



PEBKAC

Gruppo: 11

Email: pebkacswe@gmail.com

Docs: <https://pebkac-swe-group-11.github.io>

GitHub: <https://github.com/PEBKAC-SWE-Group-11>



Università degli Studi di Padova

Corso di Laurea: Informatica

Corso: Ingegneria del Software

Anno Accademico: 2024/2025

Piano di qualifica

Informazioni sul documento:

Verificatore	Derek Gusatto Alessandro Benin
Redattore	Matteo Piron Matteo Gerardin Derek Gusatto Davide Martinelli
Uso	Esterno
Destinatari	Tullio Vardanega Riccardo Cardin Vimar S.p.A.

Registro delle versioni

Versione	Data	Autore	Ruolo	Descrizione
2.0.0	2025-04-05	Derek Gusatto	Responsabile	Approvazione e rilascio
1.3.0	2025-04-05	Derek Gusatto	Verificatore	Verifica Finale
1.2.1	2025-04-05	Derek Gusatto	Amministratore	Cruscotto delle metriche
1.2.0	2025-04-05	Derek Gusatto	Verificatore	Verificati Unit Test e Test di Integrazione
1.1.1	2025-04-04	Davide Martinelli	Programmatore	Unit Test e Test di Integrazione relativi al frontend
1.1.0	2025-04-03	Derek Gusatto	Verificatore	Verificati Unit Test e Test di Integrazione Backend
1.0.1	2025-03-30	Matteo Piron	Programmatore	Aggiunti Unit Test e Test di Integrazione Backend
1.0.0	2025-01-25	Derek Gusatto	Responsabile	Approvazione e rilascio
0.2.0	2025-01-24	Alessandro Benin	Verificatore	Verifica completa
0.1.6	2025-01-24	Derek Gusatto	Responsabile	§1.4 Riferimenti
0.1.5	2025-01-24	Derek Gusatto	Responsabile	§5.5 Considerazioni finali in vista della revisione RTB
0.1.4	2025-01-23	Derek Gusatto	Responsabile	Correzioni tabelle metriche, aggiunta grafici, varie
0.1.3	2025-01-22	Derek Gusatto	Responsabile	Correzioni minori
0.1.2	2025-01-19	Matteo Gerardin	Amministratore	Test da TS-21 a TS-32 e da TA-21 a TA-25, eliminazione test end-to-end
0.1.1	2025-01-18	Derek Gusatto	Progettista	Test da TS-1 a TS-20 e da TA-1 a TA-20
0.1.0	2025-01-15	Derek Gusatto	Verificatore	Verifica §1, §2, §3, §4
0.0.3	2024-12-23	Matteo Gerardin	Amministratore	§4 Strategie di testing (Struttura e definizioni)
0.0.2	2024-12-06	Matteo Piron	Amministratore	§2 Qualità di Processo, §3 Qualità di Prodotto
0.0.1	2024-12-04	Matteo Piron	Amministratore	§1 Introduzione

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	Qualità di processo	8
2.1	Processi primari	8
2.1.1	Fornitura	8
2.1.2	Sviluppo	8
2.1.2.1	Codifica	8
2.2	Processi di supporto	8
2.2.1	Documentazione	8
2.2.2	Gestione della qualità	9
3	Qualità del prodotto	10
3.1	Funzionalità	10
3.2	Affidabilità	10
4	Strategie di testing	11
4.1	Notazione dei test	11
4.2	$Test_G$ di unità	12
4.3	$Test_G$ di integrazione	18
4.4	$Test_G$ End-to-End	20
5	Cruscotto delle metriche	21
5.1	Qualità di processo - Fornitura	21
5.1.1	Earned Value, Actual Cost, Planned Value	21
5.1.2	Estimated To Completion, Estimated At Completion	22
5.1.3	Budget Variance	22
5.1.4	Cost Variance, Schedule Variance	23
5.1.5	Cost Performance Index	24
5.2	Qualità di processo - Documentazione	24
5.2.1	Indice di Gulpease	24
5.3	Qualità di processo - Gestione della qualità	25
5.3.1	Metriche non soddisfatte	25
5.4	Qualità di processo - Sviluppo	26
5.4.1	Statement Coverage	26
5.5	Qualità di prodotto - Funzionalità	26
5.5.1	Requisiti soddisfatti	26
5.6	Qualità di prodotto - Affidabilità	27
5.6.1	PTCP, CC	27

Elenco delle figure

1	Earned Value, Actual Cost, Planned Value	21
2	Estimated To Completion, Estimated At Completion	22
3	Estimated To Completion, Estimated At Completion	22
4	Cost Variance, Schedule Variance	23
5	Cost Performance Index	24
6	Indice di Gulpease	24
7	Metriche non soddisfatte	25
8	Statement Coverage	26
9	Requisiti soddisfatti	26
10	Passed Test Cases Percentage, Code Coverage	27

Elenco delle tabelle

1	Metriche per il processo di fornitura	8
2	Metriche per il processo di codifica	8
3	Metriche per il processo di documentazione	8
4	Metriche per il processo di gestione delle qualità	9
5	Metriche _G per la funzionalità _G del prodotto	10
6	Metriche _G per l'affidabilità _G del prodotto	10
7	Stato dei <i>test_G</i> di unità	17
8	Stato dei <i>test_G</i> di integrazione	19
9	Stato dei <i>test_G</i> End-to-End	20

1 Introduzione

1.1 Scopo del documento

Il presente documento ha l'obiettivo di definire le strategie di verifica e validazione messe in atto per garantire la qualità del prodotto e dei processi associati al progetto. La creazione del piano di qualifica è progressiva e incrementale nel tempo per consentire al team di riportare gli esiti delle verifiche effettuate nel tempo, con l'intento di mantenere ed eventualmente incrementare la qualità dell'intero progetto.

1.2 Scopo del prodotto

Il progetto "Vimar GENIALE" mira a sviluppare un'applicazione intelligente che supporti installatori elettrici nell'uso di dispositivi Vimar_G, facilitando l'accesso alle informazioni tecniche sui prodotti, rispondendo a domande poste in linguaggio naturale. La tecnologia alla base prevede l'uso di modelli di LLM_G e di tecniche RAG_G , con una struttura di gestione basata su $container_G$ e integrata in un ambiente $cloud_G$. Il sistema include tre componenti principali: una *applicativo web responsive_G*, un *applicativo server_G* e un'*infrastruttura cloud-ready_G*.

1.3 Glossario

Per evitare ambiguità relative al linguaggio utilizzato nei documenti, viene fornito il Glossario V1.0.0, nel quale si possono trovare tutte le definizioni di termini che hanno un significato specifico che vuole essere disambiguato. Tali termini sono marcati con una G a pedice.

1.4 Riferimenti

1.4.1 Riferimenti normativi

- **Norme di Progetto v2.0.0**
(Ultimo accesso 2025-04-04)
- **PD1 - Regolamento del progetto didattico**
<https://www.math.unipd.it/~tullio/IS-1/2024/Dispense/PD1.pdf>
(Ultimo accesso 2024-11-20)
- **Capitolato d'Appalto C2: Vimar GENIALE**
<https://www.math.unipd.it/~tullio/IS-1/2024/Progetto/C2.pdf>
(Ultimo accesso 2025-03-26)

1.4.2 Riferimenti informativi

- **T7 - Qualità del Software**
<https://www.math.unipd.it/~tullio/IS-1/2024/Dispense/T07.pdf>
(Ultimo accesso 2025-01-20)
- **T8 - Qualità del processo**
<https://www.math.unipd.it/~tullio/IS-1/2024/Dispense/T08.pdf>
(Ultimo accesso 2025-01-20)

- **T9 - Verifica e Validazione: Introduzione**
<https://www.math.unipd.it/~tullio/IS-1/2024/Dispense/T09.pdf>
(Ultimo accesso 2025-01-20)
- **T10 - Verifica e Validazione: Analisi statica**
<https://www.math.unipd.it/~tullio/IS-1/2024/Dispense/T10.pdf>
(Ultimo accesso 2025-01-20)
- **T11 - Verifica e Validazione: Analisi dinamica (Testing)**
<https://www.math.unipd.it/~tullio/IS-1/2024/Dispense/T11.pdf>
(Ultimo accesso 2025-01-20)
- **Glossario v1.0.0**
(Ultimo accesso 2025-04-01)

2 Qualità di processo

La qualità di processo si basa sull'idea che, per realizzare un prodotto conforme a specifici standard qualitativi, è essenziale monitorare e migliorare regolarmente i processi che lo generano. Questo principio si applica all'intera gamma di attività, pratiche e metodologie impiegate durante il ciclo di vita del *software*_G. In altre parole, la qualità dei processi ha l'obiettivo di andare a conformare la qualità del prodotto in modo tale da garantire sempre che gli standard definiti nel documento *Norme di Progetto* vengano rispettati ed eventualmente migliorati. Di seguito sono elencate le metriche che il team si impegna a rispettare per garantire l'eccellenza nei processi.

2.1 Processi primari

2.1.1 Fornitura

Metrica	Descrizione	Valore Accettabile	Valore Ideale
CV	Cost variance	± 150	0
PV	Planned Value	≥ 0	$\leq BAC$
EV	Earned Value	≥ 0	$\leq EAC$
AC	Actual Cost	≥ 0	$\leq EAC$
CPI	Cost Performance Index	tra 0.95 e 1.05	1
EAC	Estimated At Completion	$\pm 5\%$ del budget preventivato	budget preventivato
ETC	Estimated To Completion	≥ 0	$\leq EAC$
SV	Schedule Variance	± 150	0%
BV	Budget Variance	$\geq 10\%$	0%
BAC	Budget At Completion	-	-

Tabella 1: Metriche per il processo di fornitura

2.1.2 Sviluppo

2.1.2.1 Codifica

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

Tabella 2: Metriche per il processo di codifica

2.2 Processi di supporto

2.2.1 Documentazione

Metrica	Descrizione	Valore accettabile	Valore ideale
IG	Indice Gulpease	$\geq 65\%$	100

Tabella 3: Metriche per il processo di documentazione

2.2.2 Gestione della qualità

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

Tabella 4: Metriche per il processo di gestione delle qualità

3 Qualità del prodotto

La qualità del prodotto si concentra sulla valutazione del $software_G$ sviluppato, ponendo l'accento su caratteristiche come facilità d'uso, funzionalità_G, affidabilità_G, capacità di manutenzione_G e, in senso più ampio, sulle prestazioni complessive del prodotto. L'obiettivo principale del $team_G$ è anche quello non solo di soddisfare le attese del cliente fornendo un prodotto $software_G$ che implementi le funzionalità_G volute, ma che lo faccia seguendo precisi $standard_G$ di qualità. Vengono quindi fornite di seguito le metriche_G che il $team_G$ si impegna a soddisfare nel contesto della qualità del prodotto.

3.1 Funzionalità

Metrica	Descrizione	Valore accettabile	Valore ideale
ROS _G	Requisiti Obbligatori Soddisfatti	100%	0
RDS _G	Requisiti Desiderabili Soddisfatti	$\geq 0\%$	$\geq 65\%$
RPS _G	Requisiti Opzionali Soddisfatti	$\geq 0\%$	$\geq 50\%$

Tabella 5: Metriche_G per la funzionalità_G del prodotto

3.2 Affidabilità

Metrica	Descrizione	Valore accettabile	Valore ideale
PTCP _G	Passed Test Cases Percentage _G	$\geq 75\%$	100%
CC _G	Code Coverage _G	$\geq 75\%$	100%

Tabella 6: Metriche_G per l'affidabilità_G del prodotto

4 Strategie di testing

Per dimostrare che i *requisiti_G* individuati dagli analisti ed elencati nella sezione omonima dell'Analisi dei Requisiti siano soddisfatti, è necessario che vengano realizzati dei *test_G* appositi che verranno eseguiti sul prodotto sia in fase di codifica che in fase di *verifica_G* e *revisione_G*.

I *test_G* realizzabili possono essere suddivisi in quattro categorie principali:

- **Test_G di unità:** verificano il corretto funzionamento di una singola unità di codice indipendente (ad esempio una funzione), assicurandosi che produca i risultati attesi al variare dei possibili input_G, e vengono generalmente automatizzati per facilitare l'individuazione degli errori durante la fase di sviluppo;
- **Test_G di integrazione:** verificano il corretto funzionamento delle interazioni tra diverse unità di codice o componenti di un sistema, assicurandosi che, una volta integrati, i vari moduli lavorino insieme senza problemi, rilevando eventuali errori nelle interfacce e nei flussi di dati tra di essi;
- **Test_G End-to-End:** verificano il funzionamento complessivo di un sistema eseguendo casi d'uso dall'inizio alla fine, comprese le interazioni con altri sistemi o applicazioni esterne, per garantire il corretto funzionamento di tutti gli elementi insieme, simulando l'esperienza dell'utente in un contesto realistico.

4.1 Notazione dei test

È stato deciso come notazione per identificare univocamente i *test_G* la seguente:

T[Tipologia][Numero]

Tipologia indica la tipologia del *test_G*:

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

Ogni *test_G* si trova in uno **Stato**, che può essere:

- **V:** verificato. Questo stato indica che il *test_G* ha fornito un esito positivo;
- **NV:** non verificato. Questo stato indica che il *test_G* ha fornito un esito negativo;
- **NI:** non implementato. Questo stato indica che il *test_G* non è ancora stato implementato, e quindi non fornisce nessun esito.

4.2 $Test_G$ di unità

I $test_G$ di unità sono una tipologia di $test_G$ utilizzata per verificare singoli componenti o unità di codice in isolamento, al fine di garantire che funzionino correttamente. Un'unità di codice può essere una funzione, un metodo, una classe o un modulo, a seconda del livello di granularità scelto. I $test_G$ di unità vengono solitamente scritti dagli sviluppatori durante o immediatamente dopo la scrittura del codice e vengono utilizzati per:

- Validare il comportamento del codice, assicurandosi che ogni unità fornisca risultati corretti per un determinato insieme di input;
- Facilitare la manutenzione del $software_G$, individuando rapidamente errori introdotti da modifiche;
- Promuovere la modularità, progettando concettualmente componenti indipendenti e riutilizzabili.

Per la realizzazione di questa categoria di $test_G$ per questo progetto sarà utilizzato il $framework_G$ $unittest_G$ per Python, dato che quest'ultimo è il linguaggio scelto per la realizzazione del $backend_G$.

I $test_G$ di unità, insieme ai $test_G$ di integrazione, come richiesto nel capitolato, devono avere un $coverage_G$ minimo pari al 75% (opzionalmente un $coverage_G$ minimo pari al 90%).

Codice	Descrizione	Stato
TU-1	Verifica che il metodo <code>create_session()</code> inserisca correttamente una nuova sessione nel database	V
TU-2	Verifica che il metodo <code>read_session()</code> recuperi correttamente una sessione dal database	V
TU-3	Verifica che il metodo <code>create_conversation()</code> crei correttamente una nuova conversazione nel database	V
TU-4	Verifica che il metodo <code>read_conversations()</code> recuperi correttamente tutte le conversazioni associate a una sessione dal database	V
TU-5	Verifica che il metodo <code>read_conversation_by_id()</code> recuperi correttamente una conversazione dal database tramite il suo ID	V
TU-6	Verifica che il metodo <code>delete_conversation()</code> elimini correttamente una conversazione dal database	V
TU-7	Verifica che il metodo <code>add_message()</code> aggiunga correttamente un nuovo messaggio a una conversazione	V
TU-8	Verifica che il metodo <code>read_messages()</code> recuperi correttamente tutti i messaggi associati a una conversazione	V
TU-9	Verifica che il metodo <code>read_feedback()</code> recuperi correttamente tutti i feedback associati a un messaggio	V
TU-10	Verifica che il metodo <code>add_feedback()</code> aggiunga correttamente un feedback a un messaggio	V

TU-11	Verifica che il metodo <code>read_num_positive_feedback()</code> recuperi correttamente il numero di feedback positivi	V
TU-12	Verifica che il metodo <code>read_num_negative_feedback()</code> recuperi correttamente il numero di feedback negativi	V
TU-13	Verifica che il metodo <code>read_num_conversations()</code> recuperi correttamente il numero totale di conversazioni	V
TU-14	Verifica che l'endpoint restituisca correttamente gli header CORS (Cross-Origin Resource Sharing)	V
TU-15	Verifica che l'endpoint restituisca un codice di stato 404 se non definito	V
TU-16	Verifica che il metodo <code>execute_query()</code> venga chiamato correttamente con la query specificata	V
TU-17	Verifica che il metodo <code>fetch_one()</code> venga chiamato correttamente con la query e i parametri specificati	V
TU-18	Verifica che il metodo <code>fetch_all()</code> venga chiamato correttamente con la query specificata e restituisca tutti i risultati attesi	V
TU-19	Verifica che il metodo <code>close()</code> venga chiamato correttamente	V
TU-20	Verifica che una sessione venga creata correttamente senza specificare la data di creazione	V
TU-21	Verifica che una sessione venga creata correttamente specificando la data di creazione	V
TU-22	Verifica che una conversazione venga creata correttamente	V
TU-23	Verifica che un messaggio venga creato correttamente	V
TU-24	Verifica che venga generato correttamente un embedding per una query quando la richiesta HTTP ha successo	V
TU-25	Verifica che venga generato correttamente un errore quando la risposta HTTP non contiene l'embedding	V
TU-26	Verifica che venga generato correttamente un errore quando la richiesta HTTP fallisce	V
TU-27	Verifica che una query venga eseguita correttamente	V
TU-28	Verifica che una query venga eseguita correttamente e restituisca un solo risultato	V
TU-29	Verifica che una query venga eseguita correttamente e restituisca tutti i risultati	V
TU-30	Verifica che la connessione al database venga chiusa correttamente	V
TU-31	Verifica che la connessione al database venga creata correttamente utilizzando <code>psycopg2</code>	V
TU-32	Verifica che l'endpoint: <code>/api/test</code> restituisca correttamente il messaggio di successo	V
TU-33	Verifica che l'endpoint: <code>/api/question/1</code> gestisca correttamente una richiesta di domanda	V
TU-34	Verifica che l'endpoint: <code>/api/session/1</code> recuperi correttamente una sessione	V

TU-35	Verifica che l'endpoint <code>/api/conversation</code> crei correttamente una nuova conversazione	V
TU-36	Verifica che l'endpoint: <code>/api/conversation</code> recuperi correttamente tutte le conversazioni associate a una sessione	V
TU-37	Verifica che l'endpoint: <code>/api/conversation/1</code> recuperi correttamente una conversazione tramite il suo ID	V
TU-38	Verifica che l'endpoint: DELETE <code>/api/conversation/1</code> elimini correttamente una conversazione	V
TU-39	Verifica che l'endpoint: POST <code>/api/message</code> aggiunga correttamente un nuovo messaggio a una conversazione	V
TU-40	Verifica che l'endpoint: GET <code>/api/message</code> recuperi correttamente tutti i messaggi associati a una conversazione	V
TU-41	Verifica che la vengano create correttamente le directory necessarie	V
TU-42	Verifica che un PDF venga scaricato correttamente e salvato	V
TU-43	Verifica che il download fallisca correttamente in caso di errore HTTP	V
TU-44	Verifica che un PDF venga scaricato e convertito correttamente in un file di testo	V
TU-45	Verifica che un file di testo venga suddiviso in chunk e che vengano generati i relativi vettori di embedding	V
TU-46	Verifica che la connessione al database venga stabilita correttamente	V
TU-47	Verifica che la funzione calcoli correttamente la dimensione del vettore di embedding	V
TU-48	Verifica che vengano create correttamente le tabelle e i trigger nel database	V
TU-49	Verifica che i chunk vengano inseriti correttamente nel database	V
TU-50	Verifica che i prodotti vengano elaborati e inseriti correttamente nel database	V
TU-51	Verifica che i documenti vengano inseriti correttamente nel database	V
TU-52	Verifica che i prodotti vengano letti da un file JSON e scritti correttamente nel database	V
TU-53	Verifica che la funzione restituisca un vettore di embedding valido per un input corretto	V
TU-54	Verifica che la funzione gestisca correttamente un input vuoto restituendo un vettore vuoto	V
TU-55	Verifica che la funzione gestisca correttamente un input molto grande restituendo un vettore di embedding valido	V
TU-56	Verifica che le tabelle vengano create correttamente nel database	V
TU-57	Verifica che venga gestito correttamente un errore durante la creazione delle tabelle	V

TU-58	Verifica che i dati vengano importati correttamente nel data-base	V
TU-59	Verifica che venga gestito correttamente un errore durante l'importazione dei dati	V
TU-60	Verifica che i link vengano estratti correttamente dai prodotti e raggruppati per ID	V
TU-61	Verifica che i prodotti vengano elaborati correttamente e che vengano generati i vettori di embedding	V
TU-62	Verifica che venga fatta una richiesta di creare una sessione	V
TU-63	Verifica che venga fatta una richiesta di aggiornamento della sessione	V
TU-64	Verifica che venga fatta una richiesta per creare una conversazione	V
TU-65	Verifica che venga fatta una richiesta per recuperare le conversazioni	V
TU-66	Verifica che venga fatta una richiesta per eliminare una conversazione secondo l'id	V
TU-67	Verifica che venga fatta una richiesta per inviare un messaggio	V
TU-68	Verifica che venga fatta una richiesta per recuperare i messaggi	V
TU-69	Verifica che venga fatta una richiesta per inviare un feedback	V
TU-70	Verifica che venga fatta una richiesta per recuperare il commento relativo al feedback	V
TU-71	Verifica che venga fatta una richiesta per verificare che la password sia corretta	V
TU-72	Verifica che venga fatta una richiesta per recuperare le statistiche degli admin	V
TU-73	Verifica che la funzione askQuestion() funzioni correttamente	V
TU-74	Verifica che la funzione readSession() funzioni correttamente	V
TU-75	Verifica che venga creata un'istanza di ChatService senza errori	V
TU-76	Verifica che la sessione possa essere aggiornata correttamente	V
TU-77	Verifica che le conversazioni possano essere correttamente recuperate	V
TU-78	Verifica che non venga creata una conversazione senza la creazione di una sessione	V
TU-79	Verifica che venga creata correttamente una conversazione	V
TU-90	Verifica che non venga eliminata una conversazione attiva	V
TU-91	Verifica che non venga eliminata alcuna conversazione senza la presenza di una sessione	V
TU-92	Verifica che venga correttamente inviato un feedback	V
TU-93	Verifica che non venga inviato un feedback senza la presenza di una sessione	V
TU-94	Verifica del raggiungimento del limite di conversazioni	V
TU-95	Verifica che venga inviato un messaggio	V
TU-96	Verifica che venga creata correttamente un'istanza della componente Sidebar	V

TU-97	Verifica che l'implementazione di <code>ngOnInit()</code> istanzi attributi e membri privati del componente Sidebar	V
TU-98	Verifica che la funzione <code>hasReachedLimit()</code> segua il comportamento aspettato	V
TU-99	Verifica che venga creata una conversazione nella Sidebar	V
TU-100	Verifica che se l'app è in attesa di una risposta, non venga creata una nuova conversazione	V
TU-101	Verifica che al raggiungimento del limite di conversazioni venga visualizzato un avviso interno al browser che notifica il problema all'utente	V
TU-102	Verifica che l'app non si trovi nello stato in cui non sta aspettando una risposta dal LLM	V
TU-103	Verifica che l'app si trovi nello stato in cui non sta aspettando una risposta dal LLM	V
TU-104	Verifica che si possa selezionare una conversazione	V
TU-105	Verifica che se l'app sta aspettando una risposta, non venga selezionata alcuna conversazione oltre a quella attiva c	V
TU-106	Verifica che possa essere possibile eliminare una conversazione	V
TU-107	Verifica che selezionando una conversazione essa diventi effettivamente quella attiva	V
TU-108	Verifica che venga creata una istanza della componente Chat-box	V
TU-109	Verifica che il metodo <code>ngOnInit()</code> inserisca dei messaggi fittizi nella variabile <code>messages</code>	V
TU-110	Verifica che il metodo <code>ngOnInit()</code> inserisca un vettore di messaggi vuoto nella variabile <code>messages</code>	V
TU-111	Verifica che il metodo <code>ngAfterViewChecked()</code> assuma il comportamento atteso	V
TU-112	Verifica che venga correttamente inviato un messaggio	V
TU-113	Verifica che un messaggio vuoto non venga inviato	V
TU-114	Verifica che un messaggio che sorpassa il limite di caratteri non venga inviato	V
TU-115	Verifica che se l'app è in attesa di risposta, non venga inviato un messaggio	V
TU-116	Verifica che venga inviato un feedback positivo	V
TU-117	Verifica che venga inviato un feedback negativo	V
TU-118	Verifica che venga effettivamente inviato un messaggio di feedback	V
TU-119	Verifica che non venga inviato un feedback senza che la variabile <code>feedbackMessageId</code> abbia un valore	V
TU-120	Verifica che venga eseguito il metodo <code>closeFeedbackPopup()</code>	V
TU-121	Verifica il numero di caratteri rimanenti per il messaggio di feedback	V
TU-122	Verifica che venga creata correttamente un'istanza della componente AdminDashboard	V
TU-123	Verifica che l'implementazione del metodo <code>ngOnInit()</code> iniziizzi statistiche e commenti di feedback	V

TU-124	Verifica che avvenga il calcolo del rapporto di soddisfazione	V
TU-125	Verifica che il calcolo del rapporto di soddisfazione sia uguale a 0	V
TU-126	Verifica che senza alcuna statistica caricata, il rapporto di soddisfazione sia uguale a 0	V
TU-127	Verifica che il metodo formatDate() funzioni correttamente	V
TU-128	Verifica che venga creata una istanza della componente AdminLogin	V
TU-129	Verifica che la password fornita venga sottoposta ad un processo di crittografia secondo hash	V
TU-130	Verifica che il login avvenga correttamente	V
TU-131	Verifica che il login non avvenga se non è presente la password	V
TU-132	Verifica che con una risposta negativa da parte della relativa API, non venga effettuato il login	V
TU-133	Verifica che il token dell'admin sia autenticato	V
TU-134	Verifica che la guardia abbia un token e che esso sia autenticato	V
TU-135	Verifica che la guardia ritorni false se non vi è alcun token	V
TU-136	Verifica del corretto funzionamento di AuthInterceptor	V
TU-137	Verifica del corretto funzionamento di AuthInterceptor senza la presenza di un token	V
TU-138	Verifica che venga creata correttamente un'istanza di AppComponent	V
TU-139	Verifica che venga correttamente attivata la componente Sidebar	V
TU-140	Verifica che venga correttamente chiusa la Sidebar entro una dimensione della finestra su dispositivi mobili	V

Tabella 7: Stato dei $test_G$ di unità

4.3 $Test_G$ di integrazione

I $test_G$ di integrazione sono una tipologia di $test_G$ progettata per verificare la capacità di diversi componenti o moduli di un sistema di funzionare insieme. I $test_G$ di integrazione non mirano a testare singoli moduli in modo indipendente, come fa il $test_G$ di unità, che si concentra su unità di codice isolate. Le caratteristiche principali dei $test_G$ di integrazione sono:

- Monitorare i problemi di comunicazione tra moduli;
- Garantire la corretta configurazione e gestione delle dipendenze tra moduli; Testare il sistema in condizioni più vicine a quelle reali rispetto a quanto avviene con i $test_G$ di unità.

I $test_G$ di integrazione, insieme ai $test_G$ di unità, come richiesto nel capitolato, devono avere un $coverage_G$ minimo pari al 75% (opzionalmente un $coverage_G$ minimo pari al 90%).

Codice	Descrizione	Stato
TI-1	Verifica che l'endpoint restituisca correttamente il messaggio di successo	V
TI-2	Verifica che l'endpoint gestisca correttamente una richiesta di domanda	V
TI-3	Verifica che l'endpoint crei correttamente una nuova conversazione	V
TI-4	Verifica che l'endpoint recuperi correttamente tutte le conversazioni associate a una sessione	V
TI-5	Verifica che l'endpoint recuperi correttamente una conversazione tramite il suo ID	V
TI-6	Verifica che l'endpoint elimini correttamente una conversazione	V
TI-7	Verifica che l'endpoint aggiunga correttamente un nuovo messaggio a una conversazione	V
TI-8	Verifica che l'endpoint recuperi correttamente tutti i messaggi associati a una conversazione	V
TI-9	Verifica che l'endpoint restituisca correttamente il messaggio di successo	V
TI-10	Verifica che l'endpoint gestisca correttamente una richiesta di domanda	V
TI-11	Verifica che l'endpoint crei correttamente una nuova conversazione	V
TI-12	Verifica che l'endpoint recuperi correttamente tutte le conversazioni associate a una sessione	V
TI-13	Verifica che l'endpoint recuperi correttamente una conversazione tramite il suo ID	V
TI-14	Verifica che l'endpoint elimini correttamente una conversazione	V
TI-15	Verifica che l'endpoint aggiunga correttamente un nuovo messaggio a una conversazione	V

TI-16	Verifica che l'endpoint recuperi correttamente tutti i messaggi associati a una conversazione	V
TI-17	Verifica che una query venga eseguita correttamente	V
TI-18	Verifica che una query venga eseguita correttamente e restituisca un solo risultato	V
TI-19	Verifica che una query venga eseguita correttamente e restituisca tutti i risultati	V
TI-20	Verifica che l'input dell'utente venga processato correttamente	V
TI-21	Verifica che vengano caricati i prodotti strutturati corretti per la RAG	V
TI-22	Verifica che vengano caricati i chunk corretti per la RAG	V
TI-23	Verifica che venga creata correttamente una sessione	V
TI-24	Verifica che venga utilizzata la sessione corretta	V
TI-25	Verifica che venga correttamente aggiornata la sessione	V
TI-26	Verifica che venga creata correttamente una conversazione	V
TI-27	Verifica che venga caricate tutte le conversazioni di una sessione	V
TI-28	Verifica che venga salvato correttamente un nuovo messaggio	V
TI-29	Verifica che venga caricato il messaggio corretto	V
TI-30	Verifica che venga salvato il giusto feedback per il giusto messaggio	V
TI-31	Verifica che venga caricato il corretto numero di feedback positivi	V
TI-32	Verifica che venga caricato il corretto numero di feedback negativi	V
TI-33	Verifica che il modello LLM generi una risposta con input validi	V
TI-34	Verifica che il modello LLM generi una risposta con contesto vuoto	V
TI-35	Verifica che venga gestito l'utilizzo di un modello non valido	V

Tabella 8: Stato dei *test_G* di integrazione

4.4 *Test_G* End-to-End

I *test_G* End-to-End sono una tipologia di *test_G* che valida il funzionamento complessivo di un sistema, testando l'intero flusso dal punto A al punto B. Questo tipo di *test_G* riguarda l'interazione tra le parti del sistema stesso e con sistemi esterni, per assicurarsi che tutte le parti funzionino correttamente insieme, emulando una situazione online da testare tramite l'utente finale. In questo modo, si garantisce che il sistema sia privo di errori, soddisfi i *requisiti_G* funzionali e contribuisca a una buona esperienza utente. I *test_G* End-to-End, come richiesto nel capitolato, devono avere un *coverage_G* minimo pari all'80%.

Codice	Descrizione	Casi d'uso	Stato
TE-1	Verifica della creazione di una nuova conversazione e dell'invio di una domanda con successiva visualizzazione della risposta	UC1, UC5	V
TE-2	Verifica della visualizzazione dello storico dei messaggi all'interno di una conversazione	UC3.4	V
TE-3	Verifica della possibilità di fare richieste in italiano e di ottenere risposte pertinenti	UC5, UC8	V
TE-4	Verifica della possibilità di fornire un feedback sulla risposta fornita dal sistema	UC10	V
TE-5	Verifica della funzionalità di eliminazione di una conversazione	UC11	V
TE-6	Verifica dell'accesso al cruscotto informativo per amministratori e della visualizzazione delle informazioni nel cruscotto informativo	UC14, UC15, UC16	V

Tabella 9: Stato dei *test_G* End-to-End

5 Cruscotto delle metriche

5.1 Qualità di processo - Fornitura

5.1.1 Earned Value, Actual Cost, Planned Value

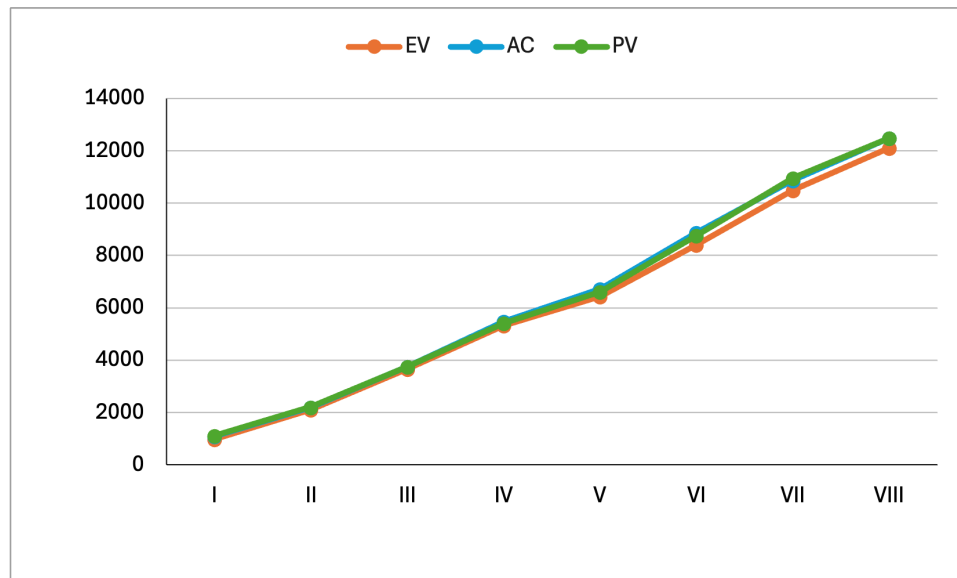


Figura 1: Earned Value, Actual Cost, Planned Value

Analisi

Queste metriche indicano un buon andamento essendo sempre abbastanza sovrapposte su tutti gli *sprint_G*, anche se si nota un graduale distaccamento negli ultimi periodi dovuto alle imprecisioni nei costi preventivati.

5.1.2 Estimated To Completion, Estimated At Completion

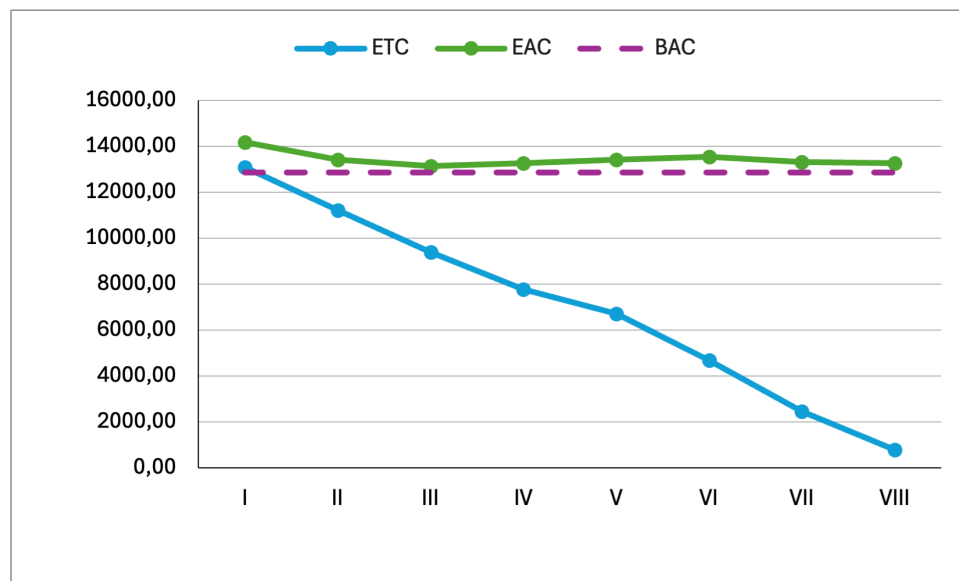


Figura 2: Estimated To Completion, Estimated At Completion

Analisi

Dopo un inizio non ottimale è possibile notare un riallineamento. L'Estimation At Completion si è riallineato al Budget At Completion, ma è sempre rimasto superiore, anche se di poco, a causa del costo maggiore per il raggiungimento di tutti gli obiettivi degli *sprint_G*. L'Estimated To Completion è sempre stato gradualmente discendente.

5.1.3 Budget Variance

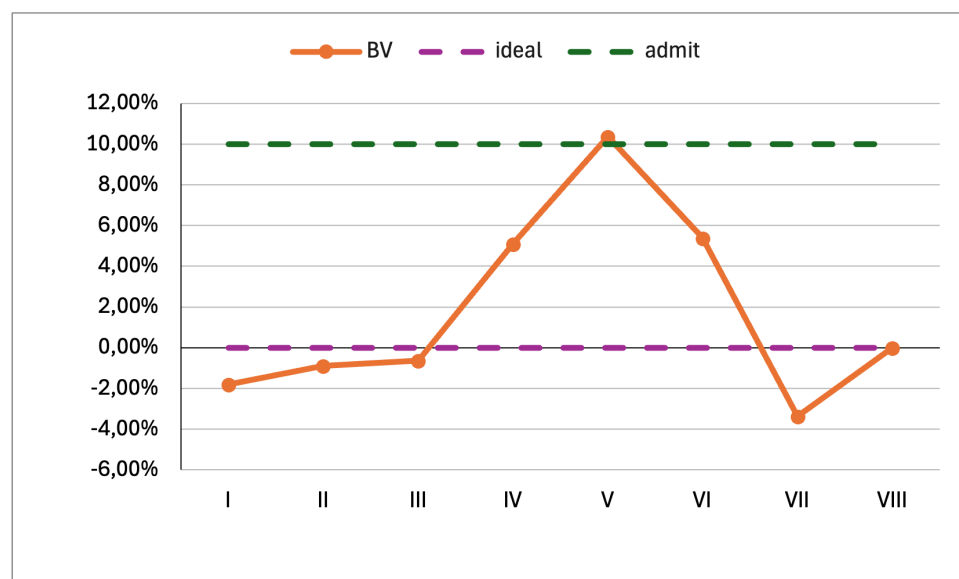


Figura 3: Estimated To Completion, Estimated At Completion

Analisi

La Budget Variance non sempre è rimasta nei limiti ammissibili come è evidente come nel quarto, quinto e sesto *sprint_G*. La variazione è stata notevole essendo le ore preventivate insufficienti e questo è stato un forte segnale di allarme per il gruppo che ha reagito adeguatamente come si può vedere dai due successivi periodo.

5.1.4 Cost Variance, Schedule Variance

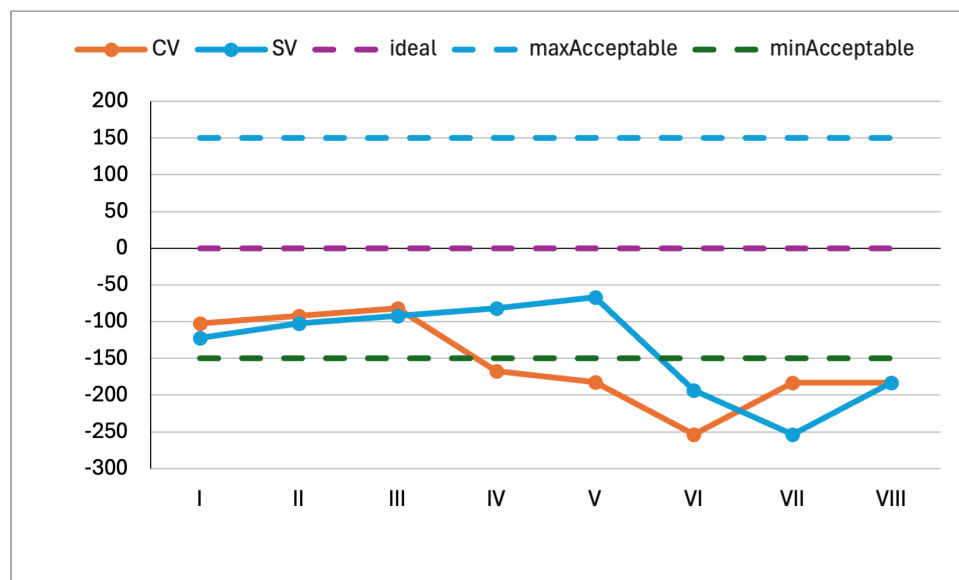


Figura 4: Cost Variance, Schedule Variance

Analisi

Per quanto i valori non si notano dei netti miglioramenti, ma anzi un peggioramento, a cui il gruppo ha cercato di fare fronte, riuscendoci in parte. Questo indica una not ottimale gestione delle *risorse_G* per il raggiungimento degli obiettivi entro i costi preventivati.

5.1.5 Cost Performance Index

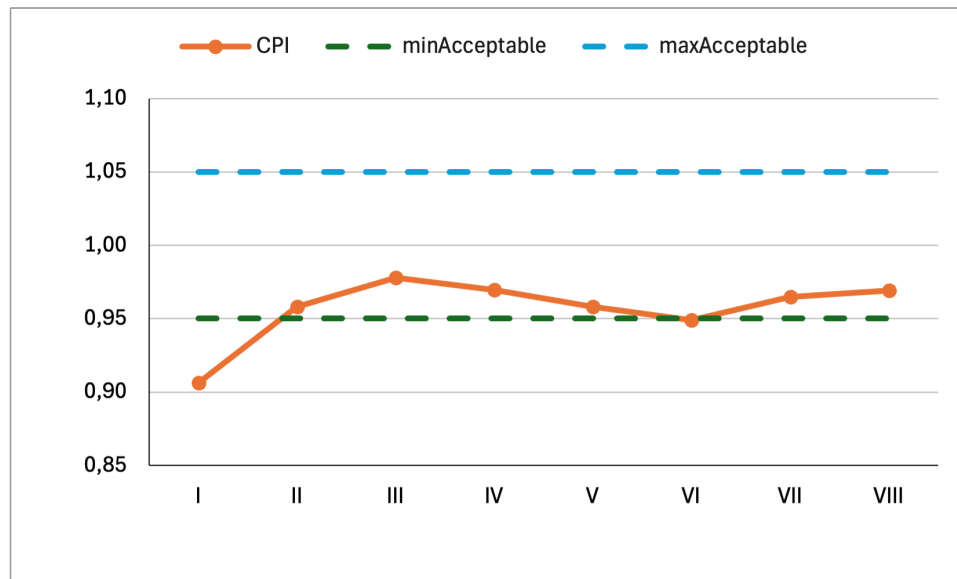


Figura 5: Cost Performance Index

Analisi

È evidente come il Cost Performance Index sia migliorato nel corso degli *sprint_G*, fino a rientrare nei limiti ammissibili e successivamente il gruppo è riuscito a mantenerlo entro tali limiti.

5.2 Qualità di processo - Documentazione

5.2.1 Indice di Gulpease

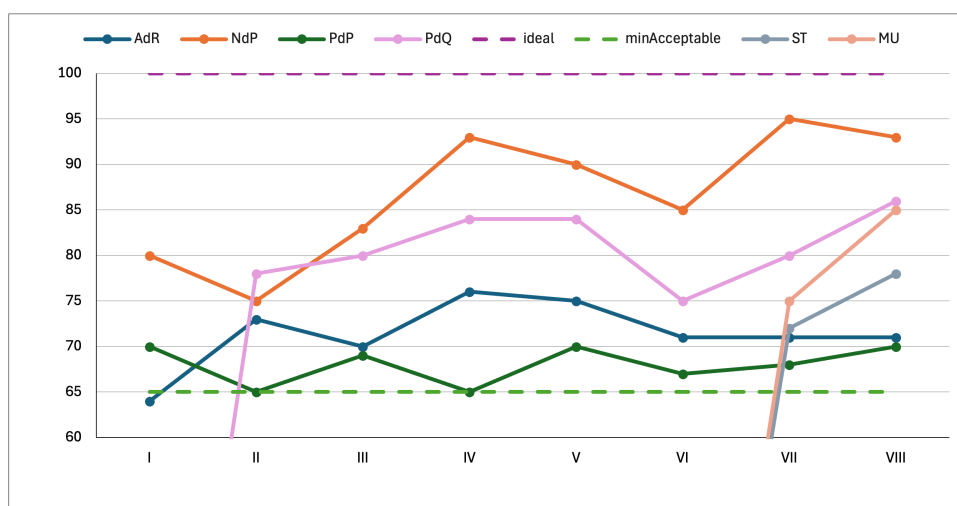


Figura 6: Indice di Gulpease

Analisi

Non solo tutti i documenti rientrano nei limiti accessibili, ma sono anche praticamente tutti in miglioramento nel corso degli ultimi periodi, per quanto il Piano di Progetto che era il documento più carente è rimasto tale. Si nota invece un buon inizio per gli ultimi due documenti creati.

5.3 Qualità di processo - Gestione della qualità

5.3.1 Metriche non soddisfatte

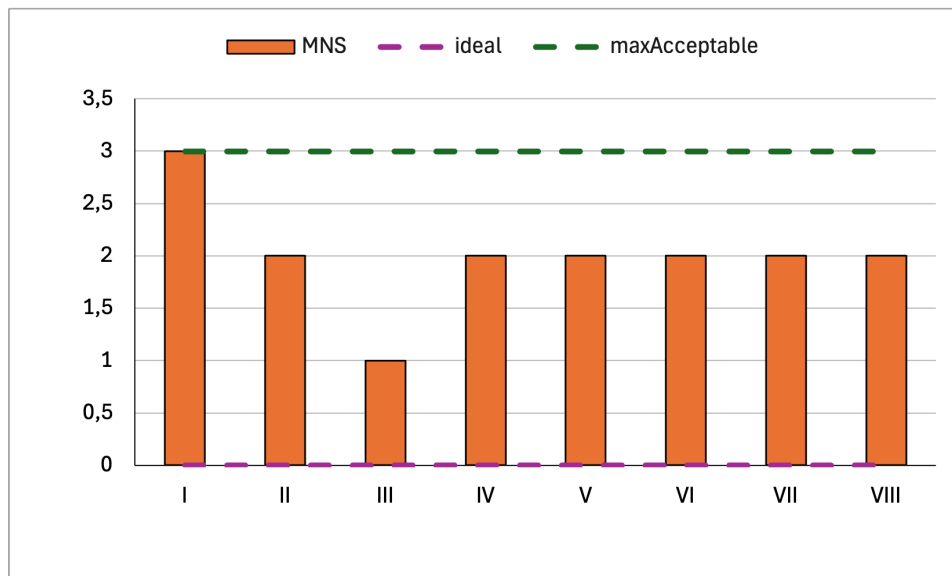


Figura 7: Metriche non soddisfatte

Analisi

La metrica che non risulta mai soddisfatta è EAC per la quale si rimanda all'analisi specifica. Altre metriche che risultano non soddisfatte sono ETC (primi due *sprint_G*), CPI (primo *sprint_G*) ed infine CV ed SV (dal quinto *sprint_G* per i quali si rimanda all'analisi specifica).

5.4 Qualità di processo - Sviluppo

5.4.1 Statement Coverage

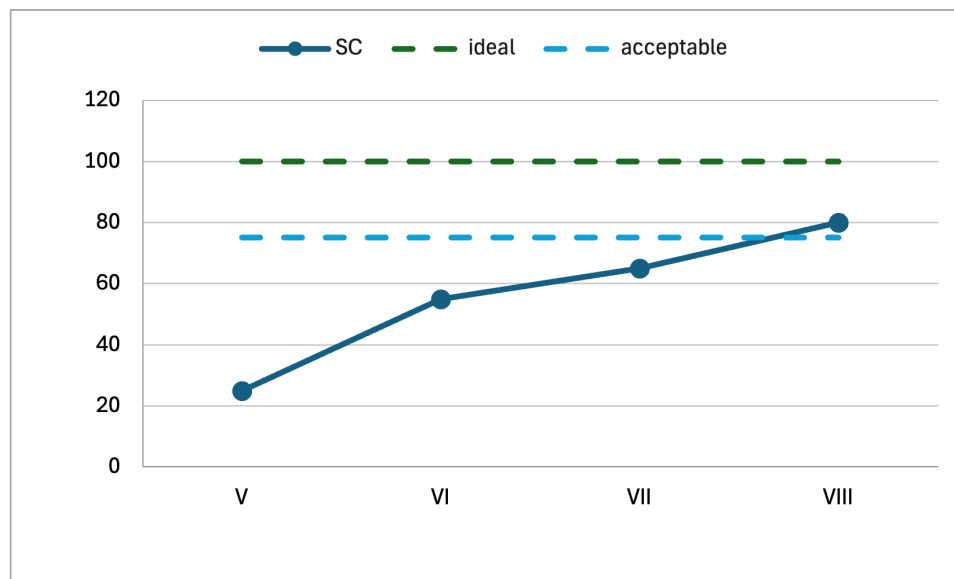


Figura 8: Statement Coverage

Analisi

L'incremento della SC indica che non solo vengono testate le funzioni più evidenti, ma si stanno anche affrontando dettagli e scenari più complessi, garantendo che il codice sia eseguito e validato in ogni sua parte.

5.5 Qualità di prodotto - Funzionalità

5.5.1 Requisiti soddisfatti

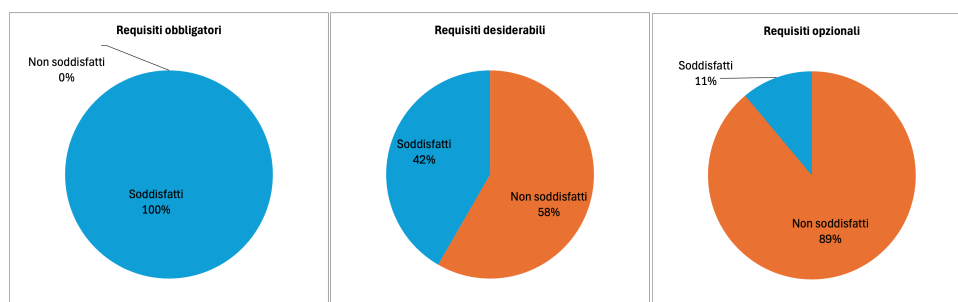


Figura 9: Requisiti soddisfatti

Analisi

La copertura dei requisiti obbligatori_G è del 100%, a conferma del pieno rispetto dei criteri minimi di accettazione e dei vincoli progettuali. I requisiti desiderabili_G risultano soddisfatti al 42%, indicando un buon livello di implementazione oltre la soglia minima,

con margini di miglioramento per potenziare ulteriormente il valore del prodotto. I requisiti opzionali_G hanno una copertura del 11%, coerente con la loro natura non prioritaria; rappresentano possibili spunti per evoluzioni future.

5.6 Qualità di prodotto - Affidabilità

5.6.1 PTCP, CC

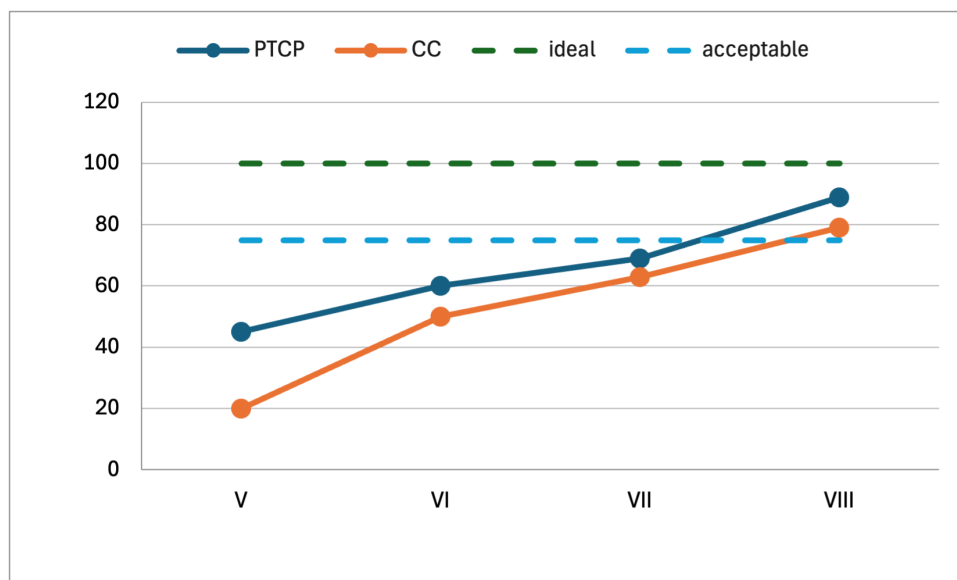


Figura 10: Passed Test Cases Percentage, Code Coverage

Analisi

Nei primi periodi di lavoro sull'MVP, ci sono stati molti test falliti (probabile fase di sviluppo iniziale), ma il miglioramento in seguito dimostra un'ottima gestione dei bug e la crescita della stabilità del codice. Anche la copertura è migliorata notevolmente, questo suggerisce che, a partire dal periodo VI, si è iniziato a testare in modo più sistematico, coprendo le feature principali.