

Piano di Qualifica

versione 2.0.0



7commits@gmail.com

Progetto di Ingegneria del Software

A.A. 2024/2025

Destinatari	Responsabile	Redattori	Verificatori
Prof. Tullio Vardanega	Marco Cola	Stefano Dal Poz	Giulia Hu
Prof. Riccardo Cardin		Marco Cola	Ruize Lin
Gruppo 🇮🇹 7Commits		Ruize Lin	Marco Cola
		Mattia Piva	Giada Rossi

Registro delle modifiche

Versione	Data	Autori	Verificatori	Descrizione
v2.0.0	2025-08-25	Marco Cola	Ruize Lin	Approvazione per PB
v1.1.4	2025-08-25	Marco Cola	Giada Rossi, Ruize Lin	Completata la sezione 5 relativa alla copertura dei requisiti
v1.1.3	2025-08-24	Giada Rossi	Marco Cola	Popolata tabella dei test di regressione
v1.1.2	2025-08-22	Marco Cola	Ruize Lin, Giada Rossi	Corretta impostazione delle sezioni e profondità della numerazione
v1.1.1	2025-08-19	Marco Cola	Ruize Lin, Giada Rossi	Aggiunti collegamenti al Glossario v2.0.0
v1.1.0	2025-08-19	Giada Rossi	Marco Cola	Popolata tabella dei test di integrazione
v1.0.6	2025-08-18	Giada Rossi	Marco Cola	Terminata tabella test di unità
v1.0.5	2025-08-17	Giada Rossi	Marco Cola	Inizio stesura test di unità
v1.0.4	2025-08-14	Giada Rossi	Marco Cola	Popolata e completata tabella dei test di accettazione
v1.0.3	2025-08-13	Giada Rossi	Marco Cola	Popolata e completata tabella dei test di sistema
v1.0.2	2025-08-09	Giada Rossi	Marco Cola	Iniziata sistemazione e integrazione del capitolo «Metodologie di testing» e relativi sottocapitoli
v1.0.1	2025-08-09	Giada Rossi	Marco Cola	Rimossa sezione «Iniziative di automiglioramento» e modificata struttura del capitolo 4
v1.0.0	2025-06-12	Marco Cola	Ruize Lin	Approvazione per RTB
v0.1.1	2025-06-04	Marco Cola	Ruize Lin	Aggiunta sezione 5 «Considerazioni finali»
v0.1.0	2025-06-03	Marco Cola	Ruize Lin	Corretti grafici e struttura dei grafici, con unionie grafici CV/SV, ETC/AC, EV/PV. Aggiunti NCR e RSI + Gulpease
v0.0.9	2025-05-30	Giada Rossi	Ruize Lin, Giulia Hu	Aggiunti grafici ETC, EAC
v0.0.8	2025-05-29	Giada Rossi	Ruize Lin, Giulia Hu	Aggiunti grafici AC, PV, EV

v0.0.7	2025-05-19	Stefano Dal Poz	Ruize Lin	Aggiunti riferimenti al Glossario
v0.0.6	2025-05-18	Mattia Piva	Ruize Lin	Ultimata sezione test di sistema e aggiunti valori sugli obiettivi metrici di qualità
v0.0.5	2025-05-16	Giada Rossi	Ruize Lin	Prima bozza test di sistema
v0.0.4	2025-05-12	Giada Rossi	Michele Ogniben, Ruize Lin	Inizio paragrafo 3 Metodologia di testing
v0.0.3	2025-05-09	Giada Rossi	Michele Ogniben, Ruize Lin	Obiettivi metrici di qualità
v0.0.2	2025-05-06	Giada Rossi	Michele Ogniben	Capitolo Introduzione
v0.0.1	2025-04-14	Marco Cola	Michele Ogniben	Prima Bozza

Indice

1. Introduzione	6
1.1. Scopo del documento	6
1.2. Scopo del prodotto	6
1.3. Glossario	6
1.4. Riferimenti	6
1.4.1. Riferimenti normativi	6
1.4.2. Riferimenti informativi	6
2. Obiettivi metrici di qualità	8
2.1. Qualità di processo	8
2.1.1. Processi primari	8
2.1.1.1. Fornitura	8
2.1.1.2. Sviluppo	8
2.1.2. Processi di supporto	8
2.1.2.1. Documentazione	8
2.1.2.2. Gestione della qualità	8
2.2. Qualità di prodotto	9
2.2.1. Funzionalità	9
2.2.2. Affidabilità	9
2.2.3. Manutenibilità	9
2.2.4. Efficienza	9
3. Metodologie di testing	10
3.1. Codifica dei test	10
3.1.1. Test di unità	11
3.1.2. Test di integrazione	16
3.1.3. Test di sistema	17
3.1.4. Test di regressione	20
3.1.5. Test di accettazione	21
4. Cruscotto di valutazione	23
4.1. EAC (Estimated At Completion)	23
4.1.1. RTB	23
4.1.2. PB	23
4.2. Earned Value (EV) e Planned Value (PV)	24
4.2.1. RTB	24
4.2.2. PB	24
4.3. Actual Cost (AC) e Estimate to Complete (ETC)	25
4.3.1. RTB	25
4.3.2. PB	25
4.4. Cost Variance (CV) e Schedule Variance (SV)	26
4.4.1. RTB	26
4.4.2. PB	26
4.5. Requirements stability index (RSI)	27
4.5.1. RTB	27
4.5.2. PB	27
4.6. Indice Gulpease	28
4.6.1. RTB	28
4.6.2. PB	28
4.7. Non-Calculated Risk	29
4.7.1. RTB	29

4.7.2. PB	29
5. Copertura dei requisiti	30
5.1. Copertura dei requisiti obbligatori	30
5.1.1. PB	30
5.2. Copertura dei requisiti desiderabili	31
5.2.1. PB	31
5.3. Copertura dei requisiti opzionali	32
5.3.1. PB	32

1. Introduzione

1.1. Scopo del documento

Il presente documento ha l'obiettivo di delineare le procedure di verifica e validazione adottate per assicurare la *qualità_G* sia dei processi che del prodotto software. Essendo il miglioramento continuo un principio fondamentale dei sistemi di *qualità_G*, il contenuto del documento sarà aggiornato e ampliato progressivamente, con l'intento di riflettere i risultati delle verifiche eseguite e di garantire il costante miglioramento del progetto nel tempo.

1.2. Scopo del prodotto

Lo scopo principale del *prodotto_G* è quello di fornire all'azienda *proponente_G*, Zucchetti S.p.A., una piattaforma *software_G* denominata Artificial QI, pensata per supportare la valutazione delle prestazioni di sistemi basati su Large Language Models (LLM). Il *sistema_G* nasce dall'esigenza di disporre di uno strumento centralizzato, automatizzato e configurabile che consenta di testare, analizzare e confrontare le risposte generate da diversi modelli *LLM_G* rispetto a risposte attese predefinite.

La piattaforma dovrà permettere all'utente sviluppatore di inserire e gestire domande e risposte attese, interfacciarsi con *LLM_G* esterni tramite chiamate *API_G* conformi allo standard *OpenAPI 3.1_G*, ed eseguire valutazioni automatiche della coerenza tra input e output tramite metriche personalizzabili (come *BM25_G* o modelli neurali). Inoltre, dovrà offrire *funzionalità_G* avanzate di visualizzazione, ordinamento e confronto dei risultati ottenuti, favorendo anche l'archiviazione e l'analisi storica delle esecuzioni.

Il *prodotto_G* finale sarà un'unica *applicazione_G* integrata, che racchiude tutte le *funzionalità_G* necessarie in un flusso di lavoro continuo e organico. Tale soluzione dovrà garantire semplicità d'uso, flessibilità nella configurazione e supporto per l'analisi evolutiva delle prestazioni dei modelli, diventando così un valido supporto nelle *attività_G* di test e selezione degli *LLM_G* più adatti per specifici scenari applicativi.

1.3. Glossario

Per evitare ambiguità relative alla terminologia utilizzata, il documento fa riferimento a un Glossario, che raccoglie definizioni chiare e univoche dei termini specifici del dominio d'uso. I termini inclusi nel glossario sono contrassegnati nel testo con una G a pedice, in *questo modo_G*.

1.4. Riferimenti

1.4.1. Riferimenti normativi

- Norme di Progetto v2.0.0
- Capitolato d'appalto: <https://www.math.unipd.it/~tullio/IS-1/2024/Progetto/C1.pdf>
- Regolamento del progetto didattico: <https://www.math.unipd.it/~tullio/IS-1/2024/Dispense/PD1.pdf>

1.4.2. Riferimenti informativi

- Glossario v2.0.0
- Qualità del software: <https://www.math.unipd.it/~tullio/IS-1/2024/Dispense/T07.pdf>
- Qualità del processo: <https://www.math.unipd.it/~tullio/IS-1/2024/Dispense/T08.pdf>
- Verifica e validazione - Introduzione: <https://www.math.unipd.it/~tullio/IS-1/2024/Dispense/T09.pdf>

- Verifica e validazione - Analisi dinamica: <https://www.math.unipd.it/~tullio/IS-1/2024/Dispense/T11.pdf>
- Verifica e validazione - Analisi statica: <https://www.math.unipd.it/~tullio/IS-1/2024/Dispense/T10.pdf>

2. Obiettivi metrici di qualità

Ogni *processo_G* viene valutato mediante l'applicazione di metriche specifiche, le cui specifiche dettagliate sono riportate nel documento *Norme di Progetto v2.0.0*. Questa sezione delinea i criteri che le metriche devono rispettare per essere valutate come accettabili o eccellenti.

2.1. Qualità di processo

La qualità di processo ha come obiettivo quello di garantire che gli standard definiti nel documento *Norme di Progetto v2.0.0* vengano rispettati sempre in modo da svolgere al meglio l'intera gamma di *attività_G*. Questo serve per monitorare costantemente l'*efficacia_G* e l'*efficienza_G* dei processi.

2.1.1. Processi primari

2.1.1.1. Fornitura

Metrica	Descrizione	Valore accettabile	Valore ideale
CV	Cost Variance	$\geq -10\%$	≥ 0
PV	Planned Value	≥ 0	$\leq \text{BAC}$
EV	Earned Value	≥ 0	$\leq \text{EAC}$
AC	Actual Cost	≥ 0	$\leq \text{EAC}$
CPI	Cost Performance Index	$\pm 10\%$	0%
SPI	Schedule Performance Index	≥ 0.9	≥ 1
BAC	Budget At Completion	-	-
EAC	Estimated At Completion	$\leq 1.1 \times \text{BAC}$	$= \text{BAC}$
VAC	Variance At Completion	$\geq -10\%$	≥ 0
ETC	Estimated To Completion	≥ 0	$\leq \text{EAC}$
SV	Schedule Variance	$\geq -10\%$	≥ 0
BV	Budget Variance	$\pm 10\%$	0%

2.1.1.2. Sviluppo

Metrica	Descrizione	Valore accettabile	Valore ideale
SC	Statement Coverage	$\geq 70\%$	100%

2.1.2. Processi di supporto

2.1.2.1. Documentazione

Metrica	Descrizione	Valore accettabile	Valore ideale
IG	Indice Gulpease	$\geq 60\%$	$\geq 80\%$

2.1.2.2. Gestione della qualità

Metrica	Descrizione	Valore accettabile	Valore ideale
MNS	Metriche Non Soddisfatte	≤ 3	0

2.2. Qualità di prodotto

La qualità del prodotto indica quanto un *software_G* soddisfa requisiti ed esigenze, sia esplicite che implicite. Garantirla significa offrire un *prodotto_G* stabile, efficiente e usabile. L'uso di metriche specifiche consente di valutare aspetti chiave come *funzionalità_G*, affidabilità e manutenibilità, permettendo interventi rapidi in caso di scostamenti dagli standard attesi.

2.2.1. Funzionalità

Metrica	Descrizione	Valore accettabile	Valore ideale
ROS	Requisiti Obbligatori Soddisfatti	100%	100%
RDS	Requisiti Desiderabili Soddisfatti	≥ 50%	100%
RPS	Requisiti Opzionali Soddisfatti	≥ 0%	≥ 50%

2.2.2. Affidabilità

Metrica	Descrizione	Valore accettabile	Valore ideale
PTCP	Passed Test Cases Percentage	≥ 75%	100%
CC	Code Coverage	≥ 80%	100%

2.2.3. Manutenibilità

Metrica	Descrizione	Valore accettabile	Valore ideale
SFIN	Structure Fan IN	-	Va massimizzato
SFOUT	Structure Fan OUT	-	Va minimizzato

2.2.4. Efficienza

Metrica	Descrizione	Valore accettabile	Valore ideale
TDE	Tempo di Elaborazione	≤ 15 minuti	≤ 5 minuti

3. Metodologie di testing

In questa sezione vengono illustrate le modalità con cui sono stati individuati e classificati i *test_G* da applicare al progetto per verificare il rispetto delle metriche previste e le specifiche del documento *Analisi dei Requisiti_G* v2.0.0.

I *test_G* sono suddivisi nelle seguenti categorie:

- **Test di Unità:** riguardano le singole unità *software_G* sviluppate dai programmatori del team ed eseguite in modo automatizzato;
- **Test di Integrazione:** verificano l'integrazione corretta tra le varie parti del *sistema_G*, evidenziando eventuali difetti a livello architetturale, per questo sono svolti successivamente ai precedenti *test_G*
- **Test di Sistema:** assicurano che l'intero prodotto software funzioni correttamente nel suo insieme, verificando il rispetto del documento *Analisi dei Requisiti_G*
- **Test di regressione:** verificano che modifiche apportate a specifiche unità non causino danni al resto del sistema. Essi consistono nella ripetizione mirata di test di unità, integrazione e sistema;
- **Test di Accettazione:** verificano che il *prodotto_G* finale sia conforme a quanto richiesto nel *capitolato_G* per questo vengono fatti alla presenza del *committente_G*.

3.1. Codifica dei test

Ogni *test_G* viene associato ad un codice univoco nel formato **T[Tipologia]-[Numero]**, dove Tipologia indica la tipologia del *test_G*:

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

Inoltre, ogni *test_G* appartiene ad uno Stato, che può essere:

- **V:** verificato, cioè il test ha esito positivo;
- **E:** non verificato, cioè il test ha esito negativo;
- **NI:** non implementato.

3.1.1. Test di unità

I test di unità si concentrano sulla verifica del corretto comportamento delle singole parti del codice. Con “unità” si fa riferimento a funzioni, classi o qualsiasi componente isolata che esegue compiti specifici all’interno del programma.

Codice	Descrizione	Stato
TU-01	Verificare che la funzione <code>start_new_preset_edit()</code> inizializzi correttamente lo stato della sessione per un nuovo preset	V
TU-02	Verificare che la funzione <code>start_existing_preset_edit(preset_id)</code> carichi correttamente un preset esistente nello stato della sessione	V
TU-03	Verificare che la funzione <code>save_preset_from_form()</code> salvi correttamente un preset se valido e aggiorni lo stato della sessione oppure generi un messaggio di errore	V
TU-04	Verificare che la funzione <code>delete_preset_callback(preset_id)</code> elimini correttamente un preset e resettì lo stato della sessione	V
TU-05	Verificare che la funzione <code>validate_preset(preset)</code> controlli che il preset sia valido (nome non vuoto, non duplicato)	V
TU-06	Verificare che la funzione <code>save_preset(preset)</code> salvi un nuovo preset o ne aggiorni uno esistente, aggiornando poi l’elenco dei preset	V
TU-07	Verificare che la funzione <code>delete_preset(preset_id)</code> elimini un preset dall’elenco e aggiorni lo stato	V
TU-08	Verificare che la funzione <code>test_api_connection</code> provi la connessione all’API e restituisca esito positivo o errore	V
TU-09	Verificare che il metodo <code>APIPreset.load_all()</code> carichi tutti i preset dal database e li restituisca come oggetti <code>APIPreset</code> .	V
TU-10	Verificare che il metodo <code>APIPreset.save(presets)</code> inserisca nuovi preset o aggiorni quelli già esistenti nel database.	V
TU-11	Verificare che il metodo <code>APIPreset.delete(preset_id)</code> elimini dal database il preset con l’id specificato	V
TU-12	Verificare che il metodo <code>st.set_page_config</code> configuri la pagina principale dell’app (titolo, layout, ecc.)	V
TU-13	Verificare che il metodo <code>st.sidebar.radio</code> gestisca la navigazione tra le pagine dell’app	V
TU-14	Verificare che la funzione <code>get_questions()</code> restituisca la lista delle domande, sfruttando la cache	V
TU-15	Verificare che la funzione <code>refresh_questions()</code> aggiorni e ricarichi la cache delle domande	V
TU-16	Verificare che la funzione <code>get_question_sets()</code> restituisca i set di domande, sfruttando la cache.	V
TU-17	Verificare che la funzione <code>refresh_question_sets()</code> aggiorni e ricarichi la cache dei set di domande	V
TU-18	Verificare che la funzione <code>get_api_presets()</code> restituisca i preset API, sfruttando la cache	V
TU-19	Verificare che la funzione <code>refresh_api_presets()</code> aggiorni e ricarichi la cache dei preset API	V

TU-20	Verificare che la funzione <code>get_results()</code> restituisca i risultati dei test, sfruttando la cache	V
TU-21	Verificare che la funzione <code>refresh_results()</code> aggiorni e ricarichi la cache dei risultati dei test	V
TU-22	Verificare che il metodo <code>create_card(title, content, icon, is_success)</code> crei una card HTML con un titolo, contenuto e icona	V
TU-23	Verificare che il metodo <code>create_metrics_container(metrics)</code> crei un contenitore HTML per visualizzare le metriche	V
TU-24	Verificare che il metodo <code>format_questions_for_view(df)</code> prenda in input un DataFrame di domande e lo formatta in modo che ogni domanda abbia una categoria	V
TU-25	Verificare che il metodo <code>build_questions_detail(question_map, question_ids)</code> costruisca i dettagli per un elenco di domande	V
TU-26	Verificare che la funzione <code>set_llm_mode_callback()</code> cambi lo stato della modalità di test in «Valutazione Automatica con LLM» e segni che la modalità è stata modificata	V
TU-27	Verificare che la funzione <code>run_llm_mode_callback()</code> aggiorni lo stato del test automatico LLM impostandolo su True per indicare che il test deve partire	V
TU-28	Verificare che la funzione <code>evaluate_answer</code> restituisca punteggi riguardanti la valutazione effettuata su risposta data e risposta attesa relative ad una domanda	V
TU-29	Verificare che la funzione <code>read_questions</code> legga i file CSV e i file JSON di domande e risposte attese e li organizzi in colonne con id, domanda, risposta_attesa, categoria	V
TU-30	Verificare che la funzione <code>read_question_sets</code> legga file CSV di set di domande raggruppandoli sotto un nome	V
TU-31	Verificare che la funzione <code>read_test_results</code> legga file CSV o JSON con i risultati dei test e li organizzi in tabella	V
TU-32	Verificare che la funzione <code>read_question_sets</code> legga file CSV di set di domande raggruppandoli sotto un nome	V
TU-33	Verificare che la funzione <code>filter_new_rows</code> filtri correttamente i dati presenti ed elimini i duplicati restituendo il conteggio corretto	V
TU-34	Verificare che la funzione <code>write_dataset</code> scriva correttamente i dati su file CSV e JSON	V
TU-35	Verificare che la funzione <code>create_save_question_callback</code> registri correttamente il successo o il fallimento del salvataggio di una domanda e se l'interfaccia venga aggiornata di conseguenza	V
TU-36	Verificare che la funzione <code>import_questions_callback</code> registri correttamente il successo o il fallimento dell'importazione di domande, evidenziando quanti elementi sono stati importati ed eventuali avvisi/errori	V
TU-37	Verificare che la funzione <code>confirm_delete_question_dialog</code> registri correttamente il successo o il fallimento dell'eliminazione di una domanda	V
TU-38	Verificare che la funzione <code>create_save_set_callback</code> registri correttamente il successo o il fallimento del salvataggio di un set di domande	V

TU-39	Verificare che la funzione <code>import_set_callback</code> registri correttamente il successo o il fallimento dell'importazione di un set di domande da file, evidenziando quanti elementi sono stati importati ed eventuali avvisi/errori	V
TU-40	Verificare che la funzione <code>confirm_delete_set_dialog_success</code> registri correttamente il successo o il fallimento dell'eliminazione di un set di domande	V
TU-41	Verificare che la funzione <code>home_render_injects_styles_and_content</code> inserisca gli stili chiamando la funzione apposita e mostri il contenuto previsto	V
TU-42	Verificare che la funzione <code>import_from_file_skips_duplicates_and_saves</code> importi correttamente i risultati da file, quindi controllando i duplicati, salvando correttamente i dati e aggiornando la cache	V
TU-43	Verificare che la funzione <code>question_gather_data</code> importi correttamente le domande	V
TU-44	Verificare che la funzione <code>question_set_gather_data</code> importi correttamente i set di domande	V
TU-45	Verificare che la funzione <code>test_result_gather_data</code> restituisca i risultati dei test correttamente in forma tabellare	V
TU-46	Verificare che la funzione <code>initialize_db_logs</code> verifichi la correttezza creazione o no del Database con conseguente log	V
TU-47	Verificare che la funzione <code>get_engine()</code> controlli che il client DB venga creato usando la config, e che il motore venga riutilizzato	V
TU-48	Verificare che la funzione <code>ensure_database_error</code> verifichi che il database esista e, se necessario, lo crei	V
TU-49	Verificare che la funzione <code>orm_tables_and_relationships</code> controlli struttura, relazioni e salvataggio dei dati per tutte le tabelle ORM principali	V
TU-50	Verificare che la funzione <code>add_and_update_question</code> inserisca e aggiorni correttamente le domande	V
TU-51	Verificare che la funzione <code>test_persist_entities_handles_duplicates</code> gestisca correttamente duplicati e avvisi nel caso di caricamenti massivi di domande	V
TU-52	Verificare che la funzione <code>create_and_update_question_set</code> crei e aggiorni i set di domande correttamente, anche in caso di id inesistenti	V
TU-53	Verificare che la funzione <code>resolve_question_ids</code> riconosca domande già presenti, aggiunga quelle nuove, salti quelle incomplete e restituisca id e avvisi	V
TU-54	Verificare che la funzione <code>add_and_persist_entities</code> verifichi che i risultati dei test vengano salvati correttamente, evitando duplicati già presenti	V
TU-55	Verificare che la funzione <code>calculate_statistics</code> calcoli statistiche aggregate dai risultati delle singole domande	V
TU-56	Verificare che le funzioni <code>get_openai_client</code> creino il client OpenAI solo con chiavi valide, gestendo errori ed endpoint personalizzati.	V
TU-57	Verificare che le funzioni <code>get_available_models</code> filtrino via i modelli di tipo embedding e mostri solo quelli pertinenti alla chat	V
TU-58	Verificare che le funzioni <code>get_openai_client</code> generino errori nei casi di mancanza di chiavi API e il client non parte	V
TU-59	Verificare che le funzioni <code>generate_answer</code> generino correttamente una risposta o errore nel caso di mancanza del client	V

TU-60	Verificare che le funzioni test_api_connection generino connessione o, in caso contrario, errore	V
TU-61	Verificare che le funzioni add_question gestiscano l'aggiunta di nuove domande, controllino se esistono già e aggiornino la cache	V
TU-62	Verificare che la funzione update_question aggiorni una domanda esistente e successivamente la cache	V
TU-63	Verificare che le funzioni delete_question eliminino una domanda e aggiornino la cache	V
TU-64	Verificare che la funzione delete_question esporti correttamente le domande in un file in formato CSV	V
TU-65	Verificare che le funzioni add_question_if_not_exists gestiscano l'aggiunta di nuove domande, controllino se esistono già e aggiornino la cache	V
TU-66	Verificare che le funzioni save_question_action salvino modifiche a una domanda e aggiornino la cache	V
TU-67	Verificare che le funzioni import_questions_action importino domande da file e gestiscano risultati o errori	V
TU-68	Verificare che la funzione create_set_controller gestisca la creazione e l'aggiornamento dei set di domande, con refresh della cache	V
TU-69	Verificare che la funzione update_set_controller aggiorni set esistente e cache	V
TU-70	Verificare che la funzione delete_set_controller elimini set e aggiorni la cache	V
TU-71	Verificare che la funzione prepare_sets_for_view filtri i set per categoria e aggiunga dettagli delle domande	V
TU-72	Verificare che la funzione export_sets_action esporti i set in file in formato CSV	V
TU-73	Verificare che la funzione import_from_file_handles_duplicates importi set e domande, segnali duplicati, aggiunga nuove domande e generi avvisi	V
TU-74	Verificare che la funzione register_page_prevents_duplicates gestisca il registro delle pagine e prevenga i duplicati	V
TU-75	Verificare che la funzione get_results_filters filtri correttamente i risultati per set e modello	V
TU-76	Verificare che la funzione list_set_names restituisca i nomi corretti dei set dai risultati	V
TU-77	Verificare che la funzione prepare_select_options generi le opzioni di selezione formattate correttamente con data, set, media e metodo	V
TU-78	Verificare che la funzione initialize_session_state scriva tutte le chiavi richieste nello stato della sessione con i valori iniziali corretti	V
TU-79	Verificare che la funzione ensure_keys non sovrascriva chiavi già presenti e aggiunga solo quelle mancanti	V
TU-80	Verificare che la funzione get_initial_state() costruisca correttamente lo stato iniziale dell'applicazione	V
TU-81	Verificare che la funzione setup_logging crei effettivamente un file di log e scriva i messaggi	V
TU-82	Verificare che la funzione load_default_config() carichi correttamente la configurazione di default	V
TU-83	Verificare che la funzione add_global_styles() inietti CSS globale per personalizzare componenti Streamlit	V

TU-84	Verificare che la funzione <code>add_page_header</code> aggiunga un'intestazione di pagina con icona e descrizione, richiamando gli stili globali	V
TU-85	Verificare che la funzione <code>add_section_title</code> renda un titolo di sezione con eventuale icona	V
TU-86	Verificare che la funzione <code>add_home_styles()</code> aggiunga stili specifici per la home page	V
TU-87	Verificare che la funzione <code>aimport_results_action(file)</code> importi risultati da un file JSON	V
TU-88	Verificare che la funzione <code>run_test</code> esegua un test, generi risposte e valuti le domande.	V
TU-89	Verificare che la funzione <code>export_results_action</code> esporti i risultati su file	V

3.1.2. Test di integrazione

I test di integrazione servono a controllare che i vari componenti di un'*applicazione_G* funzionino integrandosi tra di loro senza errori. Questi *test_G* hanno l'obiettivo di individuare errori che possono emergere quando singoli moduli o unità di codice vengono collegati tra loro.

Codice	Descrizione	Stato
TI-01	Verificare che il metodo <code>controllers.test_controller.import_results_action</code> legga correttamente i risultati con il metodo <code>test_result_importer.import_from_file</code> , li elabori e aggiorni lo stato con il metodo <code>load_results</code>	V
TI-02	Verificare che il metodo <code>controllers.test_controller.export_results_action</code> prenda i risultati presenti e li salvi su file tramite il metodo <code>test_result_importer.export_to_file</code>	V
TI-03	Verificare che il metodo <code>controllers.test_controller.run_test</code> , dopo aver generato e valutato le risposte, salvi i risultati nel database con il metodo <code>TestResult.add_and_refresh</code>	V
TI-04	Verificare che il metodo <code>utils.startup_utils.initialize_database</code> avvii correttamente il Database	V
TI-05	Verificare che il metodo <code>models.Question.load_all</code> recuperi per il database tutte le domande disponibili per un set	V
TI-06	Verificare che il metodo <code>models.TestResult.load_all_df</code> restituisca i risultati dal database come DataFrame pandas	V
TI-07	Verificare che il metodo <code>models.TestResult.add_and_refresh</code> inserisca un nuovo test nel database e restituisca l'id	V
TI-08	Verificare che il metodo <code>views.visualizza_risultati.render</code> carichi i risultati dal database con il metodo <code>get.results</code> e i set con il metodo <code>load_sets</code> , mostrandoli a interfaccia	V
TI-09	Verificare che il metodo <code>Question.load_all</code> legga tutte le domande presenti nel database	V

3.1.3. Test di sistema

I *test_G* di sistema rappresentano una *fase_G* fondamentale del *processo_G* di verifica del *software_G*, in cui viene valutato il comportamento dell'intero applicativo una volta che tutte le sue componenti sono state integrate. L'obiettivo è accertarsi che il *sistema_G*, nel suo complesso, soddisfi i requisiti funzionali e non funzionali definiti nell'*Analisi dei Requisiti_G*. Questa *attività_G* viene svolta in un ambiente che riproduce il più fedelmente possibile le condizioni d'uso reali, simulando l'interazione dell'utente finale con il *sistema_G*.

Codice	Descrizione	Requisito	Stato
TS-01	Verificare che l'utente possa visualizzare la lista delle domande presenti	RFO01	V
TS-02	Verificare che l'utente possa visualizzare una singola domanda tra quelle presenti nella lista	RFO02	V
TS-03	Verificare che l'utente possa modificare una singola domanda	RFO03	V
TS-04	Verificare che l'utente possa modificare una risposta attesa	RFO04	V
TS-05	Verificare che l'utente possa eliminare una singola domanda tra quelle presenti nella lista	RFO05	V
TS-06	Verificare che l'utente possa annullare l'operazione di eliminazione di una singola domanda	RFD06	V
TS-07	Verificare che l'utente possa filtrare le domande in base alla categoria di appartenenza	RFD07	V
TS-08	Verificare che l'utente possa ricevere un messaggio di errore nel caso in cui l'aggiunta manuale della domanda non sia corretta e/o completa	RFO08	V
TS-09	Verificare che l'utente possa aggiungere manualmente una domanda nel sistema	RFO09	V
TS-10	Verificare che l'utente possa caricare nel sistema un file CSV contenente le domande e le relative risposte attese	RFO10	V
TS-11	Verificare che l'utente possa caricare nel sistema un file JSON contenente le domande e le relative risposte attese	RFO11	V
TS-12	Verificare che l'utente possa ricevere una notifica di errore nel caso in cui il file di importazione delle domande non sia corretto	RFO12	V
TS-13	Verificare che l'utente possa visualizzare la lista dei set di domande che si trovano salvati nel sistema	RFO13	V
TS-14	Verificare che l'utente possa visualizzare un singolo set di domande tra quelli presenti nella lista	RFO14	V
TS-15	Verificare che l'utente possa modificare un set di domande	RFO15	V
TS-16	Verificare che l'utente possa eliminare un set di domande tra quelli presenti	RFO16	V
TS-17	Verificare che l'utente possa annullare l'operazione di eliminazione di un set di domande	RFD17	V
TS-18	Verificare che l'utente possa filtrare i set di domande in base alla categoria di appartenenza	RFD18	V
TS-19	Verificare che l'utente possa modificare e aggiornare manualmente un set di domande tra quelli presenti	RFP19	V

TS-20	Verificare che l'utente possa ricevere una notifica di errore nel caso in cui il nome del set di domande non sia stato inserito correttamente	RFO20	V
TS-21	Verificare che l'utente possa caricare nel sistema un file JSON contenente un set di domande	RFO21	V
TS-22	Verificare che l'utente possa caricare nel sistema un file CSV contenente un set di domande	RFO22	V
TS-23	Verificare che l'utente possa ricevere una notifica di errore nel caso in cui il file di importazione del set di domande non sia corretto e valido	RFO23	V
TS-24	Verificare che l'utente possa accedere alla sezione di configurazione dell'API e creare un preset	RFO24	V
TS-25	Verificare che l'utente possa ricevere una notifica di errore nel caso in cui il nome del preset dell'API non sia stato inserito	RFO25	V
TS-26	Verificare che l'utente possa effettuare un test di connessione dopo aver compilato i campi necessari per la creazione di un preset API	RFP26	V
TS-27	Verificare che l'utente possa ricevere una notifica di errore nel caso in cui l'esito del test effettuato sia risultato negativo	RFP27	V
TS-28	Verificare che l'utente possa visualizzare la lista dei preset che sono stati configurati	RFO28	V
TS-29	Verificare che l'utente possa modificare un preset precedentemente configurato	RFO29	V
TS-30	Verificare che l'utente possa eliminare un preset API configurato precedentemente	RFO30	V
TS-31	Verificare che sistema sia in grado di inviare domande a un LLM esterno tramite API e riceverne le risposte	RFO31	V
TS-32	Verificare che l'utente possa visualizzare tutti i preset API salvati all'interno della sezione di esecuzione test	RFO32	V
TS-33	Verificare che l'utente possa visualizzare tutti i set di domande salvati all'interno della sezione di esecuzione test	RFO33	V
TS-34	Verificare che l'utente possa visualizzare il punteggio o il giudizio qualitativo assegnato dal sistema in seguito al confronto tra la risposta attesa e la risposta generata dall'LLM	RFO34	V
TS-35	Verificare che l'utente possa visualizzare i risultati ottenuti dal confronto tra risposte attese e risposte dell'LLM	RFO35	V
TS-36	Verificare che l'utente possa visualizzare i risultati ottenuti dal confronto tra risposte attese e risposte dell'LLM relativi ad una singola domanda	RFO36	V
TS-37	Verificare che l'utente possa visualizzare le metriche applicate ad una singola risposta	RFO37	V
TS-38	Verificare che l'utente possa salvare e consultare i risultati del test (risposte, valutazioni e metadati)	RFO38	V
TS-39	Verificare che l'utente possa effettuare il download dei risultati di un test effettuato	RFP39	V
TS-40	Verificare che l'utente possa caricare nel sistema un file di risultati di test effettuati precedentemente	RFP40	V

TS-41	Verificare che l'utente possa ricevere una notifica di errore nel caso in cui il caricamento del file contenente i risultati non sia andato a buon fine	RFO41	V
TS-42	Verificare che l'utente possa visualizzare i test effettuati in precedenza	RFO42	V
TS-43	Verificare che l'utente possa visualizzare tutti i dettagli di un determinato test effettuato in precedenza	RFP43	V
TS-44	Verificare che l'utente possa filtrare lo storico dei test in base a diversi criteri	RFD44	V
TS-45	Verificare che l'utente possa confrontare tra loro due diverse esecuzioni di test	RFP45	V
TS-46	Verificare che l'utente possa confrontare tra loro due diverse risposte LLM per una singola domanda specifica	RFP46	V

3.1.4. Test di regressione

I *test_G* di regressione hanno lo scopo di garantire che, dopo modifiche o aggiornamenti del *software_G*, il *sistema_G* continui a funzionare correttamente senza introdurre nuovi problemi. In pratica servono a scoprire se errori già risolti si ripresentano o se ne compaiono di nuovi a causa delle modifiche effettuate.

Per farlo, si riutilizza un insieme di casi di test già definiti precedentemente, di unità, di integrazione e di sistema, che coprono le parti più importanti dell'applicazione. Ogni volta che il codice viene aggiornato, questo insieme viene rieseguito per confermare che le funzionalità già validate in passato mantengano il comportamento previsto.

Test di unità selezionati
TU-03, TU-08, TU-14, TU-20, TU-46, TU-56, TU-88

Test di integrazione selezionati
TI-01, TI-03, TI-08

Test di sistema selezionati
TS-01, TS-09, TS-13, TS-24, TS-26, TS-31, TS-34, TS-38, TS-42, TS-45

I test di unità, integrazione e sistema scelti coprono le parti più importanti dell'applicazione, garantendo che le funzionalità principali non vengano compromesse dopo modifiche o aggiornamenti.

Principalmente riguardano quindi:

- la gestione dei dati, quindi delle domande, dei set e dei preset;
- la comunicazione con *LLM_G* e *API_G*
- l'esecuzione e la valutazione dei test;
- la persistenza e l'archivio dei dati;

3.1.5. Test di accettazione

I *test_G* di accettazione hanno lo scopo di verificare che il prodotto rispetti pienamente i requisiti utente specificati nel *capitolato_G*. Questi test vengono condotti con la presenza del committente e servono a dimostrare che il prodotto soddisfi le aspettative attraverso l'esecuzione delle prove previste. Un esito positivo di questi test nel collaudo finale è solitamente il passaggio decisivo per il rilascio ufficiale del prodotto.

Codice	Descrizione	Stato
TA-01	Verificare che l'utente possa visualizzare la lista delle domande presenti nel sistema	V
TA-02	Verificare che l'utente possa visualizzare una singola domanda	V
TA-03	Verificare che l'utente possa inserire una nuova domanda nel sistema	V
TA-04	Verificare che l'utente possa modificare una domanda già presente	V
TA-05	Verificare che l'utente possa eliminare una domanda presente nel sistema	V
TA-06	Verificare che l'utente possa modificare una risposta attesa già presente	V
TA-07	Verificare che l'utente possa annullare l'operazione di modifica di una domanda presente nel sistema	V
TA-08	Verificare che l'utente possa annullare l'operazione di modifica di una risposta attesa nel sistema	V
TA-09	Verificare che l'utente possa creare un set di domande presente nel sistema	V
TA-10	Verificare che l'utente possa visualizzare un set di domande presente nel sistema	V
TA-11	Verificare che l'utente possa modificare il nome di un set di domande presente nel sistema	V
TA-12	Verificare che l'utente possa cancellare un set di domande presente nel sistema	V
TA-13	Verificare che l'utente possa gestire i modelli LLM, visualizzando, aggiungendo, selezionando e rimuovendo un LLM	V
TA-14	Verificare che il test venga avviato correttamente e che il sistema registri le risposte	V
TA-15	Verificare che l'utente possa visualizzare e confrontare i risultati dei test effettuati	V
TA-16	Verificare che l'utente possa scaricare i risultati	V
TA-17	Verificare che sia presente un'interfaccia user-friendly e che l'utente possa archiviare una lista di domande e risposte attese	V
TA-18	Verificare che sia presente un'interfaccia user-friendly e che l'utente possa importare un file di dati nell'archivio	V
TA-19	Verificare che l'utente possa visualizzare lo storico dei test e dei risultati effettuati precedentemente	V
TA-20	Verificare che sia presente un'interfaccia user-friendly e che l'utente possa modificare i dati presenti nell'archivio	V
TA-21	Verificare che sia presente un'interfaccia user-friendly e che l'utente possa eliminare i dati presenti nell'archivio.	V
TA-22	Verificare che sia presente un'interfaccia user-friendly e che l'utente possa annullare l'operazione di eliminazione dei dati presenti nell'archivio.	V
TA-23	Verificare che sia stata prodotta tutta la documentazione minima richiesta per il corso di Ingegneria del Software	V
TA-24	Verificare che l'utente possa consultare il manuale per l'utilizzo del prodotto	V

TA-25	Verificare che un utente sviluppatore possa consultare un manuale tramite cui estendere l'applicazione	V
-------	--	---

4. Cruscotto di valutazione

4.1. EAC (Estimated At Completion)

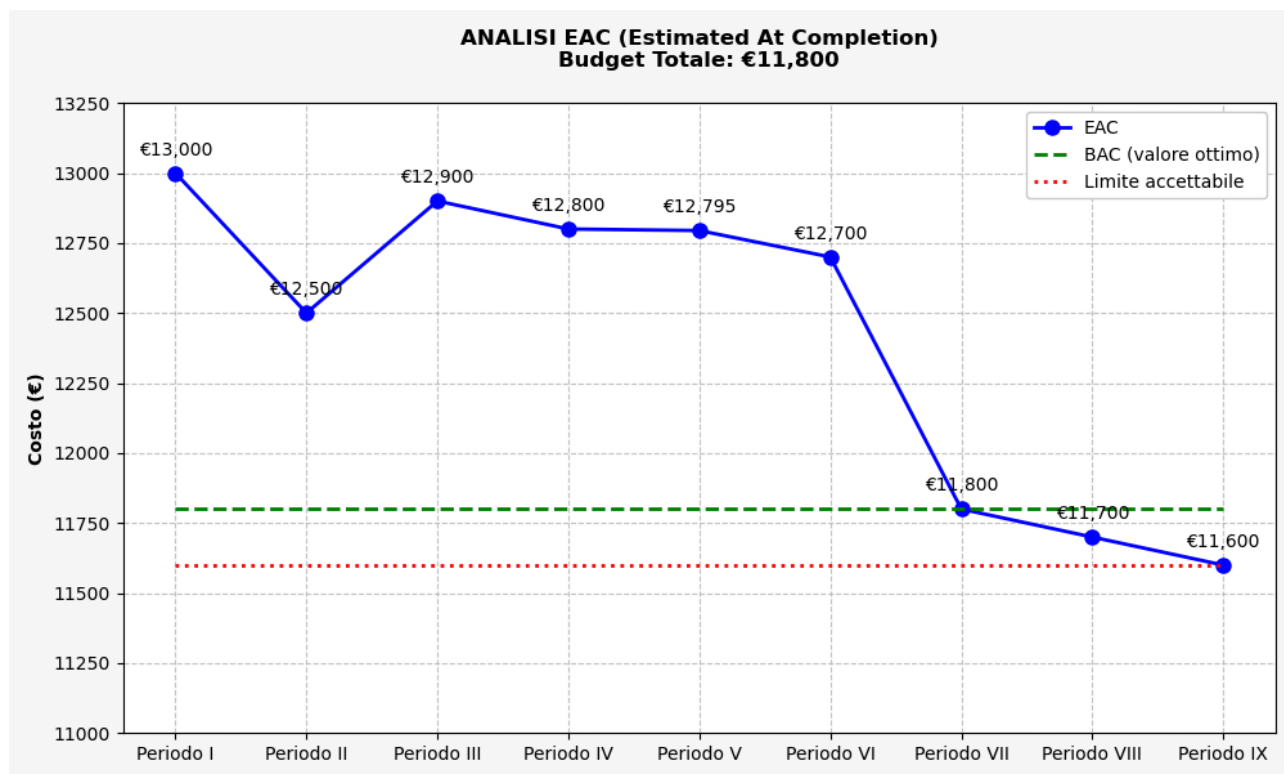


Figura 1: Proiezione della stima del costo totale nei vari periodi di progetto

4.1.1. RTB

Il grafico confronta il budget totale (€12,700) con l'EAC nei diversi periodi. Mentre il BAC (valore ottimo) rappresenta l'obiettivo ideale, l'EAC si avvicina progressivamente al limite accettabile, indicando una gestione dei costi efficace. I dati mostrano un lieve sfioramento iniziale, corretto nei periodi successivi grazie a revisioni delle stime. Ciò riflette l'adattamento del gruppo alle dinamiche di progetto, mantenendo i costi sotto controllo.

4.1.2. PB

Nel Periodo 6 (transizione RTB-PB), l'EAC inizia ad allinearsi maggiormente al budget rivisto di €11,800, mostrando un aggiustamento attivo dopo la riduzione del team. Nei Periodi 7-9 (fase PB_G), l'EAC si stabilizza vicino al nuovo budget, confermando l'efficacia delle misure correttive applicate. La sovrastima iniziale è stata compensata da una riallocazione delle risorse e una migliore precisione nelle previsioni, dimostrando resilienza nonostante le variazioni nel team.

4.2. Earned Value (EV) e Planned Value (PV)

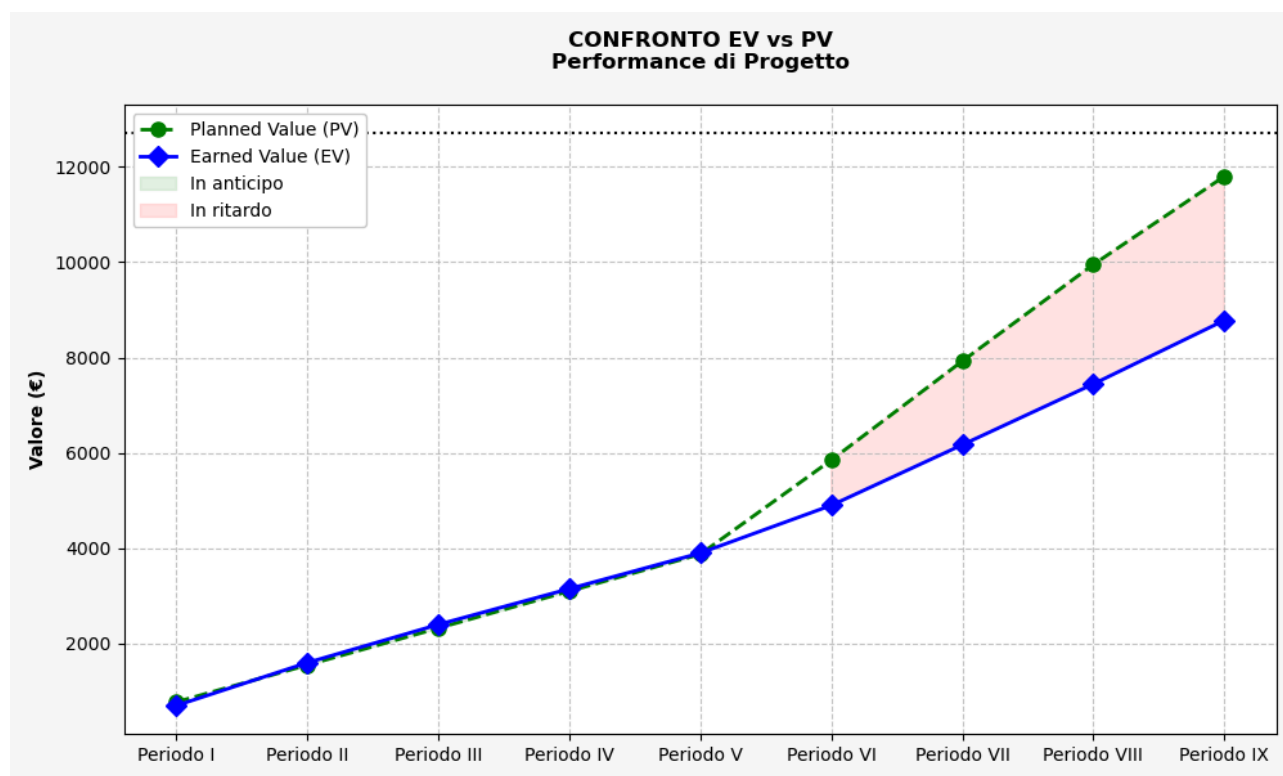


Figura 2: Proiezione dell'EV e del PV nei vari periodi di progetto.

4.2.1. RTB

Il grafico EV vs PV mostra un sostanziale allineamento tra Earned Value e Planned Value, confermando che l'avanzamento reale del progetto rispecchia fedelmente la pianificazione iniziale. Questa sovrapposizione delle curve indica una gestione efficace, con il team che ha mantenuto una produttività costante senza significativi scostamenti temporali o quantitativi.

4.2.2. PB

A partire dal Periodo 6 (transizione RTB-PB), si osserva un significativo scostamento tra PV ed EV, direttamente riconducibile alla sovrastima del budget in seguito all'abbandono di un membro del team. Il Planned Value (PV) risulta ora sovradimensionato rispetto alle effettive capacità produttive del gruppo ridotto. L'Earned Value (EV), sebbene in linea con le nuove risorse disponibili, mostra chiaramente come la pianificazione originaria non fosse più realistica. Nei Periodi 7-9 (fase PB), questo divario persiste, evidenziando la necessità di un ri-baselining del PV per allinearli alla reale capacità produttiva del team. Il progetto procede comunque con un EV costante, dimostrando un buon utilizzo delle risorse rimanenti nonostante la pianificazione non aggiornata.

4.3. Actual Cost (AC) e Estimate to Complete (ETC)

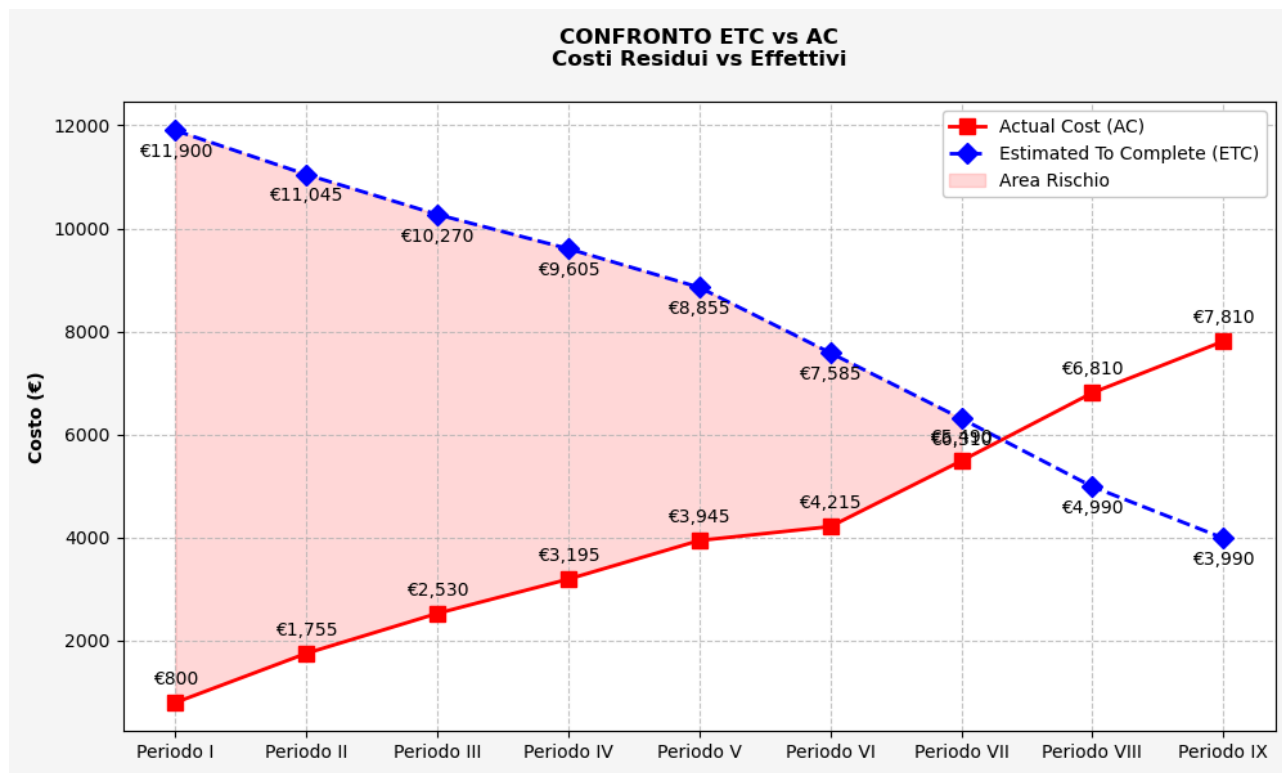


Figura 3: Proiezione dell'AC e dell'ETC nei vari periodi di progetto.

4.3.1. RTB

Il grafico mostra l'andamento dell'Estimate to Complete (ETC) e dell'Actual Cost (AC). L'ETC diminuisce progressivamente, indicando la riduzione del costo residuo previsto, mentre l'AC cresce in modo proporzionale, riflettendo i costi effettivi sostenuti periodo dopo periodo. Questo andamento contrapposto conferma un avanzamento del progetto coerente con le previsioni.

4.3.2. PB

A partire dal Periodo 6, in seguito all'abbandono di un membro del team, si osserva una significativa revisione dell'ETC. La riduzione delle risorse ha portato a un aggiustamento delle stime dei costi residui, che risultano ora più contenuti rispetto alla pianificazione originaria. L'AC mostra un andamento più contenuto nella fase PB (Periodi 7-9), riflettendo i minori costi effettivi sostenuti dal team ridotto. Sebbene l'Area Rischio si sia ampliata temporaneamente durante la transizione, nei periodi successivi l'ETC e l'AC tornano a mostrare un andamento coerente, sebbene su scale diverse rispetto alla fase RTB, dimostrando una corretta riprogrammazione dei costi dopo il cambiamento nel team.

4.4. Cost Variance (CV) e Schedule Variance (SV)

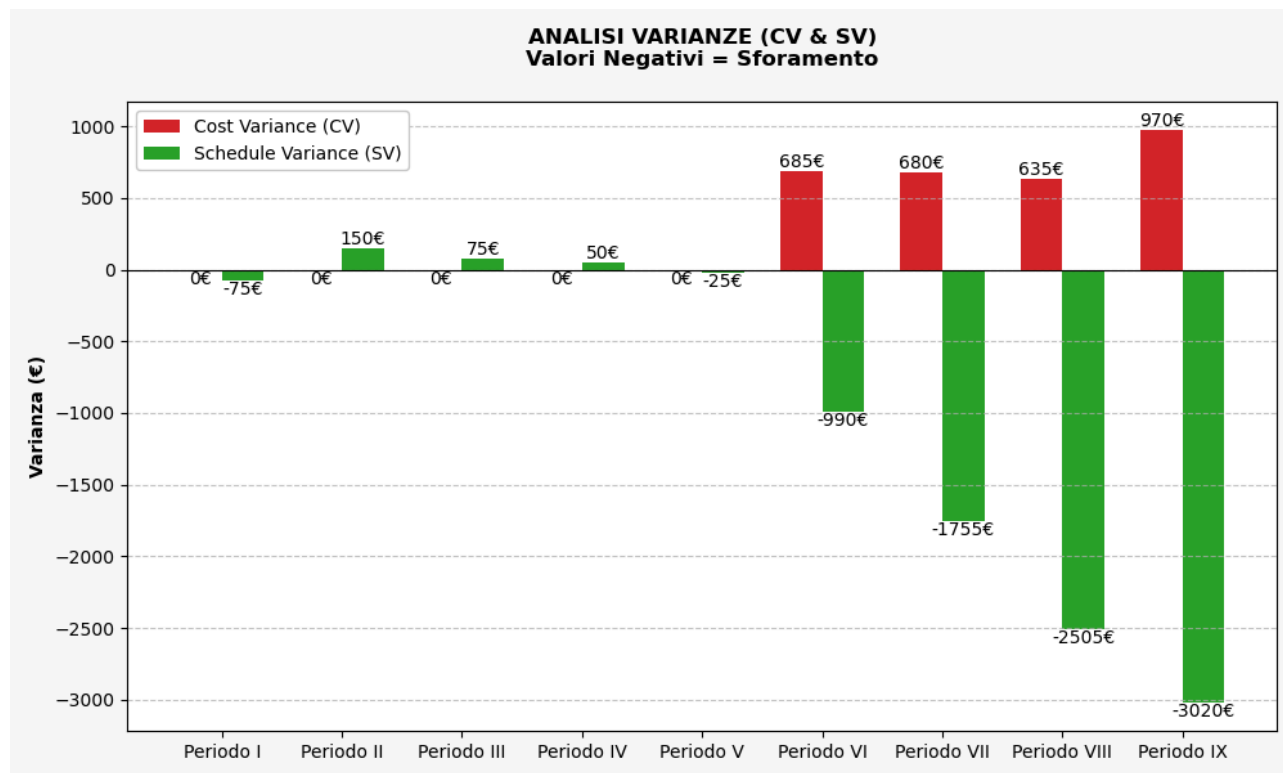


Figura 4: Proiezione della CV e della SV nei vari periodi di progetto.

4.4.1. RTB

Le varianze di costo (CV) e schedulazione (SV) rivelano andamenti distinti. Il CV costante a 0€ indica un perfetto allineamento tra costi e avanzamento. Lo SV, invece, mostra fluttuazioni: negativo nei Periodi I e V (ritardi per incomprensioni e revisioni del PoC), positivo nei Periodi II-III-IV. Questa analisi aiuta a identificare le fasi critiche del progetto.

4.4.2. PB

A partire dal Periodo 6, in seguito all'abbandono di un membro del team, si osservano significativi scostamenti sia nella CV che nella SV. La SV diventa negativa, indicando che i costi effettivi (AC) superano il valore guadagnato (EV), riflettendo l'inefficienza iniziale del team ridotto e le sovrastime del budget. La SV mostra un picco negativo, evidenziando ritardi schedulistici dovuti al riassetto delle attività e alla ridotta capacità produttiva. Nei Periodi 7-9, nonostante un parziale recupero, le varianze rimangono instabili, confermando l'impatto prolungato del cambiamento nella composizione del team sulla performance complessiva del progetto.

4.5. Requirements stability index (RSI)

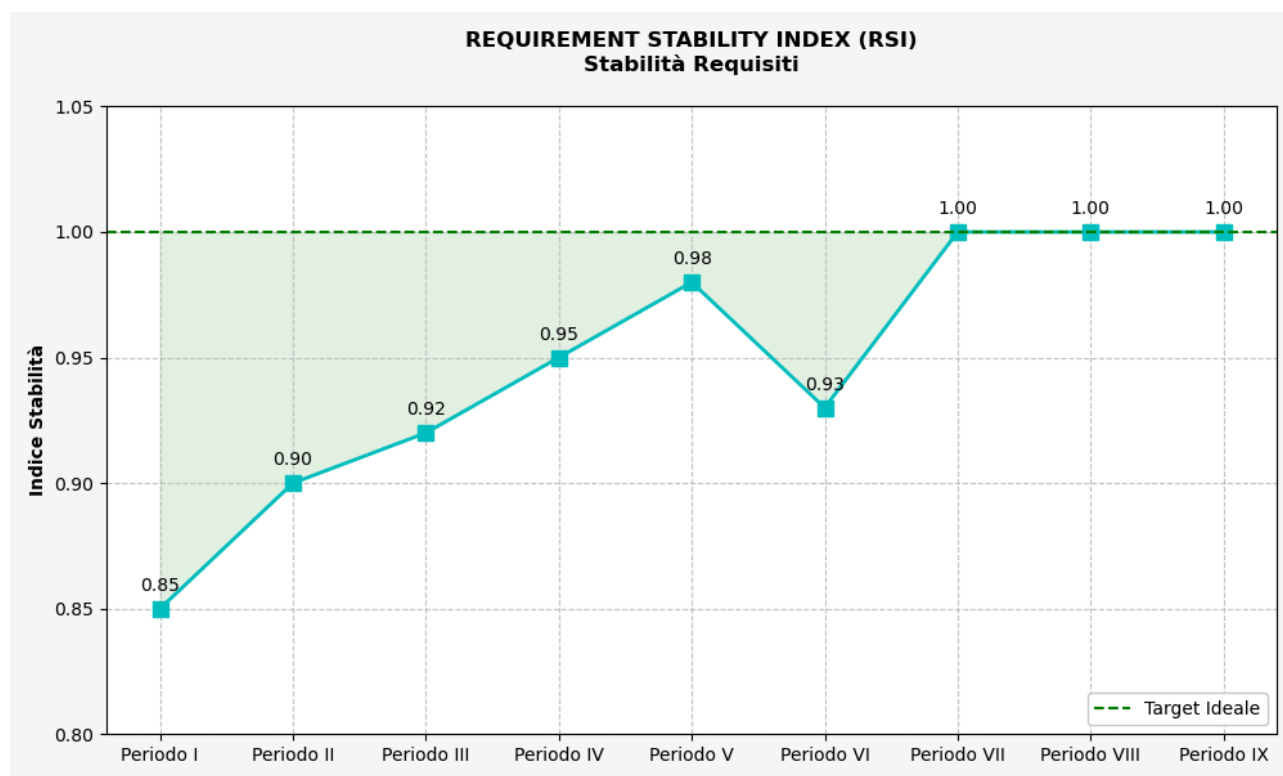


Figura 5: Proiezione del RSI nei vari periodi di progetto.

4.5.1. RTB

L'RSI misura la stabilità dei requisiti nel tempo. Il picco veloce iniziale (Periodo I) riflette l'attività intensiva di analisi, seguito da cali (Periodi II-III) per aggiornamenti e maggior dettaglio. Il raggiungimento del 98% nel Periodo V indica la maturità dei requisiti, senza ulteriori modifiche.

4.5.2. PB

Durante il Periodo 6 (transizione RTB-PB), l'RSI ha subito un calo significativo, toccando lo 0.93 (93%) a seguito delle modifiche richieste dal Prof. Cardin durante la revisione di avanzamento *RTB_G*. Questa revisione ha introdotto nuovi requisiti e modifiche specifiche che hanno temporaneamente destabilizzato la baseline. Tuttavia, durante la fase *PB_G* (Periodi 7-9), l'indice è rapidamente risalito raggiungendo e mantenendo il target ideale di 1.0 (100%), dimostrando che non sono state necessarie ulteriori modifiche ai requisiti e confermando la piena maturità e stabilità del progetto nella sua fase finale.

4.6. Indice Gulpease

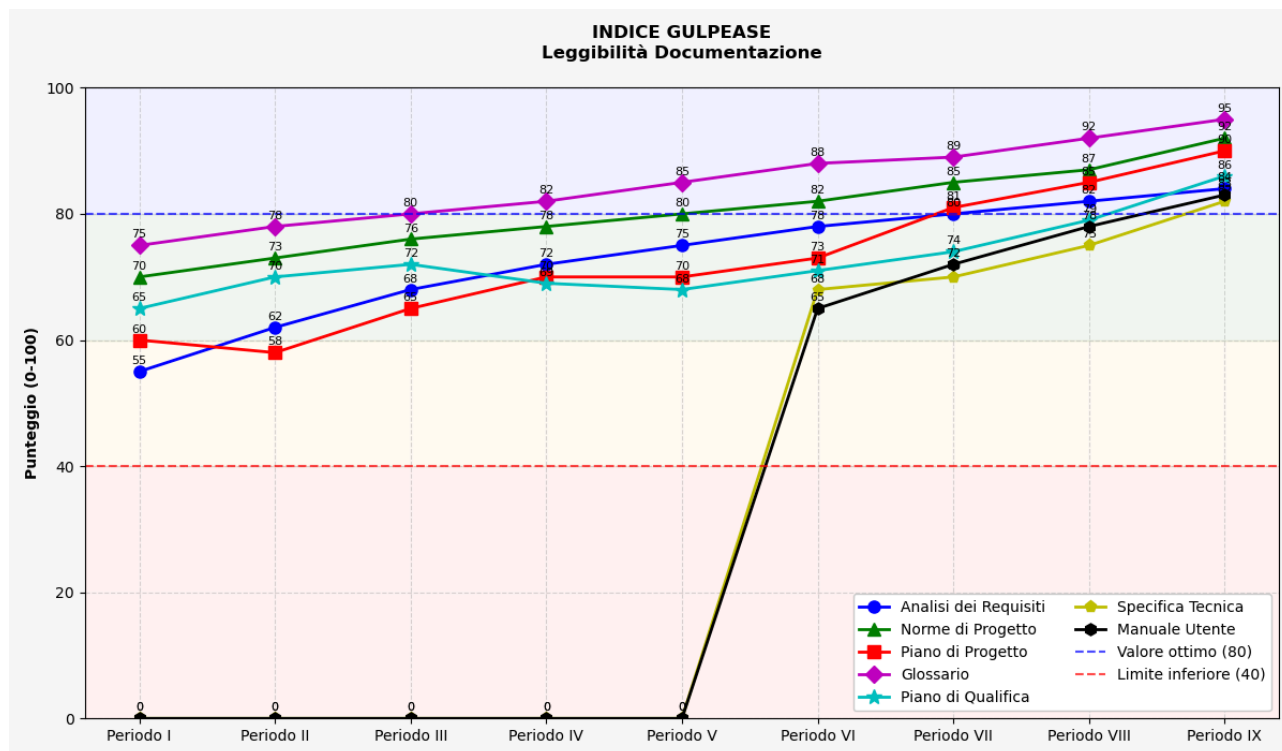


Figura 6: Proiezione dell'indice Gulpease per ogni documento nei vari periodi di progetto.

4.6.1. RTB

L'indice valuta la leggibilità dei documenti. La tendenza generale è positiva, con tutti i documenti sopra la soglia di accettabilità.

4.6.2. PB

Nella fase PB (Periodi VII-IX), i documenti esistenti continuano a mantenere un indice di Gulpease stabile, confermando la coerenza nella qualità della scrittura. Il Glossario si conferma come il documento più accessibile, mantenendo un indice costantemente superiore alla soglia ottimale di 80, adatto a un pubblico generale, in quanto non contiene tabelle, formule o immagini complesse. La Specifica Tecnica, sebbene posizionata su valori più bassi, rimane comunque ben al di sopra del limite inferiore di 40, risultando appropriata per il suo target tecnico specializzato. L'aggiunta di nuovi contenuti non ha compromesso la leggibilità generale. Alla conclusione del progetto, tutti i documenti prodotti risultano comprensibili per chi possiede almeno la licenza media, soddisfacendo pienamente gli standard di comunicazione richiesti.

4.7. Non-Calculated Risk

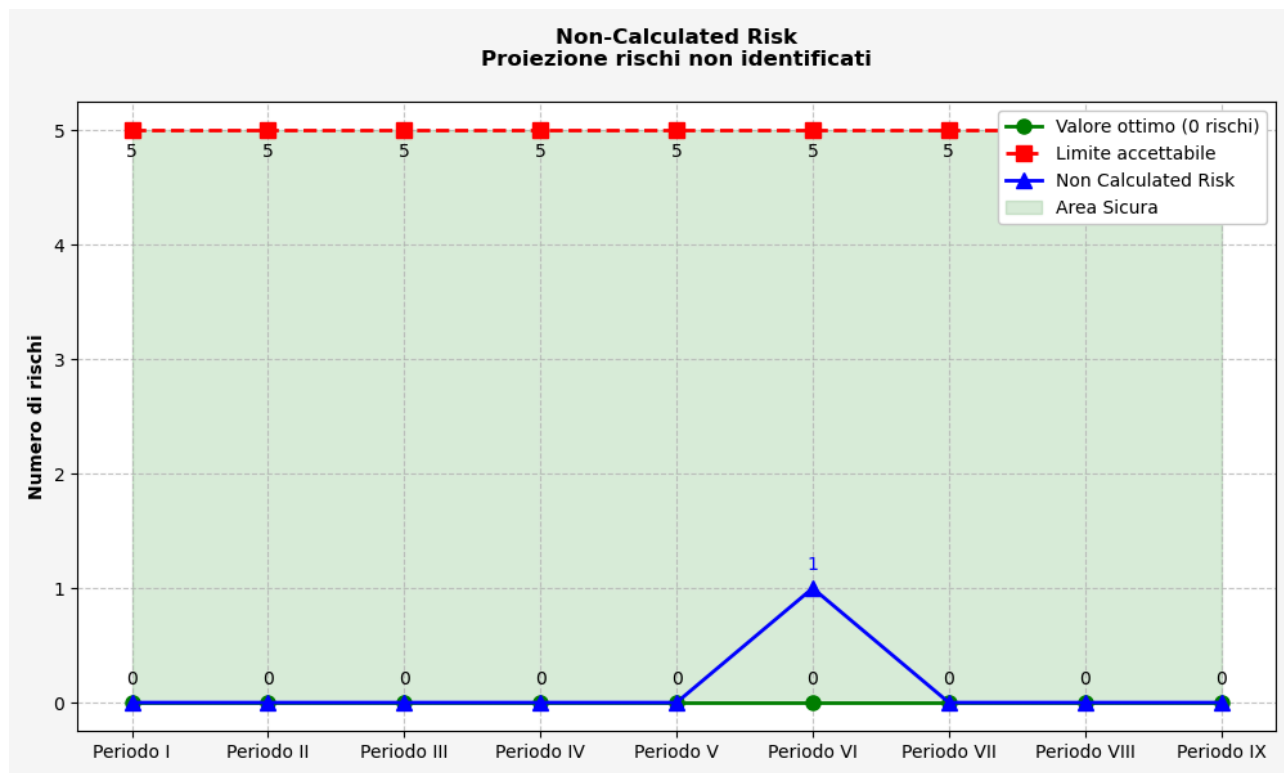


Figura 7: Proiezione del NCR (Non Calculated Risk) nei vari periodi di progetto.

4.7.1. RTB

Il grafico illustra la proiezione dei rischi non identificati durante i vari periodi del progetto. La linea del Valore ottimo (0 rischi) e il Limite accettabile superiore mostrano un'area sicura (verde) in cui il gruppo ha mantenuto un controllo efficace. L'assenza di picchi evidenzia una previsione accurata dei rischi.

4.7.2. PB

Nella fase PB, il grafico mostra un picco isolato nel Periodo VI, corrispondente all'unico rischio non atteso: l'abbandono di un membro del gruppo. Questo evento ha temporaneamente portato la proiezione fuori dall'area sicura, riflettendo l'impatto immediato sulla pianificazione delle attività. Tuttavia, nei periodi successivi (VII-IX), la curva ritorna rapidamente nell'area sicura, dimostrando una efficace gestione dell'emergenza attraverso la riorganizzazione del lavoro e la redistribuzione dei compiti. Il gruppo ha quindi mostrato resilienza, contenendo le conseguenze di questo rischio imprevisto e mantenendo il controllo complessivo del progetto.

5. Copertura dei requisiti

5.1. Copertura dei requisiti obbligatori

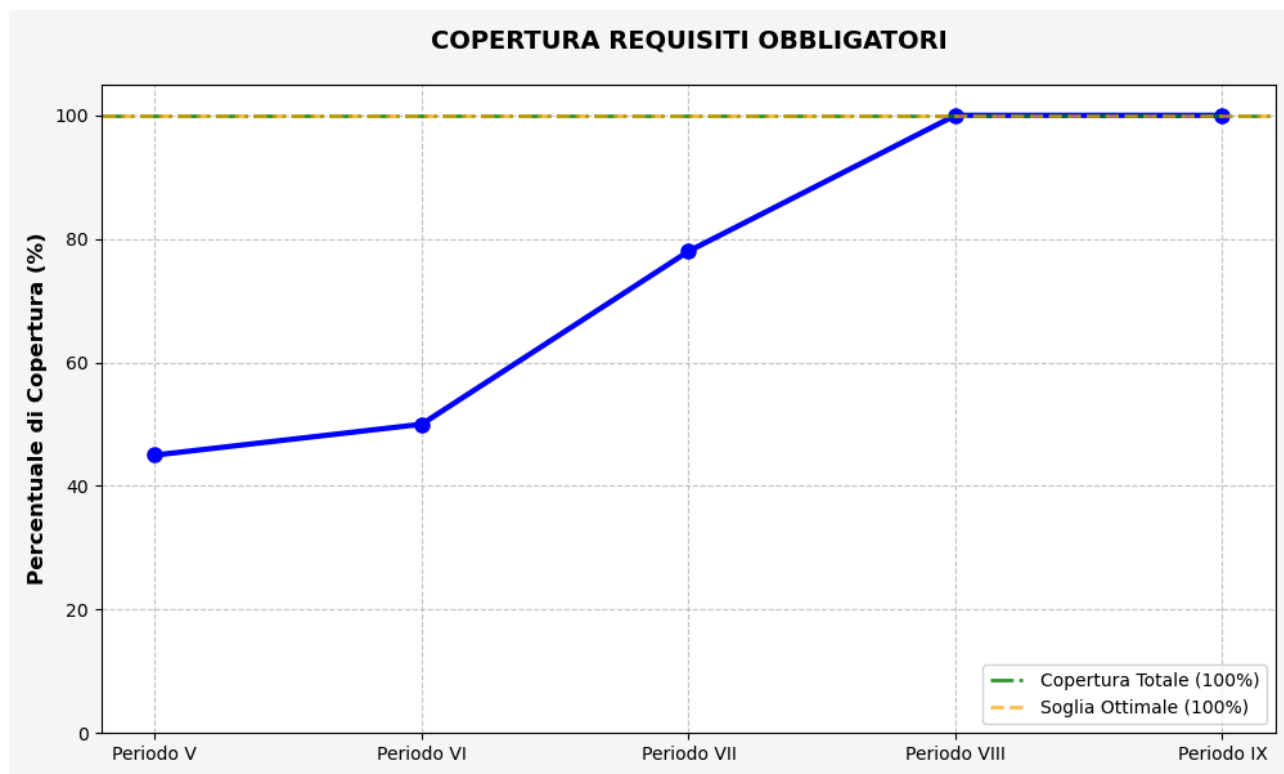


Figura 8: Copertura dei Requisiti Obbligatori

5.1.1. PB

Il grafico mostra l'andamento della copertura dei requisiti obbligatori durante la fase PB. Nei Periodi VII-IX, il team si è concentrato prioritariamente sull'implementazione completa di tutti i requisiti obbligatori, raggiungendo progressivamente la copertura totale del 100%. Questo risultato dimostra l'efficacia della strategia di sviluppo, che ha privilegiato il soddisfacimento dei requisiti fondamentali prima di eventuali funzionalità opzionali. La curva in costante crescita conferma l'impegno del gruppo nel garantire la completezza funzionale del prodotto nonostante le sfide organizzative affrontate.

5.2. Copertura dei requisiti desiderabili

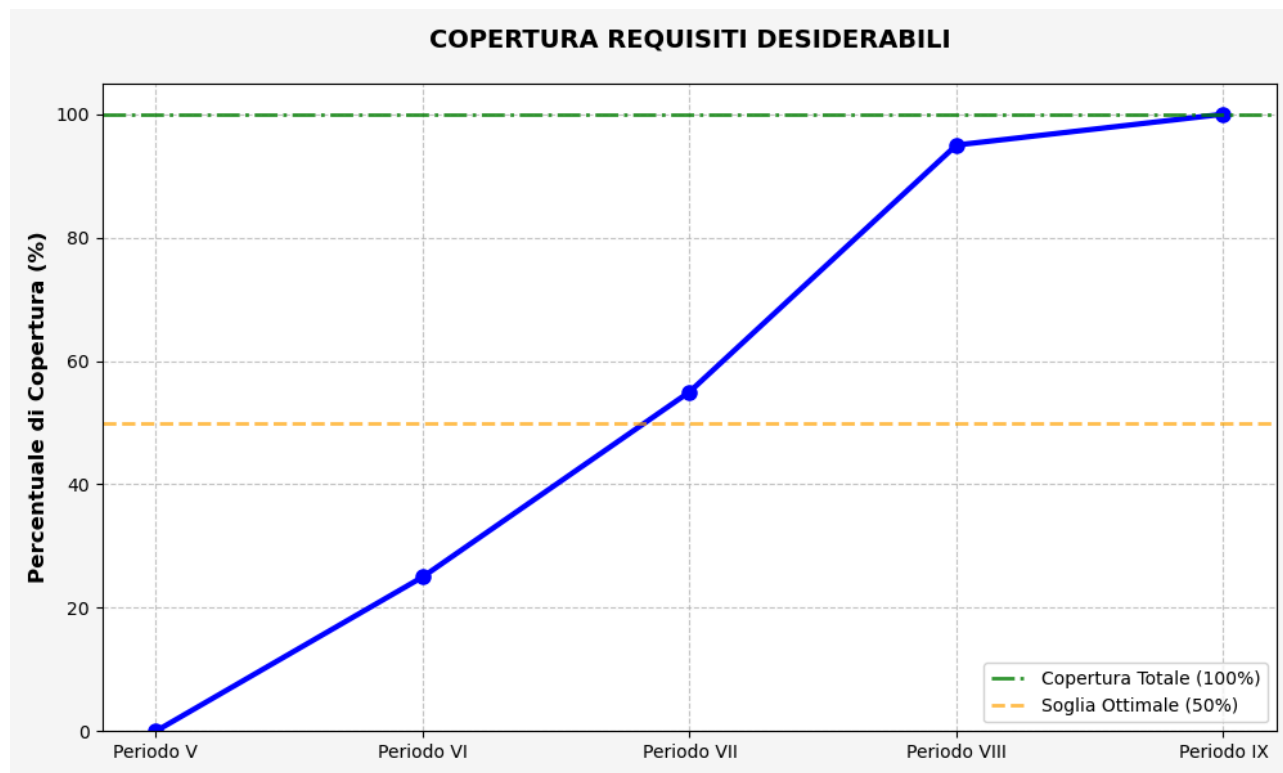


Figura 9: Copertura dei Requisiti Desiderabili

5.2.1. PB

Il grafico mostra l'andamento della copertura dei requisiti desiderabili durante la fase PB. Nei Periodi VII-IX, nonostante le difficoltà organizzative, il team è riuscito a implementare tutti i requisiti desiderabili, superando ampiamente la soglia ottimale del 50%. Questo risultato dimostra la capacità del gruppo di mantenere un alto standard qualitativo e di valore aggiunto nel prodotto finale, andando oltre gli obblighi minimi. L'approccio incrementale ha permesso di integrare progressivamente queste funzionalità aggiuntive senza compromettere il core obbligatorio del progetto.

5.3. Copertura dei requisiti opzionali

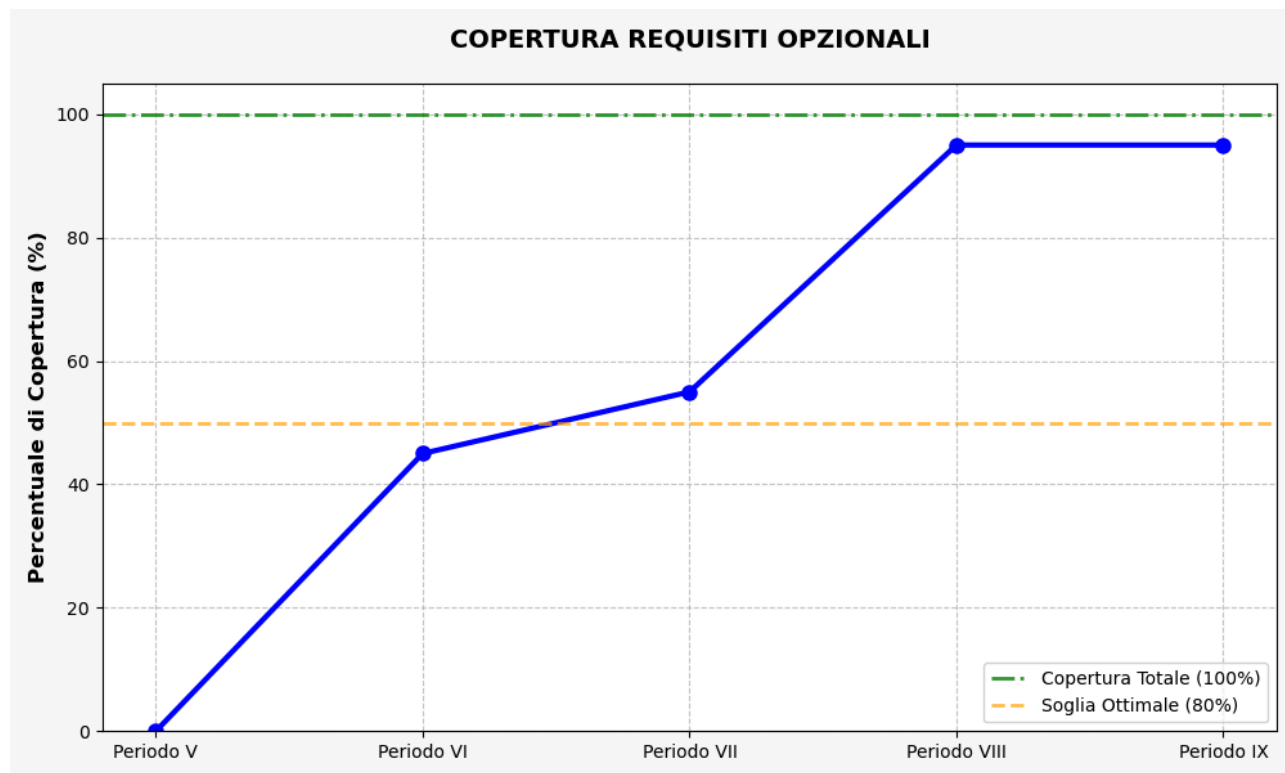


Figura 10: Copertura dei Requisiti Opzionali

5.3.1. PB

Il grafico mostra l'andamento della copertura dei requisiti opzionali durante la fase PB. Nonostante la riduzione del team, nei Periodi VII-IX il gruppo è riuscito a implementare una percentuale significativa di requisiti opzionali, superando la soglia ottimale dell'80%. Questo risultato evidenzia l'impegno del team nel fornire valore aggiunto al prodotto finale, dimostrando capacità di ottimizzazione delle risorse disponibili. L'approccio selettivo ha permesso di concentrarsi sulle funzionalità opzionali più rilevanti, massimizzando l'impatto positivo sul progetto nonostante le limitazioni organizzative.

