



ARGO

Norme di Progetto

Gruppo Argo — Progetto ChatSQL

Informazioni sul documento

Versione	0.0.6
Approvazione	TODO
Uso	Interno
Distribuzione	Prof. Tullio Vardanega Prof. Riccardo Cardin Gruppo Argo



UNIVERSITÀ
DEGLI STUDI
DI PADOVA

Registro delle modifiche

Ver.	Data	Redazione	Verifica	Descrizione
0.0.6	2024-05-06	Riccardo Cavalli	Tommaso Stocco	Automazione chiusura <i>ticket</i> _o
0.0.5	2024-05-06	Riccardo Cavalli	Tommaso Stocco	<i>Ticket</i> _o su Jira e integrazione <i>Jira</i> _o / <i>GitHub</i> _o
0.0.4	2024-05-04	Riccardo Cavalli	Martina Dall'Amico	Formalizzazione della struttura dei verbali e aggiornamento della tabella Todo dopo il passaggio a <i>Jira Software</i> _o
0.0.3	2024-04-28	Riccardo Cavalli	Martina Dall'Amico	Dashboard <i>Google Sheets</i> _o
0.0.2	2024-04-26	Riccardo Cavalli	Martina Dall'Amico, Mattia Zecchinato	Convenzioni per l'uso delle lettere maiuscole
0.0.1	2024-04-20	Tommaso Stocco	Martina Dall'Amico, Mattia Zecchinato	Creazione e stesura iniziale del documento

Indice

1	Introduzione	4
1.1	Scopo del documento	4
1.2	Scopo del prodotto	4
1.3	Glossario	4
1.4	Riferimenti	4
1.4.1	Riferimenti normativi	4
1.4.2	Riferimenti informativi	4
2	Processi primari	4
2.1	Fornitura	4
2.1.1	Descrizione	4
2.1.1.1	Selezione e studio fattibilità	5
2.1.1.2	Candidatura	5
2.1.1.3	Pianificazione	5
2.1.1.4	Esecuzione e controllo	5
2.1.1.5	Revisione e valutazione	5
2.1.2	Rapporti con la Proponente	5
2.1.3	Documentazione _e fornita	6
2.1.3.1	Piano di Progetto	6
2.1.3.2	Analisi dei Requisiti	6
2.1.3.3	Piano di Qualifica	6
2.1.3.4	Lettera di Presentazione	6
2.1.3.5	Glossario	6
2.1.4	Strumenti	6
2.1.4.1	Dashboard Google Sheets	6
2.2	Sviluppo	7
2.2.1	Descrizione	7
2.2.2	Analisi dei Requisiti	7
2.2.2.1	Descrizione	7
2.2.3	Progettazione	7
2.2.3.1	Descrizione	7
2.2.4	Codifica e testing	7
2.2.4.1	Descrizione	7
3	Processi di supporto	8
3.1	Documentazione	8
3.1.1	Descrizione	8
3.1.1.1	Implementazione del processo	8
3.1.1.2	Progettazione e sviluppo	8
3.1.1.3	Rilascio	8
3.1.2	Lista documenti	8
3.1.3	Ciclo di vita	9
3.1.4	Ambiente di lavoro	9
3.1.4.1	LaTeX _e	9
3.1.4.2	Docker _e	9
3.1.4.3	Google Docs	10



3.1.5	Struttura documenti	10
3.1.5.1	Verbali	10
3.1.6	Stile	12
3.1.6.1	Utilizzo del femminile	12
3.1.6.2	Formattazione testo	12
3.1.7	Strumenti	13
3.2	Gestione della configurazione	13
3.2.1	Scopo	13
3.2.2	Descrizione	13
3.2.3	Issue tracking system (ITS)	13
3.2.3.1	Ticket	14
3.2.4	Automazione chiusura ticket	15
3.2.5	Versionamento	16
3.2.6	Repository	16
3.2.6.1	Repository <i>Docs</i>	16
3.2.6.2	Repository <i>ChatSQL</i>	17
4	verifica	17
5	Validazione	17
6	Processi organizzativi	17
6.1	Gestione	17
6.1.1	Descrizione	17
6.1.1.1	Pianificazione	17
6.1.1.2	Esecuzione e controllo	17
6.1.1.3	Valutazione e approvazione	17
6.1.2	Ruoli	18
6.1.2.1	Responsabile	18
6.1.2.2	Amministratore	18
6.1.2.3	Analista	18
6.1.2.4	Progettista	18
6.1.2.5	Programmatore	18
6.1.2.6	Verificatore	18
6.1.3	Comunicazione	18
6.1.3.1	Comunicazione interna	18
6.1.3.2	Comunicazione esterna	18

1 Introduzione

1.1 Scopo del documento

Il presente documento ha lo scopo di delineare le *best practices*_e e il *way of working*_e che il gruppo Argo ha individuato e adotta durante tutto lo svolgimento del progetto didattico. Poiché il way of working è definito incrementalmente durante il corso del progetto, questo documento non è da considerarsi un testo definitivo o completo.

1.2 Scopo del prodotto

1.3 Glossario

Allo scopo di evitare incomprensioni relative al linguaggio utilizzato nella documentazione di progetto, viene fornito un *Glossario*, nel quale ciascun termine è corredato da una spiegazione che mira a disambiguare il suo significato. I termini tecnici, gli acronimi e i vocaboli ritenuti ambigui vengono formattati in corsivo all'interno dei rispettivi documenti e marcati con una lettera _e in pedice. Tutte le ricorrenze di un termine definito nel *Glossario* subiscono la formattazione sopracitata.

1.4 Riferimenti

1.4.1 Riferimenti normativi

- C9 ChatSQL: creare frasi SQL da linguaggio naturale (Zucchetti S.p.A.):
<https://www.math.unipd.it/~tullio/IS-1/2023/Progetto/C9.pdf>
<https://www.math.unipd.it/~tullio/IS-1/2023/Progetto/C9p.pdf>;
- Standard ISO/IEC 12207:1995:
https://www.math.unipd.it/~tullio/IS-1/2009/Approfondimenti/ISO_12207-1995.pdf;

1.4.2 Riferimenti informativi

TODO

2 Processi primari

2.1 Fornitura

2.1.1 Descrizione

Il *processo*_e di fornitura consiste nell'insieme di attività e compiti svolte dal *Fornitore*_e nel rapporto con la *Proponente*_e Zucchetti S.p.A.. Il processo parte dalla candidatura al *capitolato*_e d'appalto e prosegue con la determinazione di procedure e risorse richieste per la gestione e assicurazione del progetto, incluso lo sviluppo e l'esecuzione di un *Piano di Progetto*_e. L'obiettivo principale del processo è confrontare le aspettative della Proponente con i risultati del Fornitore durante il periodo del progetto,

mantenendo dunque una metrica oggettiva tra il preventivato e lo stato corrente. Il processo consiste nelle seguenti attività:

- Selezione e studio fattibilità;
- Candidatura;
- Pianificazione;
- Esecuzione e controllo;
- Revisione e valutazione;
- Consegna e completamento.

2.1.1.1 Selezione e studio fattibilità Il Fornitore esamina i capitolati d'appalto e arriva a una decisione sulla candidatura per uno di essi.

2.1.1.2 Candidatura Il Fornitore definisce e prepara una candidatura al capitolato d'appalto scelto producendo i seguenti documenti:

- **Lettera di Candidatura:** presentazione del gruppo rivolta al *Committente_e*;
- **Stima dei Costi e Assunzione Impegni:** documento che contiene un preventivo sulla distribuzione ore del progetto, il suo costo, una pianificazione generale e una iniziale analisi dei rischi;
- **Valutazione Capitolati:** documento che contiene l'analisi e la valutazione da parte del gruppo dei capitolati disponibili.

2.1.1.3 Pianificazione Il Fornitore stabilisce i requisiti per la gestione, lo svolgimento e la misurazione della qualità del progetto. In seguito, sviluppa e documenta attraverso il *Piano di Progetto_e* i risultati attesi.

2.1.1.4 Esecuzione e controllo Il Fornitore esegue il Piano di Progetto sviluppato, attenendosi alle norme definite nella sezione 2.2 e monitora la qualità del *prodotto software_e* nei seguenti modi:

- Controllo del progresso di *performance_e*, costi, rendicontazione dello stato del progetto e organizzazione;
- Identificazione, tracciamento, analisi e risoluzione dei problemi.

2.1.1.5 Revisione e valutazione Il Fornitore coordina la revisione interna ed esegue verifica e validazione secondo le norme definite in 4 e 5. Questo avviene in modo continuo e iterativo.

2.1.2 Rapporti con la Proponente

TODO

2.1.3 Documentazione_e fornita

Di seguito viene descritta la documentazione che il gruppo si impegna a rendere disponibile alla Proponente e ai Committenti.

2.1.3.1 Piano di Progetto TODO

2.1.3.2 Analisi dei Requisiti TODO

2.1.3.3 Piano di Qualifica TODO

2.1.3.4 Lettera di Presentazione TODO

2.1.3.5 Glossario Raccolta esaustiva di tutti i termini tecnici utilizzati nella documentazione. Permette di eliminare ambiguità e fraintendimenti fornendo una definizione univoca ed esaustiva per l'intero gruppo e per chi consulta la documentazione prodotta.

2.1.4 Strumenti

Gli strumenti impiegati nel processo di fornitura sono:

- **LaTeX:** *markup language_e* utilizzato per la redazione della documentazione;
- **Git:** *Version Control System_e* utilizzato dal gruppo;
- **Zoom:** Strumento per videochiamate utilizzato nei rapporti con la Proponente;
- **Google Sheets:** Strumento per la creazione di *spreadsheet_e* condivisi, utilizzato per la pianificazione di *sprint_e* e *rendicontazione ore_e*.

2.1.4.1 Dashboard Google Sheets

Lo *spreadsheet_e* condiviso, realizzato tramite *Google Sheets_e* e accessibile su *Google Drive_e*, ha lo scopo di automatizzare la generazione di preventivi e consuntivi (orari ed economici). Il foglio di calcolo principale è suddiviso in più fogli interni:

- Un foglio contenente le variabili globali da utilizzare nei fogli successivi, incluso il preventivo definito in fase di candidatura;
- Un foglio nascosto per ciascun periodo, che include le risposte al questionario di valutazione dello sprint;
- Un foglio per ogni sprint con le seguenti informazioni:
 - Numero dello *sprint_e*;
 - Date di inizio e termine dello *sprint_e*;
 - Tabella di assegnazione delle ore produttive per ciascun membro del team, accumulate in totali per persona e per ruolo;



- Distribuzione delle ore per ruolo, sotto forma di donut chart;
- Distribuzione delle ore per la coppia risorsa-ruolo, sotto forma di grafico a barre sovrapposte (così da poter assumere più ruoli per sprint);
- Preventivo economico dello *sprint_e*;
- Tabella riassuntiva con ore e budget spesi e restanti;
- Pie chart con la stima delle ore spese sul totale;
- Pie chart con la stima del budget sul totale;
- Ore rimanenti per la coppia risorsa-ruolo.

2.2 Sviluppo

2.2.1 Descrizione

Il *processo_e* di sviluppo contiene le attività e compiti dello *sviluppatore_e* sotto elencate:

- *Analisi dei requisiti_e*;
- *Progettazione_e*;
- *Codifica_e* e *testing_e*.

2.2.2 Analisi dei Requisiti

2.2.2.1 Descrizione L'Analisi dei Requisiti è eseguita dall'Analista, che redige l'omonimo documento *Analisi dei Requisiti*. Il documento considera i seguenti aspetti:

- TODO

TODO

2.2.3 Progettazione

2.2.3.1 Descrizione L'attività, svolta dal Progettista segue quella di analisi e ha il compito di impostare un'*architettura_e* del software capace di soddisfare i requisiti definiti. Il Progettista sviluppa l'architettura attraverso la creazione di unità e di relazioni tra loro, utilizzando opportunamente dei *design pattern_e* architettureali.

TODO

2.2.4 Codifica e testing

2.2.4.1 Descrizione La codifica segue l'attività di progettazione e viene svolta dal Programmatore. Ha lo scopo di trasformare l'architettura prodotta dal Progettista in codice rispettando le norme definite per ottenere codice mantenibile e di qualità. Il testing è una parte stessa dell'attività di codifica, necessaria ad assicurare la correttezza di ciascuna unità software.

TODO

3 Processi di supporto

3.1 Documentazione

3.1.1 Descrizione

Il processo di documentazione registra l'informazione generata da altri processi o attività. Il processo contiene l'insieme di attività che pianificano, producono, modificano, rilasciano e mantengono i documenti legati al progetto.

Il processo consiste nelle seguenti attività:

- Implementazione del processo;
- Progettazione e sviluppo;
- Rilascio.

3.1.1.1 Implementazione del processo Questa attività definisce quali documenti saranno generati durante il progetto, definendo per ciascuno:

- Titolo;
- Scopo;
- Descrizione;
- Responsabilità per contribuzione, redazione, verifica e approvazione;
- Pianificazione per versioni provvisorie e finali.

3.1.1.2 Progettazione e sviluppo Questa attività consiste nel progettare e redarre ciascun documento nel rispetto degli standard definiti per formato e contenuto, successivamente controllati dal Verificatore.

3.1.1.3 Rilascio Questa attività comincia con l'approvazione finale del documento da parte del Responsabile in carica, e della Proponente nel caso di verbali ad uso esterno. Prosegue con la pubblicazione del documento nel *repository*, apposito della documentazione.

3.1.2 Lista documenti

I documenti da produrre e mantenere durante il corso del progetto sono:

- *Piano di Progetto*;
- *Norme di Progetto*;
- *Piano di Qualifica*;
- *Analisi dei Requisiti*;

- *Manuale Utente*;
- *Glossario*;
- *Verbalì Interni*;
- *Verbalì Esterni*.

3.1.3 Ciclo di vita

Il ciclo di vita di un documento è composto dai seguenti eventi:

1. Vengono definite le caratteristiche di base del documento o di una sua parte come da sezione 3.1.1.1;
2. Il Redattore stila una bozza iniziale. Se è necessario l'input di più persone in maniera sincrona, tale bozza viene prodotto in un ambiente condiviso;
3. Prodotta una bozza di tutto il contenuto necessario, il Redattore produce una versione del documento con la forma e i metodi stabiliti in queste norme;
4. Viene sottoposto a verifica il risultato della redazione. Se il Verificatore propone delle modifiche, vengono attuate ritornando alla fase precedente;
5. In seguito a un esito positivo della verifica, se il risultato è un documento completo e che richiede rilascio, viene sottoposto ad un'approvazione finale del responsabile, bloccante in modo analogo alla verifica.

3.1.4 Ambiente di lavoro

3.1.4.1 LaTeX_e. Per lo sviluppo della documentazione del gruppo viene utilizzato un *template_e LaTeX_e* personalizzato. All'interno del template è definito lo stile della pagina iniziale, delle intestazioni e della formattazione generale. Parte del template permette l'uso di comandi personalizzati per favorire la consistenza di termini specifici spesso utilizzati (es.: nomi di documenti, nomi dei membri), inoltre è gestita sempre attraverso il template l'interazione con i termini per il Glossario.

L'utilizzo del template garantisce:

- Il disaccoppiamento di forma e contenuto della documentazione;
- L'uniformità dello stile della documentazione;
- La responsabilità del Redattore è il solo contenuto;
- La possibilità di creare documenti in maniera modulare, conciliata in modo uniforme.

3.1.4.2 Docker_e. La compilazione di file LaTeX può differire in base al compilatore utilizzato, il sistema operativo o altre caratteristiche del sistema locale. Per garantirne l'uniformità, la compilazione dei documenti viene effettuata all'interno di un container Docker costruito a partire da un'immagine comune.

3.1.4.3 Google Docs Per scrivere un documento è spesso necessario lavorare in maniera sincrona, Google Docs permette la condivisione e il lavoro contemporaneo di più persone. I limiti del software tuttavia non permettono di generare un documento finale adeguato, per cui le produzioni tramite questo mezzo sono da considerarsi bozza da cui eseguire la conversione.

3.1.5 Struttura documenti

Ciascun documento è fornito di questi elementi:

- Prima pagina:
 - Logo del gruppo;
 - Titolo;
 - Nome del gruppo;
 - Nome del progetto;
 - Versione attuale;
 - Approvatore;
 - Uso del documento (Interno/Esterno);
 - Destinatari del documento;
 - Logo dell'Università di Padova.
- Registro delle modifiche:
 - Versione del documento in seguito alla modifica;
 - Data della modifica;
 - Redattore della modifica (coincide con il Verificatore nel caso di riga associata alla verifica generale, col responsabile del caso di riga associata al rilascio);
 - Verificatore della modifica (coincide con il Responsabile nel caso di riga associata al rilascio);
 - Descrizione della modifica.
- Indice dei contenuti;

3.1.5.1 Verbali

Ad eccezione dei capitoli dichiarati in precedenza, i verbali presentano una struttura differente rispetto a quella degli altri documenti di progetto. Il contenuto dei verbali, sia interni che esterni, è infatti suddiviso nelle seguenti sezioni:

1. Informazioni:

- Orario di inizio dell'incontro;
- Orario di fine dell'incontro;



- Mezzo di pianificazione dell'incontro (Mail, Telegram, riunioni precedenti, ecc.);
 - Tipo di incontro (in presenza/da remoto);
 - Descrizione dell'incontro;
 - **Partecipanti:** sezione che include l'elenco dei partecipanti interni e, in caso di riunione con la *Proponente_e*, anche esterni. Per ciascun membro del team, si riporta la durata (in ore e minuti) della partecipazione;
 - **Glossario:** paragrafo finalizzato a stabilire la modalità di formattazione dei termini definiti nel *Glossario* e il numero di occorrenze identificate.
2. **Riunione:** i meeting vengono organizzati con lo scopo di rendicontare il lavoro svolto da ciascun membro del gruppo, chiarire eventuali dubbi, mitigare i rischi, intraprendere azioni correttive e pianificare le attività future. Il capitolo relativo alla riunione è suddiviso in due sezioni:
- **Ordine del giorno:** scaletta degli argomenti generali affrontati durante la riunione;
 - **Discussione e decisioni:** sezione contenente l'elenco cronologico degli argomenti trattati nel corso del meeting. La discussione di ciascuna tematica non è mai fine a sé stessa, ma mira a prendere decisioni consapevoli e a definire un piano d'azione (vedere tabella Todo / In Progress). Durante le riunioni, si valuta anche il progresso delle attività assegnate negli incontri precedenti. Il team può quindi decidere di considerare un task completato, di prolungare la sua data di scadenza o, se necessario, di suddividere l'attività in sotto-task. Nei verbali esterni, alcune sezioni sono organizzate secondo lo schema "domanda-risposta", per formalizzare l'interazione tra il team e la *Proponente_e*.
3. **Tabella di task Todo / In Progress:** Durante le riunioni, interne ed esterne, il team pianifica le attività a breve e medio-lungo termine. Al fine di ottimizzare la fase di creazione dei *ticket_e* su *Jira Software_e*, viene redatta una tabella con le azioni da intraprendere o, in alternativa, i task da portare a termine. Ogni voce è affiancata dal codice univoco del *ticket_e* correlato (se presente) nell'*ITS_e* di *Jira_e*. L'ID del *ticket_e* è composto dalla stringa ARGO- seguita da un numero univoco autoincrementante. I campi della tabella sono i seguenti:
- ID del *ticket_e* *Jira_e* associato all'incarico;
 - Descrizione dell'attività;
 - Nome del componente o dei componenti a cui è assegnato il task;
 - Data di scadenza, in formato AAAA-MM-GG per mantenere la coerenza con il *repository_e* documentale e il registro delle modifiche.
4. **Prossima riunione:** sezione contenente la data ed, eventualmente, l'orario della prossima riunione (se pianificata), con annessa breve descrizione dell'ordine del giorno. Nel caso in cui un meeting sia stato organizzato durante l'incontro precedente (e non tramite chat Telegram), il verbale interno deve includere un link al verbale appropriato come mezzo di pianificazione;

5. **Firma del documento:** spazio per la firma del responsabile. In caso di verbale esterno, l'approvazione finale è a carico della *Proponente*_e, che produce in output un documento, in formato PDF, firmato e timbrato.

3.1.6 Stile

Di seguito sono elencate le convenzioni stilistiche adottate dalla documentazione del gruppo.

3.1.6.1 Utilizzo del femminile Quando è necessario fare riferimento tramite ruolo di progetto ad un membro del gruppo con il genere femminile, si utilizzano i seguenti termini:

- **Responsabile** è invariato;
- **Amministratrice** al posto di Amministratore;
- **Analista** è invariato;
- **Progettista** è invariato;
- **Programmatrice** al posto di Programmatore;
- **Redattrice** al posto di Redattore;
- **Verificatrice** al posto di Verificatore.

3.1.6.2 Formattazione testo

- **Termini nel Glossario:** indicati in *corsivo* e con una lettera _e in pedice alla fine della parola. In base a ciascun documento tale formattazione può comparire alla sola prima occorrenza (quando il documento ha lo scopo di essere letto dall'inizio alla fine), o in maniera più frequente (quando il documento può essere letto in maniera più frammentata);
- **Nomi di documento:** scritti in *corsivo* con le iniziali di parola maiuscole eccetto preposizioni (es.: *Piano di Progetto*, non *Piano Di Progetto*). Questo per mantenere la coerenza con le *abbreviazioni*_e (es: AdR, PdP, PdQ, NdP);
- **Way of Working:** indicato con le iniziali di parola maiuscole eccetto preposizioni, per mantenere la coerenza con l'abbreviazione WoW usata anche come comando in *LaTeX*_e;
- **Nomi di ruolo:** scritti con la lettera iniziale minuscola; anche vocaboli come team, gruppo e fornitore seguono la medesima regola. L'unica eccezione è rappresentata dai seguenti termini:
 - **Proponente:** declinato al femminile e indicato sempre con la lettera iniziale maiuscola, per garantire la massima formalità possibile;
 - **Cliente e Committente:** scritti con la lettera iniziale maiuscola quando si desidera instaurare un rapporto formale con un attore esterno, altrimenti mantengono l'iniziale minuscola. Per esempio, nella frase "il ruolo di cliente

è rivestito dall'azienda Zucchetti S.p.A.", la parola "cliente" non richiede la lettera maiuscola.

- **Data:** indicata in formato YYYY-MM-DD nelle tabelle riassuntive, nei titoli e nei nomi dei file. Il formato esteso (esempio: 20 aprile 2024) si utilizza quando la data rientra in un testo discorsivo.

3.1.7 Strumenti

Gli strumenti impiegati nel processo di documentazione sono:

- **Git:** *Version Control System*_e utilizzato dal gruppo;
- **GitHub:** Piattaforma ospite del repository del gruppo;
- **LaTeX:** *markup language*_e per la scrittura di documenti;
- **Docker:** Software per *containerizzazione*_e utilizzato dal gruppo per uniformare la generazione di documenti;
- **Google Docs:** Strumento per la creazione di documenti condivisi, utilizzato per la collaborazione nella redazione di un documento.

3.2 Gestione della configurazione

3.2.1 Scopo

La seguente sezione viene redatta con lo scopo di formalizzare e automatizzare le procedure applicate dal team, durante tutto il ciclo di vita del software, nell'ambito del processo di "configuration management".

3.2.2 Descrizione

Il processo di gestione della configurazione si occupa di definire e gestire le componenti software utilizzate durante l'intero corso del progetto per mantenere la tracciabilità e gestire il versionamento e i rilasci di prodotti software e documenti. Si tratta di un processo di applicazione di procedure amministrative e tecniche finalizzate a:

- Identificare le componenti software del sistema e stabilire un punto di riferimento da cui misurare i progressi nel tempo;
- Controllare le modifiche e i rilasci degli item;
- Mantenere la tracciabilità delle modifiche;
- Garantire la completezza, coerenza e correttezza degli item.

3.2.3 Issue tracking system (ITS)

Al fine di registrare, gestire e monitorare le attività e sottoattività lungo l'intero ciclo di vita del software, il team impiega l'*Issue Tracking System*_e sviluppato da Atlassian: *Jira*_e.

3.2.3.1 Ticket

I *ticket*_e possono essere di quattro tipi:

- **Task**: un'attività o un compito specifico da portare a termine entro la fine di uno *sprint*_e e assegnato a un unico membro del team;
- **Sottotask**: un *ticket*_e subordinato che consente di orientarsi verso una scomposizione più granulare del lavoro. Un'attività, ritenuta troppo onerosa per una singola risorsa, può quindi essere suddivisa in più sottotask, associabili a diversi componenti del gruppo;
- **Bug**_e: un *ticket*_e marcato come *bug*_e segnala la presenza di un'anomalia da risolvere tempestivamente, relativamente al prodotto software, alla documentazione o all'infrastruttura di gestione del progetto;
- **Story**_e: detta anche "User Story", rappresenta una funzionalità del sistema, un requisito espresso dal punto di vista dell'utente.

Il layout di un *ticket*_e è formato dai seguenti campi:

- **Riepilogo**: un titolo che riassume brevemente l'incarico associato al *ticket*_e;
- **ID**: un codice univoco autoincrementante generato automaticamente dal sistema secondo la formula ARGO-ID;
- **Descrizione**: una descrizione esaustiva dei risultati attesi al completamento del *ticket*_e;
- **Assegnatario**: il componente del gruppo a cui è stata assegnato il compito di risolvere il *ticket*_e;
- **Epic**: esprime obiettivi generali o grandi porzioni di lavoro che devono essere frammentati. Ogni *ticket*_e può essere associato a un epic;
- **Sprint**_e: ciascun *ticket*_e può essere correlato a uno *sprint*_e, a sua volta scomposto in più epic;
- **Ticket collegati**: *Jira*_e offre una funzionalità, sia nei campi di contesto che nella timeline di pianificazione, per collegare i *ticket*_e tra loro. Di seguito sono riportate le associazioni predefinite:
 - blocca/è bloccato da (queste sono le due dipendenze più comuni tra i task);
 - clona/è clonato da;
 - duplica/è duplicato da;
 - item correlato a.
- **Sviluppo**: un campo di integrazione tra *Jira*_e e *GitHub*_e che permette di monitorare lo stato di avanzamento dello sviluppo, con annessi link ai *branch*_e, *commit*_e, *pull request*_e, *build*_e e *distribuzioni*_e associati al *ticket*_e;
- **Stato**: durante il suo ciclo di vita, un *ticket*_e attraversa tre stati: "To Do", "Doing" e "Done".

- **Versioni di correzione:** le versioni rappresentano punti temporali e traguardi intermedi nello svolgimento del progetto. Ciascun *ticket_e* può essere associato a una determinata versione. L'associazione *ticket_e*/versione può essere realizzata direttamente dal *backlog_e* del progetto. Le versioni possono trovarsi in uno dei seguenti tre stati:
 - Non rilasciate;
 - Rilasciate;
 - Archivate.
- **Commenti:** elenco di commenti da affiancare ai messaggi di *commit_e* per contestualizzare le modifiche e ottimizzare il lavoro di *verifica_e*;
- **Aggiungi allegato:** oltre ai commenti, è possibile allegare file di vario formato che non necessitano del controllo di versione;
- **Aggiungi un ticket figlio:** ogni *ticket_e* può avere uno o più *ticket_e* subordinati;
- **Azioni:** *Jira_e* offre la possibilità di creare, gestire e monitorare automazioni, come ad esempio la chiusura di un *ticket_e* una volta effettuato il *merge_e* di una *pull request_e*;
- **Rilasci:** elenco delle versioni rilasciate a cui il *ticket_e* è associato.

3.2.4 Automazione chiusura ticket

Su *Jira_e*, nelle impostazioni del progetto, è presente una sezione denominata "Automazione", a sua volta suddivisa in quattro sottosezioni:

- **Regole:** elenco delle regole definite dall'amministratore; ciascuna regola richiede un trigger di innesco, ossia un evento che avvia l'esecuzione della procedura definita nel corpo della regola. Una volta stabilito il trigger di attivazione, l'amministratore può scegliere una delle seguenti opzioni:
 - THEN: aggiungi un'azione (es: transizione di un *ticket_e* da uno stato all'altro);
 - IF: aggiungi una condizione (es: verifica se lo stato del *ticket_e* è diverso da "Completato");
 - FOR EACH: applica le azioni e le condizioni ad ogni task (es: esamina tutti i *ticket_e* collegati al *ticket_e* che ha attivato la regola ed esegue per ciascuno di essi una determinata azione);
- **Audit log:** cronologia delle automazioni avviate, con dettagli sullo stato di esecuzione della regola, la data di attivazione e gli elementi associati;
- **Modelli:** set di regole predefinite da importare nel progetto;
- **Utilizzo:** numero di automazioni attivate mensilmente.

Il team ha deciso di introdurre una regola personalizzata per effettuare automaticamente la transizione dello stato dei *ticket_e*. Quando viene aperta una *pull request_e* finalizzata alla chiusura di un *ticket_e*, il titolo della richiesta deve essere corredato

dal codice identificativo del *ticket*_e (ARGO-ID). In alternativa, è possibile menzionare il *ticket*_e nel nome del *branch*_e o nei *commit*_e associati alla *pull request*_e. Inoltre, l'assegnatario può lasciare un commento nella forma [ARGO-ID], affinché un bot, denominato *jira*[bot], possa convertire il commento in un link al *ticket*_e *Jira*_e. Una volta effettuato il *merge*_e della *pull request*_e su *GitHub*_e, il *ticket*_e collegato passerà in automatico allo stato "Completato".

Utilizzando i modelli predefiniti, il gruppo ha aggiunto altre due regole, rispettivamente per:

- Chiudere automaticamente un *ticket*_e quando tutti i *ticket*_e subordinati (task, story, bug, sottotask) sono completati;
- Chiudere automaticamente un *ticket*_e quando tutti i sottotask sono completati.

3.2.5 Versionamento

Il gruppo mantiene un versionamento per la documentazione nel formato:

X.Y.Z

X Avanza alla approvazione del Responsabile, corrisponde per cui ad ogni rilascio;

Y Avanza ad ogni verifica completa del documento;

Z Avanza ad ogni modifica verificata di un documento.

3.2.6 Repository

Il gruppo utilizza due repository, disponibili in *GitHub*_e:

- Repository della documentazione: <https://github.com/argo-swe/docs>
- Repository del codice sorgente: <https://github.com/argo-swe/chatsql>

Il gruppo utilizza inoltre, per hosting del sito argo-swe.github.io, un repository, da non considerare all'interno del workflow in quanto aggiornata e mantenuta solo come "vetrina" del gruppo.

- Repository del sito github.io: <https://github.com/argo-swe/argo-swe.github.io>

3.2.6.1 Repository Docs Il repository contiene il codice sorgente in LaTeX di tutta la documentazione ufficiale generata durante il progetto, oltre all'ambiente utile alla generazione dei file PDF corrispondenti.

Il repository include un file *README.md* che illustra brevemente lo scopo del repository e i componenti del gruppo, un file *.gitignore* per escludere il tracciamento di file ausiliari o artefatti di compilazione.

La directory *Logo* contiene le versioni ufficiali del logo del gruppo, in formato SVG o PNG.

La directory *sources* contiene il codice sorgente per la documentazione, separato in due directory: *model* contiene i file di utilizzo globale all'interno della documentazione, *documents* contiene, in maniera ordinata per fasi di progetto, la documentazione ufficiale.

La *directory tools* contiene gli strumenti *Docker*₆ per utilizzare un ambiente unico nella compilazione e uno script per compilare automaticamente uno o più documenti. Il repository contiene un ramo base, in cui vengono inserite le versioni verificate dei

documenti caricate nel repository attraverso *feature branch*₆ su cui viene eseguita la verifica prima di eseguire *merge*₆.

3.2.6.2 Repository ChatSQL TODO

4 verifica

TODO

5 Validazione

TODO

6 Processi organizzativi

6.1 Gestione

6.1.1 Descrizione

Il processo di gestione contiene le attività e i task che vengono adottati dal Responsabile per il coordinamento del processo.

Il processo consiste nelle seguenti attività:

- Pianificazione;
- Esecuzione e controllo;
- Valutazione e approvazione;

6.1.1.1 Pianificazione Questa attività comprende tutta la programmazione di assegnazione ruoli e attività, scadenze e previsione del periodo corrente e dei successivi.

6.1.1.2 Esecuzione e controllo Il Responsabile provvede a far eseguire e mantenere il risultato della pianificazione, analizzando e risolvendo i problemi sorti durante l'avanzamento. Problemi e soluzioni saranno documentate.

Il Responsabile inoltre si occupa di comunicare con gli *stakeholder*₆.

6.1.1.3 Valutazione e approvazione Il Responsabile assicura la soddisfazione dei requisiti del software o la completezza e correttezza della documentazione durante e alla fine dell'esecuzione dei rispettivi processi.

6.1.2 Ruoli

Questo progetto didattico prevede l'assegnazione dei seguenti ruoli, con una rotazione costante e bilanciata che va considerato nella pianificazione.

6.1.2.1 Responsabile TODO

6.1.2.2 Amministratore TODO

6.1.2.3 Analista TODO

6.1.2.4 Progettista TODO

6.1.2.5 Programmatore TODO

6.1.2.6 Verificatore TODO

6.1.3 Comunicazione

6.1.3.1 Comunicazione interna La comunicazione tra i membri del gruppo è gestita attraverso Telegram e Discord.

Attraverso Telegram il gruppo comunica in modo asincrono e generale, pertanto è opportuno per comunicazioni di interesse di tutto il gruppo e di breve contenuto.

Attraverso Discord il gruppo partecipa a chiamate di gruppo, riunioni o meno, e tramite canali testuali divisi per ruolo è l'organizzazione interna di gruppi ristretti è favorita.

Questi strumenti non devono sovrapporsi tuttavia a mezzi di comunicazione e coordinamento, come ad esempio un *Issue Tracking System*, in quanto le informazioni riportate tramite questi strumenti sono più difficilmente tracciabili e riferibili in momenti futuri.

6.1.3.2 Comunicazione esterna La comunicazione esterna è gestita dal Responsabile, attraverso il recapito di posta elettronica del gruppo argo.unipd@gmail.com.