



Università degli Studi di Padova

Laurea: Informatica

Corso: Ingegneria del Software

Anno Accademico: 2024/2025



Gruppo: SWEg Labs

Email: gruppo.sweg@gmail.com

Piano di Qualifica

Versione 2.0.0

Stato	Approvato
Redazione	Federica Bolognini Michael Fantinato Giacomo Loat Filippo Righetto Riccardo Stefani Davide Verzotto
Verifica	Federica Bolognini Michael Fantinato Giacomo Loat Filippo Righetto Riccardo Stefani Davide Verzotto
Proprietario	Michael Fantinato
Uso	Esterno
Destinatari	Prof. Tullio Vardanega Prof. Riccardo Cardin <i>AzzurroDigitale Srl</i>

Registro delle modifiche

Versione	Data	Descrizione	Autore	Verificatore
2.0.0	01-04-25	Corretti alcuni errori di battitura	Giacomo Loat	Michael Fantinato
1.2.0	30-03-25	Corretta la didascalia della tabella 3	Davide Verzotto	Giacomo Loat
1.1.5	28-03-25	Aggiornamento della sezione §4.10 con i valori aggiornati dell'Indice di Gulpease	Giacomo Loat	Riccardo Stefani
1.1.4	22-03-25	Aggiornamento della sezione §4 con i nuovi diagrammi della Product Baseline	Riccardo Stefani	Giacomo Loat
1.1.3	18-03-25	Sistemazione della sezione §3.3	Filippo Righetto	Riccardo Stefani
1.1.2	17-03-25	Aggiunti dei test di unità alla sezione §3.1 e test di integrazione alla sezione §3.2	Filippo Righetto	Riccardo Stefani
1.1.1	15-03-25	Iniziata la scrittura della sezione §3.1	Filippo Righetto	Riccardo Stefani
1.1.0	11-03-25	Inserita la didascalia alla tabella dei test di integrazione, nella sezione §3.2	Davide Verzotto	Riccardo Stefani
1.0.5	04-03-25	Iniziata la scrittura della sezione §3.2	Filippo Righetto	Riccardo Stefani
1.0.4	25-02-25	Sistemata la sezione §2	Federica Bolognini	Riccardo Stefani
1.0.3	13-02-25	Corretto errore per cui le figure del paragrafo §4.10 non portavano didascalia nell'elenco delle figure	Giacomo Loat	Federica Bolognini
1.0.2	13-02-25	Eliminata sezione <i>Valutazioni per il miglioramento</i>	Giacomo Loat	Davide Verzotto
1.0.1	11-02-25	Sistemati i link presenti nella sezione §1.4 seguendo i consigli del professor Vardanega	Riccardo Stefani	Giacomo Loat
1.0.0	23-01-25	Approvazione del documento	Michael Fantinato	Michael Fantinato
0.3.0	23-01-25	Verifica del documento	Davide Verzotto	Davide Verzotto
0.2.7	17-01-25	Stesura paragrafo §4.10	Giacomo Loat	Riccardo Stefani
0.2.6	17-01-25	Stesura di gran parte della sezione §4	Riccardo Stefani	Giacomo Loat
0.2.5	17-01-25	Stesura sezione <i>Valutazioni per il miglioramento [sezione poi rimossa]</i> e stesura paragrafo §4.9	Giacomo Loat	Riccardo Stefani
0.2.4	15-01-25	Stesura sezione §3.3	Davide Verzotto	Riccardo Stefani
0.2.3	14-01-25	Stesura sezione §3.5	Federica Bolognini	Davide Verzotto
0.2.2	12-01-25	Stesura sezione §3.4	Filippo Righetto	Davide Verzotto
0.2.1	31-12-24	Continuata stesura sezione §2.2	Davide Verzotto	Riccardo Stefani
0.2.0	06-12-24	Verifica del documento allo stato attuale	Davide Verzotto	Davide Verzotto

Versione	Data	Descrizione	Autore	Verificatore
0.1.3	02-12-24	Continuata stesura sezione § <u>2</u>	Michael Fantinato	Riccardo Stefani
0.1.2	01-12-24	Inizio stesura sezione § <u>4</u>	Riccardo Stefani	Giacomo Loat
0.1.1	09-11-24	Inizio stesura sezioni § <u>1</u> , § <u>2</u> e § <u>3</u>	Michael Fantinato	Federica Bolognini
0.1.0	05-11-24	Creazione del documento	Riccardo Stefani	Davide Verzotto

Tabella 1: Registro delle modifiche

Indice

1	Introduzione	1
1.1	Scopo del documento	1
1.2	Glossario	1
1.3	Maturità e miglioramenti	1
1.4	Riferimenti	1
1.4.1	Riferimenti normativi	1
1.4.2	Riferimenti informativi	1
2	Metriche di qualità	3
2.1	Qualità di processo	3
2.2	Qualità di prodotto	3
3	Strategie di testing	4
3.1	Test di unità	4
3.2	Test di integrazione	14
3.3	Test di sistema	18
3.4	Test di accettazione	22
3.5	Checklist	24
3.5.1	Struttura della documentazione	24
3.5.2	Errori ortografici	24
3.5.3	Non conformità con le Norme di Progetto	24
3.5.4	Analisi dei Requisiti	25
4	Cruscotto di valutazione della qualità	26
4.1	Varianza dell'impegno orario	26
4.2	Varianza di budget	27
4.3	Estimate to Complete ed Estimate at Completion	28
4.4	Planned Value, Earned Value e Actual Cost	29
4.5	Schedule Variance e Cost Variance	30
4.6	Schedule Performance Index e Cost Performance Index	31
4.7	Misure di mitigazione insufficienti	32
4.8	Rischi inattesi	33
4.9	Requisiti soddisfatti	34
4.10	Indice di Gulpease	35
4.11	Code Coverage	38

Elenco delle figure

1	Varianza dell'impegno orario per sprint	26
2	Varianza di budget per sprint	27
3	Progressione Estimate to Complete e Estimate at Completion in relazione al Budget at Completion	28
4	Progressione Planned Value, Earned Value e Actual Cost	29
5	Progressione Schedule Variance e Cost Variance	30
6	Progressione Schedule Performance Index e Cost Performance Index	31
7	Progressione occorrenza di rischi con misure mitigative insufficienti	32
8	Progressione occorrenza di rischi inattesi	33
9	Requisiti obbligatori, desiderabili e opzionali soddisfatti	34
10	Andamento indice di Gulpease nei verbali - RTB	35
11	Andamento indice di Gulpease nei verbali - PB	35
12	Indice di Gulpease Norme di Progetto	36
13	Indice di Gulpease Piano di Progetto	36
14	Indice di Gulpease Analisi dei Requisiti	36
15	Indice di Gulpease Piano di Qualifica	36
16	Indice di Gulpease Manuale Utente	36
17	Indice di Gulpease Specifica Tecnica	36
18	Code Coverage	38

Elenco delle tabelle

1	Registro delle modifiche	ii
2	Metriche di qualità di processo	3
3	Metriche di qualità di prodotto	3
4	Insieme dei test di unità	13
5	Insieme dei test di integrazione	17
6	Insieme dei test di sistema	21
7	Insieme dei test di accettazione	23
8	Struttura documentazione	24
9	Errori ortografici	24
10	Non conformità con Norme di Progetto	25
11	Aspetti di qualità per i casi d'uso	25

1 Introduzione

1.1 Scopo del documento

Per i progetti di sviluppo software che puntano a soddisfare gli elevati standard qualitativi delineati nei principi dell'ingegneria del software, un documento dettagliato del piano di qualità è indispensabile. Comprendere e valutare la *qualità_G* del prodotto sono concetti fondamentali per effettuare confronti e determinare quanto un prodotto aderisce alle aspettative. Questo documento mira a fornire una descrizione completa e precisa delle *metriche_G* e delle metodologie utilizzate per il controllo e la misurazione della qualità nelle diverse componenti del software. Definiremo gli obiettivi di qualità, i processi e le risorse necessarie per raggiungerli, oltre a specificare i test previsti con la relativa documentazione, comprese metodologie ed esiti. Il documento sarà una risorsa preziosa per chi sviluppa il prodotto, per gli utenti finali e per coloro che ne valuteranno la qualità.

1.2 Glossario

Al fine di evitare eventuali equivoci o incomprensioni riguardo la terminologia utilizzata all'interno di questo documento, abbiamo valutato di adottare un Glossario, con file apposito, in cui vengono riportate tutte le definizioni delle parole ambigue utilizzate nell'ambito di questo progetto.

Nel documento appena descritto verranno riportati tutti i termini definiti nel loro ambiente di utilizzo con annessa descrizione del loro significato.

La presenza di un termine all'interno del Glossario sarà indicata con una "G" posizionata al pedice della parola.

1.3 Maturità e miglioramenti

Questo documento è stato redatto seguendo un approccio incrementale, con l'obiettivo di facilitare l'aggiornamento alle esigenze mutevoli, stabilite di comune accordo tra i membri del gruppo di progetto e l'azienda proponente.

Pertanto, il documento non può essere considerato definitivo o esaustivo, ma piuttosto un punto di partenza per un continuo aggiornamento e affinamento.

1.4 Riferimenti

1.4.1 Riferimenti normativi

- Norme di Progetto v.2.0.0;
- Capitolato d'appalto C9 - BuddyBot (slide 3-18):
<https://www.math.unipd.it/tullio/IS-1/2024/Progetto/C9.pdf> (Ultimo accesso: 03/04/2025);
- PD1 - Regolamento del Progetto Didattico (slide 2-25):
<https://www.math.unipd.it/tullio/IS-1/2024/Dispense/PD1.pdf> (Ultimo accesso: 03/04/2025);
- Standard ISO/IEC 31000:2018 (sezioni 1-6):
<https://www.iso.org/obp/ui#iso:std:iso:31000:ed-2:v1:en> (Ultimo accesso: 03/04/2025).

1.4.2 Riferimenti informativi

- Glossario v.2.0.0;
- T7 - Qualità del software (slide 2-30):
<https://www.math.unipd.it/tullio/IS-1/2024/Dispense/T07.pdf> (Ultimo accesso: 03/04/2025);

- **T8 - Qualità di processo (slide 2-22):**
<https://www.math.unipd.it/tullio/IS-1/2024/Dispense/T08.pdf> (*Ultimo accesso: 03/04/2025*);
- **T9 - Verifica e validazione: introduzione (slide 2-25):**
<https://www.math.unipd.it/tullio/IS-1/2024/Dispense/T09.pdf> (*Ultimo accesso: 03/04/2025*);
- **T10 - Verifica e validazione: analisti statica (slide 2-26):**
<https://www.math.unipd.it/tullio/IS-1/2024/Dispense/T10.pdf> (*Ultimo accesso: 03/04/2025*);
- **T11 - Verifica e validazione: analisti dinamica (slide 2-36):**
<https://www.math.unipd.it/tullio/IS-1/2024/Dispense/T11.pdf> (*Ultimo accesso: 03/04/2025*).

2 Metriche di qualità

2.1 Qualità di processo

Codice	Nome	Accettabile	Preferibile
MPC-1	Varianza di Budget	$\pm 5\%$	$\pm 0\%$
MPC-2	Varianza dell'impegno orario	$\pm 10\%$	$\pm 0\%$
MPC-3	Earned Value	\geq MPC-4	-
MPC-4	Actual Cost	-	-
MPC-5	Planned Value	-	-
MPC-6	Cost Variance	± 150	0
MPC-7	Schedule Variance	± 150	0
MPC-8	Cost Performance Index	1 ± 0.15	1
MPC-9	Schedule Performance Index	1 ± 0.15	1
MPC-10	Estimate to Complete	-	-
MPC-11	Estimate at Completion	$\pm 5\%$ di MPC-12	MPC-12
MPC-12	Budget at Completion	-	-
MPC-13	Code Coverage	75%	100%
MPC-14	Misure di mitigazione insufficienti	3	0
MPC-15	Rischi inattesi	3	0

Tabella 2: Metriche di qualità di processo

2.2 Qualità di prodotto

Codice	Nome metrica	Valore accettabile	Valore preferibile
MPD-1	Indice di Gulpease	≥ 40	60
MPD-2	Errori ortografici	5	0
MPD-3	Requisiti obbligatori soddisfatti	100%	100%
MPD-4	Requisiti desiderabili soddisfatti	0%	100%
MPD-5	Requisiti opzionali soddisfatti	0%	100%

Tabella 3: Metriche di qualità di prodotto

3 Strategie di testing

3.1 Test di unità

I test di unità vengono utilizzati per controllare il funzionamento corretto di singoli componenti del codice in modo indipendente. Con "unità" si intendono parti del software, come funzioni, metodi o classi, che eseguono compiti ben definiti all'interno del programma.

Codice	Descrizione	Stato
TU1	Verifica che il metodo <code>get_answer</code> di <code>ChatController</code> , in caso di messaggio vuoto, restituisca un messaggio di errore.	✓
TU2	Verifica che il metodo <code>get_answer</code> di <code>ChatController</code> gestisca correttamente le eccezioni.	✓
TU3	Verifica che il metodo <code>get_answer</code> di <code>ChatService</code> restituisca la risposta corretta.	✓
TU4	Verifica che il metodo <code>get_answer</code> di <code>ChatService</code> gestisca correttamente le eccezioni.	✓
TU5	Verifica che il metodo <code>similarity_search</code> di <code>ChatService</code> gestisca correttamente le eccezioni.	✓
TU6	Verifica che il metodo <code>generate_answer</code> di <code>ChatService</code> gestisca correttamente le eccezioni.	✓
TU7	Verifica che il metodo <code>load</code> di <code>ChromaVectorStoreAdapter</code> gestisca correttamente le eccezioni.	✓
TU8	Verifica che il metodo <code>split</code> di <code>ChromaVectorStoreAdapter</code> sollevi un'eccezione <code>ValueError</code> se un documento non ha un campo <code>'id'</code> nei metadati.	✓
TU9	Verifica che il metodo <code>split</code> di <code>ChromaVectorStoreAdapter</code> inserisca correttamente la data di inserimento nel database vettoriale.	✓
TU10	Verifica che il metodo <code>split</code> di <code>ChromaVectorStoreAdapter</code> gestisca correttamente i documenti con id duplicato.	✓
TU11	Verifica che il metodo <code>split</code> di <code>ChromaVectorStoreAdapter</code> gestisca correttamente la conversione delle liste di file in stringhe.	✓
TU12	Verifica che il metodo <code>split</code> di <code>ChromaVectorStoreAdapter</code> gestisca correttamente la formattazione delle date.	✓
TU13	Verifica che il metodo <code>similarity_search</code> di <code>ChromaVectorStoreAdapter</code> gestisca correttamente le eccezioni.	✓
TU14	Verifica che il metodo <code>similarity_search</code> di <code>ChromaVectorStoreAdapter</code> salti i documenti nulli.	✓
TU15	Verifica che il metodo <code>load</code> di <code>ChromaVectorStoreRepository</code> carichi correttamente i documenti nel database vettoriale.	✓
TU16	Verifica che il metodo <code>load</code> di <code>ChromaVectorStoreRepository</code> gestisca correttamente le eccezioni.	✓
TU17	Verifica che il metodo <code>similarity_search</code> di <code>ChromaVectorStoreRepository</code> restituisca correttamente i risultati della ricerca di similarità.	✓
TU18	Verifica che il metodo <code>similarity_search</code> di <code>ChromaVectorStoreRepository</code> gestisca correttamente le eccezioni.	✓

Codice	Descrizione	Stato
TU19	Verifica che il metodo <code>load_confluence_pages</code> di <code>ConfluenceAdapter</code> gestisca correttamente le eccezioni.	✓
TU20	Verifica che il metodo <code>clean_confluence_pages</code> di <code>ConfluenceCleanerService</code> pulisca correttamente le pagine Confluence.	✓
TU21	Verifica che il metodo <code>clean_confluence_pages</code> di <code>ConfluenceCleanerService</code> gestisca correttamente le eccezioni.	✓
TU22	Verifica che il metodo <code>remove_html_tags</code> di <code>ConfluenceCleanerService</code> rimuova correttamente i tag HTML.	✓
TU23	Verifica che il metodo <code>remove_html_tags</code> di <code>ConfluenceCleanerService</code> gestisca correttamente le eccezioni.	✓
TU24	Verifica che il metodo <code>replace_html_entities</code> di <code>ConfluenceCleanerService</code> sostituisca correttamente le entità HTML.	✓
TU25	Verifica che il metodo <code>replace_html_entities</code> di <code>ConfluenceCleanerService</code> gestisca correttamente le eccezioni.	✓
TU26	Verifica che il metodo <code>get_base_url</code> di <code>ConfluenceRepository</code> ritorni l'URL base corretto.	✓
TU27	Verifica che il metodo <code>load_confluence_pages</code> di <code>ConfluenceRepository</code> ritorni le pagine correttamente.	✓
TU28	Verifica che il metodo <code>load_confluence_pages</code> di <code>ConfluenceRepository</code> gestisca correttamente le eccezioni di richiesta.	✓
TU29	Verifica che il metodo <code>load_confluence_pages</code> di <code>ConfluenceRepository</code> gestisca correttamente le eccezioni generiche.	✓
TU30	Verifica che il metodo <code>generate_answer</code> di <code>GenerateAnswerService</code> gestisca correttamente le eccezioni.	✓
TU31	Verifica che il metodo <code>get_last_load_outcome</code> di <code>GetLastLoadOutcomeController</code> gestisca correttamente le eccezioni.	✓
TU32	Verifica che il metodo <code>get_last_load_outcome</code> di <code>GetLastLoadOutcomeService</code> gestisca correttamente le eccezioni.	✓
TU33	Verifica che il metodo <code>get_messages</code> di <code>GetMessagesController</code> gestisca correttamente le eccezioni.	✓
TU34	Verifica che il metodo <code>get_messages</code> di <code>GetMessagesController</code> restituisca una lista vuota se il metodo <code>get_messages</code> di <code>GetMessagesUseCase</code> restituisce una lista vuota.	✓
TU35	Verifica che il metodo <code>get_messages</code> di <code>GetMessagesService</code> gestisca correttamente le eccezioni.	✓
TU36	Verifica che il metodo <code>get_next_possible_questions</code> di <code>GetNextPossibleQuestionsController</code> sollevi un'eccezione se il dizionario ricevuto in input non contiene le chiavi richieste.	✓
TU37	Verifica che il metodo <code>get_next_possible_questions</code> di <code>GetNextPossibleQuestionsController</code> sollevi un'eccezione se la chiave 'question' del dizionario ricevuto in input non è una stringa.	✓
TU38	Verifica che il metodo <code>get_next_possible_questions</code> di <code>GetNextPossibleQuestionsController</code> sollevi un'eccezione se la chiave 'answer' del dizionario ricevuto in input non è una stringa.	✓

Codice	Descrizione	Stato
TU39	Verifica che il metodo <code>get_next_possible_questions</code> di <code>GetNextPossibleQuestionsController</code> sollevi un'eccezione se la chiave <code>'quantity'</code> del dizionario ricevuto in input non è un intero.	✓
TU40	Verifica che il metodo <code>get_next_possible_questions</code> di <code>GetNextPossibleQuestionsController</code> sollevi un'eccezione se, nell'oggetto ricevuto da <code>GetNextPossibleQuestionsUseCase</code> , l'attributo del numero di domande non corrisponde alla lunghezza dell'attributo lista delle domande possibili.	✓
TU41	Verifica che il costruttore di <code>GetNextPossibleQuestionsService</code> sollevi un'eccezione se l'header non contiene la stringa <code>'***quantity***'</code> .	✓
TU42	Verifica che il metodo <code>get_next_possible_questions</code> di <code>GetNextPossibleQuestionsService</code> gestisca correttamente le eccezioni.	✓
TU43	Verifica che il metodo <code>load_github_commits</code> di <code>GitHubAdapter</code> gestisca correttamente le eccezioni.	✓
TU44	Verifica che il metodo <code>load_github_files</code> di <code>GitHubAdapter</code> gestisca correttamente le eccezioni.	✓
TU45	Verifica che il metodo <code>load_github_files</code> di <code>GitHubAdapter</code> gestisca correttamente le eccezioni di tipo <code>UnicodeDecodeError</code> .	✓
TU46	Verifica che il metodo <code>load_github_commits</code> di <code>GitHubRepository</code> carichi correttamente i commit di GitHub.	✓
TU47	Verifica che il metodo <code>load_github_commits</code> di <code>GitHubRepository</code> gestisca correttamente le eccezioni.	✓
TU48	Verifica che il metodo <code>load_github_files</code> di <code>GitHubRepository</code> carichi correttamente i file di GitHub.	✓
TU49	Verifica che il metodo <code>load_github_files</code> di <code>GitHubRepository</code> gestisca correttamente le eccezioni.	✓
TU50	Verifica che il metodo <code>load_github_files</code> di <code>GitHubRepository</code> gestisca correttamente le directory.	✓
TU51	Verifica che il metodo <code>load_jira_issues</code> di <code>JiraAdapter</code> gestisca correttamente le eccezioni.	✓
TU52	Verifica che il metodo <code>get_base_url</code> di <code>JiraRepository</code> restituisca l'URL base corretto.	✓
TU53	Verifica che il metodo <code>load_jira_issues</code> di <code>JiraRepository</code> restituisca correttamente le issues.	✓
TU54	Verifica che il metodo <code>load_jira_issues</code> di <code>JiraRepository</code> gestisca correttamente le eccezioni di richiesta.	✓
TU55	Verifica che il metodo <code>load_jira_issues</code> di <code>JiraRepository</code> gestisca correttamente le eccezioni generiche.	✓
TU56	Verifica che il metodo <code>generate_answer</code> di <code>LangChainAdapter</code> gestisca il superamento del limite di token escludendo i documenti che lo fanno superare.	✓
TU57	Verifica che il metodo <code>generate_answer</code> di <code>LangChainAdapter</code> gestisca correttamente le eccezioni.	✓

Codice	Descrizione	Stato
TU58	Verifica che il metodo <code>get_next_possible_questions</code> di <code>LangChainAdapter</code> gestisca correttamente le eccezioni.	✓
TU59	Verifica che il metodo <code>generate_answer</code> di <code>LangChainRepository</code> generi correttamente una risposta.	✓
TU60	Verifica che il metodo <code>generate_answer</code> di <code>LangChainRepository</code> gestisca correttamente le eccezioni.	✓
TU61	Verifica che il metodo <code>get_next_possible_questions</code> di <code>LangChainRepository</code> generi correttamente le prossime domande possibili.	✓
TU62	Verifica che il metodo <code>get_next_possible_questions</code> di <code>LangChainRepository</code> gestisca correttamente le eccezioni.	✓
TU63	Verifica che il metodo <code>load</code> di <code>LoadFilesController</code> gestisca correttamente le eccezioni.	✓
TU64	Verifica che il metodo <code>load</code> di <code>LoadFilesService</code> carichi correttamente i dati dai vari servizi e salvi i log di caricamento.	✓
TU65	Verifica che il metodo <code>load</code> di <code>LoadFilesService</code> gestisca correttamente le eccezioni.	✓
TU66	Verifica che il metodo <code>load</code> di <code>LoadFilesService</code> sollevi un'eccezione se il salvataggio nel database fallisce.	✓
TU67	Verifica che il metodo <code>load_github_commits</code> di <code>LoadFilesService</code> gestisca correttamente le eccezioni.	✓
TU68	Verifica che il metodo <code>load_github_files</code> di <code>LoadFilesService</code> gestisca correttamente le eccezioni.	✓
TU69	Verifica che il metodo <code>load_jira_issues</code> di <code>LoadFilesService</code> gestisca correttamente le eccezioni.	✓
TU70	Verifica che il metodo <code>load_confluence_pages</code> di <code>LoadFilesService</code> gestisca correttamente le eccezioni.	✓
TU71	Verifica che il metodo <code>clean_confluence_pages</code> di <code>LoadFilesService</code> gestisca correttamente le eccezioni.	✓
TU72	Verifica che il metodo <code>load_in_vector_store</code> di <code>LoadFilesService</code> gestisca correttamente le eccezioni.	✓
TU73	Verifica che il metodo <code>save_loading_attempt_in_db</code> di <code>LoadFilesService</code> gestisca correttamente le eccezioni.	✓
TU74	Verifica che il metodo <code>save_loading_attempt_in_txt</code> di <code>LoadFilesService</code> gestisca correttamente le eccezioni.	✓
TU75	Verifica che il metodo <code>get_messages</code> di <code>PostgresAdapter</code> ritorni una lista vuota se non ci sono messaggi.	✓
TU76	Verifica che il metodo <code>save_message</code> di <code>PostgresAdapter</code> gestisca correttamente le eccezioni.	✓
TU77	Verifica che il metodo <code>get_messages</code> di <code>PostgresAdapter</code> gestisca correttamente le eccezioni.	✓
TU78	Verifica che il metodo <code>save_loading_attempt</code> di <code>PostgresAdapter</code> gestisca correttamente le eccezioni.	✓
TU79	Verifica che il metodo <code>get_last_load_outcome</code> di <code>PostgresAdapter</code> gestisca correttamente le eccezioni.	✓

Codice	Descrizione	Stato
TU80	Verifica che il metodo <code>dsor_converter</code> di <code>PostgresAdapter</code> sollevi un'eccezione <code>ValueError</code> se il messaggio è vuoto.	✓
TU81	Verifica che il metodo <code>postgres_message_converter</code> di <code>PostgresAdapter</code> sollevi un'eccezione <code>ValueError</code> se il contenuto è vuoto.	✓
TU82	Verifica che il metodo <code>message_converter</code> di <code>PostgresAdapter</code> sollevi un'eccezione <code>ValueError</code> se il contenuto è vuoto.	✓
TU83	Verifica che il metodo <code>postgres_loading_attempt_converter</code> di <code>PostgresAdapter</code> sollevi un'eccezione <code>ValueError</code> se i platform logs sono vuoti.	✓
TU84	Verifica che il metodo <code>save_message</code> di <code>PostgresRepository</code> salvi correttamente un messaggio.	✓
TU85	Verifica che il metodo <code>save_message</code> di <code>PostgresRepository</code> restituisca <code>False</code> in caso di errore del database.	✓
TU86	Verifica che il metodo <code>save_message</code> di <code>PostgresRepository</code> gestisca correttamente un'eccezione generica.	✓
TU87	Verifica che il metodo <code>get_messages</code> di <code>PostgresRepository</code> recuperi correttamente i messaggi dal database.	✓
TU88	Verifica che il metodo <code>get_messages</code> di <code>PostgresRepository</code> restituisca una lista vuota nel caso in cui non ci siano messaggi nel database.	✓
TU89	Verifica che il metodo <code>get_messages</code> di <code>PostgresRepository</code> gestisca correttamente un'eccezione generica.	✓
TU90	Verifica che il metodo <code>save_loading_attempt</code> di <code>PostgresRepository</code> salvi correttamente un tentativo di caricamento nel database.	✓
TU91	Verifica che il metodo <code>save_loading_attempt</code> di <code>PostgresRepository</code> restituisca <code>False</code> in caso di errore del database.	✓
TU92	Verifica che il metodo <code>save_loading_attempt</code> di <code>PostgresRepository</code> gestisca correttamente un'eccezione generica.	✓
TU93	Verifica che il metodo <code>get_last_load_outcome</code> di <code>PostgresRepository</code> recuperi correttamente l'ultimo esito di caricamento dal database.	✓
TU94	Verifica che il metodo <code>get_last_load_outcome</code> di <code>PostgresRepository</code> restituisca <code>False</code> nel caso in cui non ci siano tuple nella tabella degli esiti di caricamento perchè ancora non è stato effettuato alcun caricamento.	✓
TU95	Verifica che il metodo <code>get_last_load_outcome</code> di <code>PostgresRepository</code> restituisca <code>Error</code> in caso di errore del database.	✓
TU96	Verifica che il metodo <code>get_last_load_outcome</code> di <code>PostgresRepository</code> gestisca correttamente un'eccezione generica.	✓
TU97	Verifica che il metodo <code>get_last_load_outcome</code> di <code>PostgresRepository</code> gestisca correttamente il caso in cui l'esito recuperato sia <code>False</code> .	✓
TU98	Verifica che il metodo <code>execute_query</code> di <code>PostgresRepository</code> gestisca correttamente le eccezioni.	✓

Codice	Descrizione	Stato
TU99	Verifica che il metodo <code>execute_query</code> di <code>PostgresRepository</code> recuperi correttamente un singolo risultato per una query di lettura.	✓
TU100	Verifica che il metodo <code>execute_query</code> di <code>PostgresRepository</code> recuperi correttamente tutti i risultati per una query di lettura.	✓
TU101	Verifica che il metodo <code>execute_query</code> di <code>PostgresRepository</code> esegua correttamente un commit per una query di scrittura.	✓
TU102	Verifica che il metodo <code>save</code> di <code>SaveMessageController</code> gestisca correttamente un messaggio con sender 'CHATBOT'.	✓
TU103	Verifica che il metodo <code>save</code> di <code>SaveMessageController</code> sollevi un'eccezione per un messaggio avente un sender non valido.	✓
TU104	Verifica che il metodo <code>save</code> di <code>SaveMessageController</code> sollevi un'eccezione per un messaggio avente timestamp non valido.	✓
TU105	Verifica che il metodo <code>save</code> di <code>SaveMessageController</code> gestisca correttamente le eccezioni generali.	✓
TU106	Verifica che il metodo <code>save</code> di <code>SaveMessageService</code> gestisca correttamente le eccezioni.	✓
TU107	Verifica che il metodo <code>similarity_search</code> di <code>SimilaritySearchService</code> salti i documenti che superano la soglia di similarità.	✓
TU108	Verifica che il metodo <code>similarity_search</code> di <code>SimilaritySearchService</code> termini e restituisca i documenti trovati finora se il distacco massimo è superato.	✓
TU109	Verifica che il metodo <code>similarity_search</code> di <code>SimilaritySearchService</code> gestisca correttamente le eccezioni.	✓
TU110	Verifica che il metodo <code>split</code> di <code>ChromaVectorStoreAdapter</code> divida correttamente in più chunk i documenti che contengono più caratteri dell'attributo <code>max_chunk_size</code> .	✓
TU111	Verifica che il metodo <code>UTC_to_CET</code> di <code>ConfluenceAdapter</code> gestisca correttamente le eccezioni.	✓
TU112	Verifica che il metodo <code>get_github_files_new_metadata</code> di <code>LoadFilesService</code> aggiorni correttamente i metadati "last_update" e "creation_date" dei file di GitHub.	✓
TU113	Verifica che il metodo <code>get_github_files_new_metadata</code> di <code>LoadFilesService</code> gestisca correttamente l'eccezione di path mancante nei metadati dei file di GitHub.	✓
TU114	Verifica che il metodo <code>get_github_files_new_metadata</code> di <code>LoadFilesService</code> gestisca correttamente le eccezioni di formato della data errato o di data assente nei metadati dei commit di GitHub.	✓
TU115	Verifica che il metodo <code>load</code> di <code>ChromaVectorStoreRepository</code> gestisca correttamente le eccezioni durante la preparazione dei dati per l'aggiunta nel database vettoriale.	✓
TU116	Verifica che il metodo <code>load</code> di <code>ChromaVectorStoreRepository</code> gestisca correttamente le eccezioni durante il recupero dei vecchi dati dal database vettoriale.	✓

Codice	Descrizione	Stato
TU117	Verifica che il metodo load di ChromaVectorStoreRepository gestisca correttamente le eccezioni durante il parsing delle date per documenti non di tipo GitHub File.	✓
TU118	Verifica che il metodo load di ChromaVectorStoreRepository gestisca correttamente le eccezioni durante il parsing delle date per documenti di tipo GitHub File.	✓
TU119	Verifica che il metodo load di ChromaVectorStoreRepository gestisca correttamente l'eccezione riguardante un documento di tipo GitHub File senza il campo 'path'.	✓
TU120	Verifica che il metodo load di ChromaVectorStoreRepository segnali correttamente la modifica di un documento di tipo GitHub File, e non la sua creazione, se la data di creazione è più vecchia di quella dell'ultimo inserimento nel database vettoriale.	✓
TU121	Verifica che il metodo load di ChromaVectorStoreRepository ignori i documenti di tipo GitHub File durante il controllo di modifica basato sul metadato 'last_update'.	✓
TU122	Verifica che il metodo load di ChromaVectorStoreRepository segnali correttamente la modifica di un documento con il metadato 'last_update' più recente rispetto al corrispondente presente nel database vettoriale.	✓
TU123	Verifica che il metodo load di ChromaVectorStoreRepository gestisca correttamente le eccezioni durante la cancellazione dei documenti dal database vettoriale.	✓
TU124	Verifica che l'istanza del componente ChatHeaderComponent sia definita.	✓
TU125	Verifica che il componente ChatHeaderComponent venga creato.	✓
TU126	Verifica che il tag header del template HTML di ChatHeaderComponent abbia la classe CSS 'chat-header'.	✓
TU127	Verifica che l'immagine del logo di ChatHeaderComponent, in HTML, abbia gli attributi src e alt corretti.	✓
TU128	Verifica che il tag app-chat-badge sia presente nel template HTML di ChatHeaderComponent.	✓
TU129	Verifica che il titolo h1 del template HTML di ChatHeaderComponent contenga 'BuddyBot'.	✓
TU130	Verifica che l'istanza del componente AppComponent sia definita.	✓
TU131	Verifica che il componente AppComponent venga creato.	✓
TU132	Verifica che il metodo loadOldMessages di ChatContainerComponent, se il recupero dei messaggi ha successo, resetti l'attributo errorMessage ad una stringa vuota.	✓
TU133	Verifica che il metodo onSuggestionClicked di ChatContainerComponent, invocato con un suggerimento, richiami il metodo onSendMessage passando come parametro il suggerimento ricevuto.	✓
TU134	Verifica che il metodo loadOldMessages di ChatContainerComponent, in caso di errore nel recupero dei messaggi, imposti l'attributo errorMessage come equivalente al messaggio di errore.	✓

Codice	Descrizione	Stato
TU135	Verifica che il metodo <code>formatResponse</code> di <code>ChatContainerComponent</code> trasformi correttamente la stringa ricevuta in input.	✓
TU136	Verifica che il metodo <code>onSendMessage</code> di <code>ChatContainerComponent</code> non faccia nulla se il testo ricevuto in input è vuoto oppure contiene solo spazi.	✓
TU137	Verifica che l'evento <code>ngOnInit</code> di <code>ChatContainerComponent</code> richiami il metodo <code>loadOldMessages</code> con l'intero 50 come parametro.	✓
TU138	Verifica che il metodo <code>onScrollChange</code> di <code>ChatContainerComponent</code> , chiamato con valore <code>true</code> , aggiorni l'attributo <code>showScrollToBottom</code> a <code>true</code> .	✓
TU139	Verifica che il metodo <code>onScrollChange</code> di <code>ChatContainerComponent</code> , chiamato con valore <code>false</code> , aggiorni l'attributo <code>showScrollToBottom</code> a <code>false</code> .	✓
TU140	Verifica che il valore iniziale dell'attributo <code>lastLoadOutcome</code> di <code>ChatBadgeComponent</code> sia <code>TRUE</code> .	✓
TU141	Verifica che il getter <code>isUpdated</code> di <code>ChatBadgeComponent</code> restituisca il valore corretto, corrispondente all'attributo <code>lastLoadOutcome</code> .	✓
TU142	Verifica che, quando si clicca il bottone 'Invia' di <code>ChatInputComponent</code> , venga inviata la stringa corretta.	✓
TU143	Verifica che la barra di input di <code>ChatInputComponent</code> non sia disabilitata se l'attributo <code>isLoading</code> è <code>false</code> .	✓
TU144	Verifica che la barra di input di <code>ChatInputContainer</code> abbia il placeholder 'Scrivi un messaggio...'.	✓
TU145	Verifica che la barra di input di <code>ChatInputComponent</code> sia disabilitata se l'attributo <code>isLoading</code> è <code>true</code> .	✓
TU146	Verifica che il componente <code>ChatInputComponent</code> venga creato.	✓
TU147	Verifica che il bottone 'Invia' di <code>ChatInputComponent</code> sia abilitato se l'attributo <code>isLoading</code> è <code>false</code> e <code>userInput</code> contiene un messaggio valido.	✓
TU148	Verifica che il bottone 'Invia' di <code>ChatInputComponent</code> sia disabilitato se l'attributo <code>userInput</code> è vuoto.	✓
TU149	Verifica che il metodo <code>onSend</code> di <code>ChatInputComponent</code> emetta il valore corretto e resettì l'input dell'utente se le condizioni di invio della domanda sono soddisfatte, cioè <code>userInput</code> contenga dei caratteri e <code>isLoading</code> sia <code>false</code> .	✓
TU150	Verifica che il bottone 'Invia' di <code>ChatInputComponent</code> sia disabilitato se l'attributo <code>userInput</code> contiene solo spazi.	✓
TU151	Verifica che il metodo <code>onSend</code> di <code>ChatInputComponent</code> non emetta alcun segnale se <code>isLoading</code> è <code>true</code> .	✓
TU152	Verifica che il metodo <code>onSend</code> di <code>ChatInputComponent</code> non emetta alcun segnale se l'input dell'utente è vuoto oppure contiene solo spazi.	✓

Codice	Descrizione	Stato
TU153	Verifica che, quando si digita nella barra di input di ChatInputComponent, il valore dell'attributo userInput si aggiorni.	✓
TU154	Verifica che, quando si preme il tasto Enter e il focus sta nella barra di input di ChatInputComponent, venga inviata la stringa corretta.	✓
TU155	Verifica che il bottone 'Invia' di ChatInputComponent sia disabilitato se l'attributo isLoading è true.	✓
TU156	Verifica che il metodo copyToClipboard di ChatMessagesComponent usi l'attributo msg.sanitizedContent del parametro msg se disponibile.	✓
TU157	Verifica che il metodo copyToClipboard di ChatMessagesComponent rimuova 'content_copy' multiplo e modifichi msg.copied.	✓
TU158	Verifica che il componente ChatMessagesComponent venga creato.	✓
TU159	Verifica che l'istanza del componente ChatMessagesComponent sia definita.	✓
TU160	Verifica che il metodo copySnippet di ChatMessagesComponent chiami il metodo navigator.clipboard.writeText e inserisca e poi rimuova opportunamente la classe CSS 'snippet-copied' all'icona di copia.	✓
TU161	Verifica che il metodo scrollToBottom di ChatMessagesComponent imposti scrollTop (la posizione di scorrimento verticale corrente) uguale a scrollHeight (l'altezza totale del contenuto dell'elemento HTML).	✓
TU162	Verifica che il metodo onScroll di ChatMessagesComponent non fallisca se l'attributo messagesContainer è undefined.	✓
TU163	Verifica che il metodo onScroll di ChatMessagesComponent emetta true se non si è in fondo.	✓
TU164	Verifica che onScroll di ChatMessagesComponent emetta false se si è in fondo.	✓
TU165	Verifica che il click sull'elemento HTML con classe CSS 'copy-snippet-icon' di ChatMessagesComponent chiami il metodo copySnippet.	✓
TU166	Verifica che stripHtml di ChatMessagesComponent converta correttamente l'HTML in testo.	✓
TU167	Verifica che il metodo scrollToBottom di ChatMessagesComponent non fallisca se l'attributo messagesContainer è undefined.	✓
TU168	Verifica che il metodo stripHtml di ChatMessagesComponent ritorni una stringa vuota se il parametro in input è vuoto.	✓
TU169	Verifica che l'evento ngAfterViewInit di ChatMessagesComponent aggiunga un event listener sull'attributo messagesContainer.	✓
TU170	Verifica che il template HTML di ChatLoadingIndicatorComponent contenga un'immagine con attributo alt 'Caricamento'.	✓

Codice	Descrizione	Stato
TU171	Verifica che il template HTML di ChatLoadingIndicatorComponent contenga un'immagine con attributo src 'assets/banana.png'.	✓
TU172	Verifica che il template HTML di ChatLoadingIndicatorComponent contenga l'elemento con classe CSS 'loading-indicator'.	✓
TU173	Verifica che l'istanza del componente ChatLoadingIndicatorComponent sia definita.	✓
TU174	Verifica che il componente ChatLoadingIndicatorComponent venga creato.	✓
TU175	Verifica che il metodo canLoadSuggestions di ChatSuggestionsComponent restituisca false se l'attributo answer è vuoto.	✓
TU176	Verifica che il metodo canLoadSuggestions di ChatSuggestionsComponent restituisca true per input validi.	✓
TU177	Verifica che il metodo onSuggestionClick di ChatSuggestionsComponent emetta il testo corretto, corrispondente alla domanda suggerita cliccata dall'utente.	✓
TU178	Verifica che il metodo ngOnChanges di ChatSuggestionsComponent non reagisca se non ci sono cambiamenti rilevanti.	✓
TU179	Verifica che l'istanza del componente ChatSuggestionsComponent sia definita.	✓
TU180	Verifica che il componente ChatSuggestionsComponent venga creato.	✓
TU181	Verifica che il metodo canLoadSuggestions di ChatSuggestionsComponent restituisca false se l'attributo question è vuoto.	✓
TU182	Verifica che, se l'attributo hideSuggestions di ChatSuggestionsComponent è true, l'attributo continuationSuggestions sia una lista e l'attributo loadError sia false.	✓
TU183	Verifica che il metodo sendMessage di ChatService aggiorni correttamente l'attributo lastMessageTimestamp.	✓
TU184	Verifica che ChatService restituisca e imposti correttamente il valore dell'attributo lastMessageTimestamp.	✓
TU185	Verifica che il metodo sendMessage di ChatService invochi il metodo setLastMessageTimestamp.	✓

Tabella 4: Insieme dei test di unità

3.2 Test di integrazione

I test di integrazione servono a verificare che i vari moduli o unità di codice dell'applicazione collaborino correttamente tra loro, individuando eventuali problemi di interazione o incompatibilità. In questo modo si garantisce che l'intero sistema operi come previsto quando i singoli componenti vengono combinati.

Codice	Descrizione	Stato
TI1	Verifica che il metodo <code>get_answer</code> di <code>ChatController</code> chiami il metodo <code>get_answer</code> di <code>ChatUseCase</code> .	✓
TI2	Verifica che il metodo <code>generate_answer</code> di <code>ChatService</code> chiami il metodo <code>generate_answer</code> di <code>GenerateAnswerService</code> .	✓
TI3	Verifica che il metodo <code>similarity_search</code> di <code>ChatService</code> chiami il metodo <code>similarity_search</code> di <code>SimilaritySearchService</code> .	✓
TI4	Verifica che il metodo <code>similarity_search</code> di <code>ChromaVectorStoreAdapter</code> chiami il metodo <code>similarity_search</code> di <code>ChromaVectorStoreRepository</code> .	✓
TI5	Verifica che il metodo <code>load</code> di <code>ChromaVectorStoreAdapter</code> chiami il metodo <code>load</code> di <code>ChromaVectorStoreRepository</code> .	✓
TI6	Verifica che il metodo <code>load_confluence_pages</code> di <code>ConfluenceAdapter</code> chiami il metodo <code>load_confluence_pages</code> di <code>ConfluenceRepository</code> .	✓
TI7	Verifica che il metodo <code>generate_answer</code> di <code>GenerateAnswerService</code> chiami il metodo <code>generate_answer</code> di <code>GenerateAnswerPort</code> .	✓
TI8	Verifica che il metodo <code>get_last_load_outcome</code> di <code>GetLastLoadOutcomeController</code> chiami il metodo <code>get_last_load_outcome</code> di <code>GetLastLoadOutcomeUseCase</code> .	✓
TI9	Verifica che il metodo <code>get_last_load_outcome</code> di <code>GetLastLoadOutcomeService</code> chiami il metodo <code>get_last_load_outcome</code> di <code>GetLastLoadOutcomePort</code> .	✓
TI10	Verifica che il metodo <code>get_messages</code> di <code>GetMessagesController</code> chiami il metodo <code>get_messages</code> di <code>GetMessagesUseCase</code> .	✓
TI11	Verifica che il metodo <code>get_messages</code> di <code>GetMessagesService</code> chiami il metodo <code>get_messages</code> di <code>GetMessagesPort</code> .	✓
TI12	Verifica che il metodo <code>get_next_possible_questions</code> di <code>GetNextPossibleQuestionsController</code> chiami il metodo <code>get_next_possible_questions</code> di <code>GetNextPossibleQuestionsUseCase</code> .	✓
TI13	Verifica che il metodo <code>get_next_possible_questions</code> di <code>GetNextPossibleQuestionsService</code> chiami il metodo <code>get_next_possible_questions</code> di <code>GetNextPossibleQuestionsPort</code> .	✓
TI14	Verifica che il metodo <code>load_github_commits</code> di <code>GitHubAdapter</code> chiami il metodo <code>load_github_commits</code> di <code>GitHubRepository</code> .	✓
TI15	Verifica che il metodo <code>load_github_files</code> di <code>GitHubAdapter</code> chiami il metodo <code>load_github_files</code> di <code>GitHubRepository</code> .	✓
TI16	Verifica che il metodo <code>load_jira_issues</code> di <code>JiraAdapter</code> chiami il metodo <code>load_jira_issues</code> di <code>JiraRepository</code> .	✓
TI17	Verifica che il metodo <code>generate_answer</code> di <code>LangChainAdapter</code> chiami il metodo <code>generate_answer</code> di <code>LangChainRepository</code> .	✓

Codice	Descrizione	Stato
TI18	Verifica che il metodo <code>get_next_possible_questions</code> di <code>LangChainAdapter</code> chiami il metodo <code>get_next_possible_questions</code> di <code>LangChainRepository</code> .	✓
TI19	Verifica che il metodo <code>load</code> di <code>LoadFilesController</code> chiami il metodo <code>load</code> di <code>LoadFilesUseCase</code> .	✓
TI20	Verifica che il metodo <code>clean_confluence_pages</code> di <code>LoadFilesService</code> chiami il metodo <code>clean_confluence_pages</code> di <code>ConfluenceCleanerService</code> .	✓
TI21	Verifica che il metodo <code>load_confluence_pages</code> di <code>LoadFilesService</code> chiami il metodo <code>load_confluence_pages</code> di <code>ConfluencePort</code> .	✓
TI22	Verifica che il metodo <code>load_github_commits</code> di <code>LoadFilesService</code> chiami il metodo <code>load_github_commits</code> di <code>GitHubPort</code> .	✓
TI23	Verifica che il metodo <code>load_github_files</code> di <code>LoadFilesService</code> chiami il metodo <code>load_github_files</code> di <code>GitHubPort</code> .	✓
TI24	Verifica che il metodo <code>load_jira_issues</code> di <code>LoadFilesService</code> chiami il metodo <code>load_jira_issues</code> di <code>JiraPort</code> .	✓
TI25	Verifica che il metodo <code>load_in_vector_store</code> di <code>LoadFilesService</code> chiami il metodo <code>load</code> di <code>LoadFilesInVectorStorePort</code> .	✓
TI26	Verifica che il metodo <code>save_loading_attempt_in_db</code> di <code>LoadFilesService</code> chiami il metodo <code>save_loading_attempt</code> di <code>SaveLoadingAttemptInDbPort</code> .	✓
TI27	Verifica che il metodo <code>save_message</code> di <code>PostgresAdapter</code> chiami il metodo <code>save_message</code> di <code>PostgresRepository</code> .	✓
TI28	Verifica che il metodo <code>get_messages</code> di <code>PostgresAdapter</code> chiami il metodo <code>get_messages</code> di <code>PostgresRepository</code> .	✓
TI29	Verifica che il metodo <code>save_loading_attempt</code> di <code>PostgresAdapter</code> chiami il metodo <code>save_loading_attempt</code> di <code>PostgresRepository</code> .	✓
TI30	Verifica che il metodo <code>get_last_load_outcome</code> di <code>PostgresAdapter</code> chiami il metodo <code>get_last_load_outcome</code> di <code>PostgresRepository</code> .	✓
TI31	Verifica che il metodo <code>save</code> di <code>SaveMessageController</code> chiami il metodo <code>save</code> di <code>SaveMessageUseCase</code> .	✓
TI32	Verifica che il metodo <code>save</code> di <code>SaveMessageService</code> chiami il metodo <code>save_message</code> di <code>SaveMessagePort</code> .	✓
TI33	Verifica che il metodo <code>similarity_search</code> di <code>SimilaritySearchService</code> chiami il metodo <code>similarity_search</code> di <code>SimilaritySearchPort</code> .	✓
TI34	Verifica che il contenuto del template HTML di <code>ChatContainerComponent</code> venga renderizzato in <code>AppComponent</code> .	✓
TI35	Verifica che il template HTML di <code>AppComponent</code> contenga il tag <code>app-chat-container</code> di <code>ChatContainerComponent</code> .	✓
TI36	Verifica che l'evento <code>ngOnInit</code> di <code>ChatContainerComponent</code> richiami il metodo <code>loadLastLoadOutcome</code> di <code>DatabaseService</code> .	✓
TI37	Verifica che il metodo <code>loadOldMessages</code> di <code>ChatContainerComponent</code> ordini e formatti correttamente i messaggi ricevuti da <code>DatabaseService</code> .	✓

Codice	Descrizione	Stato
TI38	Verifica che il metodo loadOldMessages di ChatContainerComponent imposti errorMessage in caso di errore nel recupero dei messaggi restituito da DatabaseService.	✓
TI39	Verifica che il metodo loadOldMessages di ChatContainerComponent resetti errorMessage se il recupero dei messaggi da DatabaseService ha successo.	✓
TI40	Verifica che il metodo scrollToBottom di ChatContainerComponent invochi il metodo scrollToBottom di ChatMessagesComponent.	✓
TI41	Verifica che il metodo onSendMessage di ChatContainerComponent gestisca correttamente il caso di invio di una domanda a ChatService a seguito della quale viene restituita una risposta positiva.	✓
TI42	Verifica che il metodo onSendMessage di ChatContainerComponent gestisca correttamente il caso di invio di una domanda a ChatService a seguito della quale viene restituito un errore.	✓
TI43	Verifica che, durante l'inizializzazione di ChatBadgeComponent, la sottoscrizione a lastLoadOutcome\$ di DatabaseService aggiorni l'attributo lastLoadOutcome di ChatBadgeComponent.	✓
TI44	Verifica che, al cambio di valore di lastLoadOutcome\$ in DatabaseService, l'attributo lastLoadOutcome di ChatBadgeComponent venga aggiornato.	✓
TI45	Verifica che il metodo saveMessage di DatabaseService invii il payload corretto all'endpoint POST /api/save_message e gestisca correttamente la risposta.	✓
TI46	Verifica che il metodo loadLastLoadOutcome di DatabaseService imposti TRUE se la risposta ricevuta dall'endpoint /api/get_last_load_outcome è 'True'.	✓
TI47	Verifica che il metodo loadLastLoadOutcome di DatabaseService imposti ERROR in caso di errore ricevuto dall'endpoint /api/get_last_load_outcome.	✓
TI48	Verifica che il metodo loadLastLoadOutcome di DatabaseService imposti FALSE se la risposta ricevuta dall'endpoint /api/get_last_load_outcome è 'False'.	✓
TI49	Verifica che il metodo loadLastLoadOutcome di DatabaseService imposti ERROR se la risposta ricevuta dall'endpoint /api/get_last_load_outcome non è 'True' o 'False'.	✓
TI50	Verifica che il metodo getMessages di DatabaseService mappi correttamente la risposta dell'endpoint POST /api/get_messages in un array di Message.	✓
TI51	Verifica che il metodo getContinuationSuggestions di ChatService non venga chiamato se il metodo canLoadSuggestions di ChatSuggestionsComponent restituisce false.	✓
TI52	Verifica che, se il metodo getContinuationSuggestions di ChatService fallisce, l'attributo loadError di ChatSuggestionsComponent venga impostato a true e venga impostato l'attributo errorMessage come equivalente al messaggio d'errore.	✓

Codice	Descrizione	Stato
TI53	Verifica che, se gli attributi question e answer di ChatSuggestionsComponent sono validi e l'attributo hideSuggestions è false, vengano caricate le suggestions chiamando il metodo getContinuationSuggestions di ChatService.	✓
TI54	Verifica che il metodo sendMessage di ChatService chiami l'endpoint POST /api/chat e ne gestisca la risposta.	✓
TI55	Verifica che il metodo getContinuationSuggestions di ChatService gestisca correttamente l'errore ricevuto dall'endpoint POST /api/get_next_possible_questions.	✓
TI56	Verifica che il metodo getContinuationSuggestions di ChatService chiami l'endpoint POST /api/get_next_possible_questions e ne gestisca la risposta.	✓

Tabella 5: Insieme dei test di integrazione

3.3 Test di sistema

I test di accettazione descritti di seguito rappresentano la base per la validazione del *Minimum Viable Product_G*. Questi test sono stati progettati per verificare l'implementazione delle funzionalità previste nel prodotto, con l'obiettivo di garantire che risponda agli standard di un prodotto completo e pronto per il collaudo.

Codice	Descrizione	Fonte	Stato
TS1	Verificare che l'utente possa inserire un'interrogazione in linguaggio naturale nel sistema.	ROF1	✓
TS2	Verificare che all'invio dell'interrogazione al sistema, venga generata una risposta.	ROF2	✓
TS3	Verificare che se il sistema fallisce nel generare una risposta per via di un problema interno, venga visualizzato all'utente un messaggio di errore, chiedendo di riprovare più tardi.	ROF3	✓
TS4	Verificare che se l'utente inserisce un'interrogazione che non riguarda i contenuti del database associato, il sistema risponda all'utente che la domanda inserita è fuori contesto.	ROF4	✓
TS5	Verificare che se il sistema non riesce a trovare le informazioni richieste dall'utente nonostante siano correlate al contesto, venga spiegata la mancanza dell'informazione richiesta.	ROF5	✓
TS6	Verificare che l'utente possa visualizzare i link dei file da cui il sistema ha preso i dati per la risposta.	RDF6	✓
TS7	Verificare che nel caso in cui non sia possibile recuperare i link dei file utilizzati per generare la risposta, venga visualizzato un messaggio di errore.	RDF7	✓
TS8	Verificare che sia presente un pulsante al cui click la risposta del <i>chatbot_G</i> venga copiata nel dispositivo dell'utente.	RZF8	✓
TS9	Verificare che nel caso la risposta contenga uno <i>snippet_G</i> di codice, sia presente un pulsante che permetta di copiare il singolo <i>snippet</i> nel dispositivo dell'utente.	RDF9	✓
TS10	Verificare che sia presente un sistema di archiviazione delle domande e delle risposte in un <i>database relazionale_G</i> .	RDF10	✓
TS11	Verificare che l'utente possa visualizzare lo storico della chat, recuperato dal <i>database relazionale</i> .	RDF11	✓
TS12	Verificare che nel caso il sistema fallisca nel recuperare lo storico della chat, venga visualizzato un messaggio di errore all'utente spiegando che non è stato possibile recuperare lo storico.	RDF12	✓

Codice	Descrizione	Fonte	Stato
TS13	Verificare che uno <i>scheduler_G</i> si colleghi al sistema e periodicamente aggiorni il <i>database vettoriale_G</i> con i dati più recenti.	ROF13	✓
TS14	Verificare che la risposta venga generata prendendo in considerazione i dati di contesto provenienti da <i>GitHub_G</i> , <i>Jira_G</i> e <i>Confluence_G</i> .	ROF14	✓
TS15	Verificare che il sistema possa convertire i dati ottenuti in formato vettoriale.	ROF15	✓
TS16	Verificare che il sistema possa aggiornare il <i>database vettoriale</i> con i nuovi dati ottenuti.	ROF16	✓
TS17	Verificare che se la conversazione non è ancora avviata, l'utente possa visualizzare e selezionare alcune domande di partenza proposte.	RZF17	✗
TS18	Verificare che dopo la visualizzazione di una risposta, all'utente vengano suggerite alcune interrogazioni che è possibile porre al sistema per continuare la conversazione.	RZF18	✓
TS19	Verificare che nel caso il sistema vada in errore nel tentativo di proporre altre domande per proseguire la conversazione, venga mostrato un messaggio che comunica l'errore all'utente e invita a fare altre domande.	RZF19	✓
TS20	Verificare che il sistema mostri a schermo un badge sopra alla schermata della chat.	RZF20	✓
TS21	Verificare che il sistema comunichi all'utente l'esito dell'ultimo tentativo di aggiornamento del <i>database vettoriale</i> utilizzando un badge.	RZF21	✓
TS22	Verificare che il sistema elabori i documenti caricati, generando i relativi <i>embedding_G</i> .	ROF22	✓
TS23	Verificare che il sistema salvi, in modo persistente, il contenuto dei documenti caricati.	ROF23	✓
TS24	Verificare che il sistema salvi, in modo persistente, i metadati dei documenti caricati.	ROF24	✓
TS25	Verificare che il sistema salvi, i commit e i file scritti in caratteri testuali di <i>GitHub</i> , escludendo PDF o immagini.	ROF25	✓
TS26	Verificare che il sistema permetta all'utente di digitare una domanda libera in linguaggio naturale tramite una barra di input.	ROF26	✓
TS27	Verificare che l'utente possa selezionare una delle domande proposte dal sistema e inviarla come interrogazione al <i>chatbot</i> .	RZF27	✓
TS28	Verificare che il sistema mostri a schermo i messaggi dell'utente presenti nello storico.	RDF28	✓
TS29	Verificare che il sistema mostri a schermo i messaggi del <i>chatbot</i> presenti nello storico.	RDF29	✓

Codice	Descrizione	Fonte	Stato
TS30	Verificare che i messaggi del <i>chatbot</i> e dell'utente mostrati a schermo devono essere distinguibili tra loro tramite una colorazione differente.	RDF30	✓
TS31	Verificare che il sistema carichi e mostri a schermo i messaggi cronologicamente precedenti, quando l'utente, tramite scroll verso il basso raggiunge l'inizio della lista dei messaggi mostrati.	RDF31	✓
TS32	Verificare che nel caso il sistema fallisca nel recuperare i messaggi precedenti dal database, l'utente visualizzi un messaggio che comunica che non è stato possibile recuperare altri messaggi dal database.	RDF32	✓
TS33	Verificare che nel caso non ci siano altri messaggi nel database, l'utente visualizzi un messaggio che comunica che non ci sono altri messaggi da visualizzare.	RDF33	✗
TS34	Verificare che il sistema comunichi all'utente se l'ultimo tentativo di aggiornamento del database vettoriale ha avuto successo.	RZF34	✓
TS35	Verificare che il sistema comunichi all'utente se l'ultimo tentativo di aggiornamento del database vettoriale è fallito.	RZF35	✓
TS36	Verificare che il sistema comunichi all'utente se non è stato possibile recuperare l'esito dell'ultimo aggiornamento del <i>database vettoriale</i> .	RZF36	✓
TS37	Verificare che l'utente, accanto al badge che segnala l'esito dell'aggiornamento automatico del <i>database vettoriale</i> , visualizzi un messaggio che spiega il significato di quest'ultimo nella sua forma corrente.	RZF37	✓
TS38	Verificare che se l'ultimo tentativo di aggiornamento del <i>database vettoriale</i> ha avuto successo, l'utente accanto al badge visualizzi un messaggio che spiega il significato della forma a spunta di quest'ultimo.	RZF38	✓
TS39	Verificare che se l'ultimo tentativo di aggiornamento del <i>database vettoriale</i> è fallito, l'utente accanto al badge visualizzi un messaggio che spiega il significato della forma a X di quest'ultimo.	RZF39	✓
TS40	Verificare che se non è stato possibile recuperare l'esito dell'ultimo tentativo di aggiornamento del <i>database vettoriale</i> , l'utente accanto al badge visualizzi un messaggio che spiega il significato della forma a segnale di pericolo di quest'ultimo.	RZF40	✓

Codice	Descrizione	Fonte	Stato
TS41	Verificare che ad ogni tentativo di aggiornamento del <i>database vettoriale</i> il sistema salvi in un file di testo un log che contenga le seguenti informazioni: fonte dei dati (<i>GitHub</i> , <i>Jira</i> o <i>Confluence</i>), esito, orario.	RZF41	✓
TS42	Verificare che ad ogni tentativo di aggiornamento del <i>database vettoriale</i> il sistema salvi in un file di testo un log che contenga le seguenti informazioni: fonte dei dati (<i>GitHub</i> , <i>Jira</i> o <i>Confluence</i>), esito, orario, numero di file aggiunti, numero di file modificati, numero di file eliminati.	RZF42	✓
TS43	Verificare che il sistema mostri a schermo la data e l'ora di ogni messaggio presente nello storico.	RDF43	✓
TS44	Verificare che, nel caso non sia possibile recuperare la data e l'ora di un messaggio presente nello storico, deve essere visualizzato al suo posto un avviso di errore.	RDF44	✓
TS45	Verificare che il sistema mostri un riquadro di caricamento durante la generazione di una risposta.	RZF45	✓
TS46	Verificare che il sistema mostri un riquadro di caricamento durante il caricamento dello storico dei messaggi.	RZF46	✓
TS47	Verificare che il sistema mostri un riquadro di caricamento durante il caricamento dei messaggi precedenti.	RZF47	✓
TS48	Verificare che sia presente un pulsante che, al click, permetta all'utente di tornare ai messaggi più recenti della chat.	RZF48	✓
TS49	Verificare che l'applicazione sia compatibile con la versione 131.0.6778.204/.205 di <i>Google Chrome</i> _G , ovvero l'ultima disponibile al 19 dicembre, data in cui è stato richiesto questo vincolo.	ROV1	✓
TS50	Verificare che il sistema garantisca l'integrazione con le <i>API</i> _G di <i>Confluence</i> .	ROV2	✓
TS51	Verificare che il sistema garantisca l'integrazione con le <i>API</i> di <i>Jira</i> .	ROV3	✓
TS52	Verificare che il sistema garantisca l'integrazione con le <i>API</i> di <i>GitHub</i> .	ROV4	✓

Tabella 6: Insieme dei test di sistema

3.4 Test di accettazione

I test di accettazione descritti di seguito rappresentano la base per la validazione del *Minimum Viable Product_G*. Questi test sono stati progettati per verificare l'implementazione delle funzionalità previste nel prodotto, con l'obiettivo di garantire che risponda agli standard di un prodotto completo e pronto per il collaudo.

Codice	Descrizione	Fonte	Stato
TA1	Verificare che l'utente possa inserire un interrogazione in linguaggio naturale.	UC1, UC18, UC19	✓
TA2	Verificare che l'utente possa visualizzare la risposta generata dal sistema con relativi errori nel caso non sia possibile generarla.	UC2, UC3, UC4, UC5	✓
TA3	Verificare che l'utente possa visualizzare i link dei file da cui il sistema ha preso i dati per la risposta alla domanda con relativa segnalazione nel caso in cui non siano stati rilevati i link dei file.	UC2.1, UC14	✓
TA4	Verificare che l'utente possa copiare il testo della risposta generata.	UC6	✓
TA5	Verificare che l'utente possa copiare lo snippet di codice presente nel testo della risposta generata.	UC7	✓
TA6	Verificare che l'utente possa visualizzare gli ultimi 50 messaggi della chat con relativo errore nel caso non sia possibile.	UC8, UC8.1, UC8.2, UC8.3, UC9, UC27	✓
TA7	Verificare che l'utente possa visualizzare la data e l'ora dell'invio del messaggio con relativo errore nel caso non sia possibile recuperarle.	UC8.1.1, UC28	✓
TA8	Verificare che lo scheduler effettui l'aggiornamento automatico del database vettoriale.	UC10	✓
TA9	Verificare che l'utente possa visualizzare una lista di domande ideali per poter iniziare una conversazione.	UC11, UC11.1	✗
TA10	Verificare che l'utente possa visualizzare una lista di domande ideali per poter proseguire una conversazione con relativo errore nel caso non sia possibile.	UC12, UC12.1, UC13	✓
TA11	<p>Verificare che l'utente possa visualizzare un badge che segnala l'esito dell'ultimo aggiornamento del database vettoriale, in una delle seguenti forme:</p> <ul style="list-style-type: none"> • Badge a forma di spunta se l'ultimo aggiornamento del database vettoriale è avvenuto con successo; • Badge a forma di "X" se c'è stato un fallimento nell'ultimo aggiornamento del database vettoriale; • Badge a forma di segnale di pericolo se non è stato possibile recuperare l'esito dell'ultimo aggiornamento del database vettoriale. 	UC15, UC16, UC17, UC20	✓

Codice	Descrizione	Fonte	Stato
TA12	<p>Verificare che l'utente possa visualizzare un messaggio che spiega il badge che segnala l'esito dell'ultimo aggiornamento del database vettoriale, in una delle seguenti possibilità:</p> <ul style="list-style-type: none"> • Messaggio che spiega il badge che segnala il successo dell'ultimo aggiornamento del database vettoriale; • Messaggio che spiega il badge che segnala il fallimento dell'ultimo aggiornamento del database vettoriale; • Messaggio che spiega il badge che segnala l'impossibilità di recuperare l'esito dell'ultimo aggiornamento del database vettoriale. 	UC21, UC22, UC23, UC24	✓
TA13	Verificare che l'utente possa visualizzare una lista di 50 messaggi immediatamente precedenti a quanto già presente nella vista, con relativo errore nel caso non sia possibile recuperare i messaggi.	UC25, UC26, UC27, UC8.1	✓
TA14	Verificare che l'utente possa visualizzare un riquadro di caricamento di una risposta.	UC29	✓
TA15	Verificare che l'utente possa visualizzare un riquadro di caricamento dello storico dei messaggi.	UC30	✓
TA16	Verificare che l'utente possa visualizzare un riquadro di caricamento dei messaggi precedenti.	UC31	✓
TA17	Verificare che l'utente possa, premendo un pulsante, tornare ai messaggi recenti.	UC32	✓

Tabella 7: Insieme dei test di accettazione

3.5 Checklist

Le checklist vengono costantemente aggiornate dai Verificatori durante lo sviluppo del progetto. Gli errori rilevati vengono analizzati e inclusi nella lista, contribuendo a una maggiore efficacia nella correzione e garantendo un livello di qualità sempre più elevato.

3.5.1 Struttura della documentazione

Aspetto	Spiegazione
A capo	Per facilitare la lettura, le frasi devono essere mantenute su una sola riga, evitando interruzioni non necessarie.
Ordine non alfabetico	I nomi nei documenti devono essere elencati in ordine alfabetico per una maggiore chiarezza.
Caption Assente	Ogni tabella e immagine deve includere una didascalia.
Sezioni Fantasma	Le sezioni vuote devono essere rimosse dal documento.
Documento non spezzato	I documenti devono essere creati utilizzando più file <code>.tex</code> collegati con il comando <code>input</code> nella pagina principale.

Tabella 8: Struttura documentazione

3.5.2 Errori ortografici

Aspetto	Spiegazione
Accenti invertiti	Usare l'accento grave al posto dell'acuto e viceversa.
"D" eufonica	La "d" eufonica va inserita solo quando si incontrano due vocali uguali di seguito.
Discordanza soggetto-verbo	Il verbo non concorda correttamente con il soggetto utilizzato.
Errori di battitura	La maggior parte degli errori sono dovuti a distrazione o digitazione errata.
Forma dei verbi	È preferibile l'utilizzo del presente indicativo, altre forme verbali andranno valutate opportunamente.
Forme impersonali	Il soggetto dev'essere sempre esplicito nella frase.

Tabella 9: Errori ortografici

3.5.3 Non conformità con le Norme di Progetto

Aspetto	Spiegazione
Utilizzo scorretto di ":" in grassetto	Evitare l'uso di ":" in grassetto all'interno degli elenchi puntati.
Punteggiatura scorretta negli elenchi	Ogni voce dell'elenco deve terminare con ";", tranne l'ultima che termina con ".".
Minuscolo nei ruoli	I ruoli devono essere scritti con la lettera iniziale maiuscola.
Maiuscole nei titoli	La maiuscola deve essere utilizzata solo per la prima lettera del titolo.

Aspetto	Spiegazione
Mancata segnalazione glossario	Quando viene introdotto un termine del glossario per la prima volta, deve essere segnalato.
Non aggiornare il changelog	È obbligatorio aggiornare il registro delle modifiche dopo ogni verifica.
Versione documento mancante	Quando un documento viene citato, è necessario indicare la versione attuale, se presente; in caso contrario, va specificata la versione corretta.

Tabella 10: Non conformità con Norme di Progetto

3.5.4 Analisi dei Requisiti

Aspetto	Spiegazione
Tracciamento UC - R	Ogni caso d'uso deve essere collegato a uno o più requisiti specifici.
Numerazione UC	La numerazione dei Use Case di errore deve essere al medesimo livello del caso di successo corrispondente.
Requisiti	I requisiti devono essere formulati nella forma “[soggetto] deve [verbo all’infinito]”.
<i>UML_G</i> degli UC	Estensioni, inclusioni e specializzazioni di un caso d'uso devono essere inclusi nello stesso diagramma UML del caso d'uso principale.

Tabella 11: Aspetti di qualità per i casi d'uso

4 Cruscotto di valutazione della qualità

4.1 Varianza dell'impegno orario

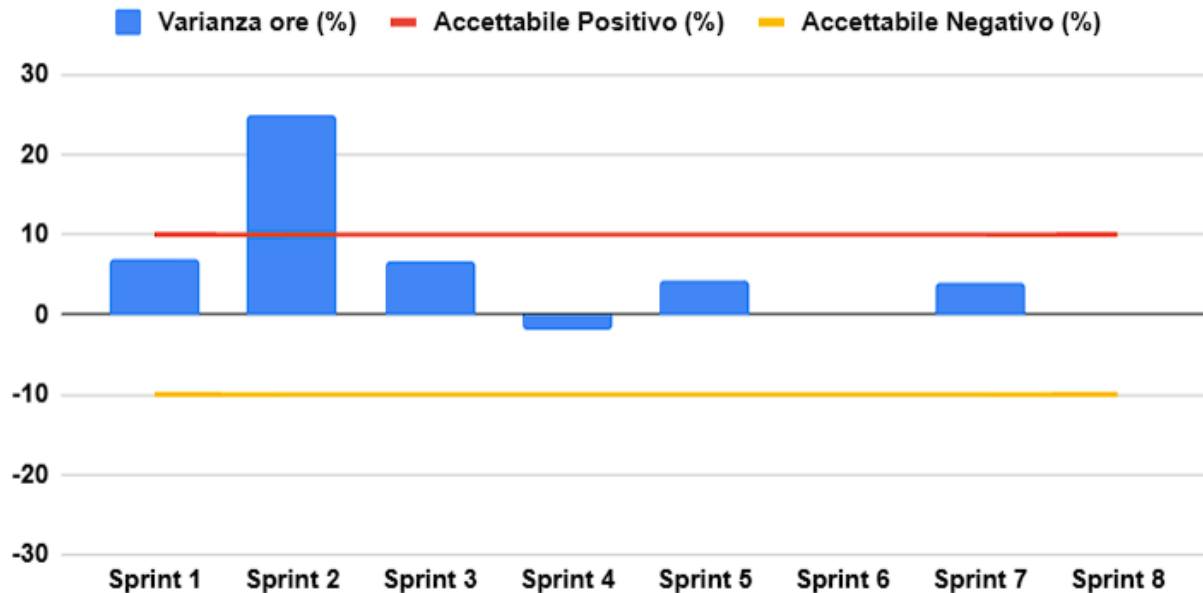


Figura 1: Varianza dell'impegno orario per sprint

Analisi

Analizzando i valori riportati nel grafico, è riscontrabile un miglioramento nel produrre preventivi orari vicini ai rispettivi consuntivi nel corso dell'andamento del progetto. Questo è dovuto alla scarsa esperienza iniziale del team nel preventivare il tempo necessario per lo svolgimento delle attività nella RTB_G , capacità che poi è migliorata di molto nel corso della PB_G . Il picco negativo si è manifestato nella seconda sprint, dove il team ha incontrato difficoltà nel modellare i *Diagrammi dei Casi d'uso* $_G$ e nel redigere i corrispettivi requisiti. Con il progredire delle sprint, il team ha acquisito maggiore esperienza nel preventivare il tempo necessario per lo svolgimento delle attività, riuscendo ad ottenere un valore di varianza oraria nullo, corrispondente al valore preferibile, nella sesta e nell'ottava e ultima sprint della PB .

4.2 Varianza di budget

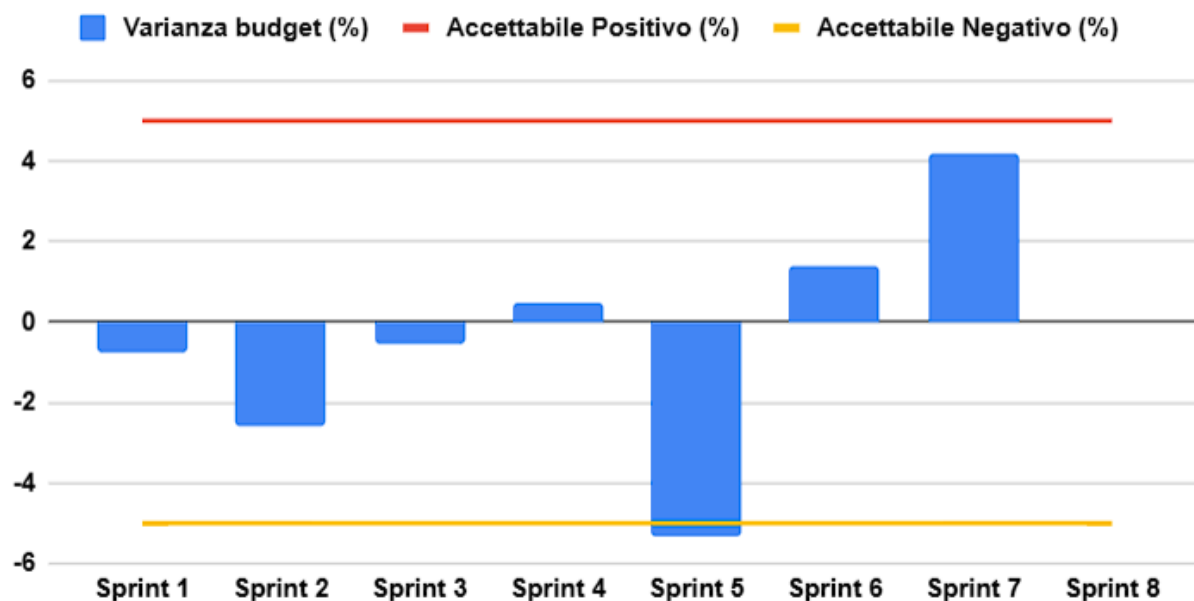


Figura 2: Varianza di budget per sprint

Analisi

Dall'analisi del grafico è immediato osservare che i valori di varianza di budget sono mediamente peggiorati nel passaggio da *RTB* a *PB*, nonostante la varianza oraria sia rimasta quasi costante. In particolare, nella Sprint 5 il team ha avuto difficoltà nell'attività di progettazione, ed allora ha concentrato la maggioranza delle forze su quest'ultima, che rappresenta una delle attività più costose del progetto: ciò ha comportato un aumento dei costi rispetto al preventivo, pur senza un incremento delle ore. Il team ha poi recuperato nella Sprint 6 e nella Sprint 7, dove il difetto è stato l'opposto: il team ha ottenuto un risparmio di budget rispetto al preventivo, poiché gli sforzi per la progettazione nella Sprint 5 hanno dato i loro frutti, consentendo di risparmiare denaro nelle sprint successive. Infine, nella Sprint 8, l'ultima sprint, il team è riuscito a mantenere esattamente il budget preventivato, ottenendo un valore di varianza di budget pari a 0, e ciò dimostra l'acquisizione di esperienza avvenuta nel corso della *PB*.

4.3 Estimate to Complete ed Estimate at Completion

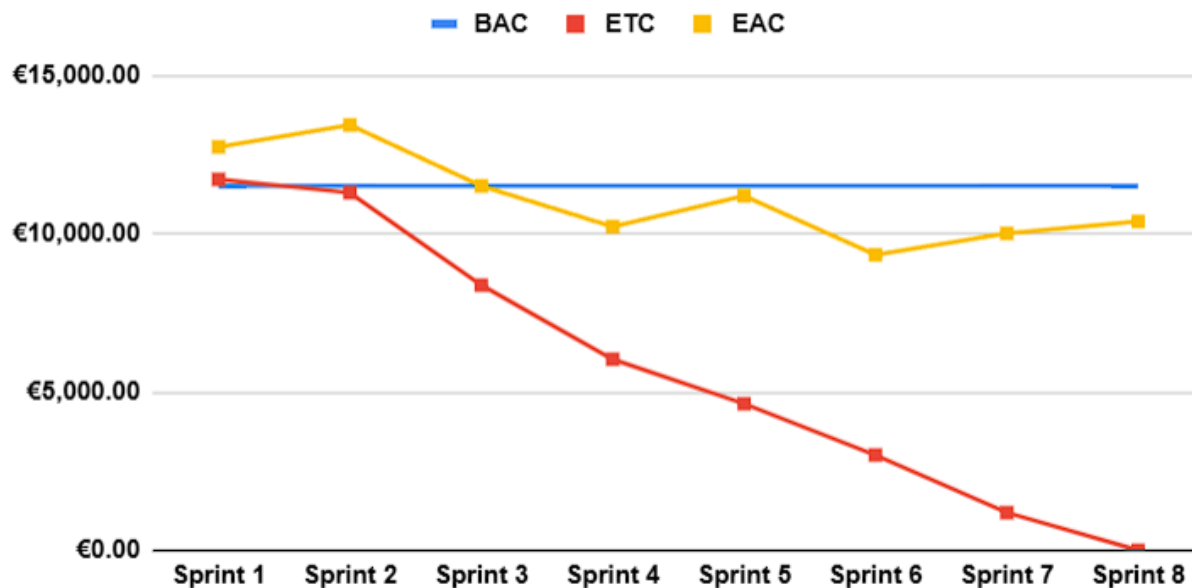


Figura 3: Progressione Estimate to Complete e Estimate at Completion in relazione al Budget at Completion

Analisi

Dall'analisi del grafico è immediato osservare come i valori siano andati a migliorare nel corso delle sprint. In particolare, l'Estimate at Completion dopo la quarta sprint è sempre stato addirittura migliore del preventivo dei costi stilato nella candidatura iniziale. Per il suddetto contrattempo dei casi d'uso, anche in questo grafico si evidenzia l'aumento dei costi tra la prima e la seconda sprint, per poi stabilizzarsi e migliorare nel corso delle sprint successive. La produttività nel corso delle ultime 5 sprint ha consentito al team di risparmiare risorse, ottenendo sempre un valore di Estimate at Completion inferiore al Budget at Completion. Per quanto riguarda l'Estimate to Complete, dopo una leggera difficoltà iniziale, esso procede scendendo regolarmente fino a zero, a dimostrazione dell'efficienza del team nel risparmiare risorse e nel completare il progetto con successo.

4.4 Planned Value, Earned Value e Actual Cost

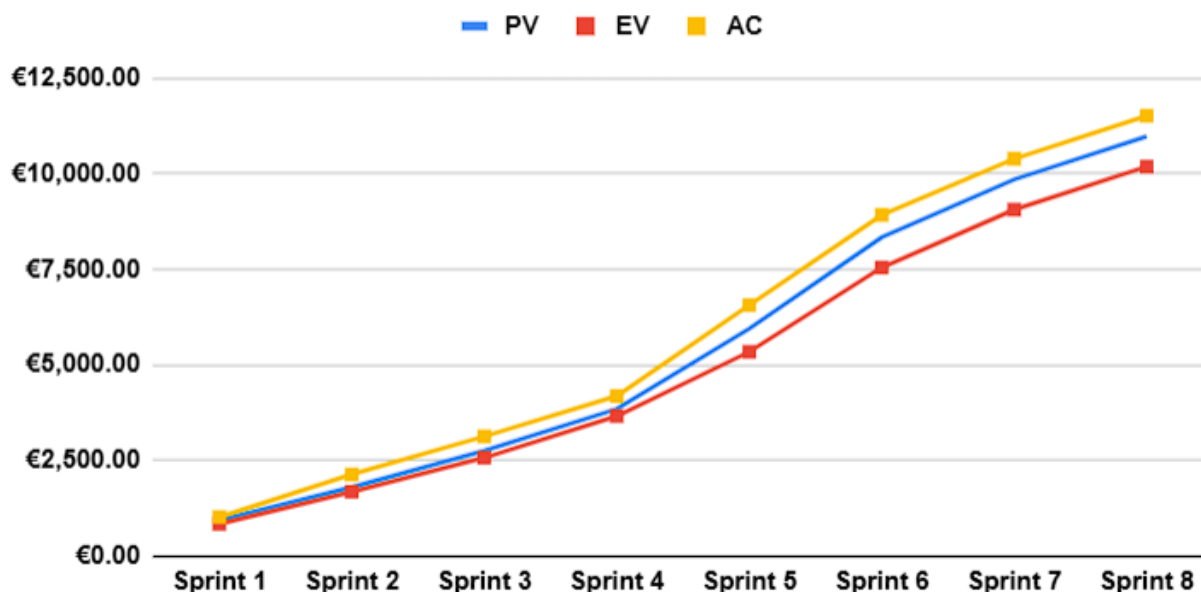


Figura 4: Progressione Planned Value, Earned Value e Actual Cost

Analisi

Analizzando il grafico, si può notare come i tre valori abbiano sempre progredito seguendo una proporzione costante. In particolare, l'Actual Cost è stato sempre superiore al Planned Value, il quale a sua volta è stato sempre superiore all'Earned Value. Questo è dovuto al fatto che, in ciascuna sprint, il team ha impiegato leggermente più tempo del previsto per svolgere leggermente meno attività preventivate.

In particolare, nelle ultime sprint vi è stato un leggero distacco tra l'Earned Value e gli altri due valori, dovuto al fatto che il team, pur rispettando il numero di ore, ha incontrato qualche difficoltà nelle attività di progettazione e programmazione.

Tuttavia, il grafico fa riferimento unicamente ai preventivi e consuntivi delle singole sprint, mentre il preventivo totale degli obiettivi da raggiungere concordato a inizio progetto è stato regolarmente rispettato al termine di quest'ultimo.

4.5 Schedule Variance e Cost Variance

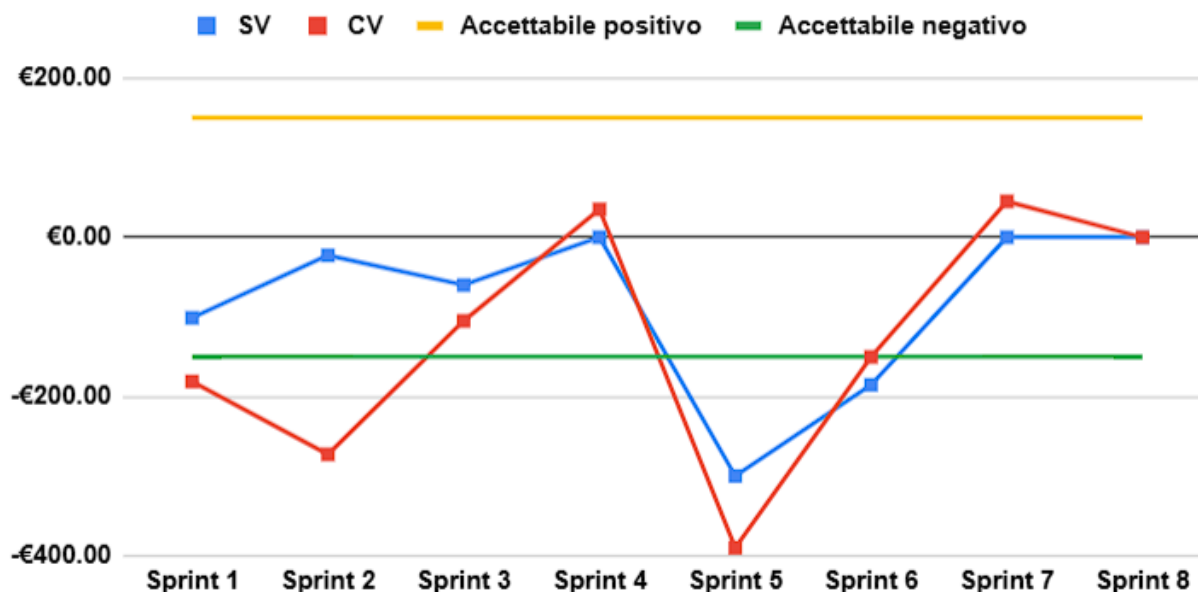


Figura 5: Progressione Schedule Variance e Cost Variance

Analisi

Dal grafico, risulta evidente come il team abbia avuto difficoltà nel rispettare i costi preventivati, seppur abbia mediamente rispettato il numero di ore. La Schedule Variance e la Cost Variance sono infatti risultate sempre negative, tranne nelle ultime sprint dei periodi *RTB* e *PB*, nelle quali si nota un miglioramento significativo. I risultati peggiori si registrano nella seconda sprint e nella quinta sprint per i suddetti motivi, legati rispettivamente all'analisi dei requisiti e alla progettazione: infatti, queste ultime rappresentano due tra le attività più costose del progetto, e quindi, quando il team si è concentrato su di esse, i costi sono saliti. Questo è il motivo per cui, seppur il numero di ore non sia mai stato largamente superato, la distribuzione delle stesse verso attività costose ha comportato una marcata varianza dei costi. Tuttavia, fortunatamente la varianza è stata bilanciata nelle sprint adiacenti al termine dei periodi *RTB* e *PB*, poiché le ore in quei periodi sono state dedicate ad attività come la codifica e la verifica, che sono meno costose.

4.6 Schedule Performance Index e Cost Performance Index

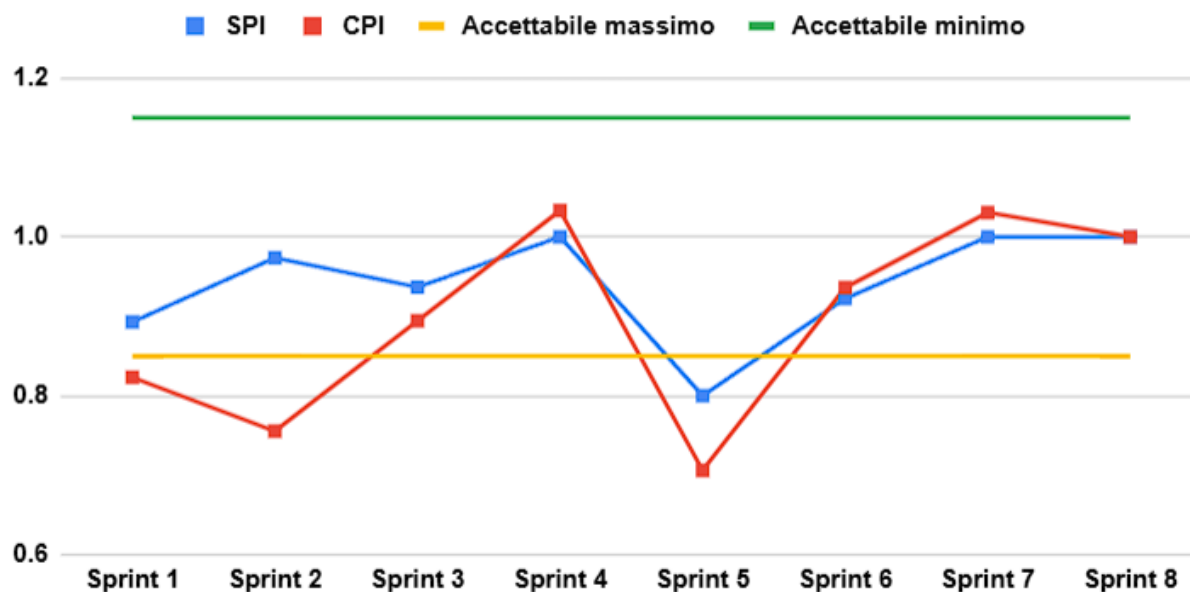


Figura 6: Progressione Schedule Performance Index e Cost Performance Index

Analisi

Similmente a quanto riscontrabile dall'analisi del grafico di Schedule e Cost Variance, anche il valore del Cost Performance Index è sceso sotto la soglia accettabile nella sprint 2 e nella sprint 5, per poi recuperare nelle sprint terminali dei periodi *RTB* e *PB*: ciò è dovuto, come suddetto, alle attività di analisi dei requisiti e di progettazione. Nella quarta sprint e nella settima sprint, invece, il team ha addirittura raggiunto un valore di Cost Performance Index superiore a 1, indicando che sono state spese meno risorse rispetto a quanto preventivato per lo svolgimento delle attività meno costose previste per le sprint finali.

Il valore dello Schedule Performance Index è invece sempre stato inferiore ad 1, addirittura scendendo sotto la soglia accettabile nella sprint 5, a indicare la difficoltà che il team ha avuto nel preventivare le ore in tale sprint, in cui, oltre alle difficoltà di progettazione, è da segnalare che metà gruppo è stato impegnato nello studio per il secondo appello di Ingegneria del Software. Nelle sprint 4, 7 ed 8, invece, lo Schedule Performance Index è risultato pari a 1, indicando che, come suddetto, verso la fine dei periodi *RTB* e *PB*, il team è riuscito a svolgere con successo tutte le attività preventivate.

4.7 Misure di mitigazione insufficienti

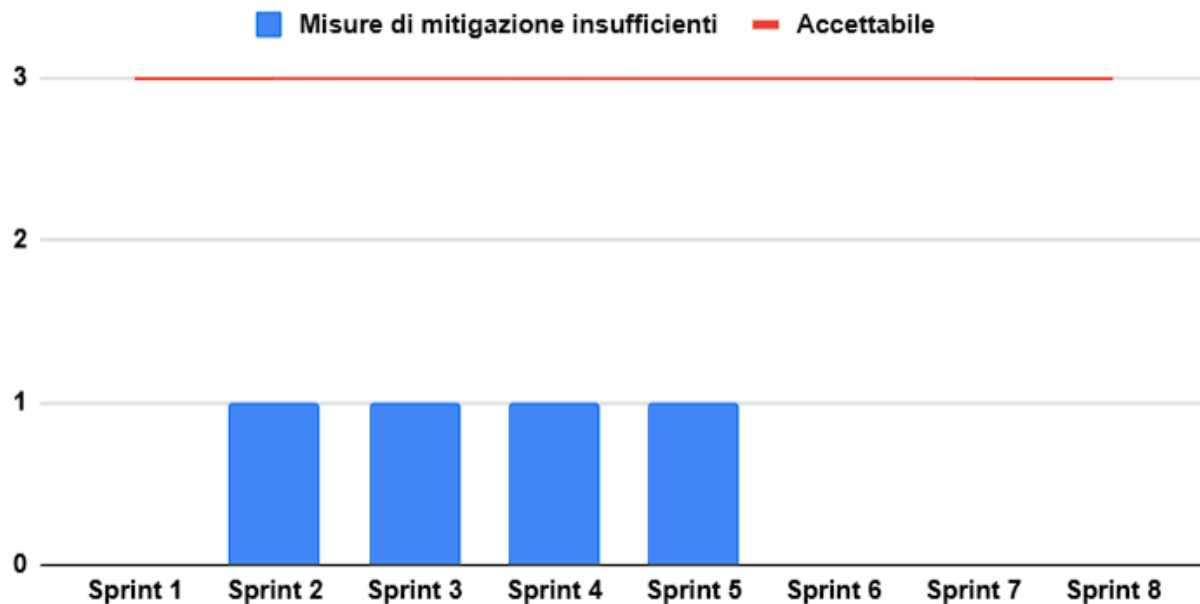


Figura 7: Progressione occorrenza di rischi con misure mitigative insufficienti

Analisi

Durante il progetto, il team ha riscontrato difficoltà nell'applicare alcune misure di mitigazione per i rischi individuati e riportati nel *Piano di Progetto_G*.

- In particolare, per gestire i rischi legati alla comunicazione interna nella seconda sprint, è stato necessario che tutti i membri del team si accordassero sia sulla sintassi che sulla semantica nella modellazione dei casi d'uso;
- Successivamente, per affrontare i rischi di non conformità rispetto agli impegni dichiarati nella terza sprint, si è dovuto considerare l'impatto delle vacanze natalizie e dello studio per la sessione d'esami;
- Nella quarta sprint, per gestire i rischi legati alla complessità delle tecnologie, è emersa la necessità di un aiuto reciproco nella programmazione. La spartizione iniziale dei compiti prevedeva esploratori solitari per ciascuna tecnologia e una bassa condivisione delle conoscenze. Questo ha reso difficile lavorare in gruppo e chiedere aiuto ai compagni al momento dell'*implementazione_G* del *PoC_G*;
- Infine, nella quinta sprint, il team ha dovuto affrontare i rischi legati agli impegni accademici, poiché più di metà del gruppo ha dovuto studiare per il secondo appello di Ingegneria del Software. Questo ha comportato un rallentamento delle attività, che è stato mitigato con una redistribuzione dei compiti e con un maggiore supporto reciproco.

Fortunatamente, si può notare come il team abbia saputo reagire in modo efficace a queste difficoltà, riuscendo a presentare misure sufficienti per ciascuna di esse nelle ultime 3 sprint.

4.8 Rischi inattesi

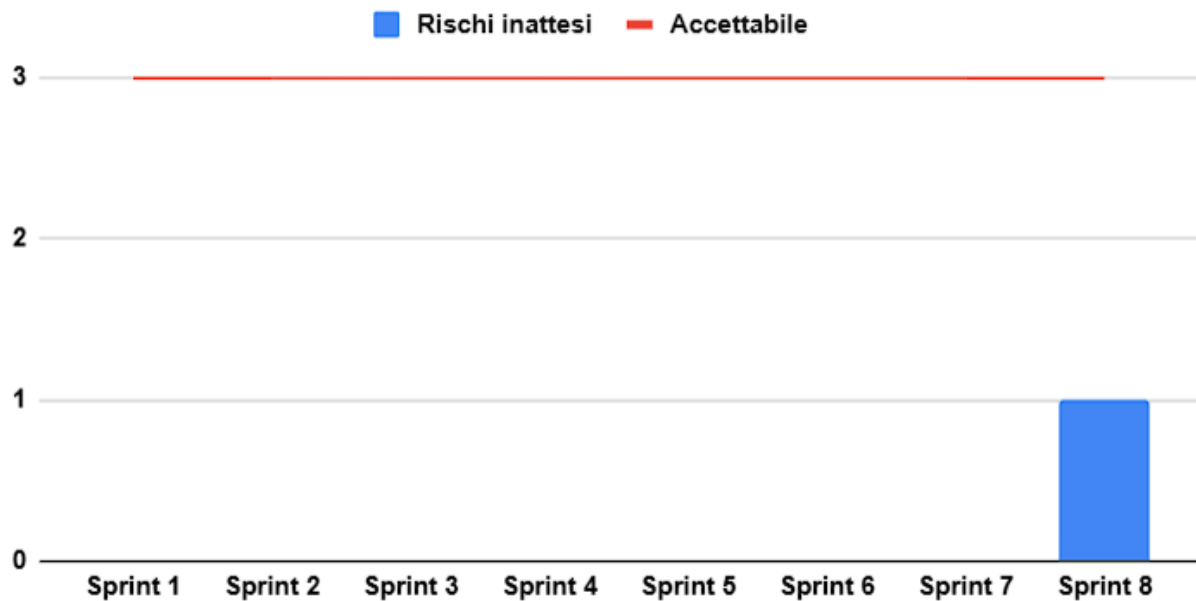


Figura 8: Progressione occorrenza di rischi inattesi

Analisi

Come osservabile dai valori riportati dal grafico, l'ampia analisi dei rischi effettuata a inizio progetto è stata soddisfacente. L'unica problematica primaria riconoscibile come rischio non già presente nella sezione dedicata del *Piano di progetto* è stata riscontrata solo al termine dei lavori: si tratta dell'assenza di un professore, in particolare del professor Cardin, nell'ultima settimana di marzo, che ha comportato il posticipo del termine del progetto. Fortunatamente, essendo questa problematica emersa solamente nella sprint finale, non ha causato ritardi su attività precedenti alla revisione *Product Baseline_G*, però potrebbe causare ritardi in caso emergano errori a valle della revisione. Inoltre, questa situazione ha costretto un componente del gruppo a posticipare l'inizio del proprio tirocinio, dimostrando l'impatto che i rischi inattesi possono avere sulla pianificazione personale dei membri del team.

Il team si è trovato inizialmente impreparato, ma ha poi saputo reagire in modo efficace, riuscendo a completare la documentazione con maggiore calma e a terminare le attività di progetto senza incidere significativamente sul costo orario.

4.9 Requisiti soddisfatti

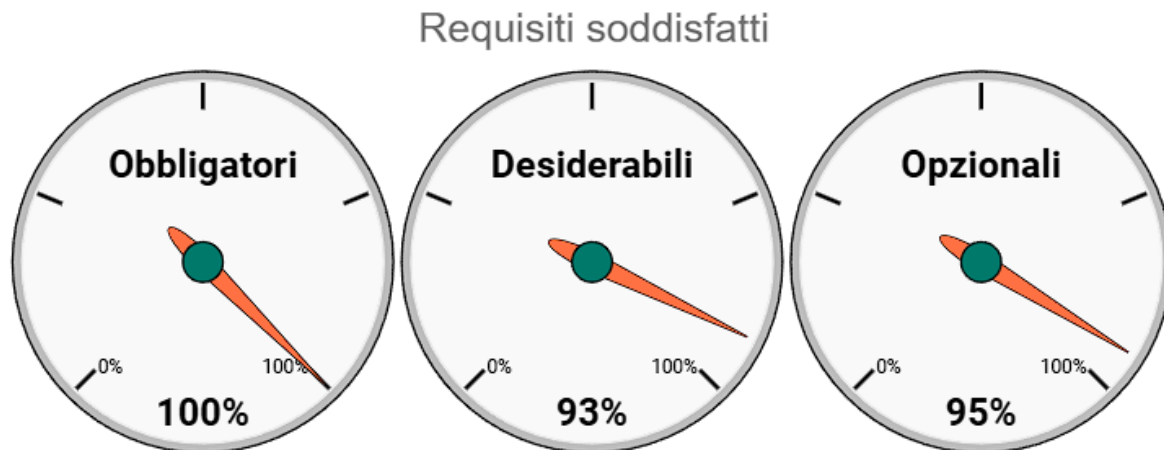


Figura 9: Requisiti obbligatori, desiderabili e opzionali soddisfatti

Analisi

L' MVP_G sviluppato nella *Product Baseline* ha soddisfatto tutti i requisiti obbligatori stabiliti assieme al *proponente_G*. Invece, non sono stati soddisfatti un requisito desiderabile e un requisito opzionale, entrambi in accordo con il proponente: il primo, relativo alla visualizzazione di un segnale di inizio dello storico dei messaggi, è stato ritenuto non necessario e troppo facilmente confondibile con un messaggio di errore, mentre il secondo, relativo alla funzionalità di selezione di alcune domande proposte per iniziare la conversazione, si è ritrovato in contrasto con la funzionalità dello storico, che ha reso concettualmente complesso capire quale fosse l'”inizio della conversazione”, ed allora è stato scartato.

4.10 Indice di Gulpease

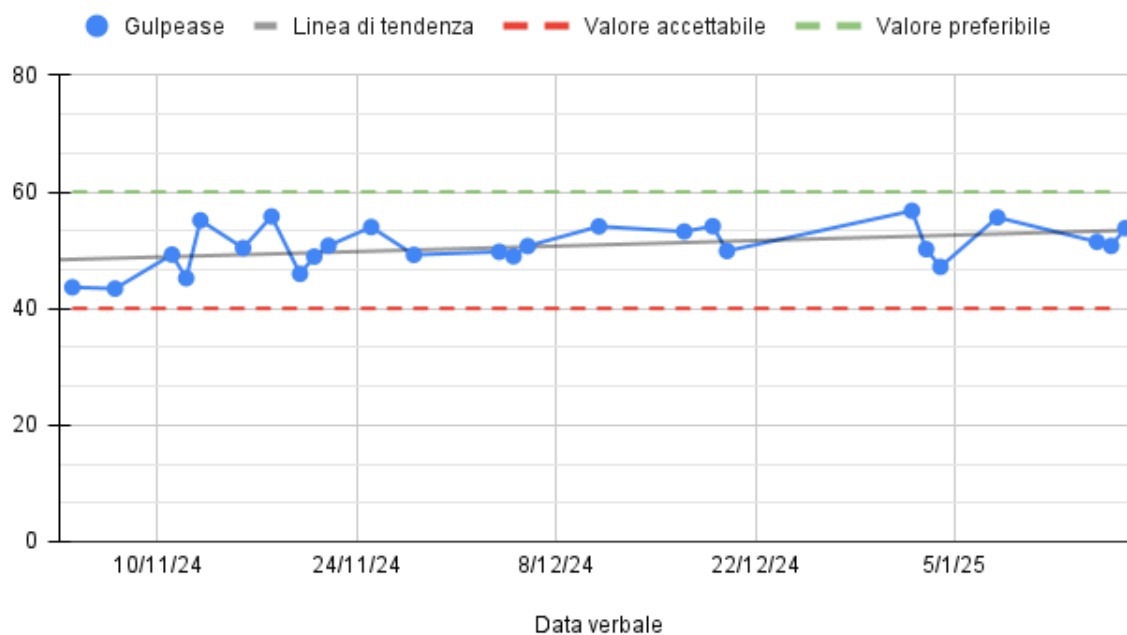


Figura 10: Andamento indice di Gulpease nei verbali - RTB

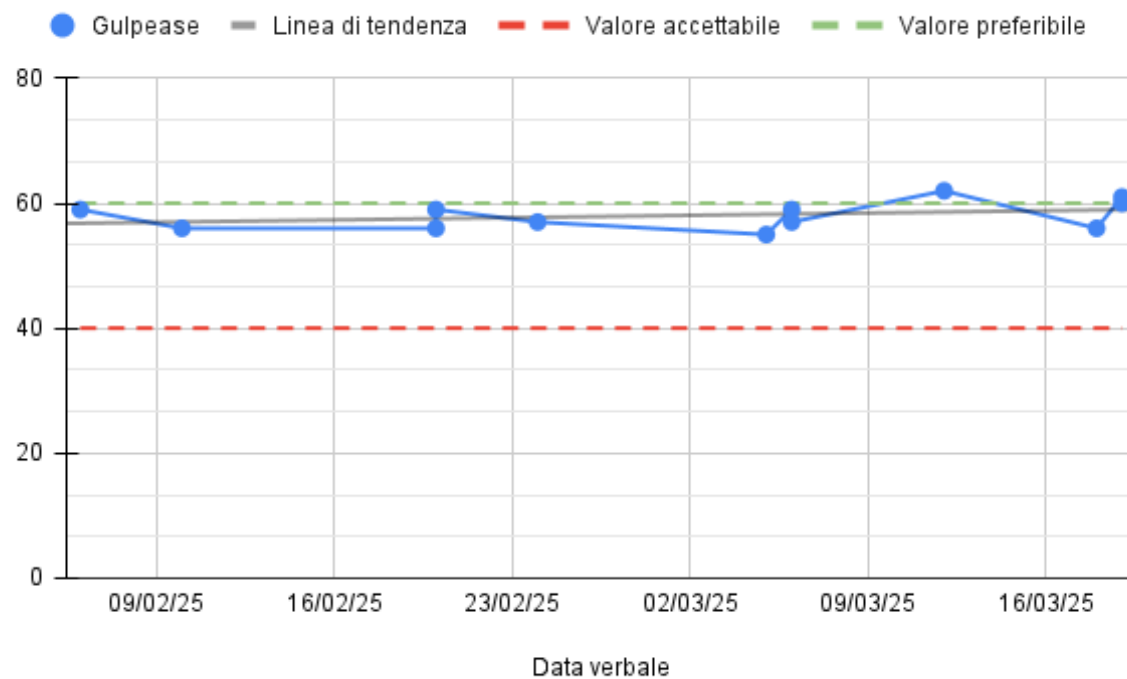


Figura 11: Andamento indice di Gulpease nei verbali - PB



Figura 12: Indice di Gulpease Norme di Progetto



Figura 13: Indice di Gulpease Piano di Progetto



Figura 14: Indice di Gulpease Analisi dei Requisiti



Figura 15: Indice di Gulpease Piano di Qualifica

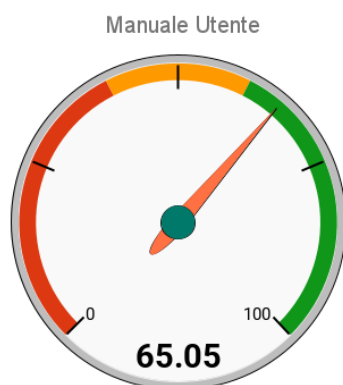


Figura 16: Indice di Gulpease Manuale Utente

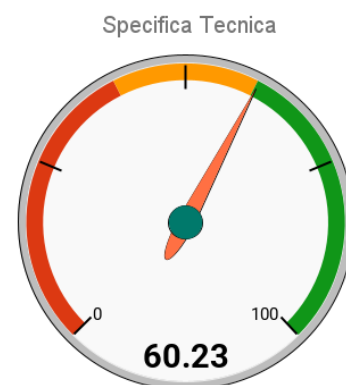


Figura 17: Indice di Gulpease Specifica Tecnica

Analisi

Questi valori mostrano che la documentazione è generalmente accessibile e comprensibile, dato che tutti i documenti superano la soglia accettabile di 40 nell'*indice di Gulpease_G*, con la maggior parte dei valori che si avvicinano al valore preferibile di 60. In particolare, è utile notare come, nel grafico relativo all'andamento dell'indice di Gulpease dei verbali durante la *RTB_G*, la linea di tendenza sia crescente, il che mostra un generale miglioramento da parte del gruppo nella qualità della documentazione prodotta. Questa tendenza è stata confermata nella *Product Baseline_G*, durante la quale il valore dell'indice per i verbali ha avuto una crescita meno evidente, ma si è stabilizzato attorno al valore preferibile di 60.

4.11 Code Coverage

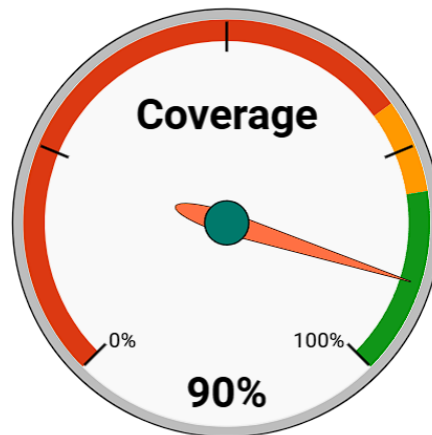


Figura 18: Code Coverage

Analisi

La *Code Coverage_G* è un indicatore che misura la percentuale di righe di codice sorgente coperte dai *test_G*. Nel tentativo di migliorare questo indicatore, il team ha scritto *test di unità_G* e *test di integrazione_G* per ogni classe del *backend_G* e per ogni componente del *frontend_G*, raggiungendo un valore di coverage pari al 90% per entrambi. Questo è un buon risultato, che dimostra l'efficacia dei test effettuati, e soprattutto che soddisfa la soglia minima richiesta dal *proponente AzzurroDigitale_G*, cioè 70/80% di copertura, che il team è stato in grado anche di superare.