



Sigma18Unipd@gmail.com

Piano di Qualifica

Responsabili	Mirco Borella Aleena Mathew Matteo Marangon Pietro Crotti Marco Egidi	
Redattori	Aleena Mathew Matteo Marangon Pietro Crotti	Versione 2.0.0 Tipo Documento Esterno
Verificatori	Carmelo Russello Carmelo Russello Matteo Marangon Pietro Crotti Alessandro Bernardello Mirco Borella Marco Egidi	Destinatari <i>Prof. Tullio Vardanega</i> <i>Prof. Riccardo Cardin</i> <i>Var Group S.p.A.</i> <i>Sigma18</i>

Abstract dei contenuti:

Piano di qualifica per il capitolato C3 proposto da Var Group S.p.A.

Registro delle versioni

Versione	Data	Autori	Verificatori	Descrizione Modifiche
2.0.0	2025/09/03	Aleena Mathew	Alessandro Bernardello	Stesura finale della sezione dei test e migliorie varie
1.3.0	2025/08/20	Carmelo Russello	Marco Egidi	Aggiornamento cruscotto di valutazione concluso il decimo sprint e aggiunta scheletro dei test
1.2.0	2025/08/06	Pietro Crotti	Matteo Marangon	Aggiornamento cruscotto di valutazione concluso il nono sprint
1.1.0	2025/07/17	Matteo Marangon	Pietro Crotti	Aggiornamento cruscotto di valutazione concluso l'ottavo sprint
1.0.0	2025/07/17	Aleena Mathew	Alessandro Bernardello	Migliorie impostazioni del documento e aggiornamento di una metrica
0.9.0	2025/07/11	Carmelo Russello	Mirco Borella	Aggiornamento cruscotto di valutazione concluso il settimo sprint
0.8.0	2025/06/26	Aleena Mathew	Carmelo Russello	Aggiornamento cruscotto di valutazione concluso il sesto sprint
0.7.0	2025/06/11	Matteo Marangon	Marco Egidi	Aggiornamento cruscotto di valutazione concluso il quinto sprint
0.6.0	2025/05/20	Pietro Crotti	Matteo Marangon	Aggiornamento di alcune metriche in accordo con le norme

Versione	Data	Autori	Verificatori	Descrizione Modifiche
0.5.0	2025/05/20	Aleena Mathew	Alessandro Bernardello	Scheletro per il cruscotto di valutazione
0.4.1	2025/05/14	Matteo Marangon	Pietro Crotti	Correzione di errori minori
0.4.0	2025/05/09	Aleena Mathew	Matteo Marangon	Correzione di errori e aggiunta elenco tabelle
0.3.0	2025/04/30	Matteo Marangon	Aleena Mathew	Aggiunta di nuove metriche
0.2.0	2025/04/30	Aleena Mathew	Matteo Marangon	Stesura qualità del prodotto
0.1.0	2025/04/17	Aleena Mathew	Carmelo Russello	Prima stesura del documento

Indice

Registro delle versioni	2
1. Introduzione	9
1.1. Scopo del documento	9
1.2. Scopo del progetto	9
1.3. Glossario	9
1.4. Riferimenti	9
1.4.1. Riferimenti normativi	9
1.4.2. Riferimenti informativi	9
2. Qualità di processo	10
2.1. Processi primari	10
2.1.1. Fornitura	10
2.1.1.1. Earned Value (MPC-EV)	10
2.1.1.2. Planned Value (MPC-PV)	10
2.1.1.3. Actual Cost (MPC-AC)	10
2.1.1.4. Estimated At Completion (MPC-EAC)	10
2.1.1.5. Estimated To Complete (MPC-ETC)	10
2.1.1.6. Cost Variance (MPC-CV)	10
2.1.1.7. Schedule variance (MPC-SV)	10
2.1.1.8. Cost performance index (MPC-CPI)	11
2.1.2. Sviluppo	11
2.1.2.1. Requirements Stability Index (MPC-RSI)	11
2.2. Processi di supporto	12
2.2.1. Documentazione	12
2.2.1.1. Correttezza ortografica (MPC-CO)	12
2.2.2. Verifica	12
2.2.2.1. Code coverage (MPC-CCO)	12
2.2.2.2. Test superati in percentuale (MPC-TSP)	12
2.2.3. Gestione della qualità	12
2.2.3.1. Satisfaction of Quality Metrics (MPC-SQM)	12
2.3. Processi organizzativi	13
2.3.1. Efficienza temporale (MPC-ET)	13
3. Qualità del prodotto	14
3.1. Funzionalità	14
3.1.1. Requisiti obbligatori soddisfatti (MPD-RO)	14
3.1.2. Requisiti desiderabili soddisfatti (MPD-RD)	14
3.1.3. Requisiti facoltativi soddisfatti (MPD-RF)	14
3.2. Affidabilità	14
3.2.1. Code coverage (MPD-CCO)	14
3.2.2. Branch coverage (MPD-BC)	14
3.2.3. Statement coverage (MPD-SC)	15
3.2.4. Failure Tolerance (MPD-FT)	15

3.2.5. Failure Frequency (MPD-FF)	15
3.3. Usabilità	15
3.3.1. Tempo di apprendimento (MPD-TA)	15
3.3.2. Facilità d'uso (MPD-FU)	15
3.4. Efficienza	16
3.4.1. Tempo medio di esecuzione (MPD-TE)	16
3.5. Manutenibilità	16
3.5.1. Complessità ciclomatica (MPD-CC)	16
3.5.2. Code Smell (MPD-CS)	16
3.5.3. Coefficient of Coupling (MPD-COC)	16
4. Metodi di testing	17
4.1. Nomenclatura	17
4.2. Test di unità	17
4.3. Test di integrazione	22
4.3.1. Test di integrazione backend	23
4.3.2. Test di integrazione frontend	25
4.4. Test di sistema	26
4.5. Test di accettazione	30
5. Cruscotto di Valutazione	32
5.1. Estimated at Completion (MPC-EAC)	32
5.1.1. RTB	32
5.1.2. PB	32
5.2. Planned Value (MPC-PV) & Earned Value (MPC-EV)	33
5.2.1. RTB	33
5.2.2. PB	33
5.3. Actual Cost (MPC-AC), Estimated To Complete (MPC-ETC) & Estimated At Completion (MPC-EAC)	34
5.3.1. RTB	34
5.3.2. PB	34
5.4. Cost Variance (MPC-CV) & Schedule Variance (MPC-SV)	35
5.4.1. RTB	35
5.4.2. PB	35
5.5. Cost Performance Index (MPC-CPI)	36
5.5.1. RTB	36
5.5.2. PB	36
5.6. Requirements Stability Index (MPC-RSI)	37
5.6.1. RTB	37
5.6.2. PB	37
5.7. Correttezza ortografica (MPC-CO)	38
5.7.1. RTB	38
5.7.2. PB	38
5.8. Satisfaction of Quality Metrics (MPC-SQM)	39
5.8.1. RTB	39

5.8.2. PB	39
5.9. Efficienza temporale (MPC-ET)	40
5.9.1. RTB	40
5.9.2. PB	40
5.10. Requisiti obbligatori soddisfatti (MPD-RO)	41
5.10.1. PB	41
5.11. Requisiti desiderabili e facoltativi soddisfatti (MPD-RD e MPD-RF)	42
5.11.1. PB	42
5.12. Code Coverage (MPD-CCO)	43
5.12.1. PB	43
5.12.2. Complessità ciclomatica (MPD-CC)	44
5.13. Test superati in percentuale (MPC-TSP)	45
5.13.1. PB	45

Elenco tabelle

Tabella 1	Valori accettabili e ottimi del processo di fornitura	11
Tabella 2	Valori accettabili e ottimi del processo di sviluppo	11
Tabella 3	Valori accettabili e ottimi del processo di documentazione	12
Tabella 4	Valori accettabili e ottimi del processo di verifica	12
Tabella 5	Valori accettabili e ottimi del processo di gestione della qualità	13
Tabella 6	Valori accettabili e ottimi dei processi organizzativi	13
Tabella 7	Valori accettabili e ottimi della funzionalità del prodotto	14
Tabella 8	Valori accettabili e ottimi dell'affidabilità del prodotto	15
Tabella 9	Valori accettabili e ottimi dell'usabilità del prodotto	16
Tabella 10	Valori accettabili e ottimi dell'efficienza del prodotto	16
Tabella 11	Valori accettabili e ottimi della manutenibilità del prodotto	16
Tabella 12	Test di unità	17
Tabella 13	Test di integrazione backend	23
Tabella 14	Test di integrazione frontend	25
Tabella 15	Test di sistema	26
Tabella 16	Test di accettazione	30

Elenco immagini

Figura 1	Stima del costo totale nel tempo	32
Figura 2	Stima dei valori EV e PV nel tempo	33
Figura 3	Stima dei valori AC, ETC e EAC nel tempo	34
Figura 4	Stima dei valori CV e SV nel tempo	35
Figura 5	Stima dei valori CPI nel tempo	36
Figura 6	Stima dei valori RSI nel tempo	37
Figura 7	Stima dei valori CO nel tempo	38
Figura 8	Stima dei valori SQM nel tempo	39
Figura 9	Stima dei valori ET nel tempo	40
Figura 10	Stima dei valori RO nel tempo	41
Figura 11	Stima dei valori RD e RF nel tempo	42
Figura 12	Stima dei valori CCO nel tempo	43
Figura 13	Stima dei valori CC nel tempo	44
Figura 14	Stima dei valori TSP nel tempo	45

1. Introduzione

1.1. Scopo del documento

Il presente documento ha lo scopo di definire le modalità di verifica e di validazione adottate dal gruppo al fine di garantire la qualità del prodotto finale.

Il documento sarà soggetto ad aggiornamenti durante il ciclo di vita del progetto per riflettere le modifiche apportate per raggiungere una migliore efficacia ed efficienza.

Sarà presente una sezione dedicata al cruscotto di valutazione, che fornirà un'analisi dell'andamento delle metriche di qualità scelte dal gruppo durante lo svolgimento del progetto.

1.2. Scopo del progetto

Il progetto ha lo scopo di creare un prodotto che permetta di automatizzare *routine_{GL}* digitali sfruttando l'intelligenza artificiale generativa in cloud.

L'utente potrà descrivere in linguaggio naturale l'automazione desiderata e il sistema genererà un flusso di lavoro che l'utente potrà modificare tramite un'interfaccia *drag & drop_{GL}*.

Sarà possibile salvare le automazioni generate e avviarle in un secondo momento.

1.3. Glossario

Data la presenza di termini tecnici e acronimi, è stato redatto un glossario per facilitare la comprensione del documento.

Alla prima occorrenza, tali termini saranno opportunamente segnalati tramite la seguente notazione: *parola_{GL}*, e sarà fornita un'accurata definizione nel Glossario.

1.4. Riferimenti

1.4.1. Riferimenti normativi

- **Norme di progetto** [versione 2.0.0]
https://sigma18unipd.github.io/documentiCompilati/3-PB/documentidiprogetto/normediprogetto_2.0.0.pdf
- **ISO/IEC 12207:1995** [ultimo accesso il: 28/08/2025]
https://www.math.unipd.it/~tullio/IS-1/2009/Approfondimenti/ISO_12207-1995.pdf
- **Capitolato C3 - Automatizzare le routine digitali tramite l'intelligenza generativa** [ultimo accesso il: 30/08/2025]
<https://www.math.unipd.it/~tullio/IS-1/2024/Progetto/C3.pdf>

1.4.2. Riferimenti informativi

- **Glossario** [versione 2.0.0]
https://sigma18unipd.github.io/documentiCompilati/3-PB/documentidiprogetto/glossario_2.0.0.pdf
- **Slide T07 - Qualità del prodotto** [ultimo accesso il: 30/08/2025]
<https://www.math.unipd.it/~tullio/IS-1/2024/Dispense/T07.pdf>
- **Slide T08 - Qualità del processo** [ultimo accesso il: 30/08/2025]
<https://www.math.unipd.it/~tullio/IS-1/2024/Dispense/T08.pdf>

2. Qualità di processo

La qualità di processo è un aspetto fondamentale per garantire il successo del progetto software poiché assicura che i processi utilizzati siano efficaci, efficienti e conformi agli standard di qualità prefissati.

Di seguito sono riportati gli obiettivi di qualità che il gruppo si impegna a raggiungere, suddivisi in tre categorie: *processi primari*, *processi di supporto* e *processi organizzativi*, come individuato dallo standard ISO/IEC 12207:1995.

2.1. Processi primari

I processi primari comprendono attività di acquisizione, gestione operativa, manutenzione del software, fornitura e sviluppo. Verranno esaminati solo gli ultimi due, data la natura didattica del progetto.

2.1.1. Fornitura

Si intendono le attività e i compiti svolti dal fornitore, in accordo con l'azienda proponente nello stabilire i vincoli e requisiti del progetto.

Con l'acronimo *MPC* (Minimum Predictive Capability) si indica il valore minimo da raggiungere.

2.1.1.1. Earned Value (MPC-EV)

È il valore del lavoro completato fino a un dato momento rispetto al budget pianificato e si ottiene dal prodotto tra *BAC* (Budget At Completion) e la percentuale di lavoro completato.

2.1.1.2. Planned Value (MPC-PV)

È il valore del lavoro pianificato da completare entro una determinata data.

Tale valore è ricavato dal prodotto tra *BAC* e la percentuale di lavoro pianificato da completare entro la data prefissata.

2.1.1.3. Actual Cost (MPC-AC)

Indica l'effettivo costo sostenuto fino ad un determinato momento per il lavoro svolto.

2.1.1.4. Estimated At Completion (MPC-EAC)

Rappresenta la stima del costo totale del progetto, basata sulle condizioni attuali.

Il suo valore è ottenuto dal rapporto tra *BAC* e *CPI* (Cost Performance Index).

2.1.1.5. Estimated To Complete (MPC-ETC)

Stima il costo finale per completare il progetto in base all'andamento attuale.

2.1.1.6. Cost Variance (MPC-CV)

Misura la differenza tra il valore guadagnato (EV) e il costo effettivo sostenuto (AC) fino a un quel momento.

2.1.1.7. Schedule variance (MPC-SV)

Misura la differenza tra il valore guadagnato (EV) e il valore pianificato (PV) fino a un quel momento. È utile per identificare eventuali ritardi o anticipi rispetto alla pianificazione, nello specifico valori negativi indicano ritardi.

2.1.1.8. Cost performance index (MPC-CPI)

Misura l'efficienza del costo per il lavoro svolto fino a un determinato momento, in base al valore ottenuto per ogni unità monetaria spesa.

Metrica	Nome	Valore Accettabile	Valore Ottimo
MPC-EV	Earned Value	≥ 0	$\leq \text{EAC}$
MPC-PV	Planned Value	≥ 0	$\leq \text{BAC}$
MPC-AC	Actual Cost	≥ 0	$\leq \text{EAC}$
MPC-EAC	Estimated At Completion	$\pm 5\% \text{ BAC}$	$= \text{BAC}$
MPC-ETC	Estimated To Complete	≥ 0	$\leq \text{EAC}$
MPC-CV	Cost Variance	$\geq -5\% \text{ BAC}$	≥ 0
MPC-SV	Schedule variance	$\pm 5\% \text{ BAC}$	≥ 0
MPC-CPI	Cost performance index	$0.9 \leq \text{CPI} \leq 1.1$	1

Tabella 1: Valori accettabili e ottimi del processo di fornitura

2.1.2. Sviluppo

Processo che ha lo scopo di descrivere le attività e i compiti necessari per creare e mantenere un sistema software, fondamentale per un prodotto finale che soddisfi i requisiti specificati nel contratto.

Di seguito sono riportate le relative metriche.

2.1.2.1. Requirements Stability Index (MPC-RSI)

Si tratta dell'indice di stabilità dei requisiti, data dalla percentuale di requisiti modificati rispetto al totale dei requisiti.

Un valore alto indica che i requisiti sono stabili e non soggetti a modifiche frequenti.

Metrica	Nome	Valore Accettabile	Valore Ottimo
MPC-RSI	Requirements Stability Index	$\geq 75\%$	100%

Tabella 2: Valori accettabili e ottimi del processo di sviluppo

2.2. Processi di supporto

I processi di supporto forniscono servizi e attività che assistono i processi primari. Tra questi rientrano la documentazione, la gestione della configurazione, il controllo della qualità, la verifica, la validazione e la risoluzione dei problemi.

2.2.1. Documentazione

La documentazione permette di avere un tracciamento delle attività svolte che facilita la comprensione, la manutenzione e l'evoluzione del prodotto durante il suo ciclo di vita.

2.2.1.1. Correttezza ortografica (MPC-CO)

La correttezza ortografica indica il numero di errori grammaticali e ortografici presenti in un documento.

Metrica	Nome	Valore Accettabile	Valore Ottimo
MPC-CO	Correttezza ortografica	2	0

Tabella 3: Valori accettabili e ottimi del processo di documentazione

2.2.2. Verifica

2.2.2.1. Code coverage (MPC-CCO)

È la quantità di codice eseguito durante i test. Consente di valutare la qualità dei test e garantire che il codice sia stato testato adeguatamente.

È stata richiesta un percentuale di copertura pari o superiore al 70%.

2.2.2.2. Test superati in percentuale (MPC-TSP)

Indica la percentuale di test automatizzati o manuali che sono stati eseguiti con successo rispetto al totale dei test previsti. Una percentuale alta indica che il sistema è stabile e che la maggior parte delle funzionalità funziona come previsto.

Metrica	Nome	Valore Accettabile	Valore Ottimo
MPC-CCO	Code coverage	$\geq 70\%$	100%
MPC-TSP	Test superati in percentuale	100%	100%

Tabella 4: Valori accettabili e ottimi del processo di verifica

2.2.3. Gestione della qualità

2.2.3.1. Satisfaction of Quality Metrics (MPC-SQM)

Indica la percentuale di metriche che soddisfano gli obiettivi minimi di qualità.

Il valore è ottenuto dal rapporto tra il numero di metriche di qualità soddisfatte e il numero totale di metriche di qualità.

Metrica	Nome	Valore Accettabile	Valore Ottimo
MPC-SQM	Satisfaction of Quality Metrics	≥80%	100%

Tabella 5: Valori accettabili e ottimi del processo di gestione della qualità

2.3. Processi organizzativi

I processi organizzativi riguardano la gestione e l'organizzazione del progetto come la gestione dei processi, il miglioramento e la formazione.

2.3.1. Efficienza temporale (MPC-ET)

Valuta l'efficienza con cui il tempo disponibile viene impiegato in attività produttive, ovvero quelle contribuiscono al raggiungimento degli obiettivi del progetto.

Metrica	Nome	Valore Accettabile	Valore Ottimo
MPC-ET	Efficienza temporale	≥50%	100%

Tabella 6: Valori accettabili e ottimi dei processi organizzativi

3. Qualità del prodotto

La qualità del prodotto valuta la capacità del software sviluppato di soddisfare i requisiti e le aspettative concordate. La valutazione della qualità del prodotto avviene considerando vari aspetti, tra cui la funzionalità, l'affidabilità, l'usabilità, l'efficienza e la manutenibilità.

L'obiettivo è produrre un software che non solo soddisfi le richieste del cliente, ma che rispetti specifici standard di qualità.

Il gruppo si impegna a rispettare le metriche di qualità del prodotto definite di seguito per garantire un elevato livello di qualità del prodotto finale.

3.1. Funzionalità

La funzionalità misura la capacità del software di soddisfare i requisiti obbligatori, desiderabili e opzionali.

3.1.1. Requisiti obbligatori soddisfatti (MPD-RO)

Indica la percentuale di requisiti obbligatori soddisfatti dal prodotto. Deve sempre avere un valore pari al 100% per garantire la conformità alle specifiche.

3.1.2. Requisiti desiderabili soddisfatti (MPD-RD)

Indica la percentuale di requisiti desiderabili soddisfatti dal prodotto. Un valore alto indica una maggiore soddisfazione del cliente.

3.1.3. Requisiti facoltativi soddisfatti (MPD-RF)

Indica la percentuale di requisiti facoltativi soddisfatti dal prodotto. Un valore alto aggiunge valore al prodotto.

Metrica	Nome	Valore Accettabile	Valore Ottimo
MPD-RO	Requisiti obbligatori soddisfatti	100%	100%
MPD-RD	Requisiti desiderabili soddisfatti	≥0%	100%
MPD-RF	Requisiti facoltativi soddisfatti	≥0%	100%

Tabella 7: Valori accettabili e ottimi della funzionalità del prodotto

3.2. Affidabilità

L'affidabilità misura la capacità del software di funzionare correttamente sotto specifiche condizioni.

3.2.1. Code coverage (MPD-CCO)

Indica la percentuale di codice eseguita durante i test.

Un valore alto indica una migliore copertura del codice, suggerendo che il codice è stato testato approfonditamente e abbia una bassa probabilità di contenere errori.

3.2.2. Branch coverage (MPD-BC)

Indica la percentuale di rami decisionali del codice eseguiti durante i test, utile per identificare scenari non testati.

Un valore alto suggerisce che il codice è stato testato approfonditamente e che ha una bassa probabilità di contenere errori.

3.2.3. Statement coverage (MPD-SC)

Indica la percentuale di istruzioni eseguite durante i test.

Un valore alto suggerisce un'analisi più approfondita del codice e una bassa probabilità di contenere errori. Il valore è dato dal rapporto tra il numero di istruzioni eseguite e il numero totale di istruzioni nel codice, moltiplicato per 100.

3.2.4. Failure Tolerance (MPD-FT)

Indica la capacità del software di mantenere un livello di prestazioni accettabile anche in caso di guasti o malfunzionamenti.

Un valore alto suggerisce che il software è capace di gestire i guasti senza compromettere le funzionalità principali.

3.2.5. Failure Frequency (MPD-FF)

Indica la frequenza con cui si verificano guasti o malfunzionamenti nel prodotto.

Un valore basso suggerisce che il prodotto è affidabile.

Metrica	Nome	Valore Accettabile	Valore Ottimo
MPD-CCO	Code coverage	$\geq 70\%$	100%
MPD-BC	Branch coverage	$\geq 60\%$	$\geq 85\%$
MPD-SC	Statement coverage	$\geq 60\%$	$\geq 85\%$
MPD-FT	Failure Tolerance	100%	100%
MPD-FF	Failure Frequency	0	0

Tabella 8: Valori accettabili e ottimi dell'affidabilità del prodotto

3.3. Usabilità

L'usabilità indica la facilità con cui gli utenti possono utilizzare il software.

3.3.1. Tempo di apprendimento (MPD-TA)

Indica il tempo necessario per un utente per imparare ad utilizzare il software.

Un valore basso indica che il prodotto è facile da usare e richiede poco tempo per essere appreso offrendo un'esperienza utente migliore.

3.3.2. Facilità d'uso (MPD-FU)

Indica il numero di errori commessi dagli utenti durante l'interazione con il software.

Un valore basso indica che il prodotto è facile da usare e che gli utenti riescono a completare le azioni richieste senza errori.

Metrica	Nome	Valore Accettabile	Valore Ottimo
MPD-TA	Tempo di apprendimento	≤ 15 minuti	≤ 5 minuti
MPD-FU	Facilità d'uso	≤ 4 errori	0 errori

Tabella 9: Valori accettabili e ottimi dell'usabilità del prodotto

3.4. Efficienza

3.4.1. Tempo medio di esecuzione (MPD-TE)

Misura il tempo medio impiegato da un *routine_{GL}* per completare la propria esecuzione. Tempi di esecuzione più brevi indicano un prodotto più efficiente, migliorando l'esperienza utente.

Metrica	Nome	Valore Accettabile	Valore Ottimo
MPD-TE	Tempo medio di esecuzione	≤ 20 secondi	≤ 15 secondi

Tabella 10: Valori accettabili e ottimi dell'efficienza del prodotto

3.5. Manutenibilità

3.5.1. Complessità ciclomatica (MPD-CC)

Indica la complessità del codice sorgente, si misura contando il numero di cammini indipendenti nel grafo di controllo del flusso.

Un valore basso indica che il codice è semplice e facile da mantenere.

3.5.2. Code Smell (MPD-CS)

Indica la presenza di potenziali problemi di progettazione o codice che potrebbero richiedere manutenzione.

Denota la possibile presenza di parti di codice non ottimali, che potrebbero causare difficoltà in futuro, come un'architettura poco chiara o sezioni di codice ripetitive.

Un valore basso indica che il codice è ben strutturato e non presenta problemi tecnici.

3.5.3. Coefficient of Coupling (MPD-COC)

Indica il grado di dipendenza tra i moduli o le componenti di un sistema.

Un valore alto implica che i moduli sono strettamente interconnessi, rendendo difficile apportare modifiche a un modulo senza influenzare gli altri.

Metrica	Nome	Valore Accettabile	Valore Ottimo
MPD-CC	Complessità ciclomatica	≤ 20	≤ 10
MPD-CS	Code Smell	≤ 10	0
MPD-COC	Coefficient of Coupling	≤ 0.5	≤ 0.3

Tabella 11: Valori accettabili e ottimi della manutenibilità del prodotto

4. Metodi di testing

La seguente sezione descrive le attività di testing effettuate per garantire la qualità del prodotto.

4.1. Nomenclatura

Ciascun test è identificato univocamente da un codice composto come segue:

[T[Tipologia]-[ID]]

dove:

- **T** : indica che si tratta di un test.
- **Tipologia** : indica la tipologia del test, che può essere:
 - **U** : test di unità;
 - **I** : test di integrazione;
 - **S** : test di sistema;
 - **A** : test di accettazione;
- **ID** : un numero progressivo del test, univoco all'interno della sua tipologia.

Per ogni test viene specificato lo stato di esecuzione, che può assumere i seguenti valori:

- **S**: superato;
- **F**: fallito;
- **N**: non eseguito.

4.2. Test di unità

I test di unità verificano il funzionamento corretto di componenti software più piccoli e indipendenti, sviluppati soprattutto nella fase di progettazione.

Di seguito viene fornita una tabella contenente i test di unità effettuati, di ogni test viene riportato il codice univoco, descrizione e stato di esecuzione.

È stato raggiunto un *code coverage* del 95%.

Codice	Descrizione	Stato
TU-01	Verificare che la configurazione del logging sia in modalità produzione, in assenza di variabili d'ambiente specifiche	S
TU-02	Verificare che la configurazione del logging sia in modalità sviluppo, se la variabile d'ambiente ENV è impostata a «dev»	S
TU-03	Verificare che la variabile d'ambiente ENV sia case insensitive	S
TU-04	Verificare che modalità sviluppo sia attiva se la variabile d'ambiente DEV è impostata a «true»	S
TU-05	Verificare che modalità sviluppo sia attiva se la variabile d'ambiente DEV è impostata a «1»	S

Codice	Descrizione	Stato
TU-06	Verificare che modalità sviluppo sia attiva se la variabile d'ambiente DEV è impostata a «yes»	S
TU-07	Verificare che modalità sviluppo sia attiva se la variabile d'ambiente DEV è impostata a «TRUE»	S
TU-08	Verificare che modalità sviluppo non sia attiva se la variabile d'ambiente DEV è impostata a valori diversi da «true», «1», «yes» o «TRUE»	S
TU-09	Verificare che se ENV è posta a «production» e DEV a «1» prevalga la modalità sviluppo	S
TU-10	Verificare che le variabili, se impostate, sovrascrivano quelle di default	S
TU-11	Verificare che il valore della variabile d'ambiente LOG_LEVEL sia convertito in maiuscolo	S
TU-12	Verificare che il formatter del logging sia configurato con una stringa con il corretto formato	S
TU-13	Verificare che venga aggiunto uno <i>StreamHandler_{GL}</i> , se non presente	S
TU-14	Verificare che non venga aggiunto uno <i>StreamHandler</i> , se già presente	S
TU-15	Verificare che la classe <i>BlockFactory</i> implementi senza errori il pattern <i>singleton</i>	S
TU-16	Verificare che la creazione del <i>singleton</i> sia thread-safe	S
TU-17	Verificare che l'istanza di <i>BlockFactory</i> sia inizializzata correttamente	S
TU-18	Verificare che la funzione di registrazione dei blocchi di <i>BlockFactory</i> funzioni correttamente	S
TU-19	Verificare che la funzione di registrazione di <i>BlockFactory</i> accetti solo classi che sono sottoclassi di <i>Block</i>	S
TU-20	Verificare che la funzione di registrazione di <i>BlockFactory</i> sia <i>thread-safe</i>	S
TU-21	Verificare la corretta creazione di un blocco	S
TU-22	Verificare che la creazione di un blocco con un tipo non registrato sollevi un errore	S
TU-23	Verificare il corretto comportamento della funzione <i>get_supported_types</i> della classe <i>BlockFactory</i> quando il registro dei blocchi è vuoto	S
TU-24	Verificare il corretto comportamento della funzione <i>get_supported_types</i> della classe <i>BlockFactory</i> in presenza di blocchi registrati	S
TU-25	Verificare che la funzione <i>lookup_implemented</i> ritorni <i>true</i> per i tipi di blocco registrati e <i>false</i> per quelli non registrati	S

Codice	Descrizione	Stato
TU-26	Verificare che la registrazione di un blocco venga correttamente tracciata nei log dal sistema	S
TU-27	Verificare che la creazione di un blocco venga correttamente tracciata nei log dal sistema	S
TU-28	Verificare l'inizializzazione di un blocco con parametri di default	S
TU-29	Verificare l'inizializzazione di un blocco con parametri personalizzati	S
TU-30	Verificare che il metodo <code>_get_input</code> della classe <code>Block</code> recuperi correttamente i valori dal dizionario, gestendo chiavi mancanti e valori di default.	S
TU-31	Verificare che il metodo <code>_get_setting</code> della classe <code>Block</code> recuperi correttamente i valori dalle impostazioni	S
TU-32	Verificare che il metodo <code>_set_output</code> della classe <code>Block</code> aggiorni l'output e registri i log di esecuzione correttamente	S
TU-33	Verificare che il metodo <code>_get_output</code> della classe <code>Block</code> ritorni correttamente i dati di output	S
TU-34	Verificare il flusso di esecuzione di un blocco	S
TU-35	Verificare la funzionalità di logging all'interno di un blocco	S
TU-36	Verificare la corretta implementazione del pattern <i>visitor</i>	S
TU-37	Verificare la corretta rappresentazione degli oggetti blocco	S
TU-38	Verificare che la classe astratta <code>Block</code> non sia istanziabile	S
TU-39	Verificare il comportamento classe astratta <code>Block</code> sia corretto	S
TU-40	Verificare che i valori di <code>Status</code> usati per la rappresentazione dello stato siano corretti	S
TU-41	Verificare che la classe <code>ExecutionLog</code> venga istanziata correttamente	S
TU-42	Verificare che la classe <code>FlaskAppSingleton</code> rispetti il pattern <i>singleton</i>	S
TU-43	Verificare che la classe <code>FlaskAppSingleton</code> venga inizializzata correttamente	S
TU-44	Verificare che la classe <code>FlaskAppSingleton</code> gestisca correttamente le richieste	S
TU-45	Verifica che la classe <code>FlaskAppSingleton</code> non consenta l'istanziazione multipla	S
TU-46	Verificare che la classe <code>MongoDBSingleton</code> rispetti il pattern <i>singleton</i>	S
TU-47	Verificare che la corretta istanziazione della classe <code>MongoDBSingleton</code> in presenza di un oggetto <i>Flask app</i>	S

Codice	Descrizione	Stato
TU-48	Verificare che la corretta istanziazione della classe MongoDBSingleton in assenza di un oggetto <i>Flask</i> app	S
TU-49	Verificare che il metodo <code>get_db</code> della classe MongoDBSingleton restituisca l'oggetto database corretto	S
TU-50	Verificare che l'attributo <code>mongo</code> venga inizializzato correttamente	S
TU-51	Verificare che l'istanza <i>singleton</i> persista anche se acceduta in modi diversi	S
TU-52	Verificare che venga sollevata un'eccezione se fallisce l'inizializzazione di PyMongo	S
TU-53	Verificare che venga sollevata un'eccezione se si prova a chiamare il metodo <code>get_db</code> quando l'attributo <code>mongo</code> è <i>None</i>	S
TU-54	Verificare che la funzione <code>generateJwt</code> generi correttamente un token JWT contenente email e scadenza	S
TU-55	Verificare che la funzione <code>verifyJwt</code> decodifichi e verifichi correttamente un token	S
TU-56	Verificare che la funzione <code>generateJwt</code> gestisca correttamente il caso in cui l'email sia una stringa vuota	S
TU-57	Verificare che la funzione <code>agent_facade</code> funzioni correttamente in condizioni normali, simulando l'interazione con AWS	S
TU-58	Verificare che la funzione <code>agent_facade</code> gestisca correttamente il caso in cui il prompt sia vuoto	S
TU-59	Verificare che la funzione <code>agent_facade</code> gestisca correttamente il caso in cui il prompt contenga caratteri speciali e emoji	S
TU-60	Verificare che la classe astratta <code>JsonParserStrategy</code> non possa essere istanziata direttamente	S
TU-61	Verificare che le sottoclassi di <code>JsonParserStrategy</code> implementino il metodo <code>parse</code>	S
TU-62	Verificare che il metodo <code>parse</code> gestisca correttamente le stringhe JSON	S
TU-63	Verificare che il comportamento della classe <code>AISummarize</code> , assicurando che il metodo <code>execute</code> restituisca un riassunto corretto	S
TU-64	Verificare che la classe <code>SystemWaitSeconds</code> gestisca correttamente input di tipo <i>int</i>	S
TU-65	Verificare che la classe <code>SystemWaitSeconds</code> gestisca correttamente input di tipo <i>float</i>	S

Codice	Descrizione	Stato
TU-66	Verificare che la classe <code>SystemWaitSeconds</code> gestisca correttamente input di tipo <i>string</i>	S
TU-67	Verificare che funzione <code>validate_inputs</code> della classe <code>SystemWaitSeconds</code> restituisca <i>True</i> per i tipi di input validi come <i>int</i> , <i>float</i> e <i>stringhe</i>	S
TU-68	Verificare che funzione <code>validate_inputs</code> della classe <code>SystemWaitSeconds</code> restituisca <i>False</i> in presenza di valori negativi	S
TU-69	Verificare che funzione <code>validate_inputs</code> della classe <code>SystemWaitSeconds</code> restituisca <i>False</i> in presenza di stringhe non numeriche	S
TU-70	Verificare che funzione <code>validate_inputs</code> della classe <code>SystemWaitSeconds</code> restituisca <i>False</i> in assenza di input	S
TU-71	Verificare che funzione <code>execute</code> della classe <code>SystemWaitSeconds</code> sia eseguita correttamente	S
TU-72	Verificare che la funzione <code>validate_inputs</code> della classe <code>TelegramSendMessage</code> restituisca <i>True</i> se sono presenti tutti gli input richiesti, ovvero <code>botToken</code> , <code>chatId</code> e <code>message</code>	S
TU-73	Verificare che la funzione <code>validate_inputs</code> della classe <code>TelegramSendMessage</code> restituisca <i>False</i> se non sono presenti tutti gli input richiesti	S
TU-74	Verificare che funzione <code>execute</code> della classe <code>TelegramSendMessage</code> sia eseguita correttamente	S
TU-75	Verificare che funzione <code>execute</code> della classe <code>TelegramSendMessage</code> sia eseguita correttamente quando <code>"message": "{{LASTOUTPUT}}"</code>	S
TU-76	Verificare il comportamento di <code>execute</code> in presenza di errori di rete	S
TU-77	Verificare che la funzione <code>validate_inputs</code> della classe <code>NotionGetPage</code> restituisca <i>True</i> se sono presenti tutti gli input richiesti, ovvero <code>pageID</code> e <code>internalIntegrationToken</code>	S
TU-78	Verificare che la funzione <code>validate_inputs</code> della classe <code>NotionGetPage</code> restituisca <i>False</i> se non sono presenti tutti gli input richiesti	S
TU-79	Verificare che la funzione <code>execute</code> della classe <code>NotionGetPage</code> sia implementata correttamente	S
TU-80	Verificare che la funzione <code>execute</code> della classe <code>NotionGetPage</code> restituisca uno stato di errore in presenza di un errore nella Notion API	S
TU-81	Verificare che la funzione <code>execute</code> della classe <code>NotionGetPage</code> restituisca uno stato di errore in presenza di token invalido	S

Codice	Descrizione	Stato
TU-82	Verificare che la classe FlowIterator sia inizializzata correttamente	S
TU-83	Verificare che l'esecuzione dei blocchi avvenga in sequenza e che ogni blocco venga eseguito	S
TU-84	Verificare la gestione dell'esecuzione dei blocchi in presenza di errori nell'esecuzione di un blocco, verificando che l'esecuzione si interrompa e che siano presenti i log di errore	S
TU-85	Verificare che il metodo add_json della classe BaseSanitizationStrategy funzioni correttamente gestendo vari tipi di input	S
TU-86	Verificare che il metodo add_json della classe BaseSanitizationStrategy non sovrascriva valori esistenti	S
TU-87	Verificare che il metodo add_field_if_missing della classe BaseSanitizationStrategy imposti il tipo di default, se mancante	S
TU-88	Verificare che più chiamate a _position_counter della classe BaseSanitizationStrategy generino una griglia di posizioni corretta	S
TU-89	Verificare che l'inizializzazione della classe FlowManager crei tutti i componenti necessari	S
TU-90	Verificare che il metodo parse_json della classe FlowManager crei i blocchi corretti dopo aver ricevuto i dati JSON	S
TU-91	Verificare che il metodo parse_json della classe FlowManager sollevi un errore se si cerca di implementare un blocco di tipo non implementato	S
TU-92	Verificare che il metodo start_workflow della classe FlowManager ritorni uno stato <i>completed</i> se il flusso viene eseguito correttamente	S
TU-93	Verificare che il metodo start_workflow della classe FlowManager ritorni uno stato <i>failed</i> se il flusso non viene eseguito	S
TU-94	Verifica che il metodo get_status della classe FlowManager ritorni gli stati	S
TU-95	Verifica che la classe FlowManagerFactory rispetti il pattern <i>singleton</i>	S
TU-96	Verifica che la classe FlowManagerFactory restituisca un'istanza di FlowManager	S

Tabella 12: Test di unità

4.3. Test di integrazione

I test di integrazione vengono eseguiti successivamente ai test di unità e verificano l'interazione tra più unità software per garantire una corretta integrazione e funzionamento del sistema.

Di seguito vengono fornite due tabelle che riportano i test di integrazione effettuati, la prima relativa al backend e la seconda al frontend, di ogni test viene riportato il codice univoco, descrizione e stato di esecuzione.

4.3.1. Test di integrazione backend

Codice	Descrizione	Stato
TI-01	Verificare che il login di un utente tramite l'endpoint /login funzioni correttamente quando le credenziali sono corrette	S
TI-02	Verificare che venga lanciata un'eccezione con HTTP status 401 se si effettua il login con credenziali errate e che il messaggio d'errore sia «Invalid email or password»	S
TI-03	Verificare che venga lanciata un'eccezione con HTTP status 401 se si effettua il login con un'account non verificato e che il messaggio d'errore sia «User account not confirmed»	S
TI-04	Verificare che venga lanciata un'eccezione con HTTP status 401 se si effettua il login con un'account non esistente e che il messaggio d'errore sia «User not found»	S
TI-05	Verificare che venga lanciata un'eccezione con HTTP status 429 se si effettuano troppi tentativi di login e che il messaggio d'errore sia «Too many login attempts. Please try again later»	S
TI-06	Verificare che la registrazione di un nuovo utente tramite l'endpoint /register funzioni correttamente quando i dati sono validi	S
TI-07	Verificare che venga lanciata un'eccezione con HTTP status 409 se si effettua la registrazione con un'email già in uso e che il messaggio d'errore sia «User already exists»	S
TI-08	Verificare che venga lanciata un'eccezione con HTTP status 500 se si effettuano più registrazioni di quelle consentite e che il messaggio d'errore sia «Email quota limit exceeded»	S
TI-09	Verificare che la verifica di un utente tramite l'endpoint /confirm funzioni correttamente quando i dati sono corretti	S
TI-10	Verificare che venga lanciata un'eccezione con HTTP status 404 se si tenta di verificare un account non esistente e che il messaggio d'errore sia «User not found»	S
TI-11	Verificare che venga lanciata un'eccezione con HTTP status 404 se si inserisce un codice non valido per verificare un account e che il messaggio d'errore sia «Code not valid»	S

Codice	Descrizione	Stato
TI-12	Verificare che venga lanciata un'eccezione con HTTP status 404 se si inserisce un codice scaduto per verificare un account e che il messaggio d'errore sia «Code Expired»	S
TI-13	Verificare che l'endpoint /dashboard restituisca correttamente la lista dei workflow dell'utente autenticato	S
TI-14	Verificare che l'endpoint /dashboard neghi l'accesso e reindirizzi l'utente quando la verifica JWT fallisce, lanciando un'eccezione con HTTP status 302	S
TI-15	Verificare che l'endpoint /logout funzioni correttamente per un utente autenticato	S
TI-16	Verificare che l'endpoint /api/new permetta la creazione di un nuovo workflow ad un utente autenticato	S
TI-17	Verificare che l'endpoint /api/new lanci un'eccezione con HTTP status 400 se si cerca di creare un workflow con il campo nome vuoto e che il messaggio d'errore sia «Workflow name is required»	S
TI-18	Verificare che l'endpoint /api/new lanci un'eccezione con HTTP status 400 se si cerca di creare un workflow con nome che supera i 25 caratteri e che il messaggio d'errore sia «Workflow name must be less than 25 characters»	S
TI-19	Verificare che l'endpoint /api/flows/<id> restituisca correttamente i dati di un workflow esistente ad un utente autenticato	S
TI-20	Verificare che l'endpoint /api/flows/<id> lanci un'eccezione con HTTP status 404 se si richiede un workflow inesistente e che il messaggio d'errore sia «Workflow not found»	S
TI-21	Verificare che l'endpoint /api/flows/<id>/delete permetta ad un utente autenticato di eliminare un workflow esistente	S
TI-22	Verificare che l'endpoint /api/flows/<id>/delete lanci un'eccezione con HTTP status 404 se si tenta di eliminare un workflow inesistente e che il messaggio d'errore sia «Workflow not found»	S
TI-23	Verificare che l'endpoint /api/flows/<id>/save permetta ad un utente autenticato di salvare correttamente un workflow	S
TI-24	Verificare che l'endpoint /api/flows/<id>/save lanci un'eccezione con HTTP status 400 se si tenta di salvare un workflow con un nome troppo lungo e che il messaggio d'errore sia «Workflow name must be less than 25 characters»	S
TI-25	Verificare che l'endpoint /api/flows/<id>/run permetta l'esecuzione di un workflow esistente ad un utente autenticato	S

Codice	Descrizione	Stato
TI-26	Verificare che l'endpoint <code>/api/flows/<id>/run</code> lanci un'eccezione con HTTP status 404 se si tenta di eseguire un <i>workflow</i> inesistente e che il messaggio d'errore sia «Workflow not found»	S
TI-27	Verificare che l'endpoint <code>/api/prompt</code> generi correttamente un <i>workflow</i> tramite AI ad un utente autenticato	S
TI-28	Verificare che l'endpoint <code>/api/prompt</code> lanci un'eccezione con HTTP status 400 se si tenta di generare un <i>workflow</i> tramite AI senza il campo <code>prompt</code> e che il messaggio d'errore sia «Prompt is required»	S
TI-29	Verificare che l'endpoint <code>/api/prompt</code> lanci un'eccezione con HTTP status 400 se si tenta di generare un <i>workflow</i> tramite AI con il campo <code>prompt</code> vuoto e che il messaggio d'errore sia «Prompt is required»	S

Tabella 13: Test di integrazione backend

4.3.2. Test di integrazione frontend

Per effettuare i test di integrazione del frontend sono stati realizzati test end-to-end utilizzando il framework Cypress.

La scelta è stata motivata dalla capacità del framework di fornire un ambiente integrato per scrivere, eseguire e debuggare test, permettendo di validare funzionalità, interazioni utente e prestazioni. L'approccio end-to-end garantisce che tutte le componenti funzionino correttamente insieme, identificando errori che potrebbero sfuggire ai test di unità o di integrazione tradizionali.

Codice	Descrizione	Stato
TI-30	Verificare che la pagina di login venga visualizzata correttamente	S
TI-31	Verificare che vengano mostrati messaggi di errore se si tenta di effettuare il login lasciando i campi vuoti, verificando che il messaggio d'errore sia «Invalid email address» per il campo email e «Password cannot be empty» per il campo password	S
TI-32	Verificare che l'utente venga reindirizzato alla pagina <i>dashboard</i> dopo aver effettuato correttamente il login	S
TI-33	Verificare che vengano gestiti correttamente gli errori provenienti dal <i>backend</i> simulando un login effettuato con credenziali errate	S
TI-34	Verificare che l'utente venga reindirizzato dalla pagina <i>login</i> alla pagina di registrazione dopo aver cliccato sul bottone «Register»	S
TI-35	Verificare che l'utente venga reindirizzato dalla pagina <i>login</i> alla pagina di conferma dopo aver cliccato sul bottone «Confirm your account»	S
TI-36	Verificare che la pagina di <i>register</i> venga visualizzata correttamente	S

Codice	Descrizione	Stato
TI-37	Verificare che vengano mostrati messaggi di errore se si tenta di effettuare la registrazione lasciando i campi vuoti, verificando che il messaggio d'errore sia «Invalid email address» per il campo <i>email</i> e «Your password must have at least 8 characters» per il campo <i>password</i> e <i>conferma password</i>	S
TI-38	Verificare che venga mostrato un messaggio di errore se, in fase di registrazione, la password e la conferma password non corrispondono, verificando che il messaggio d'errore sia «Passwords don't match»	S
TI-39	Verificare che l'utente venga reindirizzato alla pagina di conferma account dopo aver effettuato correttamente la registrazione	S
TI-40	Verificare che vengano gestiti correttamente gli errori provenienti dal <i>backend</i> simulando una registrazione effettuata con un'email già in uso	S
TI-41	Verificare che l'utente venga reindirizzato dalla pagina di registrazione alla pagina di login dopo aver cliccato sul bottone «Login»	S
TI-42	Verificare che l'utente venga reindirizzato dalla pagina di registrazione alla pagina di conferma dopo aver cliccato sul bottone «Confirm your account»	S
TI-43	Verificare che la pagina di conferma venga visualizzata correttamente	S
TI-44	Verificare che vengano mostrati messaggi di errore se i dati non sono validi quando si conferma l'account	S
TI-45	Verificare che l'utente venga reindirizzato alla pagina di login dopo aver confermato correttamente l'account	S
TI-46	Verificare che vengano gestiti correttamente gli errori provenienti dal <i>backend</i> simulando una verifica effettuata con codice errato	S
TI-47	Verificare che l'utente venga reindirizzato dalla pagina di conferma alla pagina di login dopo aver cliccato sul bottone «Login»	S
TI-48	Verificare che l'utente venga reindirizzato dalla pagina di conferma alla pagina di registrazione dopo aver cliccato sul bottone «Register»	S

Tabella 14: Test di integrazione frontend

4.4. Test di sistema

I test di sistema verificano il funzionamento del sistema software nel suo complesso per garantire il soddisfacimento delle specifiche funzionali, prestazionali e di qualità concordate, presenti nel documento dell'analisi dei requisiti [versione 2.0.0].

Di seguito viene fornita una tabella contenente i test di sistema effettuati, di ogni test viene riportato il codice univoco, descrizione e stato di esecuzione.

Codice	Descrizione	Stato
TS-01	Verificare che l'utente possa effettuare <i>login</i> con il proprio account per accedere al servizio	S
TS-02	Verificare che l'utente autenticato possa inserire la sua <i>e-mail</i> per accedere all'applicativo	S
TS-03	Verificare che l'utente possa inserire la sua <i>password</i> per accedere all'applicativo	S
TS-04	Verificare che il sistema restituisca un errore se si tenta di eseguire il login con una mail non registrata	S
TS-05	Verificare che il sistema restituisca un errore se rileva ripetuti tentativi di accesso	S
TS-06	Verificare che il sistema restituisca un errore se si tenta di eseguire il login con una mail non verificata	S
TS-07	Verificare che l'utente non autenticato possa registrarsi con un nuovo account	S
TS-08	Verificare che l'utente non autenticato possa inserire la sua <i>e-mail</i> per registrarsi nell'applicativo	S
TS-09	Verificare che l'utente possa creare la sua <i>password</i> per registrarsi nell'applicativo	S
TS-10	Verificare che l'utente possa reinserire la sua password per la registrazione nell'applicativo	S
TS-11	Verificare che il sistema restituisca un errore per credenziali non valide inserite dall'utente	S
TS-12	Verificare che il sistema restituisca un errore se l' <i>e-mail</i> è già in uso in fase di registrazione	S
TS-13	Verificare che il sistema restituisca un errore se le <i>password</i> non corrispondono tra loro in fase di registrazione	S
TS-14	Verificare che il sistema restituisca un errore se la <i>password</i> creata è inferiore a 8 caratteri in fase di registrazione	S
TS-15	Verificare che il sistema restituisca un errore se l' <i>e-mail</i> è già in uso in fase di verifica	S
TS-16	Verificare che il sistema restituisca un errore se si lascia il campo password vuoto	S
TS-17	Verificare che il sistema restituisca un errore nel caso si riscontrino problemi	S
TS-18	Verificare che l'utente possa verificare l'account creato tramite codice di verifica ricevuto per <i>e-mail</i>	S

Codice	Descrizione	Stato
TS-19	Verificare che il sistema restituisca un errore se l'utente tenta di concludere la registrazione senza inserire il codice di verifica	S
TS-20	Verificare che il sistema restituisca un errore se il codice di conferma inserito dall'utente è scaduto	S
TS-21	Verificare che il sistema restituisca un errore se il codice di conferma inserito dall'utente è errato	S
TS-22	Verificare che l'utente possa visualizzare la dashboard in seguito al login nell'applicativo	S
TS-23	Verificare che l'utente autenticato possa vedere i <i>workflow</i> creati nella dashboard	S
TS-24	Verificare che l'utente possa creare una nuova <i>routine</i>	S
TS-25	Verificare che l'utente possa modificare il nome di una <i>routine</i>	S
TS-26	Verificare che il sistema restituisca un errore se il nome del <i>workflow</i> viene lasciato vuoto	S
TS-27	Verificare che il sistema restituisca un errore se il nome del <i>workflow</i> ha più di 25 caratteri	S
TS-28	Verificare che l'utente possa generare una <i>routine</i> tramite linguaggio naturale	S
TS-29	Verificare che il sistema restituisca un errore se il prompt di generazione di una <i>routine</i> tramite linguaggio naturale viene lasciato vuoto	S
TS-30	Verificare che l'utente possa visualizzare il nome di una <i>routine</i> esistente	S
TS-31	Verificare che l'utente possa visualizzare i dettagli di una <i>routine</i> esistente	S
TS-32	Verificare che l'utente possa visualizzare il diagramma dei blocchi di una <i>routine</i> esistente	S
TS-33	Verificare che l'utente possa eliminare una <i>routine</i> esistente	S
TS-34	Verificare che il sistema restituisca un errore se si tenta di interagire con un <i>workflow</i> inesistente	S
TS-35	Verificare che l'utente possa avviare una <i>routine</i> esistente	S
TS-36	Verificare che l'utente possa avviare una <i>routine</i> esistente dalla dashboard	S
TS-37	Verificare che l'utente possa avviare una <i>routine</i> esistente dalla pagina di modifica del flusso	S
TS-38	Verificare che il sistema restituisca un errore se l'esecuzione del flusso non va a buon fine	S
TS-39	Verificare che l'utente possa aggiungere un blocco ad una <i>routine</i> esistente	S

Codice	Descrizione	Stato
TS-40	Verificare che l'utente possa aggiungere un blocco del tipo «Telegram - Send Bot Message» ad una <i>routine</i> esistente	S
TS-41	Verificare che l'utente possa aggiungere un blocco del tipo «AI - Summarize» ad una <i>routine</i> esistente	S
TS-42	Verificare che l'utente possa aggiungere un blocco del tipo «System - Wait Second(s)» ad una <i>routine</i> esistente	S
TS-43	Verificare che l'utente possa aggiungere un blocco del tipo «Notion - Get Page» ad una <i>routine</i> esistente	S
TS-44	Verificare che l'utente possa visualizzare le impostazioni di un singolo blocco	S
TS-45	Verificare che l'utente possa visualizzare le impostazioni di un blocco del tipo «Telegram - Send Bot Message»	S
TS-46	Verificare che l'utente possa visualizzare le impostazioni di un blocco del tipo «System - Wait Second(s)»	S
TS-47	Verificare che l'utente possa visualizzare le impostazioni di un blocco del tipo «Notion - Get Page»	S
TS-48	Verificare che l'utente possa modificare le impostazioni di un singolo blocco	S
TS-49	Verificare che l'utente possa modificare le impostazioni di un blocco del tipo «Telegram - Send Bot Message»	S
TS-50	Verificare che l'utente possa modificare le impostazioni di un blocco del tipo «System - Wait Second(s)»	S
TS-51	Verificare che l'utente possa modificare le impostazioni di un blocco del tipo «Notion - Get Page»	S
TS-52	Verificare che il sistema salvi le modifiche apportate dall'utente alla <i>routine</i> se viene premuto il tasto di salvataggio	S
TS-53	Verificare che l'utente possa eliminare un blocco da una <i>routine</i> esistente	S
TS-54	Verificare che l'utente possa eliminare un blocco da una <i>routine</i> esistente da tastiera premendo il tasto «backspace»	S
TS-55	Verificare che l'utente possa eliminare un blocco da una <i>routine</i> esistente da interfaccia grafica	S
TS-56	Verificare che l'utente possa collegare due blocchi di una <i>routine</i> esistente	S
TS-57	Verificare che l'utente possa scollegare due blocchi di una <i>routine</i> esistente	S
TS-58	Verificare che l'utente possa ritornare alla dashboard dalla pagina di modifica flusso	S

Codice	Descrizione	Stato
TS-59	Verificare che l'utente possa impostare la modalità del client in dark mode o light mode	S
TS-60	Verificare che l'utente possa effettuare il <i>logout</i> dall'applicativo	S

Tabella 15: Test di sistema

4.5. Test di accettazione

I test di accettazione sono condotti per verificare che il sistema soddisfi i requisiti e le aspettative del contrattuali, motivo per cui sono condotti insieme al committente.

Il loro successo è fondamentale per garantire il rilascio definitivo del prodotto.

Di seguito viene riportata una tabella contenente i test di accettazione effettuati in presenza dell'azienda proponente, di ogni test viene riportato il codice univoco, descrizione e stato di esecuzione.

Codice	Descrizione	Stato
TA-01	Verificare che l'utente non autenticato possa effettuare la registrazione per accedere al servizio, inserendo mail, password e conferma password	S
TA-02	Verificare che venga restituito un messaggio di errore se le due password non corrispondono	S
TA-03	Verificare che venga mostrato un messaggio d'errore se la password non contiene almeno 8 caratteri	S
TA-04	Verificare che l'utente non autenticato possa verificare il proprio account inserendo il codice di verifica ricevuto via email	S
TA-05	Verificare che l'utente non autenticato possa effettuare il login inserendo email e password	S
TA-06	Verificare che l'utente non autenticato riceva un messaggio di errore se tenta di effettuare il login con credenziali errate	S
TA-07	Verificare che l'utente autenticato acceda direttamente alla dashboard	S
TA-08	Verificare che l'utente autenticato possa vedere i <i>workflow</i> creati nella dashboard	S
TA-09	Verificare che l'utente autenticato possa creare nuovi <i>workflow</i>	S
TA-10	Verificare che l'utente autenticato possa inserire il nome del <i>workflow</i> che desidera creare	S
TA-11	Verificare che l'utente autenticato riceva un messaggio di errore se prova a creare un nuovo <i>workflow</i> lasciando vuoto il campo nome o se inserisce un nome con più di 25 caratteri	S

Codice	Descrizione	Stato
TA-12	Verificare che l'utente possa visualizzare il diagramma dei blocchi di una <i>routine</i> esistente	S
TA-13	Verificare che l'utente autenticato possa generare un <i>workflow</i> usando il linguaggio naturale	S
TA-14	Verificare che l'utente autenticato possa creare un <i>workflow</i> manualmente selezionando i blocchi desiderati	S
TA-15	Verificare che l'utente autenticato possa modificare il <i>workflow</i> selezionato	S
TA-16	Verificare che l'utente autenticato possa aggiungere un blocco manualmente tramite l'interfaccia <i>drag & drop</i>	S
TA-17	Verificare che l'utente autenticato possa cercare il blocco di interesse nel menù laterale, a cui si accede tramite il bottone <i>Add a Block</i>	S
TA-18	Verificare che l'utente autenticato possa modificare le impostazioni di un singolo blocco	S
TA-19	Verificare che l'utente autenticato possa eliminare un blocco da interfaccia grafica	S
TA-20	Verificare che l'utente autenticato possa eliminare un blocco da tastiera, premendo il tasto <i>backspace</i>	S
TA-21	Verificare che l'utente autenticato possa salvare un <i>workflow</i>	S
TA-22	Verificare che l'utente autenticato possa rinominare un <i>workflow</i>	S
TA-23	Verificare che l'utente autenticato possa eseguire un <i>workflow</i> dalla pagina <i>dettagli flusso</i>	S
TA-24	Verificare che l'utente autenticato possa eseguire un <i>workflow</i> dalla <i>dashboard</i>	S
TA-25	Verificare che l'utente autenticato possa eliminare un <i>workflow</i>	S
TA-26	Verificare che l'utente autenticato possa rinominare un <i>workflow</i>	S
TA-27	Verificare che l'utente autenticato possa effettuare il logout	S
TA-28	Verificare che l'utente autenticato possa selezionare il tema che preferisce tra scuro e chiaro	S
TA-29	Verificare che siano stati implementati almeno 3 blocchi di automazione	S

Tabella 16: Test di accettazione

5. Cruscotto di Valutazione

La seguente sezione riporta un'analisi dell'andamento delle metriche di qualità del prodotto e del processo, in modo da monitorare il progresso del progetto e verificare se gli obiettivi di qualità sono stati raggiunti.

5.1. Estimated at Completion (MPC-EAC)

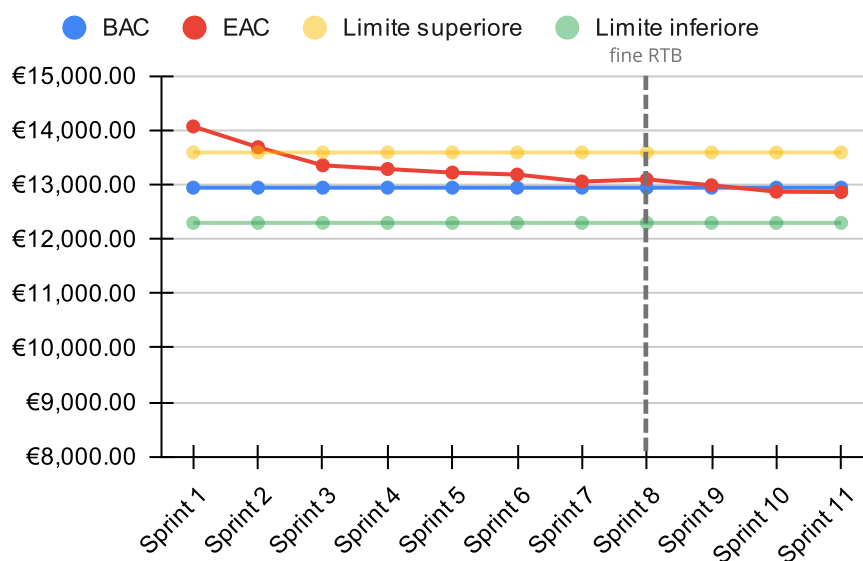


Figura 1: Stima del costo totale nel tempo

5.1.1. RTB

Dal grafico si osserva che il valore di **MPC-EAC** rimane sempre entro una variazione massima del **5% rispetto al BAC**, indicato come *limite superiore*, quindi il progetto è in linea con le aspettative di budget.

5.1.2. PB

Osservando il grafico emerge che il valore di **MPC-EAC** si è leggermente alzato in seguito al colloquio RTB, in concomitanza dell'ottavo sprint, a causa di alcune correzioni da effettuare che hanno portato ad un rallentamento. Tuttavia, si è riusciti a terminare portando il valore al di sotto del **BAC**.

5.2. Planned Value (MPC-PV) & Earned Value (MPC-EV)

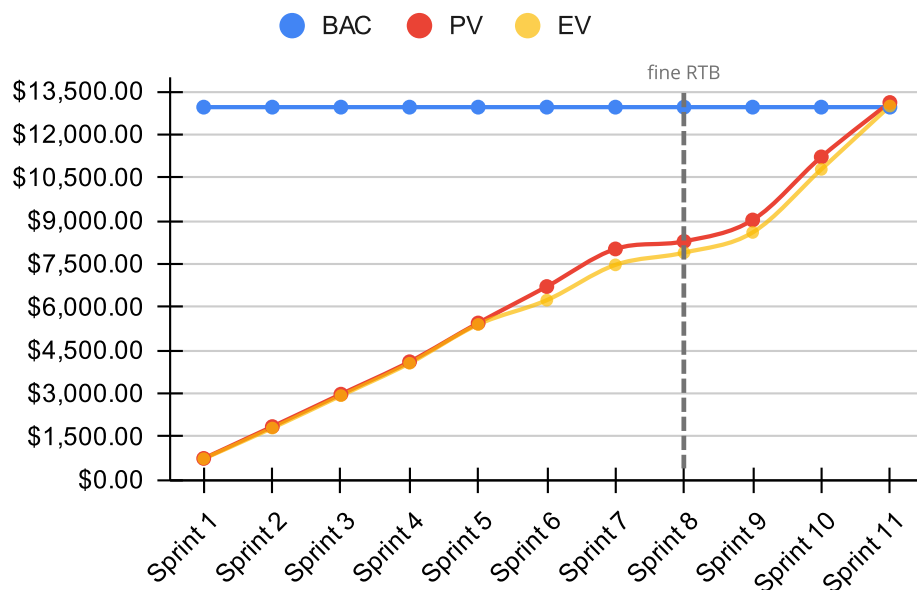


Figura 2: Stima dei valori EV e PV nel tempo

5.2.1. RTB

Dal grafico si può notare che il valore di **MPC-EV** è in linea con il valore **MPC-PV**, indicando che il lavoro viene effettivamente svolto rispettando la pianificazione.

5.2.2. PB

Il grafico mostra come il valore del **MPC-EV** abbia continuato a crescere in modo proporzionale al valore del **MPC-PV**, dimostrando che il gruppo è riuscito a pianificare correttamente le attività da svolgere.

I due valori si avvicinano sempre di più a quello di **BAC** indicando che il gruppo è rimasto coerente con il preventivo prodotto.

5.3. Actual Cost (MPC-AC), Estimated To Complete (MPC-ETC) & Estimated At Completion (MPC-EAC)

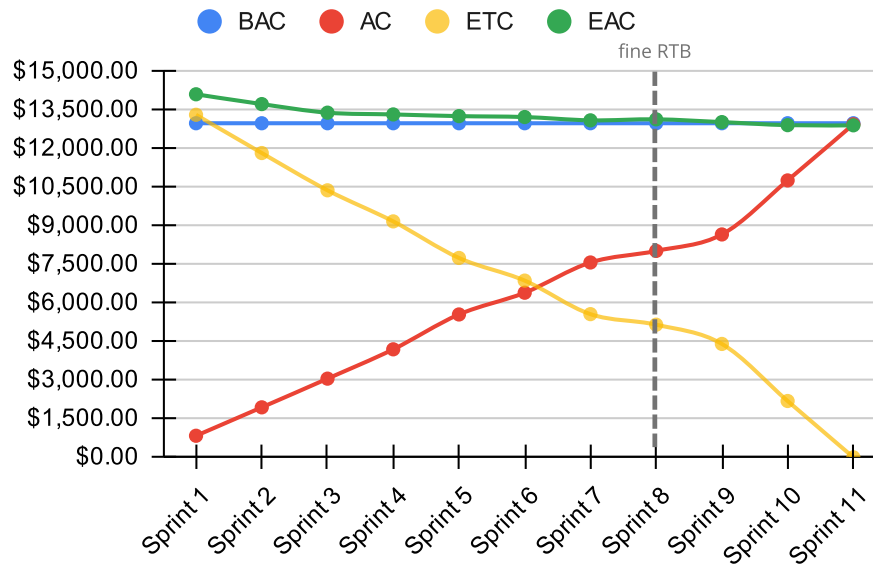


Figura 3: Stima dei valori AC, ETC e EAC nel tempo

5.3.1. RTB

Il grafico mostra l'andamento dei valori di **MPC-AC**, **MPC-ETC** e **MPC-EAC** nel tempo.

Si può osservare che il valore di **MPC-AC** cresce in modo costante, mentre il valore di **MPC-ETC** decresce progressivamente con l'avanzare degli sprint, entrambi i valori sono inferiori al valore di **MPC-EAC**.

Si può affermare quindi che il progetto è in linea con le aspettative di budget.

5.3.2. PB

Dal grafico si può notare come il valore di **MPC-AC** cresca in maniera inversamente proporzionale alla crescita di **MPC-ETC**.

Si osserva che nell'ultimo sprint i valori di **MPC-AC** e **MPC-EAC** si sovrappongono e quello di **MPC-ETC** diventa 0.

Si può quindi affermare che il progetto è stato completato rispettando il budget preventivato.

5.4. Cost Variance (MPC-CV) & Schedule Variance (MPC-SV)

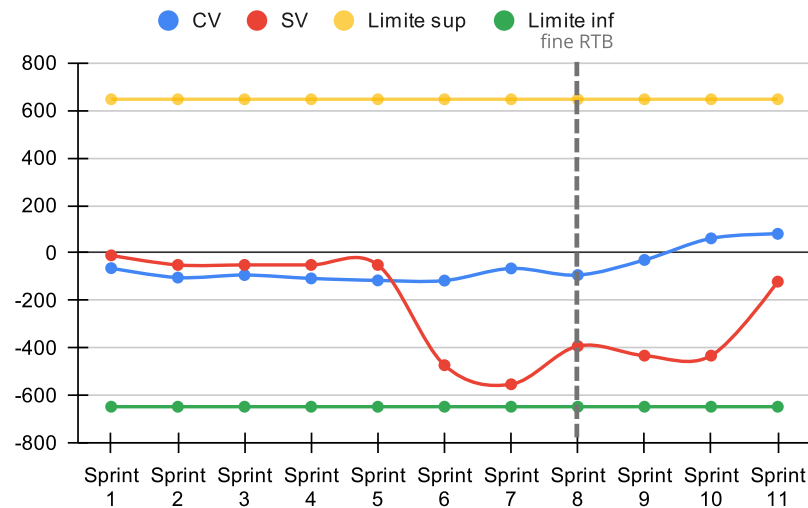


Figura 4: Stima dei valori CV e SV nel tempo

5.4.1. RTB

Dal grafico si può notare che i valore di **MPC-CV** e **MPC-SV** sono sempre entro i limiti accettabili, indicati come *limite superiore* e *limite inferiore*. Si può notare che il valore di **MPC-SV** ha subito un picco negativo durante il sesto e settimo sprint causato dalla concomitanza con la sessione di esami estiva che ha causato un rallentamento soprattutto negli sprint in questione. Tuttavia, il valore è tornato a crescere, indicando una ripresa del progetto e un ritorno alla pianificazione iniziale.

5.4.2. PB

Si può notare che i valori di **MPC-SV** e **MPC-CV** si sono stabilizzati subendo un piccolo picco verso la fine, indicando un recupero di eventuali ritardi.

5.5. Cost Performance Index (MPC-CPI)

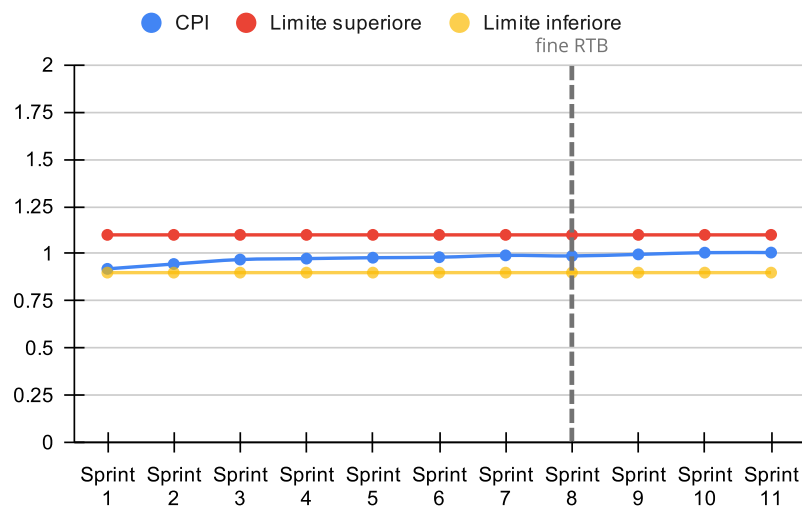


Figura 5: Stima dei valori CPI nel tempo

5.5.1. RTB

Il grafico mostra l'andamento del valore di **MPC-CPI** nel tempo, si può notare che è sempre superiore al limite inferiore e che si avvicina sempre di più al valore ottimo 1, indicando che i costi per completare i lavori sono in linea con quanto stabilito.

5.5.2. PB

Osservando l'andamento del valore di **MPC-CPI** nel tempo, si può notare che rimasto costante anche in fase di PB, indicando che i costi per completare i lavori sono in linea con quanto stabilito.

5.6. Requirements Stability Index (MPC-RSI)

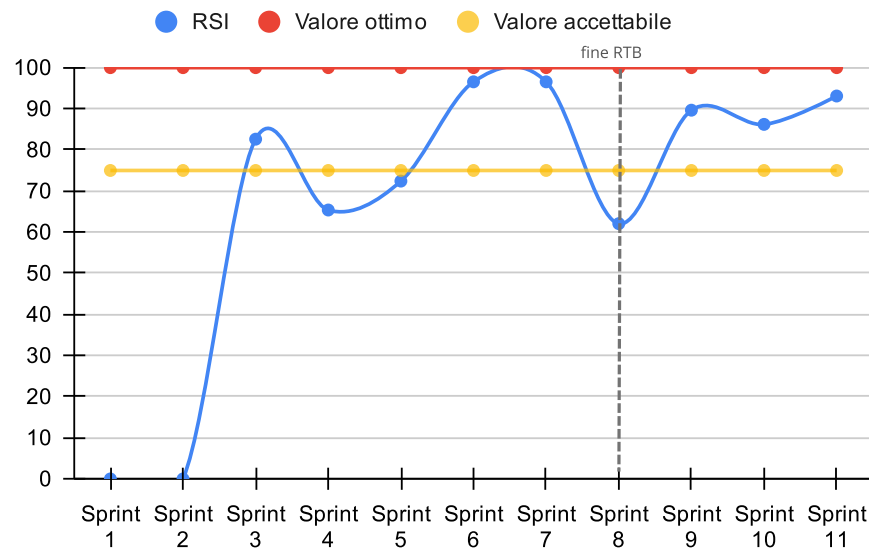


Figura 6: Stima dei valori RSI nel tempo

5.6.1. RTB

Dal grafico si può notare che nei due sprint iniziali il valore di **MPC-RSI** è pari a 0, dato dal fatto che i requisiti sono stati definiti solo a partire dal terzo sprint. Con l'avanzare del progetto, il valore cresce fino a raggiungere stabilità e valori accettabili, indicando che i requisiti sono stati stabiliti e non sono soggetti a modifiche frequenti.

Si può osservare una diminuzione del valore nell'ottavo sprint, a seguito della revisione con il professor Cardin, il quale ha richiesto un maggior dettaglio nella definizione dei requisiti.

5.6.2. PB

Si può notare che il valore di **MPC-RSI** ha subito delle variazioni a causa di alcune modifiche ai requisiti emersi dopo un'analisi più approfondita in vista dello sviluppo del MVP, tuttavia il valore rimane sempre entro i limiti accettabili.

5.7. Correttezza ortografica (MPC-CO)

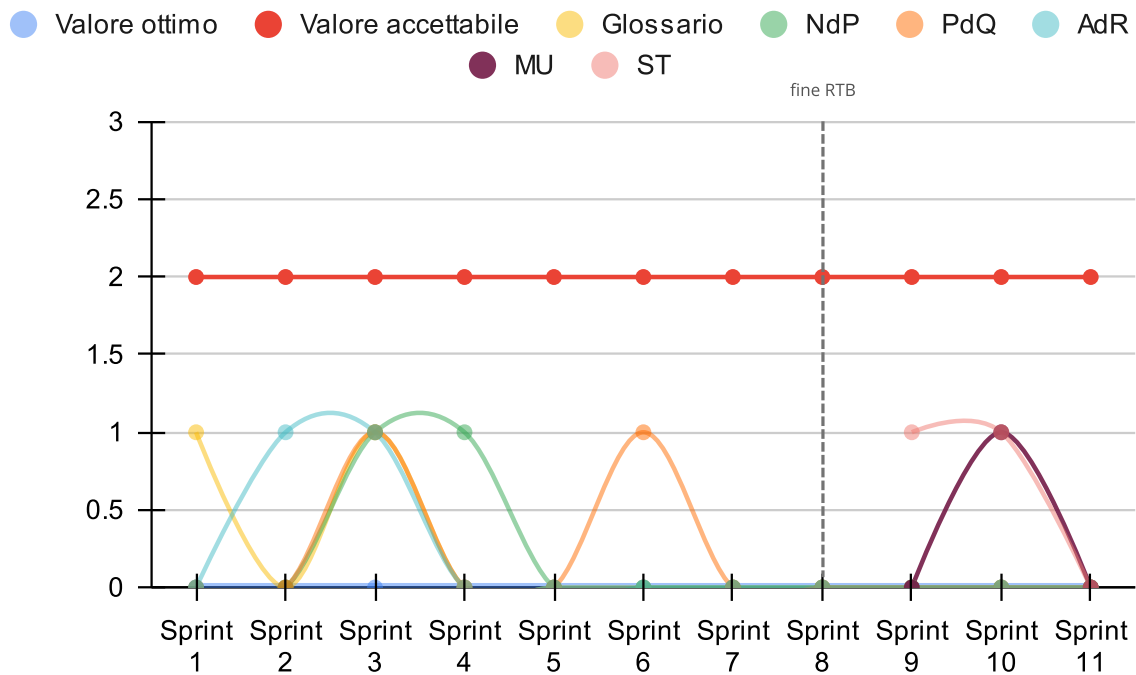


Figura 7: Stima dei valori CO nel tempo

5.7.1. RTB

Si può notare che il gruppo si è sempre impegnato a correggere gli eventuali errori ortografici presenti inizialmente nei documenti redatti, per raggiungere un valore pari a 0.

5.7.2. PB

Si osserva che il gruppo ha mantenuto alta l'attenzione alla correttezza ortografica per raggiungere un valore pari a 0 anche nei documenti redatti in fase di PB.

5.8. Satisfaction of Quality Metrics (MPC-SQM)

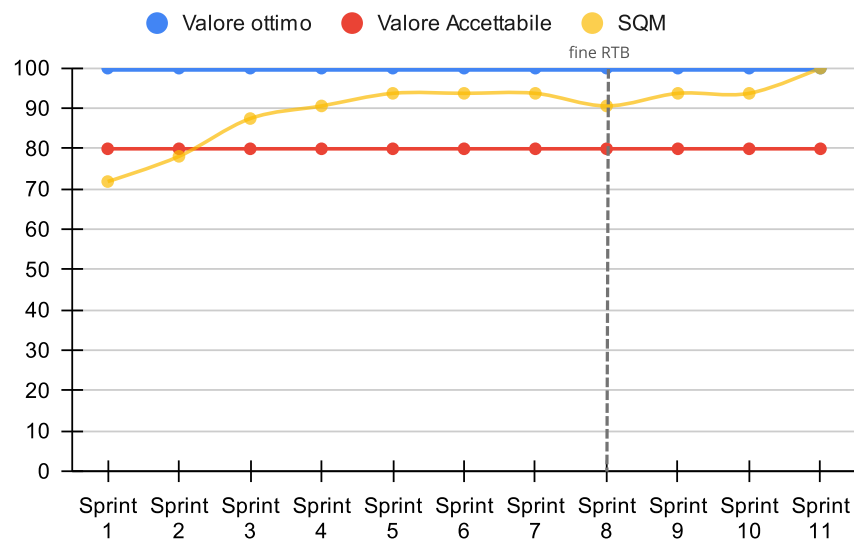


Figura 8: Stima dei valori SQM nel tempo

5.8.1. RTB

Dal grafico emerge che la percentuale di metriche soddisfatte nei primi due sprint è inferiore al *limite inferiore*, ciò è dato dall'iniziale inesperienza del gruppo. Tuttavia, la percentuale cresce progressivamente con l'avanzare degli sprint raggiungendo valori accettabili.

5.8.2. PB

Osservando il grafico si può notare che la percentuale di metriche soddisfatte cresce costantemente, terminando con un valore pari al massimo, indicando che il gruppo è riuscito a mantenere un buon livello di qualità del prodotto in fase di PB.

5.9. Efficienza temporale (MPC-ET)

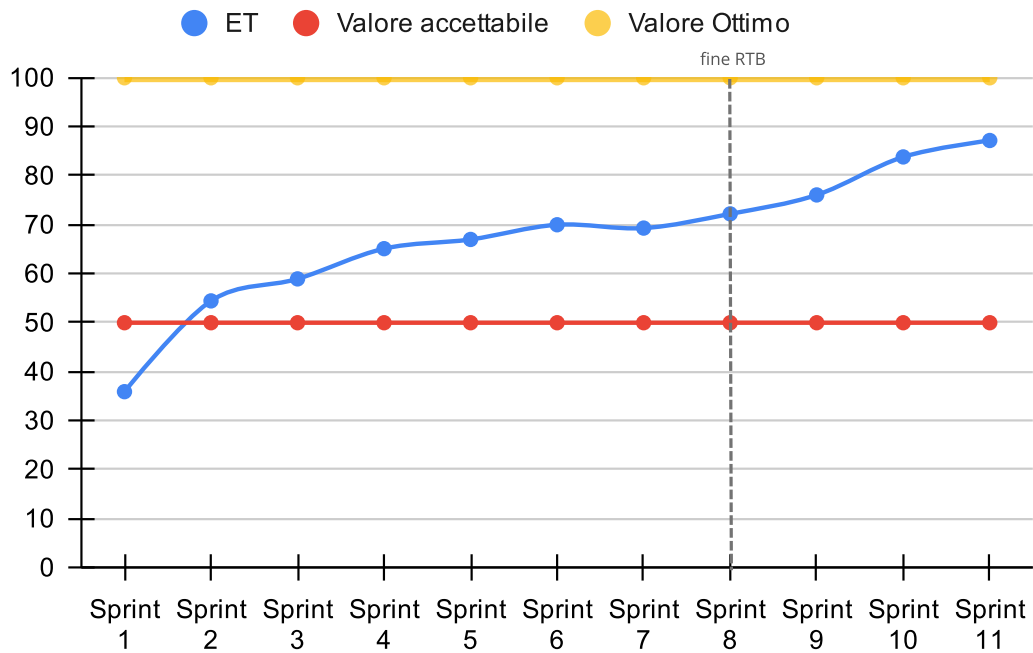


Figura 9: Stima dei valori ET nel tempo

5.9.1. RTB

Il grafico mostra l'andamento dell'efficienza temporale del gruppo, che cresce progressivamente con l'avanzare degli sprint. Nei primi sprint si ha un'efficienza temporale bassa, dovuta all'inesperienza del gruppo e alla mancanza di conoscenze sulle tecnologie utilizzate.

5.9.2. PB

Dal grafico emerge che in fase di PB il gruppo è riuscito a mantenere un'efficienza temporale elevata, dimostrando una buona gestione del tempo, grazie all'esperienza acquisita durante la fase di RTB.

5.10. Requisiti obbligatori soddisfatti (MPD-RO)

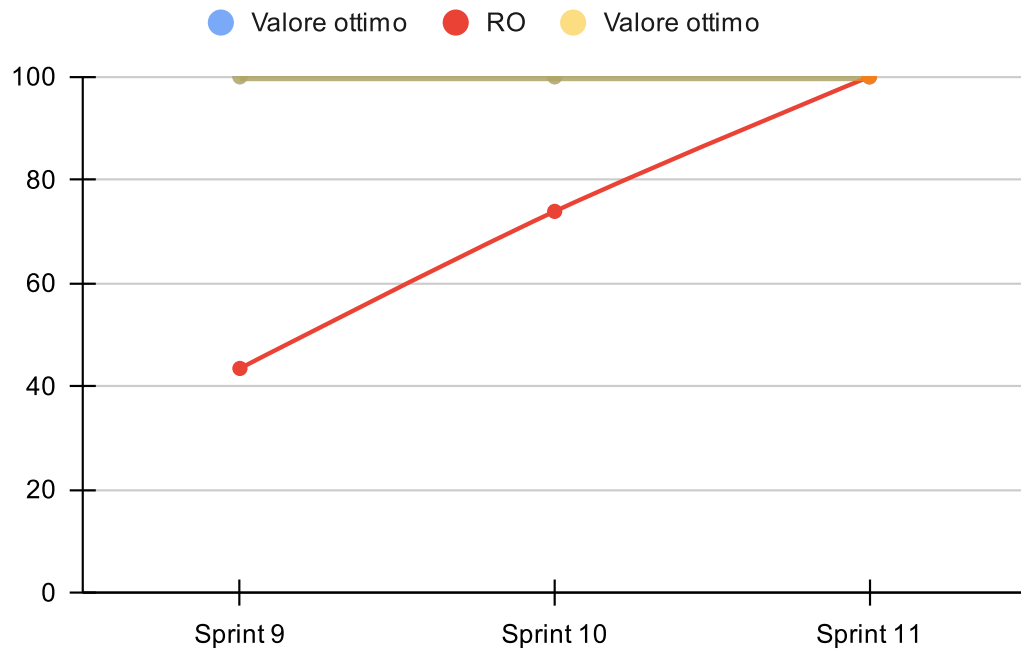


Figura 10: Stima dei valori RO nel tempo

5.10.1. PB

Dal grafico si può notare che il gruppo è partito da una buona base del PoC.

Il gruppo si è impegnato a soddisfare la maggior parte dei requisiti obbligatori entro lo sprint 10, fino al soddisfacimento di tutti i requisiti obbligatori entro la fine del progetto.

5.11. Requisiti desiderabili e facoltativi soddisfatti (MPD-RD e MPD-RF)

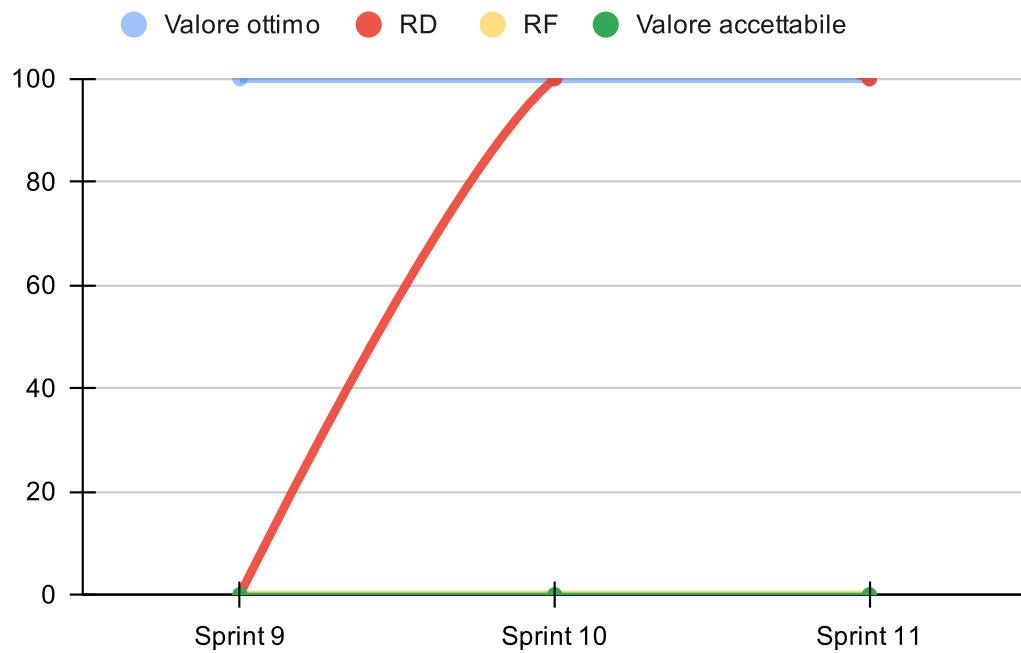


Figura 11: Stima dei valori RD e RF nel tempo

5.11.1. PB

Dal grafico si può osservare che il gruppo si è impegnato fin dall'inizio a soddisfare il requisito desiderabile individuato, ovvero la possibilità di cambiare tema, raggiungendo così una maggiore soddisfazione dell'utente.

5.12. Code Coverage (MPD-CCO)

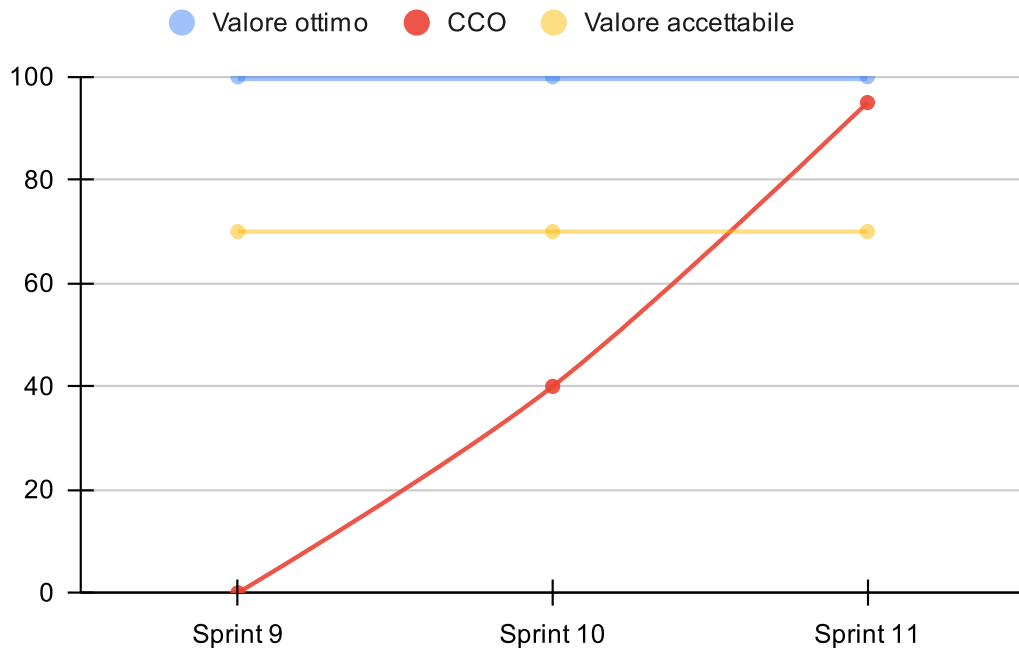


Figura 12: Stima dei valori CCO nel tempo

5.12.1. PB

Dal grafico si può notare che il gruppo ha iniziato a lavorare sulla copertura del codice appena terminata la progettazione, passo passo con la stesura del codice. Questo approccio ha permesso di ottenere una copertura del codice superiore a quella richiesta del 70%.

5.12.2. Complessità ciclomatica (MPD-CC)

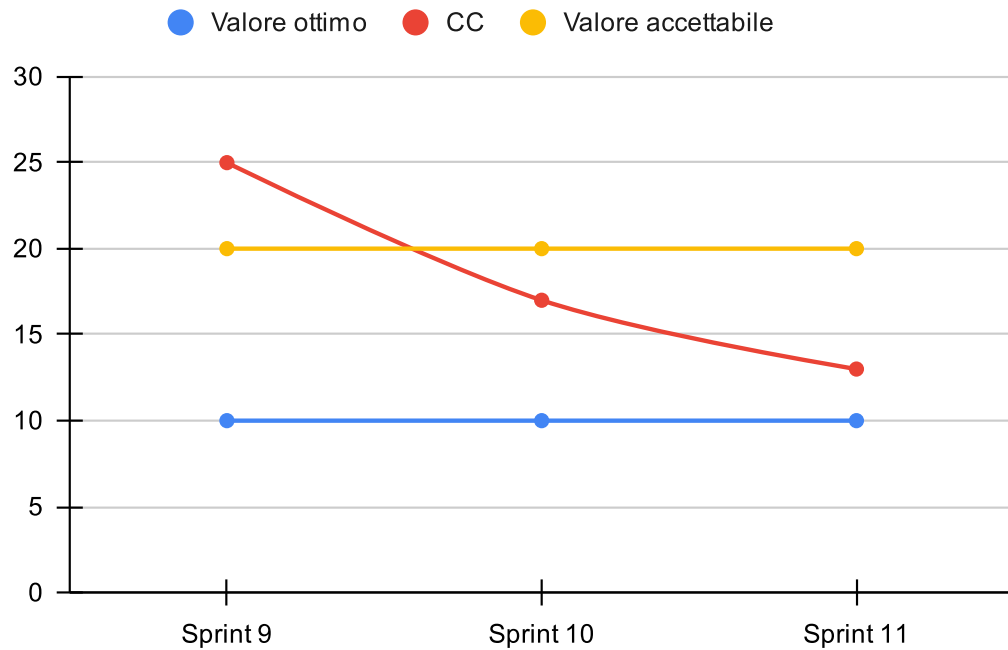


Figura 13: Stima dei valori CC nel tempo

Dal grafico si può osservare che il valore **MPD-CC** è inizialmente elevato ma ha subito una riduzione costante nel tempo, grazie ad una progettazione migliore e a pratiche di refactoring. Indica che il codice è facilmente leggibile e ben strutturato, portando a una maggiore manutenibilità e testabilità.

5.13. Test superati in percentuale (MPC-TSP)

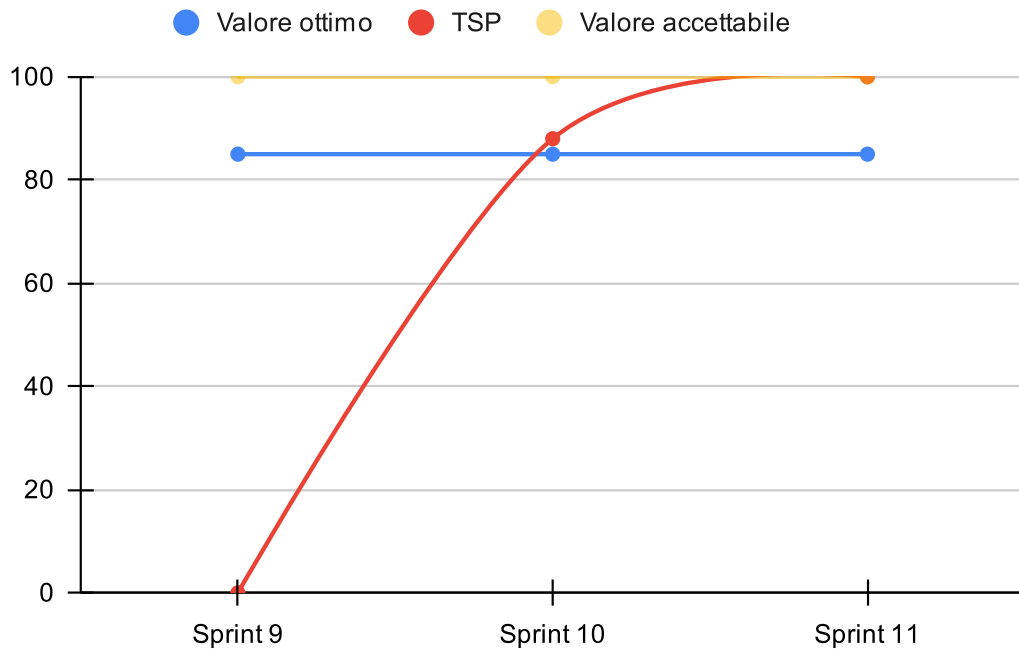


Figura 14: Stima dei valori TSP nel tempo

5.13.1. PB

Dal grafico si può notare che il valore dei test superati è sempre stato elevato. Questo dimostra che i test sviluppati sono sempre stati superati.

Si è consapevoli che una quantità elevata di test superati non sia garanzia di un software privo di errori o lacune, tuttavia è pur sempre indice di qualità del codice.