



SWEvenTeam

E-Mail

sweventeam@outlook.it

PIANO DI QUALIFICA

Informazioni documento

Versione	2.0.0
Redazione	Alessio Barraco Alessandro Damiani Yuri Lunardon Alba Hui Larrosa Serrano Matteo Mazzotti Valentina Schivo
Verifica	Alessio Barraco Alessandro Damiani Yuri Lunardon Alba Hui Larrosa Serrano Matteo Mazzotti Valentina Schivo Alessio Turetta
Approvazione	Valentina Schivo

Storia del documento

Versione	Data	Autori	Verificatori	Descrizione
2.0.0	2025-06-10	Valentina Schivo	-	Approvazione documento
1.7.0	2025-06-10	Alba Hui Larrosa Serrano	Yuri Lunardon	Controllo finale
1.6.0	2025-06-09	Matteo Mazzotti	Alessio Barraco	Aggiornamento cruscotto
1.5.0	2025-05-28	Yuri Lunardon	Valentina Schivo	Aggiornamento test
1.4.0	2025-05-24	Valentina Schivo	Alessandro Damiani	Aggiornamento test di sistema
1.3.0	2025-05-20	Alessandro Damiani	Alessio Barraco	Redazione test di integrazione
1.2.0	2025-05-10	Valentina Schivo	Alba Hui Larrosa Serrano	Redazione test di unità
1.1.0	2025-04-13	Alessio Barraco, Alessandro Damiani	Yuri Lunardon	Modifiche a test di sistema
1.0.0	2025-03-21	Valentina Schivo	-	Approvazione per RTB
0.5.0	2025-03-21	Alessio Barraco	Yuri Lunardon, Alessio Turetta	Controllo finale
0.4.4	2025-03-20	Yuri Lunardon	Matteo Mazzotti	Migliorie finali
0.4.3	2025-03-19	Yuri Lunardon	Alessio Turetta	Aggiornamento Cruscotto
0.4.2	2025-03-17	Alessio Barraco	Alessandro Damiani	Aggiornamento documento
0.4.1	2025-02-24	Alessandro Damiani	Alessio Barraco	Aggiornamento sezione Specifica di test
0.4.0	2025-02-20	Valentina Schivo	Yuri Lunardon	Scrittura sezione Cruscotto di controllo della qualità
0.3.0	2025-01-29	Alessio Barraco	Yuri Lunardon	Inizio scrittura sezione Specifica di test
0.2.2	2025-01-28	Valentina Schivo	Alessandro Damiani	Miglioramento sottosezione Qualità di prodotto
0.2.1	2025-01-27	Valentina Schivo	Yuri Lunardon	Miglioramento sottosezione Qualità di processo
0.2.0	2025-01-22	Alessio Barraco	Valentina Schivo	Scrittura sottosezione Qualità di prodotto
0.1.0	2025-01-15	Yuri Lunardon	Valentina Schivo	Scrittura sottosezione Qualità di processo
0.0.2	2025-01-05	Alessio Barraco	Valentina Schivo	Introduzione documento
0.0.1	2025-01-03	Alessandro Damiani	Alessio Barraco	Definizione struttura

Indice

1	Introduzione	5
1.1	Scopo del documento	5
1.2	Glossario	5
1.3	Riferimenti	5
1.3.1	Riferimenti normativi	5
1.3.2	Riferimenti informativi	5
2	Obiettivi metrici di qualità	6
2.1	Qualità di processo	6
2.1.1	Processi primari	6
2.1.2	Processi di supporto	7
2.1.3	Processi organizzativi	8
2.2	Qualità di prodotto	8
3	Specifica dei test	9
3.1	Denominazione dei test	10
3.2	Test di unità	10
3.3	Test di integrazione	17
3.4	Test di sistema	19
3.5	Test di accettazione	23
4	Cruscotto di controllo della qualità	25
4.1	Budget Variance	25
4.2	Planned Value e Earned Value	26
4.3	Cost Variance e Schedule Variance	27
4.4	Cost Performance Index	28
4.5	Estimate to Complete, Estimate at Completion e Actual Cost	29
4.6	Indice di Gulpease	30
4.7	Rischi	31
4.8	Percentuale di requisiti soddisfatti	32
4.9	Coverage	33

Elenco delle Figure

1	Proiezione dei valori di Budget Variance	25
2	Proiezione dei valori di Planned Value e Earned Value	26
3	Proiezione dei valori di Cost Variance e Schedule Variance	27
4	Proiezione dei valori di Cost Performance Index	28
5	Proiezione dei valori di Estimate to Complete, Estimate at Completion e Actual Cost	29
6	Proiezione dei valori di Gulpease	30
7	Proiezione dei rischi attesi e rischi occorsi	31
8	Proiezione della percentuale di requisiti soddisfatti	32
9	Proiezione dei valori percentuali di code coverage dei test	33

Elenco delle Tabelle

1	Metriche e valori per il processo di fornitura	6
2	Metriche e valori per il processo di sviluppo	7
3	Metriche e valori per il processo di documentazione	7
4	Metriche e valori per il processo di verifica	7
5	Metriche e valori per il processo di gestione della qualità	7
6	Metriche e valori per il processo di gestione di processi	8
7	Metriche e valori per la funzionalità del prodotto	8
8	Metriche e valori per l'affidabilità del prodotto	8
9	Metriche e valori per la manutenibilità del prodotto	9
10	Metriche e valori per l'usabilità del prodotto	9
11	Metriche e valori per l'efficienza del prodotto	9
12	Test di unità	17
13	Test di integrazione	18
14	Test di sistema	23
15	Test di accettazione	24

1 Introduzione

1.1 Scopo del documento

Il presente documento ha lo scopo di stabilire metriche e strategie per assicurare la qualità_G del progetto_G, definendo linee guida per la verifica e la validazione. Saranno riportati i risultati delle misure sulla qualità di processi e prodotti, permettendo la valutazione e l'accettazione degli stessi. Data la natura evolutiva del progetto e del documento stesso, i contenuti saranno ampliati e modificati nel tempo. Questo approccio incrementale è necessario per apportare frequenti miglioramenti a metodologie e regole, con l'obiettivo di massimizzare la qualità del prodotto finale. I miglioramenti saranno effettuati sulla base di evidenze concrete ed inoltre, le verifiche effettuate sul prodotto consentiranno di individuare e correggere tempestivamente eventuali problematiche.

1.2 Glossario

Per mantenere la consistenza nell'uso dei termini specifici al progetto, è presente una lista di definizioni dei termini specifici del dominio d'uso su un glossario. Il Glossario risulterà utile fornendo delle definizioni chiare e approvate da tutto il gruppo. La presenza di un termine all'interno del Glossario viene indicata con questo stile_G.

1.3 Riferimenti

1.3.1 Riferimenti normativi

- **Norme di Progetto v2.0.0**
https://sweventeam17.github.io/pdf/3-PB/Documenti%20interni/Norme_di_Progetto_v2.0.0.pdf
Ultima consultazione: 2025-06-09
- **Capitolato d'appalto C1: ArtificialQI**
<https://www.math.unipd.it/tullio/IS-1/2024/Progetto/C1.pdf>
Ultima consultazione: 2025-06-09

1.3.2 Riferimenti informativi

- **Glossario v2.0.0**
<https://sweventeam17.github.io/glossario.html>
Ultima consultazione: 2025-06-09
- **Qualità del software**
<https://www.math.unipd.it/tullio/IS-1/2024/Dispense/T07.pdf>
Ultima consultazione: 2025-06-09
- **Qualità del processo**
<https://www.math.unipd.it/tullio/IS-1/2024/Dispense/T08.pdf>
Ultima consultazione: 2025-06-09
- **Verifica e Validazione - introduzione**
<https://www.math.unipd.it/tullio/IS-1/2024/Dispense/T09.pdf>
Ultima consultazione: 2025-06-09
- **Verifica e Validazione - analisi statica**
<https://www.math.unipd.it/tullio/IS-1/2024/Dispense/T10.pdf>
Ultima consultazione: 2025-06-09
- **Verifica e Validazione - analisi dinamica**
<https://www.math.unipd.it/tullio/IS-1/2024/Dispense/T11.pdf>
Ultima consultazione: 2025-06-09

2 Obiettivi metrici di qualità

Per misurare l'efficacia_G dei processi e la qualità del software, vengono impiegate metriche il cui quadro di riferimento è definito nelle Norme di Progetto_G. In questa sezione sono riportati i criteri che le metriche devono soddisfare per essere considerate accettabili o pienamente soddisfatte.

2.1 Qualità di processo

La qualità del processo_G rappresenta un requisito_G fondamentale nello sviluppo software. Per garantire un prodotto finale di alta qualità, è indispensabile adottare e applicare una best practice_G ben definita.

2.1.1 Processi primari

Fornitura

Codice	Nome	Descrizione	Valore accettabile	Valore ottimale
MPC-01	Budget Variance	Indica la percentuale di variazione del budget tra <u>preventivo</u> _G e <u>consuntivo</u> _G in un periodo	±15%	0%
MPC-02	Earned Value	Valore del lavoro effettivamente svolto alla fine di un periodo	≥ 0	≤ <u>EAC</u> _G
MPC-03	Actual Cost	Misura i costi effettivamente sostenuti dall'inizio del progetto fino al periodo attuale	≥ 0	≤ EAC
MPC-04	Planned Value	Rappresenta il totale dei costi pianificati per periodo e viene calcolata prima che esso inizi	≥ 0	= <u>BAC</u> _G
MPC-05	Cost Variance	Rappresenta lo scostamento dai costi pianificati	±200€	0€
MPC-06	Schedule Variance	Indica di quanto si è in anticipo o in ritardo con le <u>attività</u> _G pianificate	±200€	0€
MPC-07	Cost Performance Index	Misura il rapporto tra il valore del lavoro effettivamente svolto ed il costo reale del lavoro fino al periodo di riferimento	1 ± 0.1	1
MPC-08	Estimate to Complete	Rappresenta il costo totale ancora da sostenere per il completamento del progetto	≥ 0	≤ EAC
MPC-09	<u>Estimate at Completion</u> _G	Rappresenta il costo totale alla fine del progetto in base all'andamento attuale	±10% rispetto al BAC	= BAC

Tabella 1: Metriche e valori per il processo di fornitura

Sviluppo

Codice	Nome	Descrizione	Valore accettabile	Valore ottimale
MPC-10	Variazione dei requisiti	Misura la variazione dei requisiti durante il progetto	≤ 4	0

Tabella 2: Metriche e valori per il processo di sviluppo

2.1.2 Processi di supporto
Documentazione

Codice	Nome	Descrizione	Valore accettabile	Valore ottimale
MPC-11	Indice Gulepase	Stima la comprensibilità del testo	$\geq 40\%$	$\geq 60\%$
MPC-12	Correttezza ortografica	Misura la presenza di errori ortografici nel testo	0 errori	0 errori

Tabella 3: Metriche e valori per il processo di documentazione

Verifica

Codice	Nome	Descrizione	Valore accettabile	Valore ottimale
MPC-13	<u>Code Coverage_G</u>	Stima la percentuale di codice attraversato dai <u>test_G</u> rispetto al totale	$\geq 80\%$	100%
MPC-14	Percentuale di test case superati	Percentuale di casi test superati	$\geq 80\%$	100%
MPC-15	Percentuale di test case falliti	Percentuale di casi test falliti	$\leq 20\%$	$\leq 5\%$

Tabella 4: Metriche e valori per il processo di verifica

Gestione della qualità

Codice	Nome	Descrizione	Valore accettabile	Valore ottimale
MPC-16	Percentuale di metriche soddisfatte	Misura che valuta quante metriche sono state effettivamente adottate e soddisfatte, rispetto a quelle definite	$\geq 60\%$	100%

Tabella 5: Metriche e valori per il processo di gestione della qualità

2.1.3 Processi organizzativi

Gestione dei processi

Codice	Nome	Descrizione	Valore accettabile	Valore ottimale
MPC-17	Rischi inattesi	Misura il numero di rischi non previsti nel corso del progetto	≤ 4	0

Tabella 6: Metriche e valori per il processo di gestione di processi

2.2 Qualità di prodotto

La qualità di prodotto si riferisce al grado in cui un prodotto soddisfa i requisiti, le aspettative e le esigenze degli utenti o delle parti interessate, garantendo al contempo che il prodotto sia affidabile, funzionale e conforme agli standard previsti.

Funzionalità

Codice	Nome	Descrizione	Valore accettabile	Valore ottimale
MPD-01	Copertura requisiti obbligatori	Indica la percentuale dei requisiti obbligatori soddisfatti	100%	100%
MPD-02	Copertura requisiti desiderabili	Indica la percentuale dei requisiti desiderabili soddisfatti	$\geq 0\%$	$\geq 50\%$
MPD-03	Copertura requisiti opzionali	Indica la percentuale dei requisiti opzionali soddisfatti	$\geq 0\%$	$\geq 50\%$

Tabella 7: Metriche e valori per la funzionalità del prodotto

Affidabilità

Codice	Nome	Descrizione	Valore accettabile	Valore ottimale
MPD-04	Code coverage	Stima la percentuale di codice attraversato dai test rispetto al totale	$\geq 80\%$	100%
MPD-05	<u>Branch</u> _G coverage	Indica la percentuale dei rami decisionali del codice coperti dai test	$\geq 50\%$	$\geq 80\%$
MPD-06	<u>Statement</u> coverage _G	Indica la percentuale degli statement del codice coperti dai test	$\geq 50\%$	$\geq 80\%$
MPD-07	Densità errori	Misura la frequenza con cui si verificano guasti o difetti del prodotto	$\leq 10\%$	0%

Tabella 8: Metriche e valori per l'affidabilità del prodotto

Manutenibilità

Codice	Nome	Descrizione	Valore accettabile	Valore ottimale
MPD-08	Complessità ciclomatica	Misura la complessità del codice in termini di percorsi logici	1-10	11-20
MPD-09	Coefficiente di accoppiamento	Misura il livello di dipendenza tra moduli, classi o funzioni	$\leq 30\%$	$\leq 15\%$
MPD-10	Code smell	Indica la potenziale presenza di problemi nel codice sorgente	0	0

Tabella 9: Metriche e valori per la manutenibilità del prodotto

Usabilità

Codice	Nome	Descrizione	Valore accettabile	Valore ottimale
MPD-11	Facilità di utilizzo	Misura l'usabilità di un <u>sistema</u> _G software	≤ 6 click	≤ 3 click
MPD-12	Tempo di apprendimento	Misura il tempo massimo richiesto per apprendere l'utilizzo del prodotto	≤ 15 minuti	≤ 10 minuti

Tabella 10: Metriche e valori per l'usabilità del prodotto

Efficienza

Codice	Nome	Descrizione	Valore accettabile	Valore ottimale
MPD-13	Utilizzo delle risorse	Misura quanto efficientemente vengono utilizzate le risorse disponibili	$\geq 70\%$	100%
MPD-14	Tempo medio di <u>risposta</u> _G	Misura quanto è efficiente e reattivo un sistema software	≤ 8 secondi	≤ 5 secondi

Tabella 11: Metriche e valori per l'efficienza del prodotto

3 Specifica dei test

L'esecuzione dei test è un passaggio fondamentale per verificare che il prodotto nel suo insieme rispetti pienamente tutti i requisiti e i vincoli definiti nel documento di Analisi dei Requisiti_G. Questa sezione, dunque, elenca i test eseguiti sul prodotto per dimostrare che tali vincoli siano rispettati.

I test si suddividono in:

- **Test di unità:** eseguiti sulle singole unità software autonome e vengono implementati principalmente durante la progettazione;
- **Test di integrazione:** successivi ai test di unità_G, servono a verificare che le diverse unità software interagiscano correttamente;
- **Test di sistema:** controllano il corretto funzionamento dell'intero sistema, assicurando che tutti i requisiti obbligatori siano rispettati;
- **Test di accettazione:** servono a verificare che il prodotto finale soddisfi i requisiti richiesti e i bisogni dell'utente.

3.1 Denominazione dei test

Ogni test è associato ad un codice univoco composto nel seguente modo:

$$T[\text{Tipo}]-[\text{Numero}]$$

Dove con Tipo si indica la tipologia di test:

- **U**: unità;
- **I**: integrazione;
- **S**: sistema;
- **A**: accettazione.

Vengono poi assegnate le seguenti abbreviazioni per definire lo stato di un test:

- **S**: superato;
- **NS**: non superato;
- **NI**: non implementato.

3.2 Test di unità

Codice	Descrizione	Stato del test
TU-01	Verificare che la classe LLMController, metodo <code>__init__</code> , iniziizzi correttamente l'istanza se il microservizio risponde con successo.	S
TU-02	Verificare che la classe LLMController, metodo <code>__init__</code> , sollevi <code>ConnectionError</code> se la richiesta fallisce.	S
TU-03	Verificare che la classe LLMController, metodo <code>get_answer</code> , restituisca correttamente una risposta unendo lo stream <code>LLMG</code> .	S
TU-04	Verificare che la classe LLMController, metodo <code>get_semantic_evaluation</code> , restituisca 100 in caso di risposta esattamente uguale.	S
TU-05	Verificare che la classe LLMController, metodo <code>get_semantic_evaluation_similarity</code> , calcoli correttamente la similarità con <code>SentenceTransformer</code> .	S
TU-06	Verificare che la classe LLMController, metodo <code>get_external_evaluation_env_missing</code> , sollevi <code>FileNotFoundError</code> se manca <code>.env</code> .	S
TU-07	Verificare che la classe LLMController, metodo <code>get_external_evaluation_no_key</code> , restituisca un messaggio di errore se manca la chiave <code>APIG</code> .	S
TU-08	Verificare che la classe LLMController, metodo <code>get_external_evaluation_percentage_found</code> , estragga correttamente la percentuale dalla risposta LLM.	S
TU-09	Verificare che la classe LLMController, metodo <code>get_external_evaluation_percentage_not_found</code> , ritorni un messaggio se non è presente alcuna percentuale.	S

Codice	Descrizione	Stato del test
TU-10	Verificare che la classe LLMController, metodo <code>get_external_evaluation_generic_exception</code> , gestisca eccezioni generiche restituendo un messaggio di errore.	S
TU-11	Verificare che la classe LLMController, metodo <code>get_external_evaluation_google_api_exception</code> , gestisca errori di tipo <code>ImportError</code> ritornando un messaggio.	S
TU-12	Verificare che la classe LLMController, metodo <code>get_external_evaluation_unsupported</code> , ritorni "Unsupported provider" per provider non supportati.	S
TU-13	Verificare che la classe <code>AbstractService</code> , metodo <code>create</code> , chiami correttamente il <code>repository_G</code> e restituisca l'oggetto con ID.	S
TU-14	Verificare che la classe <code>AbstractService</code> , metodo <code>read</code> , recuperi correttamente un oggetto dato il suo ID.	S
TU-15	Verificare che la classe <code>AbstractService</code> , metodo <code>read_all</code> , restituisca correttamente tutti gli oggetti presenti nel repository.	S
TU-16	Verificare che la classe <code>AbstractService</code> , metodo <code>update</code> , aggiorni un oggetto con i dati specificati e lo restituisca.	S
TU-17	Verificare che la classe <code>AbstractService</code> , metodo <code>delete</code> , elimini un oggetto richiamando il repository con l'ID corretto.	S
TU-18	Verificare che la classe <code>BlockService</code> , metodo <code>create</code> , crei correttamente un blocco e associ il prompt tramite repository.	S
TU-19	Verificare che la classe <code>BlockService</code> , metodo <code>create_no_duplicated</code> , gestisca correttamente più prompt associati a un blocco.	S
TU-20	Verificare che la classe <code>BlockService</code> , metodo <code>create_duplicated</code> , non crei un blocco se il nome è già presente nel sistema.	S
TU-21	Verificare che la classe <code>BlockService</code> , metodo <code>is_duplicated</code> , restituisca <code>True</code> se il nome del blocco è già esistente.	S
TU-22	Verificare che la classe <code>BlockService</code> , metodo <code>is_duplicated</code> , restituisca <code>False</code> se il nome non è presente.	S
TU-23	Verificare che la classe <code>BlockService</code> , metodo <code>retrieve_blocks</code> , recuperi correttamente i blocchi a partire da una lista di ID.	S
TU-24	Verificare che la classe <code>BlockService</code> , metodo <code>get_common_blocks</code> , identifichi correttamente i blocchi comuni tra due LLM.	S
TU-25	Verificare che la classe <code>BlockTestService</code> , metodo <code>interrogate</code> , invii correttamente la richiesta al microservizio LLM e ne gestisca la risposta.	S
TU-26	Verificare che la classe <code>BlockTestService</code> , metodo <code>runtest</code> , gestisca correttamente interrogazione, valutazione e calcolo delle medie per ogni blocco e LLM.	S
TU-27	Verificare che la classe <code>BlockTestService</code> , metodo <code>format_results</code> , restituisca correttamente il dizionario dei risultati formattati da un test.	S
TU-28	Verificare che la classe <code>EvaluationService</code> erediti correttamente le <u>funzionalità_G</u> da <code>AbstractService</code> .	S
TU-29	Verificare che la classe <code>LLMService</code> , metodo <code>fetch_ollama_models</code> , recuperi correttamente i modelli da <u>Ollama_G</u> e gestisca l' <u>endpoint_G</u> .	S

Codice	Descrizione	Stato del test
TU-30	Verificare che la classe LLMService, metodo sync_ollama_llms, sincronizzi i modelli ottenuti da Ollama nel <code>databaseg</code> tramite il repository.	S
TU-31	Verificare che la classe LLMService, metodo fetch_ollama_models, gestisca correttamente un timeout durante la richiesta HTTP.	S
TU-32	Verificare che la classe LLMService, metodo sync_ollama_llms, non chiami il repository se la lista dei modelli è vuota.	S
TU-33	Verificare che la classe PrevTestService, metodo get_tests_by_session, restituisca correttamente i test relativi a una sessione tramite repository.	S
TU-34	Verificare che la classe PromptService erediti correttamente le funzionalità da AbstractService.	S
TU-35	Verificare che la classe RunService, metodo get_formatted_by_prompt, restituisca correttamente i risultati formattati per un prompt, calcolando le medie.	S
TU-36	Verificare che la classe RunService, metodo get_formatted_by_prompt, restituisca None se il prompt con l'ID fornito non esiste.	S
TU-37	Verificare che la classe SessionService, metodo get_excluded_llm, restituisca correttamente la lista dei LLM esclusi da una sessione.	S
TU-38	Verificare che la classe SessionService, metodo add_llm, aggiunga correttamente un LLM a una sessione.	S
TU-39	Verificare che la classe SessionService, metodo delete_llm, rimuova correttamente un LLM da una sessione.	S
TU-40	Verificare che la AbstractView, quando riceve una richiesta GET senza id, restituisca correttamente una lista di dati simulati con status HTTP 200.	S
TU-41	Verificare che la AbstractView, quando riceve una richiesta GET con id, restituisca correttamente un dato simulato con status HTTP 200.	S
TU-42	Verificare che la AbstractView, quando riceve una richiesta GET con id tramite il metodo as_view(), restituisca correttamente un dato simulato con status HTTP 200.	S
TU-43	Verificare che la AbstractView, quando riceve una richiesta POST con dati validi, crei correttamente un nuovo oggetto simulato e restituisca la risposta con status HTTP 201.	S
TU-44	Verificare che la AbstractView, quando riceve una richiesta PUT con dati validi e un id, aggiorni correttamente l'oggetto simulato e restituisca la risposta con status HTTP 200.	S
TU-45	Verificare che la AbstractView, quando riceve una richiesta DELETE con id, elimini correttamente l'oggetto simulato e restituisca una risposta con status HTTP 204 No Content.	S
TU-46	Verifica che la AbstractView, quando riceve una richiesta PUT senza specificare un id, restituisca errore 404 e il messaggio "Sessione non trovata".	S
TU-47	Verifica che la AbstractView, quando riceve una richiesta PUT senza specificare un id, restituisca errore 404 e il messaggio "Sessione non trovata".	S

Codice	Descrizione	Stato del test
TU-48	Verifica che la AbstractView, quando riceve una richiesta POST con dati non validi (serializer non valido), restituisca errore 400 e un messaggio di errore nella risposta.	S
TU-49	Verifica che la AbstractView, quando riceve una richiesta PUT con dati non validi (serializer non valido), restituisca errore 400 e un messaggio di errore nella risposta.	S
TU-50	Verifica che la AbstractView, quando il service _G solleva un'eccezione durante una richiesta POST, restituisca errore 400 e un messaggio di errore nella risposta.	S
TU-51	Verifica che la AbstractView, quando il service solleva un'eccezione durante una richiesta PUT, restituisca errore 400 e un messaggio di errore nella risposta.	S
TU-52	Verifica che la AbstractView, quando il service solleva un'eccezione durante una richiesta DELETE, restituisca errore 400 e un messaggio di errore nella risposta.	S
TU-53	Verifica che la AbstractView, quando il service solleva un'eccezione durante una richiesta GET per tutti gli oggetti, restituisca errore 400 e un messaggio di errore nella risposta.	S
TU-54	Verifica che la AbstractView, quando il service solleva un'eccezione durante una richiesta GET con id, restituisca errore 400 e un messaggio di errore nella risposta.	S
TU-55	Verifica che la BlockView, quando riceve una richiesta POST con dati validi, crei correttamente un nuovo blocco e restituisca la risposta con status HTTP 201.	S
TU-56	Verifica che la BlockView, quando riceve una richiesta POST con un nome di blocco già esistente (duplicato), restituisca errore 500 e il messaggio "Nome duplicato" nella risposta.	S
TU-57	Verifica che la BlockView, quando il service solleva un'eccezione durante una richiesta POST, restituisca errore 400 e il messaggio di errore corrispondente nella risposta.	S
TU-58	Verifica che la BlockTestView, quando entrambi i LLM e i blocchi comuni esistono, restituisca correttamente la lista dei blocchi comuni con status HTTP 200 e i dati attesi.	S
TU-59	Verifica che la BlockTestView, quando uno dei due LLM richiesti non esiste, restituisca errore 400 e un messaggio di errore nella risposta.	S
TU-60	Verifica che la BlockTestView, quando non esistono blocchi comuni tra i due LLM richiesti, restituisca una lista vuota di blocchi comuni con status HTTP 200.	S
TU-61	Verifica che la LLMView, quando riceve una richiesta GET per tutti i LLM, restituisca correttamente una lista di LLM con status HTTP 200.	S
TU-62	Verifica che la LLMView, quando riceve una richiesta GET per un LLM specifico tramite id, restituisca correttamente i dati del LLM con status HTTP 200.	S
TU-63	Verifica che la LLMView, quando il service solleva un'eccezione durante una richiesta GET per tutti i LLM, restituisca errore 400 e un messaggio di errore nella risposta.	S
TU-64	Verifica che la LLMView, quando il service solleva un'eccezione durante una richiesta GET per un LLM specifico tramite id, restituisca errore 400 e un messaggio di errore nella risposta.	S

Codice	Descrizione	Stato del test
TU-65	Verifica che la LLMView, quando riceve una richiesta POST con dati validi, crei correttamente un nuovo LLM e restituisca la risposta con status HTTP 201.	S
TU-66	Verifica che la LLMView, quando riceve una richiesta POST con dati non validi (serializer non valido), restituisca errore 400 e segnali correttamente l'errore di validazione del campo obbligatorio nella risposta.	S
TU-67	Verifica che la LLMView, quando il service solleva un'eccezione durante una richiesta POST, restituisca errore 400 e un messaggio di errore nella risposta.	S
TU-68	Verifica che la LLMView, quando riceve una richiesta PUT con dati validi e un id esistente, aggiorni correttamente il LLM e restituisca la risposta con status HTTP 200.	S
TU-69	Verifica che la LLMView, quando riceve una richiesta PUT con dati non validi (serializer non valido), restituisca errore 400 e segnali correttamente l'errore di validazione del campo obbligatorio nella risposta.	S
TU-70	Verifica che la LLMView, quando il service solleva un'eccezione durante una richiesta PUT, restituisca errore 400 e un messaggio di errore nella risposta.	S
TU-71	Verifica che la LLMView, quando riceve una richiesta DELETE con un id valido, elimini correttamente il LLM e restituisca una risposta con status HTTP 204.	S
TU-72	Verifica che la LLMView, quando il service solleva un'eccezione durante una richiesta DELETE, restituisca errore 400 e un messaggio di errore nella risposta.	S
TU-73	Verifica che la OllamaView, quando la sincronizzazione dei modelli LLM va a buon fine, restituisca status HTTP 200 e il messaggio di successo nella risposta.	S
TU-74	Verifica che la OllamaView, quando si verifica un'eccezione durante la sincronizzazione dei modelli LLM, restituisca status HTTP 500 e il messaggio di errore nella risposta.	S
TU-75	Verifica che la prev_test_view, quando viene effettuata una richiesta GET con parametro test_id, restituisca i risultati formattati correttamente e status HTTP 200, chiamando i metodi del service e del formatter previsti.	S
TU-76	Verifica che la prev_test_view, quando viene effettuata una richiesta GET senza parametro test_id, restituisca la lista dei test precedenti relativi alla sessione richiesta con status HTTP 200, utilizzando il service e il serializer previsti.	S
TU-77	Verifica che la view_G, quando il service solleva un'eccezione di tipo "Test not found" durante una richiesta GET per i test precedenti di una sessione inesistente, restituisca errore 400 e un messaggio di errore nella risposta.	S
TU-78	Verifica che la view, quando il service solleva un'eccezione generica durante una richiesta GET per i test precedenti, restituisca errore 400 e un messaggio di errore nella risposta.	S
TU-79	Verifica che la view, quando riceve una richiesta DELETE con un id di sessione valido, elimini correttamente il test precedente e restituisca status HTTP 204.	S

Codice	Descrizione	Stato del test
TU-80	Verifica che la view, quando il service solleva un'eccezione "Test not found" durante una richiesta DELETE, restituisca errore 400 e un messaggio di errore nella risposta.	S
TU-81	Verifica che la view, quando il service solleva un'eccezione generica durante una richiesta DELETE, restituisca errore 400 e un messaggio di errore nella risposta.	S
TU-82	Verifica che la PromptView, quando riceve una richiesta GET per tutti i prompt, restituisca una lista di prompt con status HTTP 200.	S
TU-83	Verifica che la PromptView, quando il service solleva un'eccezione durante una richiesta GET per tutti i prompt, restituisca errore 400 e un messaggio di errore nella risposta.	S
TU-84	Verifica che la PromptView, quando riceve una richiesta GET per un prompt specifico tramite id, restituisca i dati del prompt richiesto con status HTTP 200.	S
TU-85	Verifica che la PromptView, quando il service solleva un'eccezione durante una richiesta GET per un prompt specifico tramite id, restituisca errore 400 e un messaggio di errore nella risposta.	S
TU-86	Verifica che la PromptView, quando riceve una richiesta POST con dati validi, crei correttamente un nuovo prompt e restituisca la risposta con status HTTP 201.	S
TU-87	Verifica che la PromptView, quando riceve una richiesta POST con dati non validi (serializer non valido), restituisca errore 400 e segnali correttamente l'errore di validazione del campo obbligatorio nella risposta.	S
TU-88	Verifica che la PromptView, quando il service solleva un'eccezione durante una richiesta POST, restituisca errore 400 e un messaggio di errore nella risposta.	S
TU-89	Verifica che la PromptView, quando riceve una richiesta PUT con dati validi e un id esistente, aggiorni correttamente il prompt e restituisca la risposta con status HTTP 200.	S
TU-90	Verifica che la PromptView, quando riceve una richiesta PUT con dati non validi (serializer non valido), restituisca errore 400 e segnali correttamente l'errore di validazione del campo obbligatorio nella risposta.	S
TU-91	Verifica che la PromptView, quando il service solleva un'eccezione durante una richiesta PUT, restituisca errore 400 e un messaggio di errore nella risposta.	S
TU-92	Verifica che la PromptView, quando riceve una richiesta DELETE con un id valido, elimini correttamente il prompt e restituisca una risposta con status HTTP 204.	S
TU-93	Verifica che la PromptView, quando il service solleva un'eccezione durante una richiesta DELETE, restituisca errore 400 e un messaggio di errore nella risposta.	S
TU-94	Verifica che la RunPromptView, quando la richiesta GET non include il parametro prompt_id, restituisca errore 400 e il messaggio "Missing prompt_id."	S
TU-95	Verifica che la RunPromptView, quando la richiesta GET include un parametro prompt_id non valido, restituisca errore 400 e il messaggio "Invalid prompt_id."	S

Codice	Descrizione	Stato del test
TU-96	Verifica che la RunPromptView, quando il prompt richiesto tramite prompt_id non esiste, restituisca errore 404 e il messaggio "Prompt not found."	S
TU-97	Verifica che la RunPromptView, quando viene richiesto un prompt esistente tramite prompt_id, restituisca correttamente i dati dei run associati con status HTTP 200.	S
TU-98	Verifica che la RunPromptView, quando la richiesta DELETE non include il parametro run_id, restituisca errore 400 e il messaggio "Missing run_id."	S
TU-99	Verifica che la RunPromptView, quando la richiesta DELETE include un parametro run_id non valido, restituisca errore 400 e il messaggio "Invalid run_id."	S
TU-100	Verifica che la RunPromptView, quando riceve una richiesta DELETE con un run_id valido, elimini correttamente il run e restituisca status HTTP 204.	S
TU-101	Verifica che la RunPromptView, quando il service solleva un ValueError perché il run non esiste, restituisca errore 404 e il messaggio "Run not found".	S
TU-102	Verifica che la RunPromptView, quando il service solleva un'eccezione generica durante una richiesta DELETE, restituisca errore 400 e il relativo messaggio di errore nella risposta.	S
TU-103	Verifica che session_llm_view, Verifica che la view, quando riceve una richiesta GET per la lista di LLM esclusi da una sessione reale, restituisca la lista corretta e che l'LLM già associato non sia presente nella risposta.	S
TU-104	Verifica che la session_view, quando riceve una richiesta GET per tutte le sessioni, restituisca la lista delle sessioni presenti con status HTTP 200.	S
TU-105	Verifica che la session_view, quando riceve una richiesta GET per una sessione specifica tramite id, restituisca i dati della sessione richiesta con status HTTP 200.	S
TU-106	Verifica che la session_view, quando il service solleva un'eccezione durante una richiesta GET per tutte le sessioni, restituisca errore 400 e un messaggio di errore nella sessione.	S
TU-107	Verifica che la session_view, quando riceve una richiesta POST con dati validi, crei correttamente una nuova sessione e restituisca la risposta con status HTTP 201.	S
TU-108	Verifica che la session_view, quando riceve una richiesta POST con dati non validi (assenza di campi obbligatori), restituisca errore 400 e segnali correttamente l'errore di validazione del campo obbligatorio nella risposta.	S
TU-109	Verifica che la session_view, quando il service solleva un'eccezione durante una richiesta POST, restituisca errore 400 e un messaggio di errore nella risposta.	S
TU-110	Verifica che la session_view, quando riceve una richiesta PUT con dati validi e un id esistente, aggiorni correttamente la sessione e restituisca la risposta con status HTTP 200.	S
TU-111	Verifica che la session_view, quando riceve una richiesta PUT con dati non validi (assenza di campi obbligatori), restituisca errore 400 e segnali correttamente l'errore di validazione del campo obbligatorio nella risposta.	S

Codice	Descrizione	Stato del test
TU-112	Verifica che la session_view, quando il service solleva un'eccezione durante una richiesta PUT, restituisca errore 400 e un messaggio di errore nella risposta.	S
TU-113	Verifica che la session_view, quando riceve una richiesta DELETE con un id valido, elimini correttamente la sessione e restituisca una risposta con status HTTP 204.	S
TU-114	Verifica che la session_view, quando il service solleva un'eccezione durante una richiesta DELETE, restituisca errore 400 e un messaggio di errore nella risposta.	S
TU-115	Verifica che la test_view, quando riceve una richiesta POST con dati validi, esegua correttamente il test sui blocchi richiesti e restituisca la risposta con status HTTP 200 e il risultato _G atteso.	S
TU-116	Verifica che la test_view, quando si verifica un errore di connessione durante l'esecuzione del test, restituisca errore 503 e il relativo messaggio di errore nella risposta.	S
TU-117	Verifica che la test_view, quando si verifica un errore di tipo FileNotFoundException durante l'esecuzione del test, restituisca errore 500 e il relativo messaggio di errore nella risposta.	S
TU-118	Verifica che la test_view, quando si verifica un'eccezione generica durante l'esecuzione del test, restituisca errore 500 e un messaggio di errore che segnala un errore inatteso.	S
TU-119	Verificare che la view LLMView, metodo POST, interroghi correttamente il modello LLM e restituisca la risposta con codice 200.	S
TU-120	Verificare che la view LLMView, metodo POST, restituisca errore 500 se LLMService.get_llm non restituisce un'istanza valida.	S

Tabella 12: Test di unità

3.3 Test di integrazione

Codice	Descrizione	Stato del test
TI-01	Verifica che la classe AbstractRepository, metodo create, salvi correttamente un'istanza nel database e ne popoli i campi.	S
TI-02	Verifica che la classe AbstractRepository, metodo get_all, recuperi correttamente le istanze esistenti nel database.	S
TI-03	Verifica che la classe AbstractRepository, metodo get_by_id, recuperi correttamente un'istanza dato il suo ID.	S
TI-04	Verifica che la classe AbstractRepository, metodo update, aggiorni correttamente il valore di un campo di un'istanza.	S
TI-05	Verifica che la classe AbstractRepository, metodo delete, elimini correttamente un'istanza e che non sia più recuperabile.	S
TI-06	Verifica che la classe BlockRepository, metodo add_prompt, associ correttamente un prompt a un blocco.	S
TI-07	Verifica che la classe BlockRepository, metodo remove_prompt, rimuova correttamente un prompt da un blocco.	S

Codice	Descrizione	Stato del test
TI-08	Verifica che la classe BlockRepository, metodo get_by_name, restituisca correttamente un blocco dato il nome.	S
TI-09	Verifica che la classe BlockRepository, metodo get_prompts, restituisca correttamente tutti i prompt associati a un blocco.	S
TI-10	Verifica che la classe BlockRepository, metodo filter_by_llm, restituisca i blocchi contenenti prompt collegati all'LLM indicato.	S
TI-11	Verifica che la classe BlockRepository, metodo filter_by_ids, restituisca solo i blocchi con ID specificati.	S
TI-12	Verifica che la classe BlockTestRepository, metodo add_run, associ correttamente uno o più run a un test	S
TI-13	Verifica che la classe BlockTestRepository, metodo remove_run, rimuova correttamente i run da un test.	S
TI-14	Verificare che la classe EvaluationRepository erediti correttamente il comportamento CRUD da AbstractRepository.	S
TI-15	Verificare che la classe LLMRepository, metodo update_or_create, crei un nuovo modello LLM se non esiste.	S
TI-16	Verifica che la classe LLMRepository, metodo update_or_create, aggiorni correttamente i parametri di un LLM esistente.	S
TI-17	Verifica che la classe LLMRepository, metodo update_or_create, non esegua modifiche se i dati forniti sono identici.	S
TI-18	Verifica che la classe PrevTestRepository, metodo get_tests_by_session, restituisca solo i test associati alla sessione specificata e non includa test di altre sessioni.	S
TI-19	Verifica che la classe PromptRepository, metodo filter_one, restituisca il prompt corretto dato il testo e la risposta attesa, oppure None se non esiste corrispondenza.	S
TI-20	Verifica che la classe PromptRepository, metodo get_or_create, restituisca il prompt esistente se presente oppure ne crei uno nuovo con i valori forniti.	S
TI-21	Verifica che la classe RunRepository, metodo get_common_runs, restituisca correttamente le run comuni a due LLM per uno stesso prompt.	S
TI-22	Verifica che la classe RunRepository, metodo get_by_prompt, restituisca tutte le run associate a un determinato prompt, includendo i riferimenti agli oggetti Run e Prompt.	S
TI-23	Verifica che la classe SessionRepository, metodo get_remaining_llm, restituisca correttamente i LLM non ancora associati alla sessione.	S
TI-24	Verifica che la classe SessionRepository, metodo get_llms, restituisca correttamente i LLM associati a una sessione.	S
TI-25	Verifica che la classe SessionRepository, metodo add_llm, associ un LLM alla sessione e lo renda disponibile nella relazione.	S
TI-26	Verifica che la classe SessionRepository, metodo delete_llm, rimuova correttamente un LLM dalla sessione.	S

Tabella 13: Test di integrazione

3.4 Test di sistema

Codice	Descrizione	Fonte	Stato del test
TS-01	Verificare che l'utente possa creare una sessione.	RF-01	S
TS-02	Verificare che l'utente, in fase di creazione di una sessione, possa inserire un Titolo.	RF-02	S
TS-03	Verificare che l'utente, in fase di creazione di una sessione, possa inserire una Descrizione.	RF-03	S
TS-04	Verificare che l'utente, in fase di creazione di una sessione, riceva un errore contestuale in seguito all'inserimento di un Titolo non accettato dal sistema.	RF-04	S
TS-05	Verificare che l'utente, in fase di creazione di una sessione, riceva un errore contestuale in seguito all'inserimento di una Descrizione non accettata dal sistema.	RF-05	S
TS-06	Verificare che l'utente possa visualizzare la lista delle sessioni presenti nel sistema.	RF-06	S
TS-07	Verificare che l'utente possa visualizzare un singolo elemento "sessione" dalla lista delle sessioni.	RF-07	S
TS-08	Verificare che l'utente possa visualizzare il Titolo di un singolo elemento "sessione" dalla lista delle sessioni.	RF-08	S
TS-09	Verificare che l'utente possa visualizzare la Descrizione di un singolo elemento "sessione" dalla lista delle sessioni.	RF-09	S
TS-10	Verificare che l'utente possa visualizzare la data di ultimo accesso di un singolo elemento "sessione" dalla lista delle sessioni.	RF-10	S
TS-11	Verificare che l'utente possa modificare i dettagli di una sessione.	RF-11	S
TS-12	Verificare che l'utente in fase di modifica dei dettagli di una sessione, possa modificarne il Titolo.	RF-12	S
TS-13	Verificare che l'utente in fase di modifica dei dettagli di una sessione, possa modificarne la Descrizione.	RF-13	S
TS-14	Verificare che l'utente possa cancellare una sessione.	RF-14	S
TS-15	Verificare che l'utente all'interno di una sessione, possa inserire manualmente una, o più, coppie domande e risposte attese.	RF-15	S
TS-16	Verificare che l'utente, durante l'inserimento di una coppia domanda e risposta attesa, possa inserire una singola domanda nell'apposito spazio.	RF-16	S

Codice	Descrizione	Fonte	Stato del test
TS-17	Verificare che l'utente, durante l'inserimento di una coppia domanda e risposta attesa, possa inserire una singola risposta attesa nell'apposito spazio.	RF-17	S
TS-18	Verificare che l'utente, durante l'inserimento di una coppia domanda e risposta attesa, riceva un errore contestuale in seguito all'inserimento di una domanda non accettata dal sistema.	RF-18	S
TS-19	Verificare che l'utente all'interno di una sessione, possa importare una lista di domande e risposte attese, tramite un file in formato <u>JSON</u> .	RF-19	S
TS-20	Verificare che l'utente, durante l'importazione di una lista domande e risposte attese, tramite un file in formato JSON, riceva un errore contestuale nel caso in cui una coppia domanda e risposta attesa risulti priva di domanda.	RF-20	S
TS-21	Verificare che l'utente, durante l'importazione di una lista domande e risposte attese, tramite un file in formato JSON, riceva un errore contestuale nel caso in cui una coppia domanda e risposta attesa risulti priva di risposta.	RF-21	S
TS-22	Verificare che l'utente, durante l'importazione di una lista domande e risposte attese, tramite un file in formato JSON, possa ricevere un errore contestuale nel caso in cui il sistema non riesca a validare il file.	RF-22	S
TS-23	Verificare che l'utente, all'interno di una sessione, possa visualizzare la lista delle domande e risposte attese attualmente presenti nella sessione.	RF-23	S
TS-24	Verificare che l'utente, all'interno di una sessione, possa visualizzare dalla lista delle domande e risposte attese una singola coppia domanda e risposta attesa.	RF-24	S
TS-25	Verificare che l'utente, mentre visualizza una singola coppia domanda e risposta attesa, possa visualizzare la domanda.	RF-25	S
TS-26	Verificare che l'utente, mentre visualizza una singola coppia domanda e risposta attesa, possa visualizzare la risposta attesa.	RF-26	S
TS-27	Verificare che l'utente, all'interno di una sessione, possa modificare il testo di una domanda inserita precedentemente.	RF-27	S
TS-28	Verificare che l'utente, durante la modifica del testo di una domanda inserita precedentemente, all'interno di una sessione, riceva un errore contestuale nel caso in cui la nuova domanda inserita non sia accettata dal sistema.	RF-28	S

Codice	Descrizione	Fonte	Stato del test
TS-29	Verificare che l'utente, all'interno di una sessione, possa modificare il testo di una risposta attesa inserita precedentemente.	RF-29	S
TS-30	Verificare che l'utente, durante la modifica del testo di una domanda inserita precedentemente, all'interno di una sessione, riceva un errore contestuale nel caso in cui la nuova risposta attesa inserita non sia accettata dal sistema.	RF-30	S
TS-31	Verificare che l'utente, all'interno di una sessione, possa eliminare una, o più coppie domanda e risposta attesa, tra quelle presenti nella lista.	RF-31	S
TS-32	Verificare che l'utente, all'interno di una sessione, possa eliminare contemporaneamente tutte le coppie domande e risposte attese, presenti nella lista.	RF-32	S
TS-33	Verificare che l'utente, all'interno dell'interfaccia di gestione dei <u>LLM di test</u> _G , possa collegare un nuovo LLM di test.	RF-33	S
TS-34	Verificare che l'utente, all'interno dell'interfaccia di gestione dei LLM di test, possa caricare dei modelli LLM Ollama predefiniti scelti in fase di sviluppo.	RF-34	S
TS-35	Verificare che l'utente, durante la fase di collegamento di un nuovo LLM di test, inserisca il nome del LLM di test desiderato.	RF-35	S
TS-36	Verificare che l'utente, durante la fase di collegamento di un nuovo LLM di test, possa inserire il numero di parametri del LLM di test desiderato.	RF-36	S
TS-37	Verificare che l'utente, all'interno dell'interfaccia di gestione dei LLM di test, possa visualizzare la lista dei LLM di test già collegati al sistema.	RF-37	S
TS-38	Verificare che l'utente, durante la visualizzazione della lista dei LLM di test già collegati al sistema, possa visualizzare un singolo LLM di test.	RF-38	S
TS-39	Verificare che l'utente, durante la visualizzazione di un singolo LLM di test, possa visualizzare il nome.	RF-39	S
TS-40	Verificare che l'utente, durante la visualizzazione di un singolo LLM di test, possa visualizzare, se presente, il numero di parametri.	RF-40	S
TS-41	Verificare che l'utente, durante il collegamento di un nuovo LLM di test, riceva un errore contestuale nel caso in cui sia già stato raggiunto il numero di LLM collegabili al sistema (25).	RF-41	S

Codice	Descrizione	Fonte	Stato del test
TS-42	Verificare che l'utente, durante il collegamento di un nuovo LLM di test, riceva un errore contestuale nel caso in cui il nome inserito non sia riconosciuto dal sistema.	RF-42	S
TS-43	Verificare che l'utente, all'interno dell'interfaccia di gestione dei LLM di test, possa rimuovere un LLM collegato precedentemente.	RF-43	S
TS-44	Verificare che l'utente, all'interno dell'interfaccia di gestione dei LLM di test, possa collegare un nuovo <u>LLM di valutazione_G</u> .	RF-44	S
TS-45	Verificare che l'utente, all'interno di una sessione, prima di eseguire un test, proceda a configurarlo, selezionando un LLM da testare.	RF-45	S
TS-46	Verificare che l'utente, all'interno di una sessione, prima di eseguire un test, proceda a configurarlo, selezionando un LLM di valutazione.	RF-46	S
TS-47	Verificare che l'utente, all'interno di una sessione, dopo aver configurato il LLM di test e il LLM di valutazione, possa eseguire un test.	RF-47	S
TS-48	Verificare che il sistema per eseguire un test invii le risposte del LLM di test al LLM di valutazione, che ritornerà una valutazione della risposta, confrontandola con la risposta attesa inserita dall'utente.	RF-48	S
TS-49	Verificare che il sistema per eseguire un test esegua una valutazione semantica, di somiglianza fra la risposta attesa e la risposta ritornata dal LLM di test.	RF-49	S
TS-50	Verificare che l'utente, durante lo svolgimento di un test, riceva un errore contestuale nel caso in cui la connessione con il LLM di test fallisca.	RF-50	S
TS-51	Verificare che l'utente, durante lo svolgimento di un test, riceva un errore contestuale nel caso in cui la connessione con il LLM di valutazione fallisca.	RF-51	S
TS-52	Verificare che il LLM di valutazione restituisca una valutazione sulle risposte generate dal LLM di test, dopo che l'utente ha eseguito un test.	RF-52	S
TS-53	Verificare che l'utente, dopo aver eseguito un test, possa visualizzare i risultati.	RF-53	S
TS-54	Verificare che l'utente, dopo aver eseguito un test, possa visualizzare la valutazione semantica per ogni coppia risposta attesa e risposta del LLM.	RF-54	S

Codice	Descrizione	Fonte	Stato del test
TS-55	Verificare che l'utente, dopo aver eseguito un test, possa visualizzare la valutazione data dal LLM di valutazione in merito alla risposta generata dal LLM di test.	RF-55	S
TS-56	Verificare che l'utente, dopo aver eseguito un test, possa visualizzare un grafico a barre orizzontali in pila, che riassume i risultati del test. Verificare che il grafico riporti sull'asse Y il nome del LLM di test e sull'asse X il numero di domande, suddivise per il livello di valutazione ottenuta (verde: >75%, giallo: 40-75%, rosso: <40%).	RF-56	S
TS-57	Verificare che l'utente, dopo aver eseguito un test, possa visualizzare per ogni domanda, un riquadro colorato, che renda visivamente immediata la comprensione della valutazione (verde: >75%, giallo: 40-75%, rosso: <40%).	RF-57	S
TS-58	Verificare che l'utente, dopo aver eseguito un test, possa visualizzare la risposta generata dal LLM di test meglio valutata dal sistema.	RF-58	S
TS-59	Verificare che l'utente, dopo aver eseguito un test, possa visualizzare la risposta generata dal LLM di test peggio valutata dal sistema.	RF-59	S
TS-60	Verificare che l'utente, all'interno di una sessione, possa visualizzare i test precedentemente eseguiti.	RF-60	S
TS-61	Verificare che il sistema permetta l'archiviazione dei risultati dei test.	RF-61	S
TS-62	Verificare che l'utente, all'interno di una sessione, possa eseguire un confronto fra test precedentemente eseguiti, visualizzando per ogni test il rispettivo grafico a barre orizzontali in pila, riassuntivo dei risultati e la lista di domande su cui avviene il confronto, con le relative valutazioni.	RF-62	S
TS-63	Verificare che l'utente, all'interno di una sessione, possa scaricare in un file in formato PDF, i risultati di un test eseguito.	RF-63	S
TS-64	Verificare che l'utente, all'interno di una sessione, possa rimuovere dal sistema un test precedentemente eseguito.	RF-64	S

Tabella 14: Test di sistema

3.5 Test di accettazione

Codice	Descrizione	Fonte	Stato del test
TA-01	Verificare che l'utente possa creare, visualizzare, modificare e cancellare sessioni	UC 1, UC 2, UC 3, UC 4, UC 5, UC 6	S
TA-02	Verificare che l'utente possa visualizzare, aggiungere, selezionare e rimuovere modelli LLM	UC 20, UC 21, UC 22, UC 23, UC 24, UC 25	S
TA-03	Verificare che l'utente possa inserire, visualizzare, modificare e rimuovere domande e risposte	UC 7, UC 8, UC 9, UC 10, UC 11, UC 12, UC 13, UC 14, UC 15, UC 16, UC 17, UC 18, UC 19	S
TA-04	Verificare che il test venga avviato correttamente e che il sistema registri le risposte	UC 27, UC 28, UC 29	S
TA-05	Verificare che i risultati siano comprensibili e utili per l'utente	UC 31	S
TA-06	Verificare che l'utente possa confrontare test passati	UC 32, UC 33	S
TA-07	Verificare che l'utente possa scaricare i risultati	UC 34	S

Tabella 15: Test di accettazione

4 Cruscotto di controllo della qualità

4.1 Budget Variance

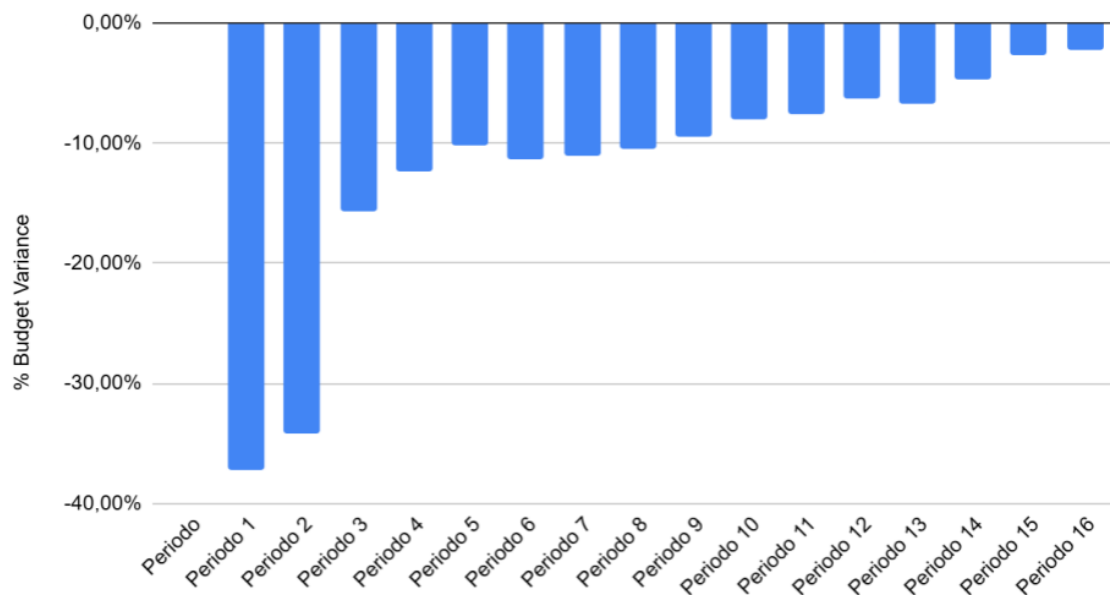


Figura 1: Proiezione dei valori di Budget Variance

RTB : Nei primi periodi ci sono stati gravi sforamenti di budget dovuti a errori di stima e pianificazione inadeguata. Tuttavia nei successivi periodi si nota un miglioramento e la tendenza negativa è diminuita notevolmente rientrando nei valori ottimali dal quarto periodo in poi. Il team si impegnerà a definire preventivi sempre più realistici, impegnandosi a utilizzare le ore e le risorse stanziare.

PB : Dopo lo scostamento marcato nella prima fase del progetto, segno di un lento avvio delle attività effettive rispetto al piano teorico, a partire dai periodi successivi si osserva una graduale riduzione della Budget Variance negativa, soprattutto dal decimo periodo, fino a convergere verso un valore prossimo allo 0% negli ultimi due periodi. Grazie agli sforzi del team nella fase di Product Baseline_G, il trend della curva in risalita indica un miglioramento significativo nella coerenza tra attività pianificate e svolte.

4.2 Planned Value e Earned Value

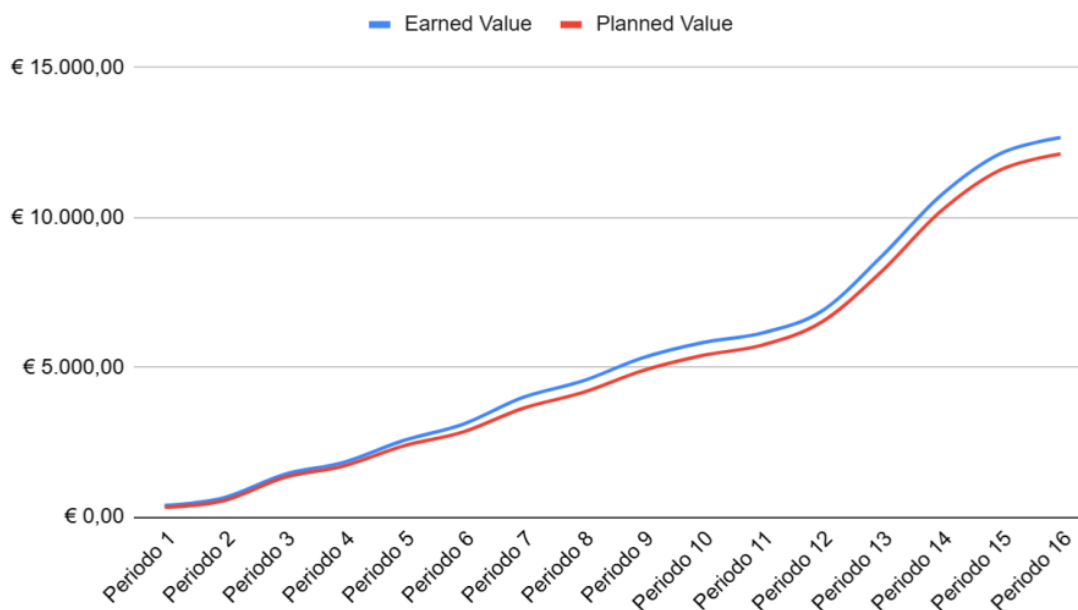


Figura 2: Proiezione dei valori di Planned Value e Earned Value

RTB : Il grafico mostra l'andamento dell'Earned Value, ovvero il valore prodotto dal progetto, rispetto al Planned Value, ovvero il valore del lavoro pianificato. Si può notare che Planned Value si mantiene al di sotto dell'Earned Value, seppur di poco, segno che i preventivi scritti finora sono risultati ottimistici rispetto alla spesa effettiva.

PB : Dall'osservazione del grafico emerge una stretta aderenza tra i valori di Earned Value e Planned Value, con le due curve che mantengono un andamento simile e pressoché parallelo lungo tutto l'arco temporale analizzato. Pur registrando in diversi momenti un Earned Value leggermente superiore rispetto al Planned Value, questo scostamento si presenta in forma contenuta e costante, suggerendo una buona coerenza tra quanto pianificato e quanto effettivamente realizzato. L'andamento parallelo dei due indicatori è un segnale positivo sotto il profilo della conformità al piano di progetto e della maturità del processo di sviluppo, in quanto indica un controllo continuo dell'avanzamento e un impegno costante nel mantenere l'equilibrio tra carico di lavoro, risorse e tempi.

4.3 Cost Variance e Schedule Variance

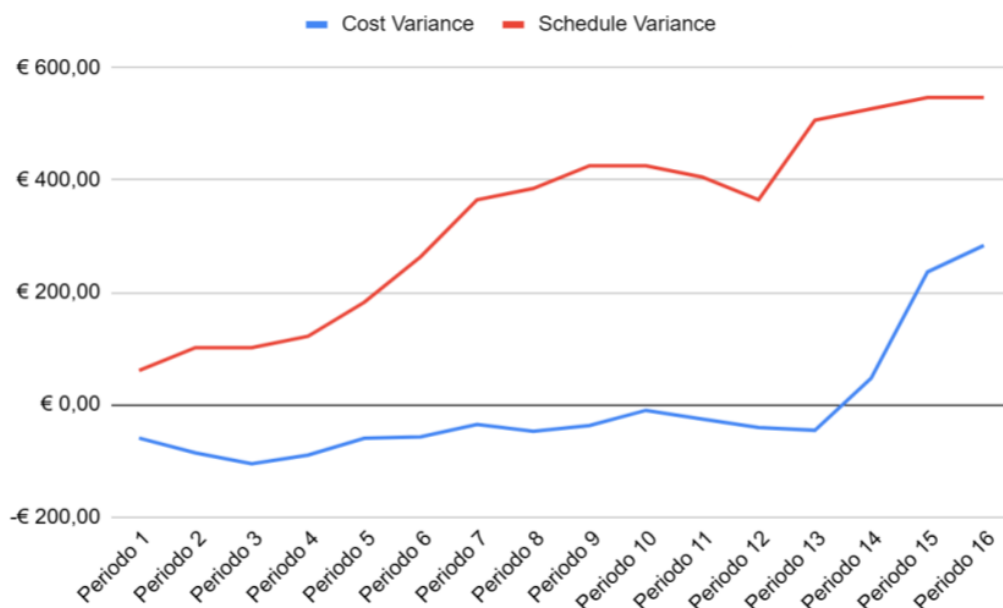


Figura 3: Proiezione dei valori di Cost Variance e Schedule Variance

RTB : Dopo un buon inizio nei primi tre periodi, è possibile notare uno sfioramento delle metriche dai valori accettabili, soprattutto per quanto riguarda la Schedule Variance. Mentre la Cost Variance negativa rispecchia il superamento dei costi nei consuntivi rispetto a quanto pianificato, la Schedule Variance positiva indica che il team ha prodotto più valore rispetto a quanto pianificato per tamponare il ritardo accumulato nei primi periodi. Il team si aspetta che nei prossimi periodi, grazie al way of working_G consolidato, si riuscirà a recuperare quanto perso.

PB : Nei primi tre periodi post RTB_G, i valori di Cost Variance restano negativi, seppur con valori contenuti rispetto ai periodi iniziali. Questo suggerisce che il progetto stava ancora operando con costi superiori al valore prodotto, ma con un miglioramento progressivo rispetto alle fasi iniziali. A partire dal periodo 14, la metrica_G diventa positiva, segnando una netta inversione di tendenza: il valore prodotto (Earned Value) supera il costo effettivo (Actual Cost) , indicando un'efficienza_G crescente. Gli ultimi due periodi mostrano valori di Cost Variance ampiamente positivi, indice di una fase finale in cui il team ha raggiunto una piena ottimizzazione delle risorse e un bilancio economico favorevole. Per quanto riguarda la Schedule Variance, la stabilità dei valori negli ultimi tre periodi (Schedule Variance \approx 547,50 €) suggerisce che il ritmo di avanzamento si è consolidato, mantenendo un equilibrio costante tra produzione di valore e obiettivi previsti. L'aumento percepito tra i periodi 12-14 segnala una spinta produttiva significativa verso il termine del progetto.

4.4 Cost Performance Index

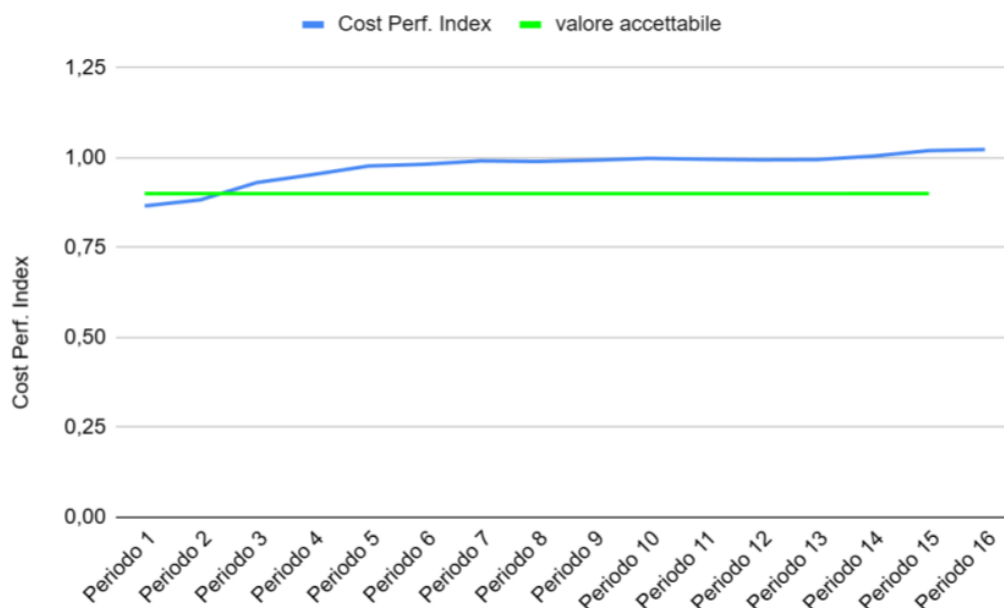


Figura 4: Proiezione dei valori di Cost Performance Index

RTB : Similmente a quanto riscontrabile nel grafico precedente di Cost Variance e Schedule variance, i valori di Cost Performance Index indicano una leggera inefficienza nei costi che però sta migliorando con l'avanzamento del progetto. Si può notare che la metrica sfiora dai valori solo nei primi due periodi, rientrando poi nel range accettabile, mentre i valori appena sotto l'unità (0,99...) indicano una fase di stabilizzazione dei costi: il progetto è vicino al pareggio tra spesa e valore generato.

PB : Nei periodi dedicati allo sviluppo e progettazione, si è registrato un aumento progressivo del Cost Performance Index da ~0,994 a oltre 1,02 indicando una crescente efficienza nella fase conclusiva del progetto, risultato di un controllo più accurato delle attività e di un affinamento del flusso di lavoro. A partire dal terzultimo periodo, il CPI supera la soglia critica di 1, segnalando che le risorse vengono impiegate in modo economicamente vantaggioso. L'andamento del Cost Performance Index nelle ultime fasi del progetto evidenzia una notevole crescita dell'efficienza economica, frutto di un miglior controllo dei costi, di un impiego più mirato delle risorse e della maturazione del processo di sviluppo.

4.5 Estimate to Complete, Estimate at Completion e Actual Cost

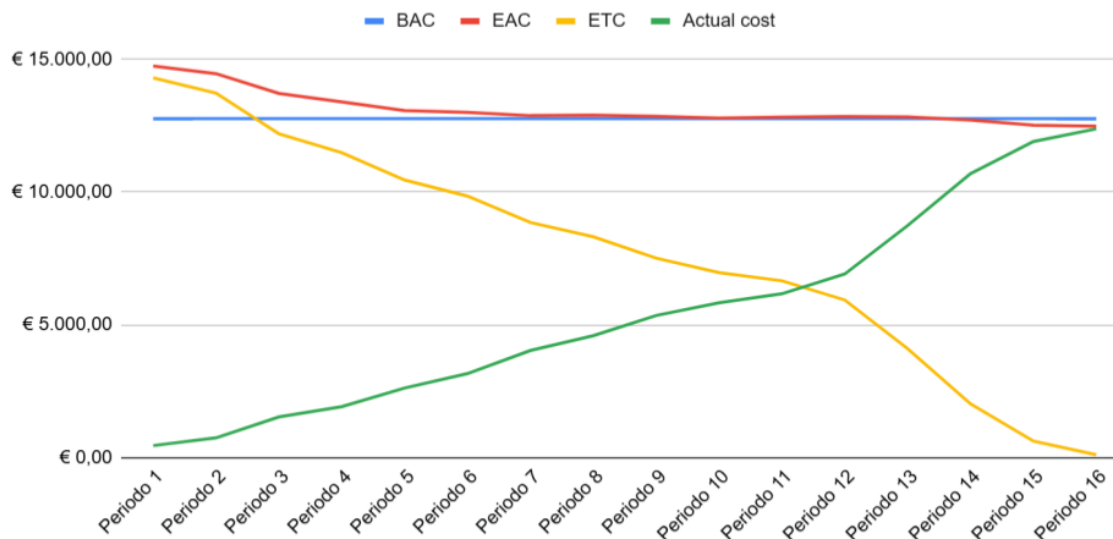


Figura 5: Proiezione dei valori di Estimate to Complete, Estimate at Completion e Actual Cost

RTB : Il grafico analizza la crescita della metrica Actual Cost, ovvero il costo effettivo fino ad un dato momento, e la decrescita di Estimate to Complete (ETC), che rappresenta la stima del costo rimanente del progetto. Di conseguenza, ci si aspetta che l'andamento delle due metriche sia speculare. Nel grafico si può notare anche la decrescita e lo stabilizzarsi di Estimate at Completion (EAC), allineandosi con le aspettative del BAC.

PB : Il progressivo allineamento di Estimate at Completion verso il BAC indica un raffinamento continuo delle stime, reso possibile da una maggiore conoscenza del progetto e dal controllo progressivo sui costi reali. La discesa sotto il BAC nelle fasi finali mostra una buona capacità di convergenza verso il budget previsto, sintomo di adattamento e recupero dell'efficienza economica dopo una fase iniziale critica, dove l'EAC superava il BAC. Il calo costante dell'ETC è coerente con l'avanzamento del progetto e riflette una riduzione graduale del lavoro residuo da completare. Anche i valori di Actual Cost hanno un andamento regolare, e ciò conferma che il progetto ha seguito una curva di spesa stabile, senza picchi anomali o scostamenti critici. La vicinanza finale tra AC e BAC è un indicatore positivo di conformità economica complessiva, pur tenendo conto delle difficoltà iniziali. Nonostante una partenza caratterizzata da sovrastime e costi proiettati superiori al previsto, il team ha dimostrato una buona capacità di controllo e adattamento. Le stime si sono affinate con l'avanzare del lavoro, portando a una conclusione in linea con il budget stabilito, in piena coerenza con i principi di qualità del processo.

4.6 Indice di Gulpease

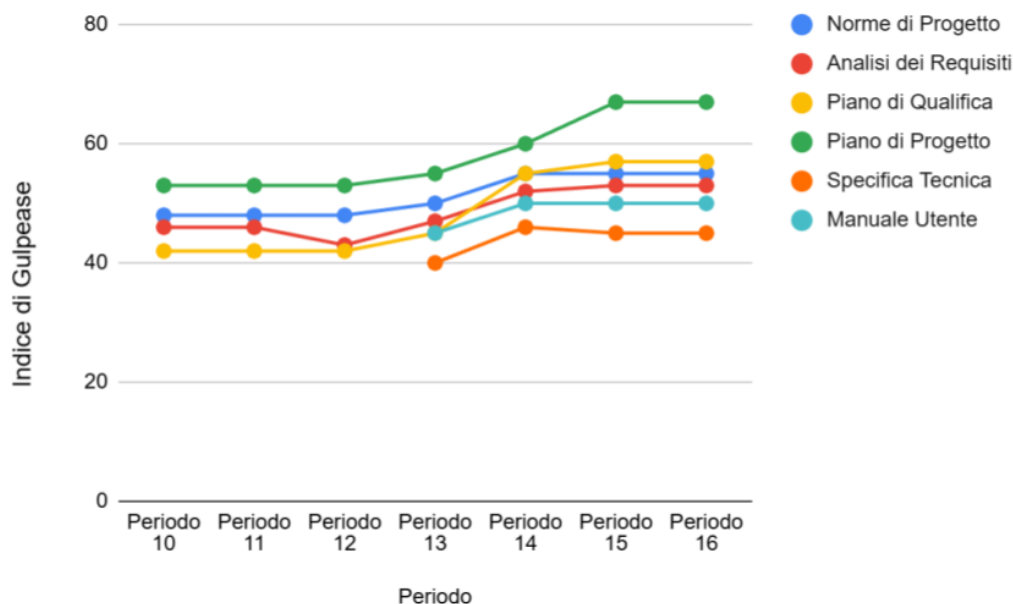


Figura 6: Proiezione dei valori di Gulpease

RTB : Analizzando il grafico, è immediato notare che l'indice di Gulpease dei diversi documenti si mantiene sopra la soglia di accettabilità. Tutti i primi quattro documenti si mantengono nel range di 40-60, indicando una complessità moderata in linea con la natura dei testi realizzati. Questo livello di leggibilità garantisce un equilibrio tra chiarezza e accuratezza, rendendo i documenti comprensibili per un pubblico con un'istruzione media senza sacrificare il rigore del contenuto.

PB : Il grafico mostra l'andamento della leggibilità dei principali documenti del progetto, valutata tramite l'indice di Gulpease, nel periodo compreso tra il 10 e il 16. A partire dal periodo 13, in corrispondenza dell'intensificarsi delle attività di sviluppo e rifinitura, si osserva un miglioramento costante nella qualità formale della documentazione. I valori rilevati indicano una leggibilità complessivamente buona, con indici generalmente superiori alla soglia raccomandata per testi tecnici. La maggior parte dei documenti raggiunge o supera il valore di 50, a conferma di una struttura testuale chiara e di un linguaggio accessibile. Fa eccezione la Specifica Tecnica, che presenta un indice leggermente più basso, comprensibile alla luce della complessità e della densità tecnica che la caratterizzano.

4.7 Rischi

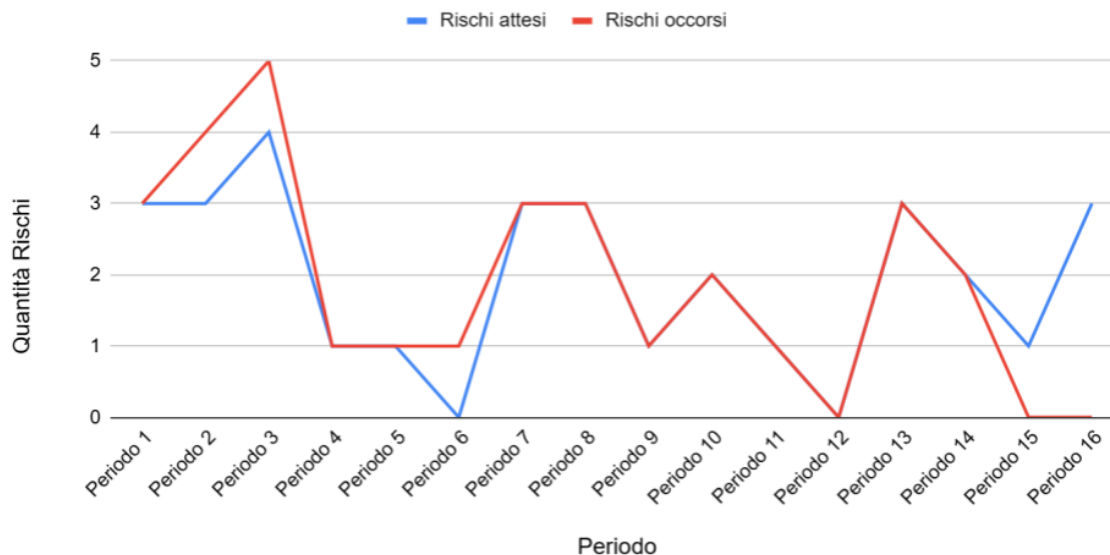


Figura 7: Proiezione dei rischi attesi e rischi occorsi

RTB : Il piano di gestione dei rischi evidenzia una coerenza complessiva soddisfacente tra i rischi previsti e quelli effettivamente verificatisi nel corso del progetto. Alcune discrepanze si riscontrano principalmente nei periodi iniziali e nel periodo di sessione invernale, quando il team era ancora in una fase di familiarizzazione con il progetto e stava acquisendo consapevolezza e competenza nell'applicazione delle buone pratiche di gestione.

PB : Nei periodi successivi al confronto RTB, le discrepanze tra i rischi attesi e quelli effettivamente occorsi risultano sovrapporsi. In particolare, nelle fasi finali del progetto si osserva una prevalenza dei rischi previsti rispetto a quelli effettivamente verificatisi. Questo andamento suggerisce l'efficacia delle misure di prevenzione adottate, che hanno contribuito a mitigare l'impatto potenziale degli eventi identificati in fase di pianificazione.

4.8 Percentuale di requisiti soddisfatti

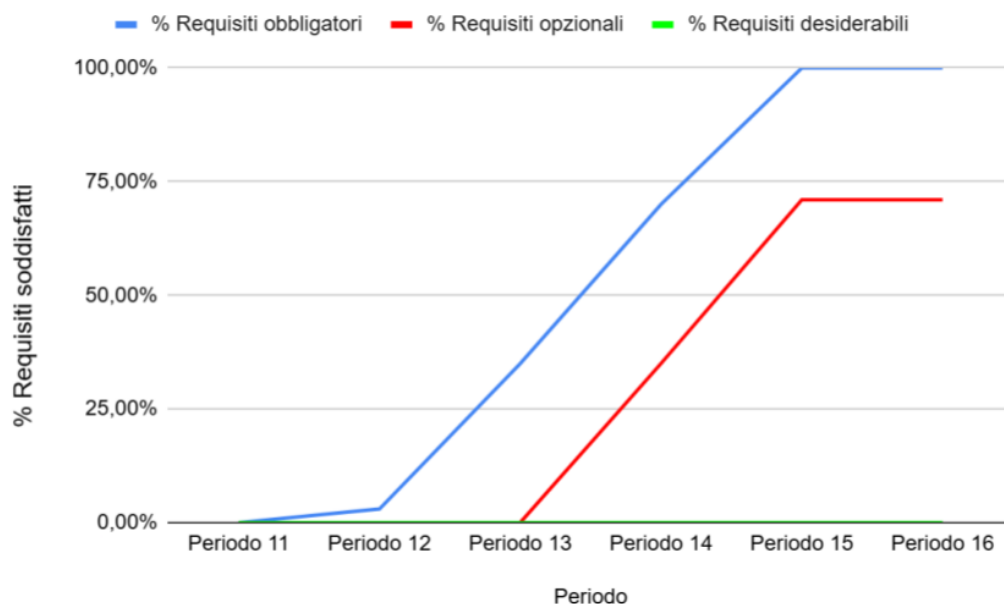


Figura 8: Proiezione della percentuale di requisiti soddisfatti

PB : La rappresentazione grafica mostra il raggiungimento del 100% di copertura dei requisiti obbligatori, accompagnato da un livello di soddisfacimento dei requisiti opzionali pari a circa il 70%. Tale risultato è stato possibile grazie all'impiego di codice riutilizzabile sviluppato durante la fase di Proof of Concept_G (POC_G), che ha consentito al gruppo di procedere con maggiore efficacia all'implementazione dei requisiti successivamente alla prima revisione RTB. A partire dal periodo 13, si osserva una crescita significativa nella copertura, favorita dall'intenso lavoro del team nella fase di progettazione, che ha permesso di integrare progressivamente le funzionalità principali e accessorie nel sistema. Nessun requisito desiderabile è stato implementato, in quanto si è preferito concentrare le risorse sulle funzionalità core e opzionali più rilevanti. L'omissione nell'implementazione di tali requisiti desiderabili è attribuibile alla priorità conferita ai requisiti obbligatori e opzionali, che hanno necessitato di un impegno considerevole. È fondamentale enfatizzare che tale omissione è stata concordata con l'azienda proponente_G.

4.9 Coverage

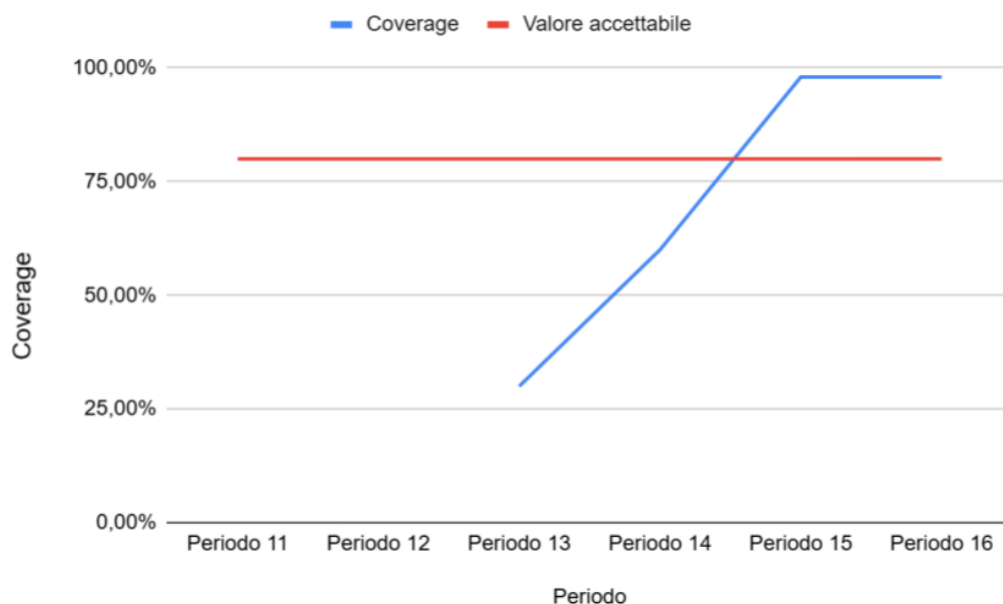


Figura 9: Proiezione dei valori percentuali di code coverage dei test

PB : Il grafico mostra la percentuale di coverage registrata, dove, a seguito dell'implementazione delle nuove funzionalità del sistema e all'introduzione di nuovo codice, è iniziata l'attività di testing. Nella prima fase di sviluppo, la copertura del codice risulta limitata, attestandosi intorno al 30%. Questo valore iniziale è coerente con l'introduzione dei primi test e con il fatto che, in quel momento, solo una porzione del sistema risultava implementata e testabile. Nei periodi successivi, la percentuale di copertura ha registrato un incremento costante e significativo, a testimonianza di una crescente estensione dei test alle funzionalità via via completate. Il miglioramento è proseguito nel periodo seguente, fino a raggiungere la copertura del 98%, superando ampiamente il valore di accettabilità. Questo risultato conferma un'evoluzione graduale ma solida dell'attività di testing, che si è dimostrata efficace e ben distribuita sull'intero codice base, garantendo un'elevata affidabilità e qualità del software prodotto.