un valore  $< 1$  indica che il progetto è fuori budget.

**Formula di calcolo:**

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

**MPC-5 - Schedule Performance Index (SPI)**

**Descrizione:** Indice di efficienza della schedulazione. Un valore  $< 1$  indica che il progetto è in ritardo.

**Formula di calcolo:**

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

**6.2.1.2 Processo di Sviluppo****MPC-11 - Work in Progress (WIP)**

**Descrizione:** Numero di task o ticket su cui il team sta lavorando attivamente in contemporanea. Un valore alto indica colli di bottiglia.

**Formula di calcolo:**

$$WIP = \sum \text{Ticket in stato "In Progress"}$$

**MPC-13 - Change Failure Rate (CFR)**

**Descrizione:** Percentuale di deployment o merge in produzione che causano un guasto richiedendo un hotfix.

**Formula di calcolo:**

$$CFR = \left( \frac{\text{Numero di deployment falliti}}{\text{Numero totale di deployment}} \right) \times 100$$

**6.2.1.3 Processo di Documentazione****MPC-14 - Indice di Gulpease**

**Descrizione:** Indice che valuta la leggibilità di un testo in lingua italiana. Valori bassi indicano bassa leggibilità.

**Formula di calcolo:**

$$G = 89 + \frac{300 \times (\text{numero frasi}) - 10 \times (\text{numero lettere})}{\text{numero parole}}$$

**MPC-15 - Correttezza Ortografica**

**Descrizione:** Misura la densità di errori ortografici presenti nella documentazione.

**Formula di calcolo:**

$$C = 1 - \left( \frac{\text{Numero errori ortografici}}{\text{Numero parole totali}} \right)$$

**6.2.1.4 Processo di Configurazione****MPC-16 - Build Success Rate (BSR)**

**Descrizione:** Percentuale di build completate con successo dalla pipeline di CI/CD rispetto al totale delle build eseguite.

**Formula di calcolo:**

$$BSR = \left( \frac{\text{Builds con esito positivo}}{\text{Builds totali}} \right) \times 100$$

**MPC-17 - Average Build Time**

**Descrizione:** Tempo medio impiegato dalla pipeline per completare il processo di build e test automatici.

**Formula di calcolo:**

$$T_{avg} = \frac{\sum_{i=1}^n T_{build\_i}}{n}$$

**6.2.1.5 Processo di Verifica****MPC-19 - Statement Coverage**

**Descrizione:** Percentuale di istruzioni (statements) del codice sorgente eseguite durante i test automatici.

**Formula di calcolo:**

$$SC = \left( \frac{\text{Linee di codice eseguite}}{\text{Linee di codice totali}} \right) \times 100$$

**MPC-20 - Branch Coverage**

**Descrizione:** Percentuale di rami decisionali (if, switch, loop) percorsi durante i test.

**Formula di calcolo:**

$$BC = \left( \frac{\text{Rami percorsi}}{\text{Rami totali}} \right) \times 100$$

**6.2.1.6 Processi Organizzativi)****MPC-21 - Retrospective Effectiveness**

**Descrizione:** Misura la capacità del team di risolvere i problemi identificati durante le retrospettive.

**Formula di calcolo:**

$$E_{retro} = \left( \frac{\text{Action items risolti}}{\text{Action items identificati}} \right) \times 100$$

**MPC-22 - Meeting Attendance**

**Descrizione:** Percentuale di presenza dei membri del team alle riunioni obbligatorie.

**Formula di calcolo:**

$$MA = \left( \frac{\text{Presenze effettive}}{\text{Presenze attese totali}} \right) \times 100$$

**6.2.2 Qualità di Prodotto**

In questa sezione vengono definite le metriche per valutare la qualità intrinseca del prodotto software, basandosi sul modello ISO/IEC 9126.

**6.2.2.1 Funzionalità****MPD-2 - Completezza dell'implementazione**

**Descrizione:** Misura la percentuale di requisiti funzionali obbligatori implementati e testati.

**Formula di calcolo:**

$$CI = \left( \frac{\text{Requisiti funzionali implementati}}{\text{Requisiti funzionali totali}} \right) \times 100$$

**MPD-3 - Accuratezza dei risultati**

**Descrizione:** Percentuale di test funzionali che producono il risultato atteso senza deviazioni.

**Formula di calcolo:**

$$Acc = \left( \frac{\text{Test funzionali passati}}{\text{Test funzionali eseguiti}} \right) \times 100$$

**6.2.2.2 Affidabilità****MPD-6 - Tasso Errori HTTP**

**Descrizione:** Percentuale di richieste al server che restituiscono codici di errore (4xx o 5xx) rispetto al totale delle richieste.

**Formula di calcolo:**

$$E_{http} = \left( \frac{\text{Richieste con status} \geq 400}{\text{Richieste totali}} \right) \times 100$$

**MPD-NEW - Availability (Disponibilità)**

**Descrizione:** Percentuale di tempo in cui il sistema è operativo e accessibile agli utenti.

**Formula di calcolo:**

$$A = \left( \frac{\text{Tempo totale} - \text{Tempo di disservizio}}{\text{Tempo totale}} \right) \times 100$$

**6.2.2.3 Usabilità****MPD-9 - Profondità di Navigazione**

**Descrizione:** Numero massimo di click necessari per raggiungere una qualsiasi pagina di contenuto partendo dalla Home Page.

**Formula di calcolo:**

$$Clicks \leq 3$$

**MPD-10 - Search Success Rate**

**Descrizione:** Percentuale di ricerche effettuate dagli utenti che portano al clic su un risultato utile.

**Formula di calcolo:**

$$SSR = \left( \frac{\text{Ricerche con clic}}{\text{Ricerche totali}} \right) \times 100$$

#### 6.2.2.4 Efficienza

##### MPD-11 - Time to First Byte (TTFB)

**Descrizione:** Tempo che intercorre tra l'invio della richiesta HTTP dal client e la ricezione del primo byte di risposta dal server.

**Formula di calcolo:**

$$TTFB = T_{ricezione\_primo\_byte} - T_{invio\_richiesta}$$

##### MPD-13 - Tempo di Rendering

**Descrizione:** Tempo necessario al browser per completare il rendering visivo della pagina (First Contentful Paint).

**Formula di calcolo:**

$$T_{render} \text{ (misurato tramite API del browser)}$$

#### 6.2.2.5 Manutenibilità

##### MPD-16 - Complessità Ciclomantica

**Descrizione:** Misura la complessità del flusso di controllo del codice. Valori superiori a 15 indicano codice difficile da testare e mantenere.

**Formula di calcolo:**

$$M = E - N + 2P \quad (\text{dove } E = \text{archi}, N = \text{nodi}, P = \text{componenti connessi})$$

##### MPD-17 - Comment Ratio

**Descrizione:** Rapporto tra le righe di commento e le righe di codice totali (LOC).

**Formula di calcolo:**

$$CR = \frac{\text{Linee di commento}}{\text{Linee di Codice (SLOC)}}$$

##### MPD-18 - Accoppiamento tra Classi (CBO)

**Descrizione:** Numero di classi a cui una determinata classe è accoppiata (usa o è usata da). Un valore alto indica scarsa modularità.

**Formula di calcolo:**

$$CBO = \text{Conteggio dipendenze esterne per classe}$$

## 6.3 Strumenti