



7zpus.swe@gmail.com

Norme di Progetto

Indice

1	Introduzione	4
1.1	Scopo	4
1.2	Glossario	4
1.3	Riferimenti	4
1.3.1	Riferimenti Normativi	4
1.3.2	Riferimenti Informativi	4
2	Processi Primari	5
2.1	Processo di Fornitura	5
2.1.1	Attività di processo	5
2.1.1.1	Avvio	5
2.1.1.2	Preparazione della proposta di fornitura	5
2.1.1.3	Accordo	5
2.1.1.4	Pianificazione	5
2.1.1.5	Esecuzione e controllo	6
2.1.1.6	Verifica e validazione	6
2.1.1.7	Consegna e terminazione	6
2.1.1.8	Accordi con l'azienda proponente	6
2.1.2	Documentazione fornita	6
2.1.2.1	Analisi dei requisiti	6
2.1.2.2	Glossario	7
2.1.2.3	Piano di progetto	7
2.1.2.4	Piano di qualifica	7
2.1.2.5	Lettera di presentazione	8
2.1.3	Strumenti	8
2.2	Processo di sviluppo	8
2.2.1	Attività di processo	8
2.3	Processo operativo	8
2.4	Processo di manutenzione	8

3	Processi di Supporto	8
3.1	Processo di documentazione	8
3.1.1	Strumenti a supporto	8
3.1.2	Attività di processo	9
3.1.3	Pianificazione della documentazione	9
3.1.4	Produzione della documentazione	10
3.1.5	Revisione e approvazione	10
3.1.6	Procedure di processo	11
3.1.6.1	Autore del documento	11
3.1.6.2	Revisore del documento	12
3.1.6.3	Responsabile di progetto	13
3.2	Processo di garanzia della qualità	13
3.3	Processo di verifica	13
3.4	Processo di validazione	13
3.5	Processo di revisione congiunta	13
3.6	Processo di risoluzione dei problemi	13
3.7	Gestione della qualità	13
4	Processi Organizzativi	13
4.1	Gestione	13
4.2	Infrastruttura	13
4.2.1	Attività di processo	13
4.2.2	Procedure di processo	14
4.2.3	Strumenti a supporto	14
4.2.3.1	GitHub	14
4.2.3.2	Jira	14
4.2.3.2.1	Automation	14
4.2.3.3	VSCode	14
4.2.3.3.1	Smart Commit	15
4.2.3.3.2	Standard per le Branch	16
4.3	Miglioramento	16
4.4	Formazione	16
5	Metriche della qualità	16

Tabella di Versionamento

Versione	Data	Autore	Verificatore	Descrizione
0.3	25/11/2025	Soligo Lorenzo	Soligo Lorenzo	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	Soligo Lorenzo	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 quanto più efficace ed efficiente, oltre che omogenea. Data la natura incrementale di redazione del documento, il *responsabile di progetto_G* in carica ha il compito di mantenere aggiornate le presenti norme e eventuali riferimenti ad altri documenti contenuti al loro interno.

1.2 Glossario

Ogni termine tecnico o con particolare significato nell'ambito dell'*Ingegneria del Software_G* utilizzato nella documentazione di progetto viene definito nell'apposito documento [Glossario 1.0](#) (ultimo accesso: 17/11/2025).

1.3 Riferimenti

Il gruppo ha deciso di redigere il presente documento in conformità con lo *standard_G* ISO/IEC 12207:1995, integrando occasionalmente con approfondimenti contenuti nella sua versione più attuale ISO/IEC/IEEE 12207:2017 per includere dettagli aggiuntivi relativi agli approcci *agili_G* e *iterativi_G* che contraddistinguono 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 considerazione di 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

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 Strumenti a supporto

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

- *GitHub_G*: repository centrale per la gestione della documentazione. Permette il versionamento dei documenti e grazie alle *pull request_G* anche la gestione di *Verifica_G* e *Approvazione_G* finale dei documenti.
- *LaTeX_G*: linguaggio di markup utilizzato per la stesura dei documenti di progetto, in quanto permette di ottenere una formattazione professionale e uniforme tra i vari documenti. 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.

3.1.2 Attività di processo

Le attività principali del processo di documentazione sono:

- **Pianificazione della documentazione:** definizione delle linee guida per la stesura dei documenti, inclusi formati, modelli e standard di qualità e assegnazione della redazione ai membri del gruppo. Più in 3.1.3
- **Produzione della documentazione:** redazione dei documenti di progetto seguendo le linee guida stabilite, assicurando chiarezza, coerenza e completezza delle informazioni. Più in 3.1.4
- **Revisione e approvazione:** ogni documento redatto viene sottoposto a un processo di revisione interna da parte di un membro del gruppo diverso dall'autore.

e seguono il seguente flusso:

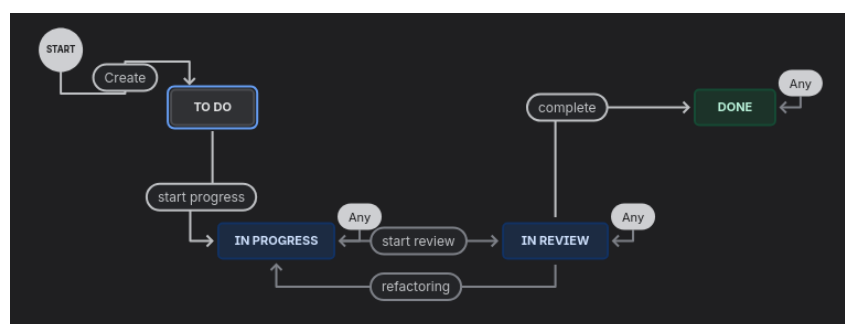


Figura 1: Flusso del processo di documentazione

3.1.3 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 (Verbalì Interni, Verbalì Esterni, Diario di Bordo) sono presenti modelli standard approvati in [/assets](#). L'aggiornamento di tali standard deve essere argomento di Verbalì Interni e risultato di una discussione e successiva decisione presa in tale sede.

3.1.4 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 bocciata** 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 4.2.3.2.

3.1.5 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 che al termine del processo può:

- **Approvare il documento.** Questo porta all'apertura di una Pull Request verso la branch principale revisionata dal *responsabile_G* che quindi confermerà l'integrazione ai documenti approvati in via definitiva.
- **Richiedere modifiche.** In questo caso il *responsabile_G* fornisce un feedback dettagliato all'autore del documento, indicando le aree che necessitano di miglioramenti o correzioni. L'autore apporta le modifiche richieste e il documento viene nuovamente sottoposto a revisione.

A questo punto il responsabile effettua il merge della feature nel ramo principale (*verbalì in lavorazione, norme in lavorazione, etc*), base per la versione ufficiale di rilascio corrispondente alla *milestone_G*.

3.1.6 Procedure di processo

Le seguenti procedure guidano il Team nella produzione di documenti, ricercando una produzione veloce, efficiente e di qualità, garantendo la tracciabilità e la divisione dei compiti. Tali procedure sono suddivise per ruolo:

- Autore del documento
- Revisore del documento
- Responsabile di progetto

3.1.6.1 Autore del documento

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

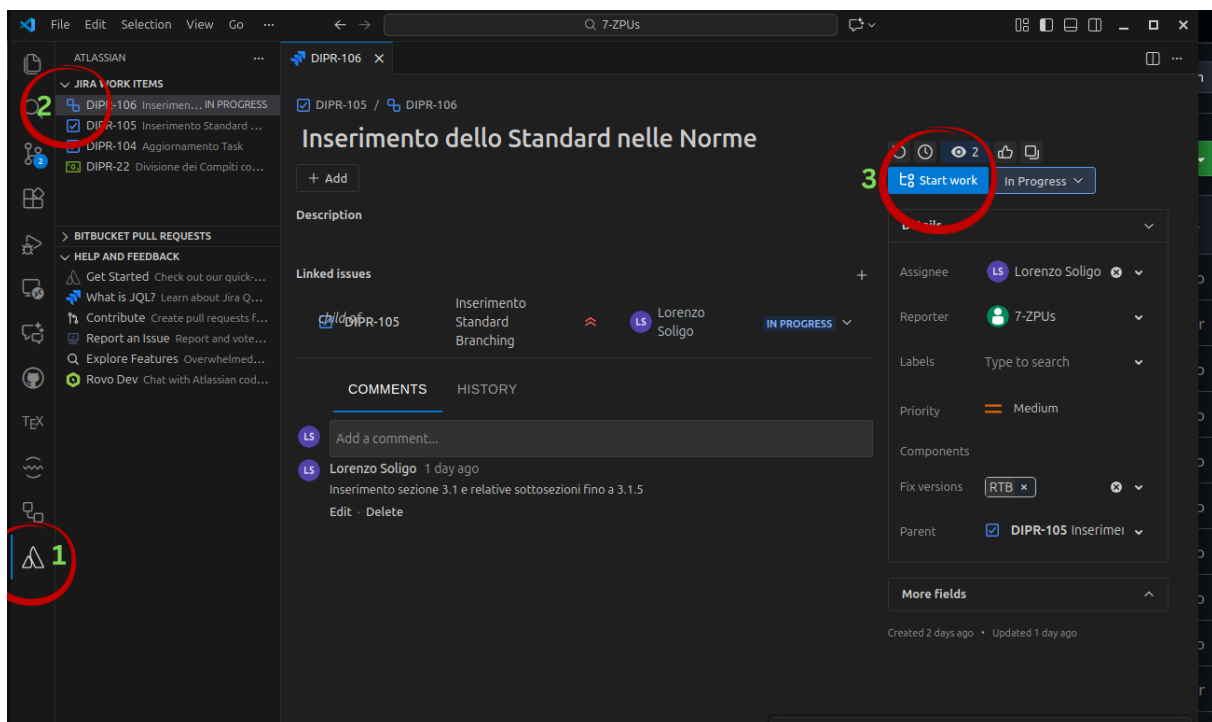


Figura 2: Estensione VSCode per Jira

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 della issue mettendo come revisore il membro del gruppo designato. La PR viene chiusa tramite commit e deve essere scritta anche in questo caso utilizzando lo Smart Commit per chiudere la propria issue e una volta approvata anche la issue principale usata come branch di controllo.

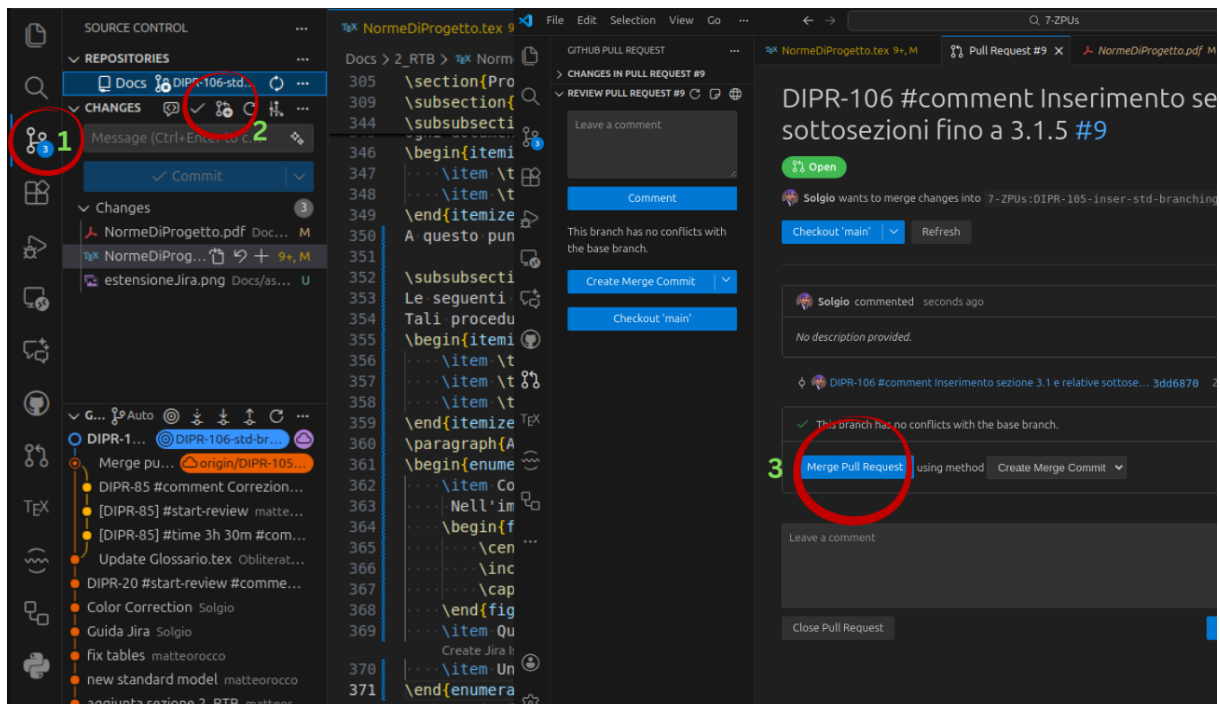


Figura 3: Creazione della PR verso la branch principale

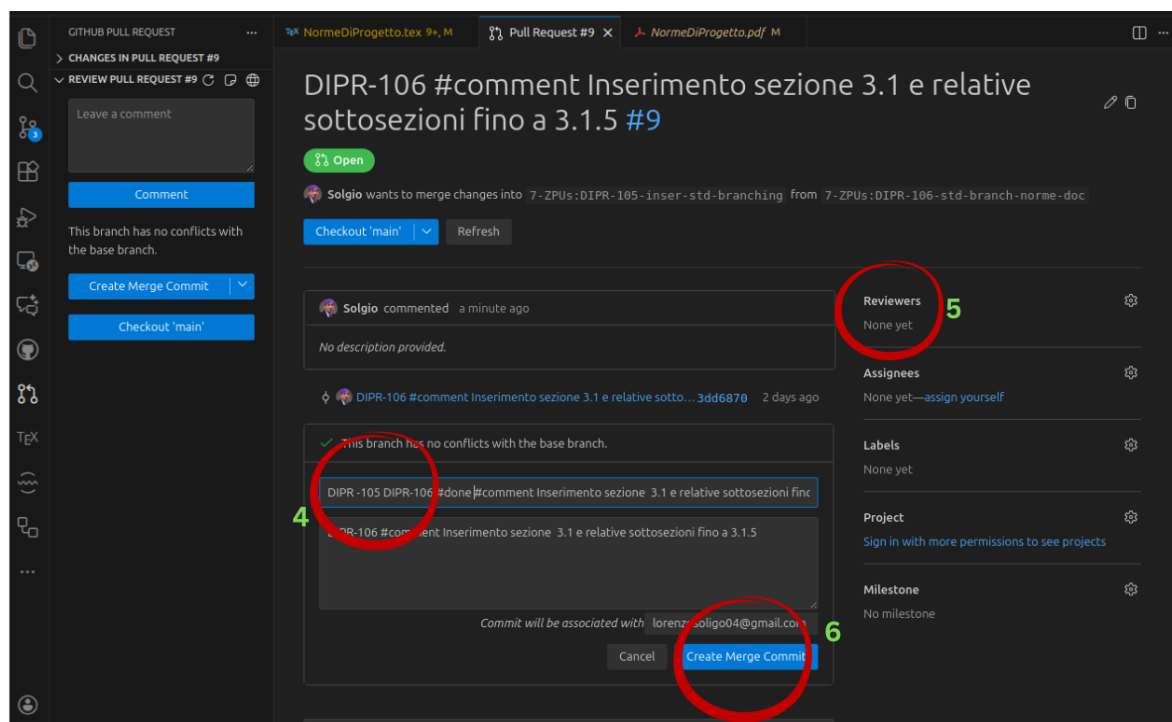


Figura 4: Commit della PR con Smart Commit verso le due issue

3.1.6.2 Revisore del documento

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. A sua volta l'autore procederà con il merge della PR verso la branch principale contenente i documenti pronti all'approvazione finale del responsabile. La procedura per la PR mima quella dell'autore (3, 3) e segue gli stessi crismi.
- Richiedere modifiche, fornendo un feedback dettagliato all'autore, bloccando la PR.

3.1.6.3 Responsabile di progetto

La funzione del responsabile di progetto è quella di approvazione finale dei documenti e 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.

Una volta approvato il documento, il responsabile approva la PR del revisore verso il branch di raccolta dei documenti ufficiali in vista del merge con il branch *main* in sede di *milestone_G*.

3.2 Processo di garanzia della qualità

3.3 Processo di verifica

3.4 Processo di validazione

3.5 Processo di revisione congiunta

3.6 Processo di risoluzione dei problemi

3.7 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

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.

4.2.3.3.1 Smart Commit

La necessità di tracciamento delle attività di progetto richiede l'adozione capillare, costante e frequente degli Smart Commit, al fine di un aggiornamento tempestivo e che rispecchi il reale avanzamento delle task.

Gli Smart Commit sono una funzionalità di Jira che permette di collegare i commit di GitHub alle issue di Jira tramite messaggi speciali nei commit. Questo consente di automatizzare la gestione delle issue, come la chiusura automatica o l'aggiornamento dello stato, direttamente dal messaggio di commit.

Lo Smart Commit segue la seguente sintassi:

```
DIPR-<numero issue> #comment <Commento descrittivo>
#time <ore lavorative consumate> #<transizione di stato>
```

Dove l'inserimento di **<ore lavorative consumate>** segue il formato **nd nh nm**, dove **n** è un numero intero, **d** indica giorni, **h** ore e **m** minuti.

Le transizioni di stato possibili sono:

- **start-progress:** inizio del lavoro sull'issue
- **start-review:** sposta in revisione l'issue
- **complete:** risoluzione dell'issue
- **refactoring:** necessità della issue di essere ritoccata

e fanno riferimento allo schema di flusso 3.1.2

Si possono compiere azioni su multiple issue in un singolo commit, basta separarle con spazi. Questo è necessario per le Pull Request, in cui si devono chiudere sia l'issue della branch di lavoro che quella della branch di verifica.

Non devono però essere inseriti *Work Log_G* in questo caso, in quanto porterebbe ad una multipla contabilizzazione delle ore.

4.2.3.3.2 Standard per le Branch

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

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

Che segue ciò che viene **automaticamente** generato dall'estensione Jira per VSCode. Al momento della creazione della branch è quindi solo di scegliere **eventuali abbreviazioni** per la descrizione breve.

4.3 Miglioramento

4.4 Formazione

5 Metriche della qualità