



7zpus.swe@gmail.com

Norme di Progetto

Indice

1 Introduzione	5
1.1 Scopo	5
1.2 Glossario	5
1.3 Riferimenti	5
1.3.1 Riferimenti Normativi	5
1.3.2 Riferimenti Informativi	5
2 Processi Primari	6
2.1 Processo di Fornitura	6
2.1.1 Attività di processo	6
2.1.1.1 Avvio	6
2.1.1.2 Preparazione della proposta di fornitura	6
2.1.1.3 Accordo	6
2.1.1.4 Pianificazione	6
2.1.1.5 Esecuzione e controllo	7
2.1.1.6 Verifica e validazione	7
2.1.1.7 Consegna e terminazione	7
2.1.1.8 Accordi con l'azienda proponente	7
2.1.2 Documentazione fornita	7
2.1.2.1 Analisi dei requisiti	7
2.1.2.2 Glossario	8
2.1.2.3 Piano di progetto	8
2.1.2.4 Piano di qualifica	8
2.1.2.5 Lettera di presentazione	9
2.1.3 Strumenti	9
2.2 Processo di sviluppo	9
2.2.1 Attività di processo	9
2.3 Processo operativo	9
2.4 Processo di manutenzione	9

3 Processi di Supporto	9
3.1 Processo di documentazione	9
3.1.1 Attività: Pianificazione della documentazione	9
3.1.1.1 Procedure di Pianificazione	10
3.1.1.2 Strumenti di Pianificazione	10
3.1.2 Attività: Produzione della documentazione	10
3.1.2.1 Procedure di Produzione	11
3.1.2.1.1 Denominazione e datazione	12
3.1.2.2 Strumenti di Produzione	13
3.1.3 Attività: Revisione e Approvazione	13
3.1.3.1 Procedure di Revisione e Approvazione	13
3.1.3.2 Strumenti di Revisione e Approvazione	14
3.2 Processo di Gestione della Configurazione	14
3.2.1 Attività: Identificazione della Configurazione	15
3.2.1.1 Procedure: Identificazione degli SCIs	15
3.2.1.1.1 Issue e SubIssue	16
3.2.1.2 Strumenti di Identificazione	16
3.2.2 Attività: Versionamento e Identificazione	16
3.2.2.1 Procedure: Standard per le Branch	16
3.2.2.2 Strumenti di Versionamento	16
3.2.3 Attività: Controllo delle Configurazioni	17
3.2.3.1 Procedure: Smart Commit	17
3.2.3.2 Strumenti	17
3.2.4 Registrazione delle Configurazioni	17
3.2.4.1 Procedure: Registrazione delle Modifiche	17
3.3 Processo di garanzia della qualità	18
3.4 Processo di verifica	18
3.5 Processo di validazione	18
3.6 Processo di revisione congiunta	18
3.7 Processo di risoluzione dei problemi	18
3.8 Gestione della qualità	18
4 Processi Organizzativi	18
4.1 Gestione	18
4.2 Infrastruttura	18
4.2.1 Attività di processo	18
4.2.2 Procedure di processo	19
4.2.3 Strumenti a supporto	19
4.2.3.1 GitHub	19
4.2.3.2 Jira	19
4.2.3.2.1 Automation	19
4.2.3.3 VSCode	19
4.3 Miglioramento	20
4.4 Formazione	20

5 Metriche della qualità	20
---------------------------------	-----------

Elenco delle figure

1	Flusso del processo di documentazione	11
2	Creazione del branch di lavoro tramite estensione Jira in VSCode	11
3	Creazione della PR verso l'issue branch <i>in_lavorazione</i>	12
4	Commit della PR con Smart Commit verso le due issue	14

Elenco delle tabelle

Tabella di Versionamento

Versione	Data	Autore	Verificatore	Descrizione
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-Sstrumenti)
0.3	25/11/2025	Soligo Lorenzo	Fattoni Antonio	Creazione e stesura sezioni Processi di Infrastruttura e sottosezioni 4.2.1-4.3 e 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_G*, ovvero le regole di lavoro che ogni membro del gruppo deve rispettare durante lo svolgimento delle *attività di progetto_G* volte allo sviluppo dell'applicativo software **DIPReader_G**, proposto dall'azienda *Sanmarco Informatica_G*. 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_G* 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_G*, 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_G* 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_G* e *iterativi_G* che caratterizzano lo *sviluppo software_G* 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^AT_EX by Lorenzo Pantieri](#) (ultimo accesso: 17/11/2025)
 - [Documentazione Jira](#)

2 Processi Primari

2.1 Processo di Fornitura

Il *processo_G* di fornitura contiene le attività e i compiti svolti dal *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 *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_G* 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 capitolati presentati dalle proponenti vengono analizzati e viene redatto il documento di [Analisi dei capitolati](#), 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_G*, in qualità di *stakeholder_G*, 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_G*) 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_G* principali: *RTB_G* (Requirements and Technology Baseline) e *PB_G* (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_G* (*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 $sprint_G$. 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_G*). 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_G (*Requirements and Technology Baseline*). Essa contiene le informazioni sul repository di progetto, il puntatore al $Proof of Concept_G$, il consuntivo di spesa e preventivo a finire del progetto.

2.1.3 Strumenti

- $GitHub_G$ per la gestione della documentazione di progetto e mezzo comunicativo nella fase di fornitura
- $Jira_G$ 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

2.4 Processo di manutenzione

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 $Agile_G$ di continuo miglioramento e adattamento tramite $feedback_G$.

3.1.1 Attività: Pianificazione della documentazione

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

Durante la pianificazione di ogni $sprint_G$, il *responsabile di progetto_G* 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 $proponente_G$, $committente_G$).

Per una più efficiente scrittura dei documenti, soprattutto di tutti quei documenti periodici (Verbali Interni, Verbali Esterni, Diario di Bordo) sono presenti modelli standard approvati in [/assets](#). L'aggiornamento di tali standard deve essere argomento di Verbali 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 $sprint_G$, il $responsabile_G$ 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, nonchè della necessità di ruotare i ruoli, per dare la possibilità a tutti i membri di acquisire esperienza in diverse aree.
3. Vengono create le $issue_G$ 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_G* per la gestione delle attività di progetto. In particolare con la *board_G Scrum_G* che viene aggiornata in automatico con i commit effettuati sui Work Item e può essere personalizzata con la creazione di *Sprint_G*.
- *DashBoard/Cruscotto_G* 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_G* grazie all'estensione Jira di VSCode 4.2.3.3. 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_G* 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 3.2.3.1.

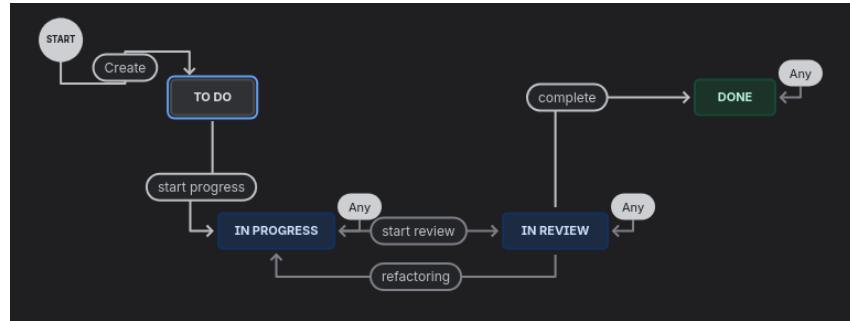


Figura 1: Flusso del processo di documentazione

3.1.2.1 Procedure di Produzione

I seguenti passaggi guidano il Team nella produzione di documenti:

1. Consultando l'estensione "[Atlassian: Jira, Rovo 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.

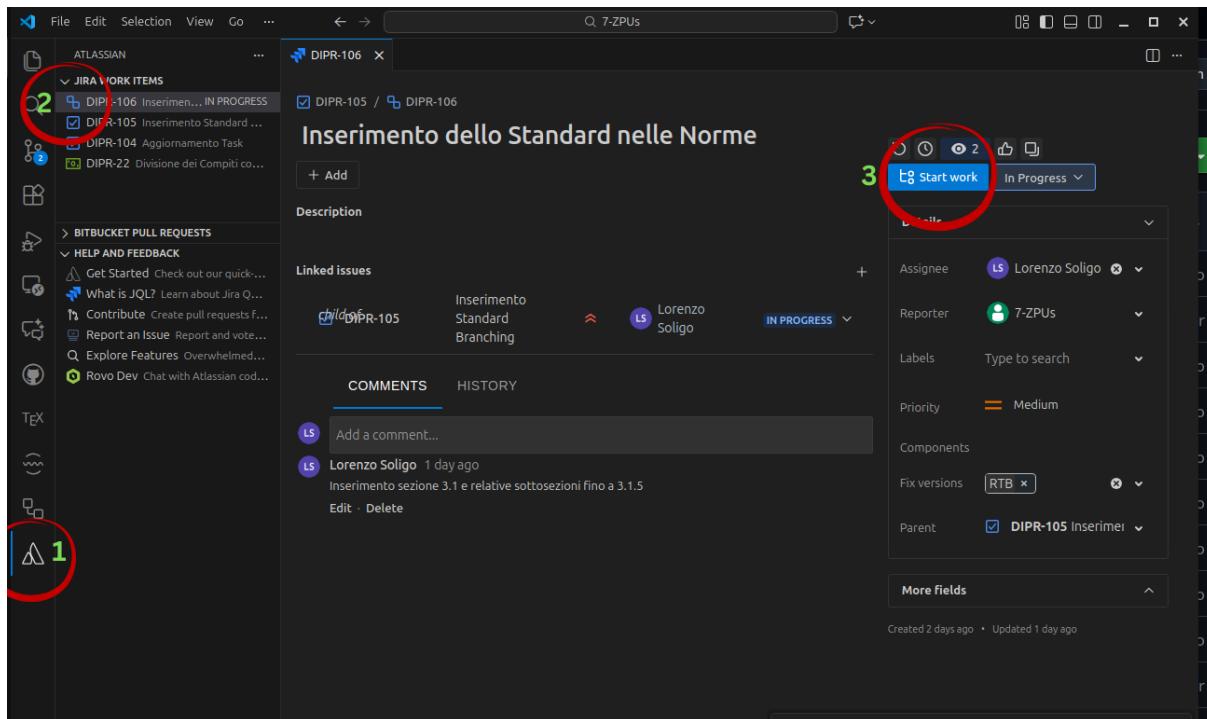


Figura 2: Creazione del branch di lavoro tramite estensione Jira in VSCode

2. Quando è necessario **effettuare un commit** è obbligatorio utilizzare lo Smart Commit. Più in 4.2.3.2.

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.

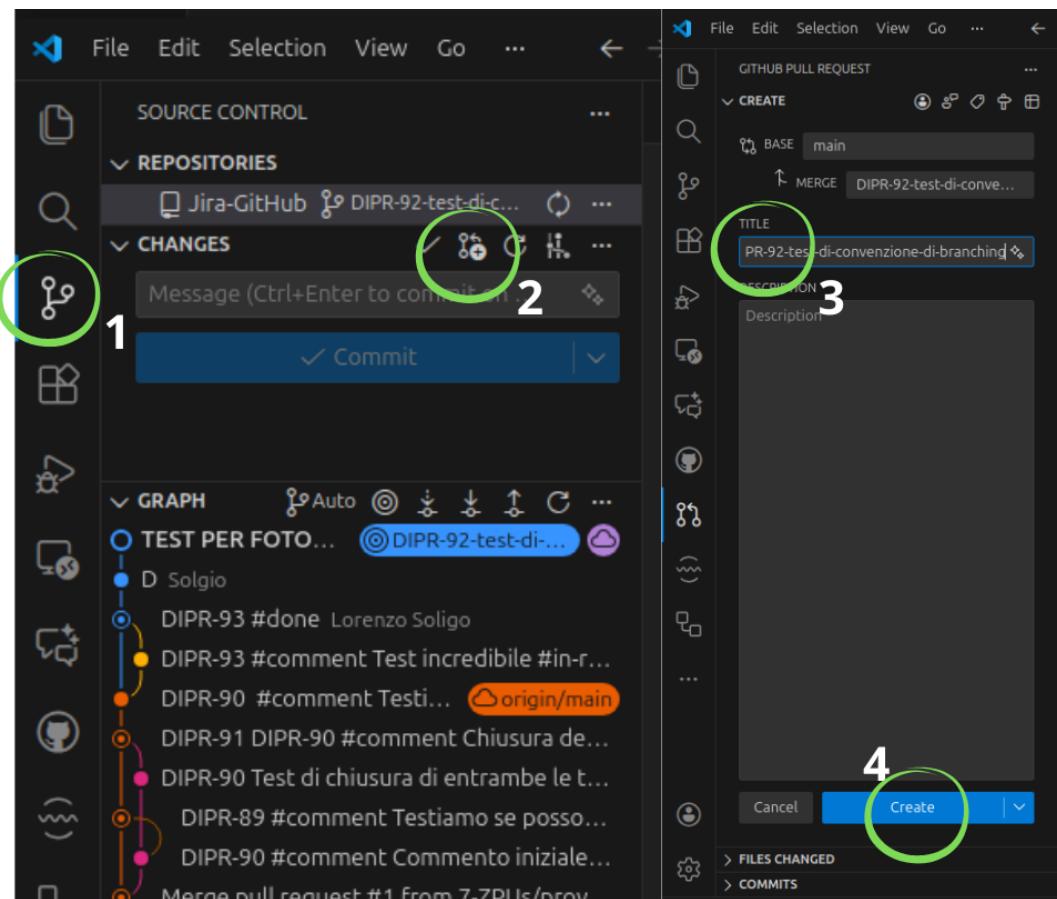


Figura 3: Creazione della PR verso l'issue branch *in_lavorazione*

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_G* come IDE principale per la stesura dei documenti in *LATEX_G*.
- *Estensione Jira per VSCode_G* per la gestione dei Work Item assegnati e la creazione automatica delle branch di lavoro.
- *GitHub_G* 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.

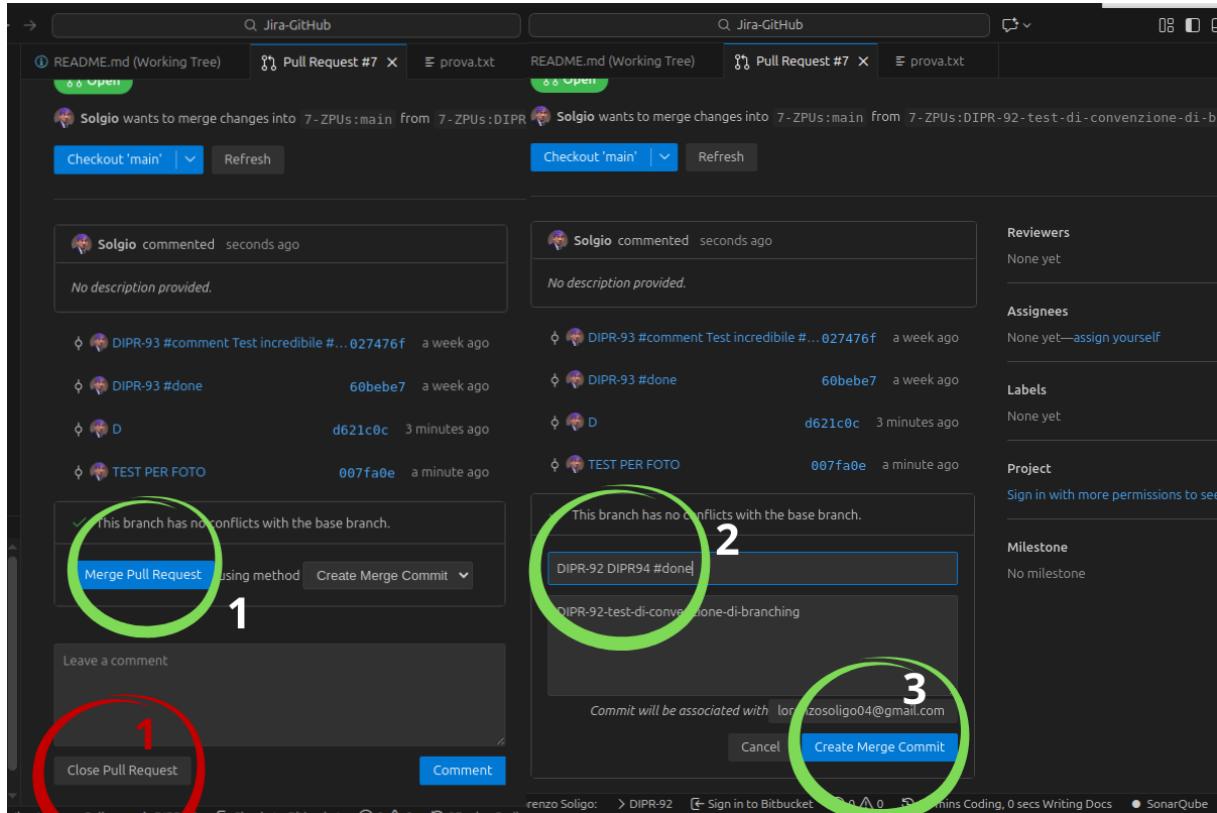


Figura 4: Commit della PR con Smart Commit verso le due issue

Per quanto riguarda l'approvazione finale del documento, questa spetta al *responsabile_G*, il quale effettua il merge da "in_lavorazione" nel ramo principale *main*, base per la versione ufficiale di rilascio corrispondente alla *milestone_G*. 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_G*, in particolare integrato in VSCode per la gestione delle versioni e delle pull request, comodamente nell'ambiente di sviluppo di VSCode 4.2.3.3. Per una stesura efficiente dei documenti il Team si è dotato di modelli predefiniti (Decisione del [2025-11-07](#)).
- *Jira_G*: strumento di gestione delle attività di progetto, utilizzato per tracciare le attività di documentazione e assegnarle ai membri del gruppo.
- *LLM_G* 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_G*) 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_G* 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_G* 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:** Similmente, 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 4.2.3.3.
- 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 regolamenta 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.

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)[cite: 97, 98].

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 Infrastruttura

Il processo di Infrastruttura fornisce il supporto tecnico necessario per lo sviluppo e la gestione del progetto.

4.2.1 Attività di processo

Il processo si articola nelle seguenti attività principali:

- **Implementazione:** scelta, configurazione e gestione degli strumenti e delle tecnologie necessarie per supportare le attività di progetto.
- **Creazione:** sviluppo e manutenzione dell'infrastruttura tecnica, strumenti, procedure e ambienti di sviluppo.

- **Manutenzione:** aggiornamento, monitoraggio e risoluzione di eventuali problemi legati all'infrastruttura esistente.

4.2.2 Procedure di processo

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

4.2.3 Strumenti a supporto

4.2.3.1 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. Vengo quindi descritti gli Smart Commit, lo standard per la scrittura di commit e la gestione dello stato di vita degli work item.

4.2.3.2 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.

4.2.3.2.1 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

4.2.3.3 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, Rovo 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.

4.3 Miglioramento

4.4 Formazione

5 Metriche della qualità