

Piano di Qualifica

2025-03-20

Responsabile	Nicolò Bovo
	Alessandro Di Pasquale
	Giovanni Battista Matteazzi
	Manuel Cinnirella
	Elia Leonetti
Redattori	Nicolò Bovo, Giovanni Battista Matteazzi, Manuel Cinnirella, Alessandro Di Pasquale, Massimo Chioru, Elia Leonetti
Verificatori	Manuel Cinnirella
	Romeo Calearo
	Elia Leonetti
	Alessandro Di Pasquale

Registro delle modifiche

Vers.	Data	Descrizione	Autore	Verificatore
2.0.0	2025-08-13	Aggiornamento con rilevazioni PB e correzioni finali	Giovanni Battista Matteazzi	Alessandro Di Pasquale
1.2.0	2025-08-09	Aggiornate Specifiche dei test	Nicolò Bovo, Massimo Chioru	Romeo Calero
1.1.0	2025-07-24	Correzioni migliorative	Giovanni Battista Matteazzi, Manuel Cinnirella	Elia Leonetti
1.0.0	2025-07-11	Selezione termini glossario e correzioni finali	Alessandro Di Pasquale, Nicolò Bovo	Romeo Calero
0.3.0	2025-07-09	Aggiunta Cruscotto di Qualità e processi di auto miglioramento	Giovanni Battista Matteazzi, Elia Leonetti, Nicolò Bovo	Romeo Calero
0.2.0	2025-03-29	Aggiunta Qualità di Prodotto e Specifiche dei Test	Alessandro Di Pasquale, Massimo Chioru, Elia Leonetti	Manuel Cinnirella
0.1.0	2025-03-20	Introduzione del documento e Qualità di Processo	Giovanni Battista Matteazzi, Nicolò Bovo, Manuel Cinnirella	Elia Leonetti

Indice

I. Introduzione	5
I - 1. Scopo del documento	5
I - 2. Scopo del prodotto	5
I - 3. Riferimenti	5
I - 3.1. Riferimenti informativi	5
I - 3.2. Riferimenti normativi	5
II. Qualità di processo	6
II - 1. Fornitura	6
II - 2. Sviluppo	7
II - 3. Documentazione	7
II - 4. Verifica	7
II - 5. Gestione della qualità	7
III. Qualità di prodotto	9
III - 1. Funzionalità	9
III - 2. Affidabilità	9
III - 3. Usabilità	10
III - 4. Efficienza	10
III - 5. Manutenibilità	10
IV. Specifiche dei test	12
IV - 1. Convenzioni di nomenclatura	12
IV - 2. Test di unità	12
IV - 3. Test di integrazione	16
IV - 4. Test di sistema	17
IV - 5. Test di accettazione	21
V. Cruscotto valutazione della qualità	22
V - 1. Estimated At Completion (MPC-EAC)	22
V - 2. Earned Value (MPC-EV) e Planned Value (MPC-PV)	23
V - 3. Actual Cost (MPC-AC) e Estimate To Completion (MPC-ETC)	24
V - 4. Budget Variance (MPC-BV) e Schedule Variance (MPC-SV)	25
V - 5. Indice di Stabilità dei Requisiti (MPC-RSI)	26
V - 6. Indice Gulpease (MPC-GP)	27
V - 7. Correttezza Ortografica (MPC-CO)	28
V - 8. Percentuale di Metriche Soddisfatte (MPC-PMS)	29
V - 9. Efficienza Temporale (MPC-ET)	30
V - 10. Copertura Codice (MPC-CC)	31
V - 11. Passed test cases percentage (MDP-PTCP)	31
V - 12. Complessità Ciclomantica Media (MDP-CCM)	32
V - 13. Code Smell (MDP-CS)	32
VI. Processi di automiglioramento	33
VI - 1. Valutazione sull'organizzazione	33
VI - 2. Valutazione sui ruoli	33

VI - 3. Valutazione sugli strumenti	34
---	----

I. Introduzione

I - 1. Scopo del documento

Il Piano di Qualifica è un documento essenziale per garantire il controllo della qualità del prodotto software e dei processi impiegati nel suo sviluppo. Definisce le strategie di verifica e validazione adottate dal gruppo, con l'obiettivo di assicurare che il progetto rispetti i requisiti, raggiunga gli obiettivi stabiliti e mantenga elevati standard qualitativi. Il documento si articola in tre componenti principali:

1. Pianificazione della qualità: stabilisce obiettivi, standard e strategie;
2. Controllo della qualità: applica metriche e verifiche per valutare la conformità;
3. Miglioramento continuo: analizza i risultati per ottimizzare metodi e processi.

Essendo un documento in continua evoluzione, verrà aggiornato nel tempo per riflettere miglioramenti o modifiche organizzative. Le sezioni di monitoraggio documenteranno l'andamento delle metriche, consentendo interventi tempestivi in caso di criticità.

I - 2. Scopo del prodotto

Lo scopo di NearYou_G è progettare, realizzare e validare una piattaforma di advertising personalizzato in tempo reale, capace di unire flussi GPS_G, profilazione utente e AI generativa_G per produrre annunci altamente contestuali.

- **Implementazione tecnica:** Il team AlphaCode_G svilupperà simulatori di posizionamento, pipeline_G Kafka_G → ClickHouse_G → PostGIS_G e modelli LLM_G per garantire un flusso dati integrato e performante.
- **Misurazione e validazione:** Verranno condotte misurazioni mirate per valutare throughput_G e latenza_G delle componenti di streaming_G (Kafka) e orchestrazione_G (Airflow_G), l'efficacia e i tempi di risposta delle query geospaziali_G in PostGIS, e la velocità di generazione e consegna degli annunci via FastAPI_G. Analizzeremo inoltre l'impatto delle operazioni AI su CPU e memoria.
- **Test di resilienza e performance:** Il team testerà la resilienza dei microservizi_G (producer_G, consumer_G, webapp_G) in caso di picchi di traffico o failure_G di rete. Per confermare scalabilità_G, robustezza delle logiche di prossimità e manutenibilità del codice, AlphaCode eseguirà stress test_G ad alto carico su tutte le pipeline, verificando la correttezza funzionale e le performance sotto condizioni estreme.

I - 3. Riferimenti

I - 3.1. Riferimenti informativi

- Norme di Progetto (v2.0.0)
- Capitolato d'appalto C4: NearYou: sistema di advertising
- Standard ISO/IEC 12207:1995_G

I - 3.2. Riferimenti normativi

- I processi di ciclo di vita del software
- Glossario (v2.0.0)

II. Qualità di processo

La qualità di processo è nota essere un fattore di fondamentale importanza per qualsiasi produzione di software che punti all'eccellenza qualitativa. Essa, infatti, influenza con un evidente rapporto di causa effetto la qualità del prodotto finale. Di seguito elenchiamo gli obiettivi di qualità che il gruppo si prefigge di raggiungere nell'ambito della qualità di processo, suddivisi per le tre categorie di processi individuate dallo standard ISO/IEC 12207:1995 (primari, di supporto e organizzativi).

II - 1. Fornitura

Insieme delle azioni e responsabilità del fornitore, con l'obiettivo di definire formalmente gli accordi con il committente in merito a vincoli e requisiti del progetto.

1. **Planned value (MPC-PV):** Valore stimato del lavoro da eseguire entro una data prefissata, secondo la pianificazione.
2. **Earned value (MPC-EV):** Quantifica il valore del lavoro realmente completato fino a un certo momento, in relazione al budget pianificato.
3. **Actual cost (MPC-AC):** Spesa reale sostenuta per le attività concluse fino alla data considerata.
4. **Cost performance index (MPC-CPI):** Indice che esprime l'efficienza della spesa, indicando quanto valore si è ottenuto per ogni unità di costo impiegata.
5. **Estimated at completion (MPC-EAC):** Proiezione del costo finale totale del progetto basata sull'andamento attuale.
6. **Schedule variance (MPC-SV):** Differenza tra il valore ottenuto (EV) e quello previsto (PV); un risultato negativo segnala un ritardo.
7. **Budget variance (MPC-BV):** Scostamento tra il valore previsto (PV) e la spesa effettiva (AC); valori negativi indicano costi superiori al preventivo.
8. **Estimate to completion (MPC-ETC):** Previsione della spesa necessaria per terminare le attività residue del progetto.

Metrica	Nome	Valore accettabile	Valore ottimo
MPC-EV	Earned Value	≥ 0	$\leq EAC$
MPC-PV	Planned Value	≥ 0	$\leq BAC$
MPC-AC	Actual Cost	≥ 0	$\leq EAC$
MPC-EAC	Estimate At Completion	≥ 0	$\leq BAC$
MPC-ETC	Estimate To Complete	≥ 0	$\leq BAC$

II - 2. Sviluppo

Insieme delle operazioni tecniche volte alla costruzione e manutenzione del sistema software, con l'obiettivo di rispettare i requisiti contrattuali.

1. **Indice di stabilità dei requisiti (MPC-RSI):** Misura quanto i requisiti siano rimasti costanti nel tempo, segnalando modifiche o instabilità progettuali.

Metrica	Nome	Valore accettabile	Valore ottimo
MPC-RSI	Indice di stabilità dei requisiti	$\geq 80\%$	100%

II - 3. Documentazione

Processo necessario per il tracciamento di tutte le attività relative al progetto.

1. **Indice Gulpease_G (MPC-GP):** Misura la facilità di lettura di un testo italiano in base alla lunghezza media delle parole e delle frasi. Il punteggio va da 0 (poco leggibile) a 100 (molto leggibile).
2. **Correttezza Ortografica (MPC-CO):** Valuta la presenza di errori ortografici e grammaticali nei testi prodotti.

Metrica	Nome	Valore accettabile	Valore ottimo
MPC-GP	Indice di Gulpease	≥ 40	≥ 80
MPC-CO	Correttezza ortografica	0	0

II - 4. Verifica

1. **Code Coverage_G (MPC-CC):** Misura la percentuale di codice eseguita durante i test. Valori più alti indicano una migliore copertura del codice.

Metrica	Nome	Valore accettabile	Valore ottimo
MPC-CC	Code coverage	$\geq 80\%$	100%

II - 5. Gestione della qualità

1. **Percentuale di metriche soddisfatte (MPC-PMS):**

Metrica	Nome	Valore accettabile	Valore ottimo
MPC-PMS	Percentuale Metriche Soddisfatte	$\geq 80\%$	100%

1. **Efficienza temporale (MPC-ET):**

Metrica	Nome	Valore accettabile	Valore ottimo
MPC-ET	Efficienza temporale	$\geq 55\%$	$\geq 75\%$

III. Qualità di prodotto

La qualità del prodotto rappresenta un aspetto fondamentale nello sviluppo software, poiché misura quanto il sistema realizzato sia in grado di soddisfare i requisiti stabiliti, le esigenze esplicite e implicite degli stakeholder e gli standard di riferimento. Essa è strettamente correlata alla qualità dei processi impiegati, poiché un prodotto di valore nasce da pratiche progettuali e tecniche adeguate e controllate lungo tutto il ciclo di vita. Un prodotto software si considera di elevata qualità quando dimostra le seguenti caratteristiche:

1. **Funzionalità:** risponde pienamente ai requisiti obbligatori, desiderabili e opzionali delineati nell'Analisi dei Requisiti, offrendo le funzionalità attese senza deviazioni.
2. **Affidabilità:** è stabile nel tempo e resistente a guasti, garantendo un comportamento prevedibile anche in condizioni anomale.
3. **Usabilità:** consente all'utente finale un'interazione chiara, semplice ed efficace, riducendo il tempo di apprendimento e minimizzando gli errori d'uso.
4. **Efficienza:** utilizza in modo ottimale le risorse disponibili, rispondendo rapidamente alle operazioni richieste, anche sotto carico.
5. **Manutenibilità:** è progettato con una struttura che facilita interventi futuri, come aggiornamenti, correzioni o estensioni, mantenendo la coerenza e la stabilità del sistema.

III - 1. Funzionalità

La funzionalità esprime quanto il software sia in grado di rispondere alle esigenze espresse nei requisiti, distinguendo tra quelli essenziali, preferibili e accessori.

1. **Requisiti desiderabili soddisfatti (MDP-RDS):** Misura quanto il software integra i requisiti considerati apprezzabili. Un valore elevato di questo indicatore contribuisce ad aumentare il livello di soddisfazione degli utenti.
2. **Requisiti opzionali soddisfatti (MDP-ROS):** Esprime la percentuale di requisiti opzionali effettivamente implementati. Una maggiore presenza di tali funzionalità può arricchire il prodotto e aumentarne il valore complessivo.

Metrica	Nome	Valore accettabile	Valore ottimo
MDP-RDS	Requisiti desiderabili soddisfatti	0%	100%
MDP-ROS	Requisiti opzionali soddisfatti	0%	100%

III - 2. Affidabilità

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

1. **Passed test cases percentage (MDP-PTCP):** Misura la percentuale di casi di test superati rispetto al totale dei test eseguiti. Un alto valore indica che il software soddisfa i requisiti funzionali e non funzionali previsti.
2. **Failure density_G (MDP-FD):** Indica il numero di fallimenti correttamente riscontrati per unità di dimensione del software, spesso misurata in linee di codice (LOC_G) o punti funzione_G. Valori più bassi denotano un software di qualità superiore, con meno difetti rispetto alla sua complessità o dimensione

Metrica	Nome	Valore accettabile	Valore ottimo
MDP-PTCP	Passed test cases percentage	$\geq 80\%$	100%
MDP-FD	Failure Density	$\leq 5\%$	0%

III - 3. Usabilità

1. **Facilità di utilizzo (MDP-FU):** Misura il numero di errori commessi dagli utenti durante l'interazione. Un valore minimo indica un'interfaccia intuitiva.
2. **Tempo di apprendimento (MDP-FA):** Valuta il tempo necessario a un utente per imparare a utilizzare il software. Tempi più brevi migliorano l'esperienza utente.

Metrica	Nome	Valore accettabile	Valore ottimo
MDP-FU	Facilità di utilizzo	≤ 2 errori	0 errori
MDP-FA	Tempo di apprendimento	≤ 5 minuti	≤ 1 minuto

III - 4. Efficienza

1. **Utilizzo delle risorse (MDP-UR):** Valuta quanto efficacemente il sistema sfrutta le risorse hardware_G, come la CPU, la memoria e altri componenti. Un impiego ottimale di queste risorse assicura il corretto funzionamento del sistema, evitando un utilizzo eccessivo o inefficiente delle capacità disponibili.

Metrica	Nome	Valore accettabile	Valore ottimo
MDP-UR	Utilizzo delle risorse	$\geq 80\%$	100%

III - 5. Manutenibilità

1. **Code smell_G (MDP-CS):** Rileva potenziali problemi di progettazione o codice che potrebbero richiedere manutenzione. Segnala parti del codice che potrebbero non essere ottimali e che potrebbero causare difficoltà nel futuro, come un'architettura poco chiara o sezioni di codice ripetitive.
2. **Complessità Ciclomantica Media (MDP-CCM)** La *Cyclomatic complexity_G* valuta la complessità del codice sorgente attraverso la misurazione del numero di cammini

indipendenti attraverso il grafo di controllo del flusso. Una complessità ciclomatica più alta indica che il codice è più difficile da comprendere e mantenere.

Metrica	Nome	Valore accettabile	Valore ottimo
MDP-CS	Code smell	0	0
MDP-CCM	Complessità Ciclomantica Media	≤ 5	≤ 3

IV. Specifiche dei test

In questa sezione vengono illustrate le attività di verifica svolte per accertare che tutti i vincoli definiti nei requisiti siano rispettati. Come previsto nel documento Norme di Progetto, il piano di test segue una metodologia organizzata, suddividendosi in quattro principali categorie: test di unità, test di integrazione, test di sistema e test di accettazione.

IV - 1. Convenzioni di nomenclatura

Ogni test è identificato in modo univoco attraverso una sigla che segue la seguente struttura:

T[tipologia]-[numero]

dove «tipologia» può assumere i seguenti valori:

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

Il campo «numero» rappresenta un identificativo progressivo all'interno della categoria di appartenenza. A ciascun test è inoltre associato uno stato che ne rappresenta l'esito:

- **Verificato**: il test è stato eseguito con successo
- **Non verificato**: il test non è ancora stato eseguito
- **Non implementato**: il test non è stato realizzato

IV - 2. Test di unità

Hanno l'obiettivo di controllare il corretto comportamento dei componenti software più piccoli e indipendenti, realizzati prevalentemente durante la fase progettuale.

Codice Test	Descrizione	Status
TU-01	Test inizializzazione Redis Cache con connessione	Verificato
TU-02	Test inizializzazione Redis Cache con password	Verificato
TU-03	Test gestione errori di connessione Redis	Verificato
TU-04	Test operazioni di cache (set/get) per valori semplici	Verificato
TU-05	Test gestione valori JSON complessi in cache	Verificato
TU-06	Test gestione chiavi non esistenti	Verificato
TU-07	Test gestione JSON malformato in cache	Verificato
TU-08	Test operazioni delete ed exists per cache	Verificato

TU-09	Test informazioni Redis server e statistiche	Verificato
TU-10	Test operazioni cache senza client connesso	Verificato
TU-11	Test gestione errori nelle operazioni Redis	Verificato
TU-12	Test gestione TTL di default per Redis	Verificato
TU-13	Test inizializzazione Memory Cache	Verificato
TU-14	Test operazioni set/get Memory Cache semplici	Verificato
TU-15	Test operazioni set/get Memory Cache complesse	Verificato
TU-16	Test scadenza TTL in Memory Cache	Verificato
TU-17	Test gestione dimensione massima Memory Cache	Verificato
TU-18	Test operazioni delete ed exists Memory Cache	Verificato
TU-19	Test pulizia Memory Cache	Verificato
TU-20	Test informazioni Memory Cache e statistiche	Verificato
TU-21	Test pulizia entries scadute Memory Cache	Verificato
TU-22	Test aggiornamento tempo accesso Memory Cache	Verificato
TU-23	Test aggiornamento chiavi esistenti Memory Cache	Verificato
TU-24	Test compatibilità interfaccia cache implementations	Verificato
TU-25	Test pattern di fallback tra Redis e Memory cache	Verificato
TU-26	Test creazione modello Offer con valori default	Verificato
TU-27	Test validazione Offer valida con tutti i constraint	Verificato
TU-28	Test validazione Offer con discount invalido	Verificato
TU-29	Test validazione Offer con date invalide	Verificato
TU-30	Test validazione Offer con range età invalido	Verificato
TU-31	Test utilizzo validator personalizzato per Offer	Verificato
TU-32	Test conversione Offer to dictionary	Verificato
TU-33	Test creazione Offer from dictionary	Verificato

TU-34	Test validazione Offer per utente con constraint età	Verificato
TU-35	Test validazione Offer per utente con constraint interessi	Verificato
TU-36	Test validazione Offer per utente con constraint date	Verificato
TU-37	Test validazione Offer per utente con constraint usage	Verificato
TU-38	Test generazione testo display per diversi tipi Offer	Verificato
TU-39	Test Builder Pattern - costruzione base Offer	Verificato
TU-40	Test Builder Pattern - interfaccia fluente	Verificato
TU-41	Test Builder Pattern - impostazione periodo validità	Verificato
TU-42	Test Builder Pattern - validità per numero giorni	Verificato
TU-43	Test Builder Pattern - reset dopo build	Verificato
TU-44	Test Builder Pattern - validazione al build	Verificato
TU-45	Test Builder Pattern - build unsafe senza validazione	Verificato
TU-46	Test Builder Pattern - reset manuale	Verificato
TU-47	Test Factory Pattern - creazione flash offer	Verificato
TU-48	Test Factory Pattern - creazione student offer	Verificato
TU-49	Test Factory Pattern - creazione senior offer	Verificato
TU-50	Test Factory Pattern - creazione category offer	Verificato
TU-51	Test Factory Pattern - categoria sconosciuta fallback	Verificato
TU-52	Test creazione modello UserVisit	Verificato
TU-53	Test calcoli post-inizializzazione UserVisit	Verificato
TU-54	Test conversione UserVisit to dictionary	Verificato
TU-55	Test UserVisit con dati temporali parziali	Verificato
TU-56	Test validator di default con Offer valida	Verificato
TU-57	Test validator di default con discount invalido	Verificato
TU-58	Test validator di default con range date invalido	Verificato

TU-59	Test validator di default con range età invalido	Verificato
TU-60	Test protocollo validator personalizzato	Verificato
TU-61	Test Factory - creazione standard strategy	Verificato
TU-62	Test Factory - creazione aggressive strategy	Verificato
TU-63	Test Factory - creazione conservative strategy	Verificato
TU-64	Test Factory - strategy sconosciuta fallback	Verificato
TU-65	Test Factory - creazione default strategy	Verificato
TU-66	Test Standard Strategy - generazione con alta probabilità	Verificato
TU-67	Test Standard Strategy - generazione con bassa probabilità	Verificato
TU-68	Test Standard Strategy - nessuna generazione quando non dovuta	Verificato
TU-69	Test Standard Strategy - numero corretto offerte generate	Verificato
TU-70	Test Standard Strategy - offerta con config categoria	Verificato
TU-71	Test Standard Strategy - targeting per età	Verificato
TU-72	Test Standard Strategy - targeting per interessi	Verificato
TU-73	Test Aggressive Strategy - generazione sempre attiva	Verificato
TU-74	Test Aggressive Strategy - generazione più offerte	Verificato
TU-75	Test Aggressive Strategy - discount migliorato	Verificato
TU-76	Test Aggressive Strategy - cap massimo discount	Verificato
TU-77	Test Conservative Strategy - probabilità ridotta	Verificato
TU-78	Test Conservative Strategy - meno offerte generate	Verificato
TU-79	Test Conservative Strategy - discount più basso	Verificato
TU-80	Test Conservative Strategy - discount minimo	Verificato
TU-81	Test OffersService - inizializzazione servizio	Verificato

TU-82	Test OffersService - impostazione strategy runtime	Verificato
TU-83	Test OffersService - ottenimento connessione database	Verificato
TU-84	Test OffersService - generazione offerte per tutti gli shop	Verificato
TU-85	Test OffersService - inserimento offerte nel database	Verificato
TU-86	Test OffersService - gestione errori inserimento	Verificato
TU-87	Test OffersService - recupero offerte attive per shop	Verificato
TU-88	Test OffersService - pulizia offerte scadute	Verificato
TU-89	Test OffersService - inserimento lista vuota	Verificato
TU-90	Test HTTP client lazy initialization	Verificato
TU-91	Test costruzione dataflow structure	Verificato

IV - 3. Test di integrazione

Seguono i test di unità e mirano a controllare la comunicazione tra differenti moduli software per assicurare che operino congiuntamente per funzionalità specifiche.

Codice Test	Descrizione	Status
TI-01	Test disconnessione utente WebSocket	Verificato
TI-02	Test verifica token JWT valido	Verificato
TI-03	Test verifica token JWT invalido	Verificato
TI-04	Test verifica token JWT con chiave sbagliata	Verificato
TI-05	Test token endpoint con credenziali valide	Verificato
TI-06	Test user dashboard endpoint	Verificato
TI-07	Test get user profile non autenticato	Verificato
TI-08	Test get user notifications con paginazione	Verificato
TI-09	Test cache posizione in Connection Manager	Verificato
TI-10	Test pattern Singleton per DatabaseConnections	Verificato
TI-11	Test inizializzazione pattern Observer	Verificato

TI-12	Test inizializzazione lazy client ClickHouse	Verificato
TI-13	Test caching messaggi	Verificato
TI-14	Test notifiche observer	Verificato
TI-15	Test raccolta metriche	Verificato
TI-16	Test validazione messaggio valido	Verificato
TI-17	Test validazione messaggio con campi mancanti	Verificato
TI-18	Test arricchimento con shop più vicino - successo	Verificato
TI-19	Test arricchimento con shop più vicino - nessun shop	Verificato
TI-20	Test costruzione struttura dataflow	Verificato

IV - 4. Test di sistema

Verificano il comportamento del sistema completo in scenari realistici, garantendo che tutti i requisiti funzionali siano soddisfatti.

Codice Test	Requisito	Descrizione	Status
TS-01	RF1.1	Test endpoint autenticazione JWT	Verificato
TS-02	RF1.2	Test scadenza token JWT configurabile	Verificato
TS-03	RF1.3	Test validazione credenziali ClickHouse	Verificato
TS-04	RF1.4	Test invalidazione sessione logout	Verificato
TS-05	RF1.5	Test autenticazione JWT WebSocket	Verificato
TS-06	RF2.1	Test simulazione movimento utente OSRM	Verificato
TS-07	RF2.2	Test eventi GPS Kafka con SSL	Verificato
TS-08	RF2.3	Test profilo ciclabile OSRM	Verificato
TS-09	RF2.4	Test struttura contenuto eventi	Verificato

TS-10	RF2.5	Test controlli prontezza producer	Verificato
TS-11	RF3.2	Test calcolo distanza PostGIS	Verificato
TS-12	RF3.3	Test Connection Manager Singleton	Verificato
TS-13	RF3.4	Test prevenzione messaggi duplicati	Verificato
TS-14	RF3.5	Test metriche realtime Performance Observer	Verificato
TS-15	RF4.1	Test servizio generazione messaggi HTTP	Verificato
TS-16	RF4.2	Test provider LLM configurabili	Verificato
TS-17	RF4.3	Test TTL cache Redis configurabile	Verificato
TS-18	RF4.4	Test Strategy Pattern per offerte	Verificato
TS-19	RF4.5	Test Builder Pattern offerte complesse	Verificato
TS-20	RF4.6	Test Factory Pattern offerte tipizzate	Verificato
TS-21	RF4.7	Test Strategy validazione constraint offerte	Verificato
TS-22	RF5.1	Test interfaccia web file statici	Verificato
TS-23	RF5.2	Test mappa Leaflet marker categorizzati	Verificato
TS-24	RF5.3	Test storico route utente polyline	Verificato
TS-25	RF5.4	Test filtri categoria mapping predefinito	Verificato
TS-26	RF5.5	Test fallback WebSocket HTTP polling	Verificato
TS-27	RF6.1	Test storage eventi ClickHouse	Verificato

TS-28	RF6.2	Test indici spaziali PostGIS	Verificato
TS-29	RF6.3	Test profili utente ClickHouse	Verificato
TS-30	RF6.4	Test vista materializzata statistiche giornaliere	Verificato
TS-31	RF6.5	Test constraint temporali offerte PostgreSQL	Verificato
TS-32	RF7.1	Test serializzazione JSON cache Redis	Verificato
TS-33	RF7.2	Test fallback cache Memory LRU	Verificato
TS-34	RF7.3	Test TTL configurabile	Verificato
TS-35	RF7.4	Test statistiche cache hit rate	Verificato
TS-36	RV1	Test deployment Docker Compose	Verificato
TS-37	RV1	Test architettura microservizi	Verificato
TS-38	RV1	Test presenza stream processor	Verificato
TS-39	RV2	Test compatibilità browser	Verificato
TS-40	RV3	Test tecnologie frontend	Verificato
TS-41	RV4	Test constraint geografici	Verificato
TS-42	RV5	Test requisiti operazionali	Verificato
TS-43	RNF2	Test requisiti scalabilità	Verificato
TS-44	RNF4	Test requisiti sicurezza	Verificato
TS-45	RNF5	Test requisiti usabilità	Verificato
TS-46	RNF6	Test requisiti manutenibilità	Verificato
TS-47	RFD1.1	Test lazy loading Intersection Observer	Verificato
TS-48	RFD1.2	Test theme switching LocalStorage	Verificato

TS-49	RFD1.3	Test cache locale frontend	Verificato
TS-50	RFD1.4	Test ottimizzazione marker max 100	Verificato
TS-51	RFD1.5	Test design responsive breakpoint mobile	Verificato
TS-52	RFD2.1	Test Prometheus FastAPI Instrumentator	Verificato
TS-53	RFD2.2	Test dashboard Grafana assemblaggio dinamico	Verificato
TS-54	RFD2.3	Test logging strutturato livelli configurabili	Verificato
TS-55	RFF1.1	Test integrazione API meteo per context-aware messaging	Non implementato
TS-56	RFF1.2	Test payment integration per offer redemption	Non implementato
TS-57	RFF2.1	Test reinforcement learning per offer optimization	Non implementato
TS-58	RFF3.1	Test A/B testing per offer strategies	Non implementato

IV - 5. Test di accettazione

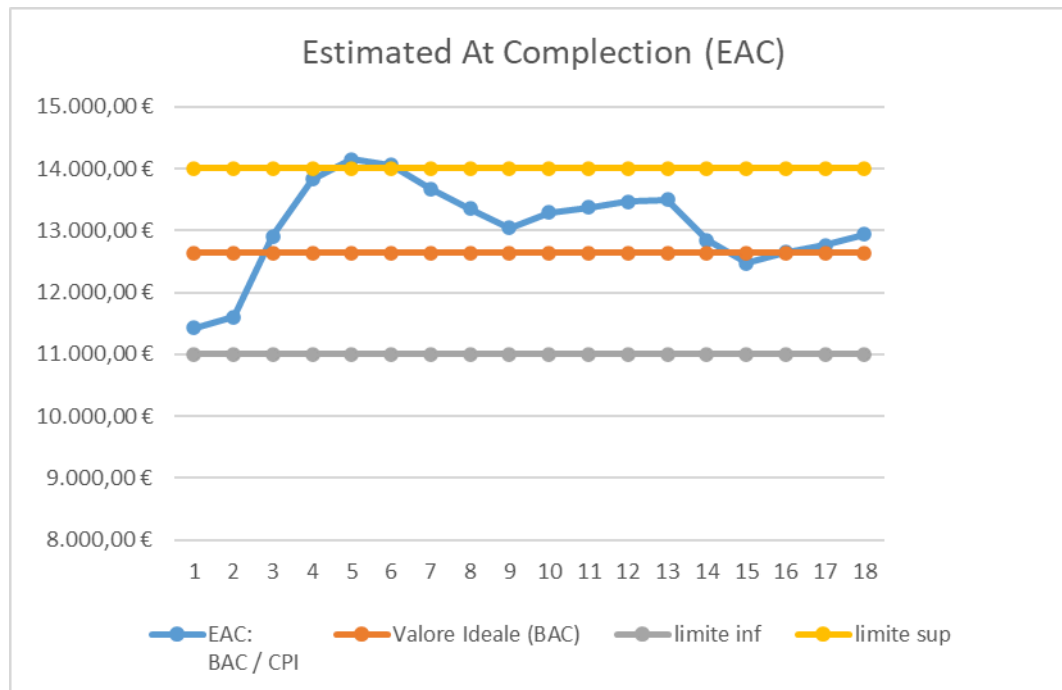
I test di accettazione verificano che i requisiti business siano soddisfatti dal punto di vista dell'utente finale. Tutti i test di accettazione sono stati effettuati in presenza di SyncLab.

Codice Test	Descrizione	Status
TA-01	Utente si avvicina a negozio e riceve offerta personalizzata in tempo reale	Verificato
TA-02	Targeting intelligente - solo utenti nel raggio ricevono notifiche	Verificato
TA-03	Personalizzazione AI - stesso negozio, messaggi diversi per età/professione	Verificato
TA-04	Scalabilità - gestione concorrente multipli utenti	Verificato
TA-05	Dashboard mobile responsive per utenti finali	Verificato
TA-06	Analytics real-time - statistiche utente e negozi visitati	Verificato
TA-07	Deploy Docker one-click con microservizi	Verificato
TA-08	Mappa interattiva con filtri categoria in tempo reale	Verificato
TA-09	Dashboard utente con statistiche personali real-time	Verificato
TA-10	Aggiornamenti real-time posizione	Verificato
TA-11	ETL automatico negozi e offerte via Airflow	Verificato
TA-12	Messaggi AI personalizzati	Verificato
TA-13	Dashboard Grafana con analytics business	Verificato
TA-14	Targeting offerte basato su interessi utente	Verificato
TA-15	Conversion rate tracking	Verificato
TA-16	Revenue tracking per categoria e negozio	Verificato
TA-17	User journey tracking con storico notifiche	Verificato

V. Cruscotto valutazione della qualità

Di seguito sono riportate le misurazioni effettuate nel periodo compreso tra l'assegnazione del capitolato e il completamento della Product Baseline_G (PB_G).

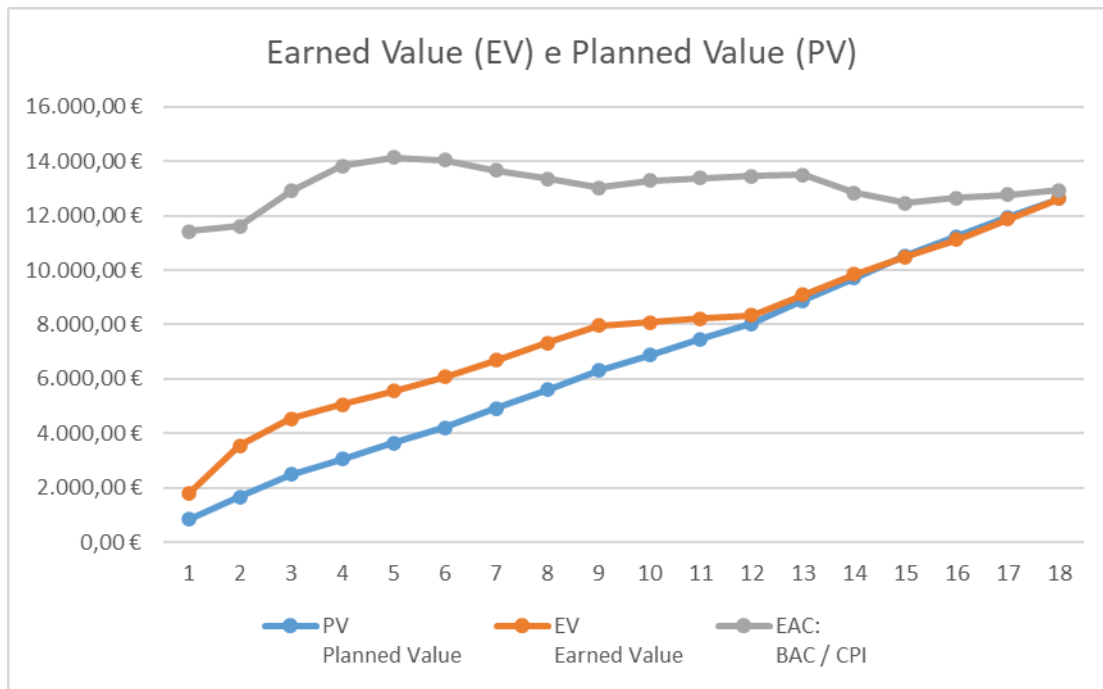
V - 1. Estimated At Completion (MPC-EAC)



Il grafico sull'EAC evidenzia un andamento complessivamente sotto controllo, con una prima fase in cui le stime di completamento si mantengono vicine o addirittura inferiori al budget previsto. Nei periodi centrali si registra un aumento dell'EAC, con un picco massimo al quinto periodo che supera temporaneamente la soglia superiore, dovuto a un momentaneo calo di efficienza. Questo scostamento, però, è stato prontamente contenuto grazie a un'attenta gestione delle risorse e a un costante monitoraggio_G. A partire dal sesto periodo, infatti, si osserva un progressivo miglioramento che porta i valori stimati a rientrare entro limiti accettabili. Negli ultimi periodi l'EAC si stabilizza, segno di una gestione efficace e coerente. Pur restando leggermente al di sopra del valore ideale, il trend complessivo dimostra una buona capacità del team di intervenire in modo tempestivo, mantenendo il progetto su un percorso sostenibile e ben governato.

Nel passaggio alla fase PB, l'EAC evidenzia un leggero incremento rispetto alla chiusura della fase RTB. Dopo aver toccato il punto minimo nel periodo 15, i valori risalgono progressivamente fino al periodo 18, rimanendo comunque entro la fascia di accettabilità e vicini al valore ideale. Questo andamento riflette una stabilizzazione complessiva della spesa prevista, con oscillazioni contenute e sotto controllo nella parte finale del progetto.

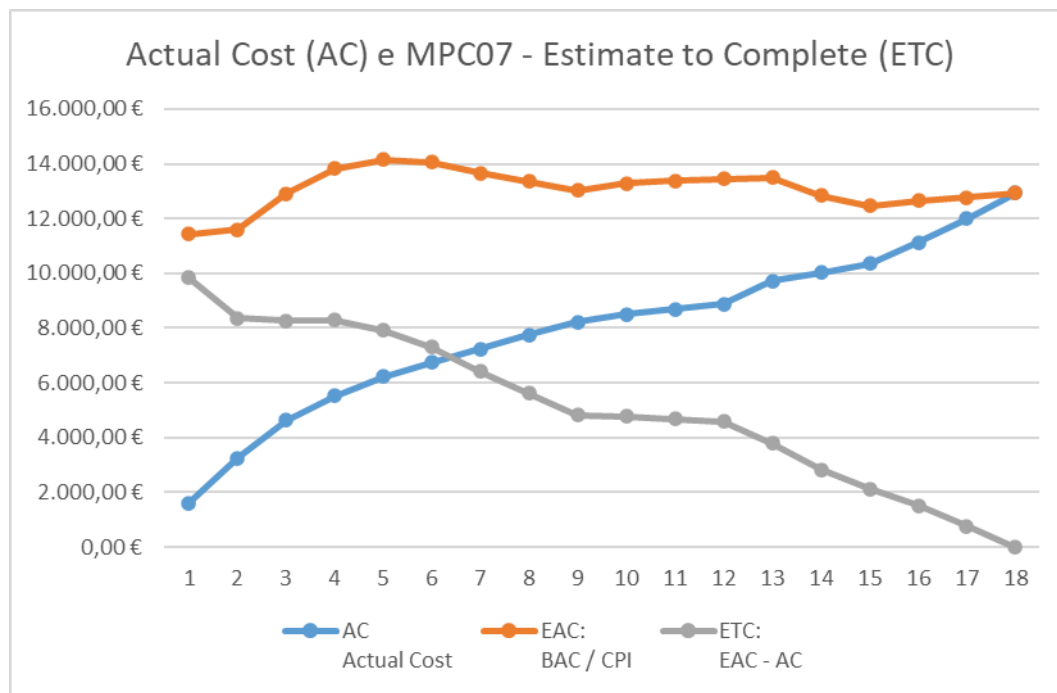
V - 2. Earned Value (MPC-EV) e Planned Value (MPC-PV)



Il grafico mostra un andamento molto solido della fase RTB, con un Earned Value (EV) costantemente superiore al Planned Value (PV) fino al dodicesimo periodo. Questo indica che il progetto ha mantenuto un buon ritmo di avanzamento, completando attività in anticipo rispetto al piano. Il margine tra EV e PV è particolarmente ampio nei primi periodi, segnalando un avvio molto efficace. A partire dal nono periodo si osserva un avvicinamento tra EV e PV, che si mantengono comunque vicini fino alla fine del periodo RTB. Questo trend evidenzia un allineamento progressivo tra quanto realizzato e quanto pianificato, segno di una pianificazione realistica e di una buona capacità di adattamento. Anche l'EAC si stabilizza nella parte alta del grafico, coerente con il buon avanzamento delle attività. Nel complesso, il team ha dimostrato un controllo efficace, riuscendo a mantenere performance superiori alle attese per la maggior parte del progetto.

Nei periodi della fase PB, EV e PV proseguono il trend di avvicinamento osservato a fine RTB, fino a coincidere sostanzialmente al termine del periodo 18. Questo indica che il lavoro completato si è mantenuto in linea con la pianificazione, con una chiusura perfettamente allineata alle previsioni. L'andamento evidenzia l'assenza di scostamenti rilevanti e una gestione regolare delle attività nella parte conclusiva del progetto.

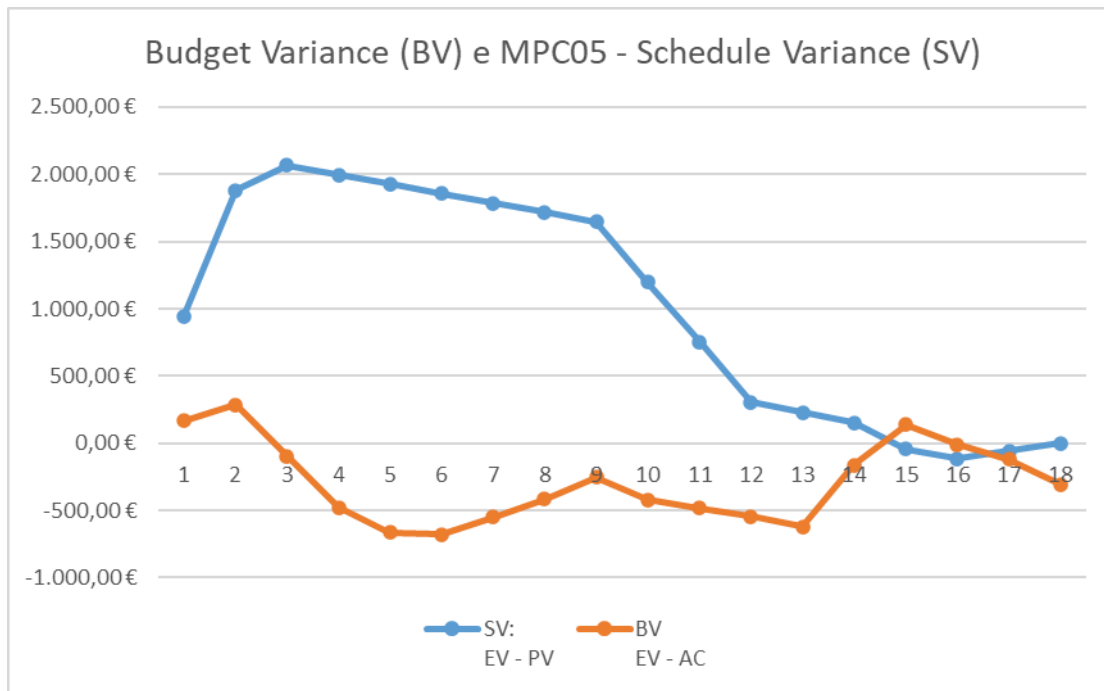
V - 3. Actual Cost (MPC-AC) e Estimate To Completion (MPC-ETC)



Il grafico evidenzia una gestione efficace dei costi lungo tutto la fase RTB. L'Actual Cost (AC) cresce in modo regolare e controllato, mantenendosi sempre ben al di sotto del valore dell'EAC, a conferma di una spesa coerente con le previsioni. L'EAC, pur presentando una fase centrale leggermente più elevata, si stabilizza verso la fine, mostrando che i costi complessivi previsti sono rimasti entro una soglia sostenibile. Il valore dell'ETC, ovvero la stima dei costi rimanenti, decresce in modo progressivo e costante: questo trend riflette una corretta distribuzione dell'impegno economico e una chiara visione sul lavoro residuo. Nel complesso, il progetto è stato condotto con attenzione e precisione, contenendo i costi e dimostrando una buona capacità di previsione e adattamento.

Durante la fase PB, l'AC continua a crescere fino a convergere con l'EAC al termine del periodo 18, segnalando l'esaurimento del budget pianificato in concomitanza con la conclusione delle attività. L'ETC prosegue la discesa costante già visibile in RTB, fino ad azzerarsi, confermando che non restano costi aggiuntivi previsti dopo la chiusura del progetto.

V - 4. Budget Variance (MPC-BV) e Schedule Variance (MPC-SV)

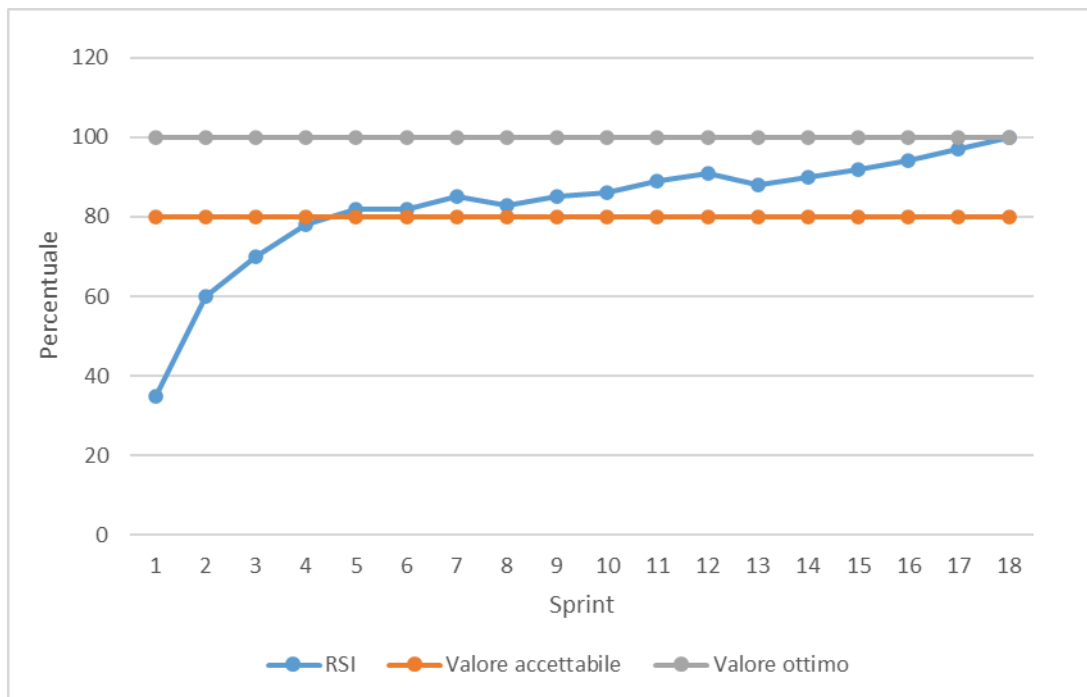


Il grafico mostra un andamento molto positivo dello Schedule Variance (SV) per la maggior parte della fase RTB: i valori restano ampiamente sopra lo zero fino al nono periodo, indicando che le attività sono state completate in anticipo rispetto alla pianificazione. Questo evidenzia una buona organizzazione del lavoro e una produttività costante. La leggera discesa di SV negli ultimi periodi rappresenta un progressivo riallineamento al piano, senza però segnalare ritardi critici. Per quanto riguarda la Budget Variance (BV), il progetto ha registrato oscillazioni contenute attorno allo zero, con alcuni periodi in lieve disavanzo. Tuttavia, la BV non ha mai superato soglie di allarme, dimostrando una gestione attenta delle risorse economiche. Nel complesso, le due curve evidenziano una performance solida, con una forte partenza, una gestione attenta dei costi e una tendenza al riequilibrio nella fase finale della fase RTB.

Nella fase PB, lo Schedule Variance si mantiene vicino allo zero, segnalando il pieno rispetto delle tempistiche pianificate senza anticipi o ritardi significativi. La Budget Variance presenta lievi variazioni positive e negative, ma sempre contenute, a conferma di un controllo costante delle spese nella parte conclusiva del progetto. L'assenza di picchi anomali indica che la fase PB è stata gestita in equilibrio sia sul fronte temporale che su quello economico.

V - 5. Indice di Stabilità dei Requisiti (MPC-RSI)

L'ISR misura la stabilità dei requisiti del progetto nel tempo, valutando quanto siano stati modificati o aggiornati durante il ciclo di vita.

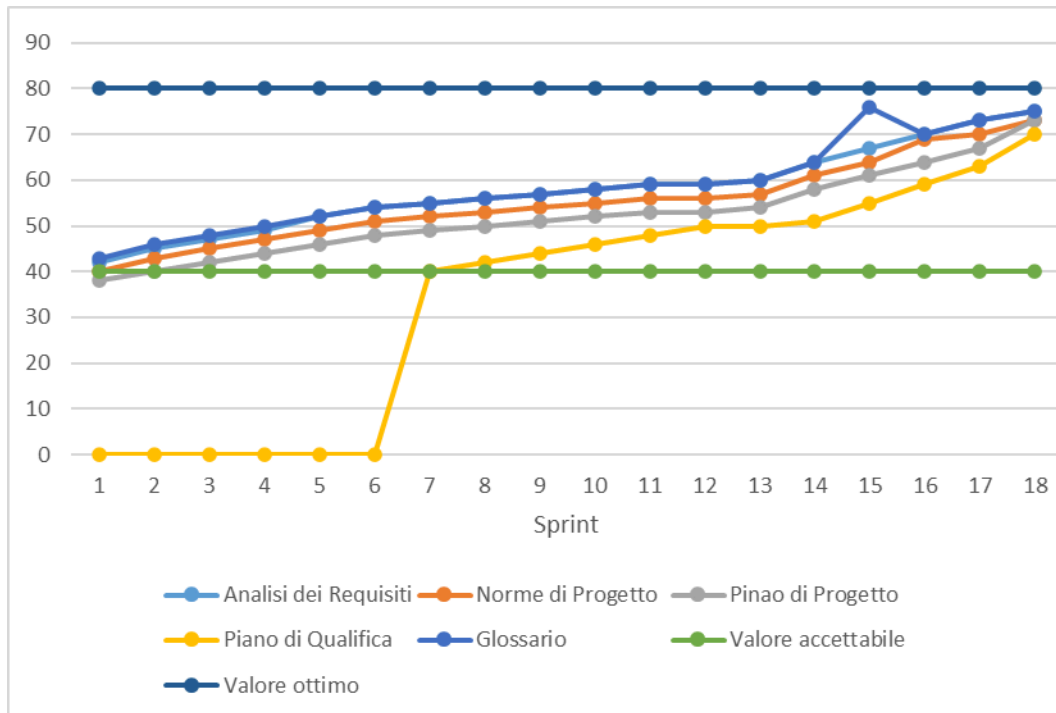


Il grafico dell'ISR mostra una crescita costante da 0% a 82% nei primi sei periodi, superando stabilmente la soglia accettabile del 75%. Nel Periodo 7 si osserva un calo a 79% dovuto alle modifiche per l'integrazione di Grafana_G emerse dai feedback aziendali. La fase finale (Periodi 8-13) presenta una stabilizzazione ottimale con valori tra 83-91%, raggiungendo il picco nel Periodo 12 con l'approvazione del P.o.C._G. Il valore finale dell'88% dimostra un controllo efficace dei requisiti, nonostante l'aggiunta tardiva dello stream processor_G nell'ultimo periodo.

Nella fase PB l'ISR si mantiene costante, con valori prossimi al risultato finale della RTB. L'assenza di nuove modifiche rilevanti ai requisiti conferma la maturità del progetto e la stabilità delle specifiche, favorendo un completamento lineare delle attività senza introdurre variazioni sostanziali.

V - 6. Indice Gulpease (MPC-GP)

- Indice che valuta la leggibilità dei documenti scritti in italiano.

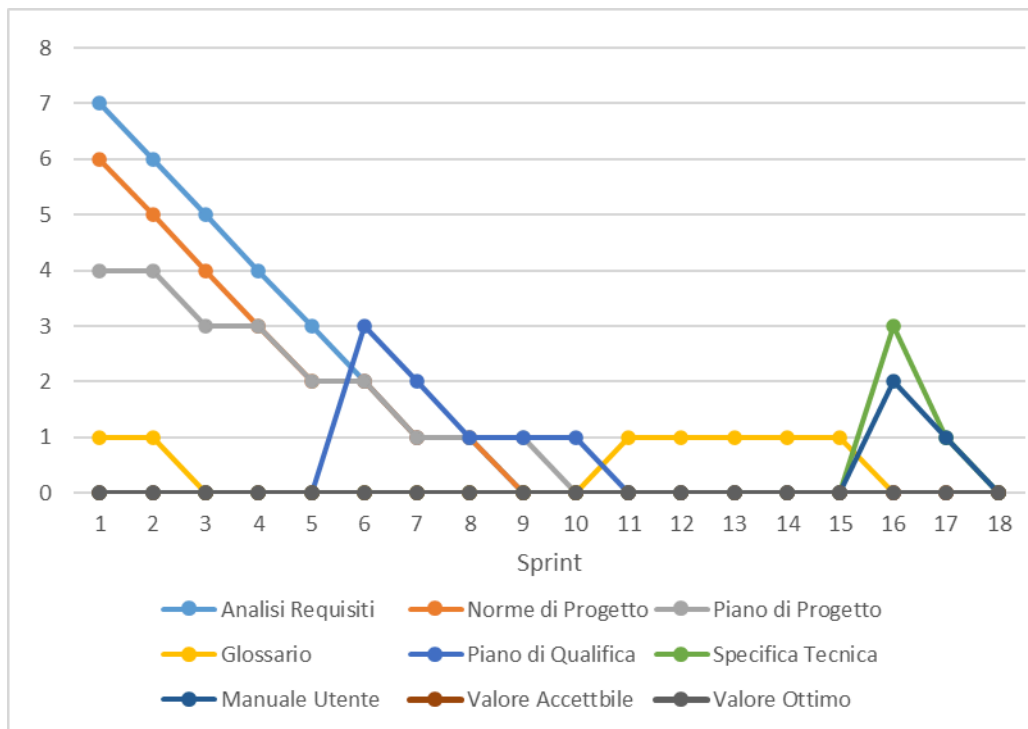


Il grafico mostra un andamento differenziato che riflette la timeline del modello incrementale_G adottato. I documenti pre-codice (Analisi dei Requisiti e Norme di Progetto) presentano crescita rapida nei primi periodi, mentre Piano di Progetto e Glossario mostrano progressione costante per tutto il ciclo di vita. Il Piano di Qualifica inizia dal Periodo 6 crescendo fino a 51, coincidendo con l'accumulo di metriche durante lo sviluppo. Tutti i documenti superano la soglia accettabile del 40, con Analisi dei Requisiti e Glossario che raggiungono 58 punti finali, dimostrando elevata leggibilità. Nei Periodi 14 e 15 si osserva un lieve incremento in quasi tutti i documenti, legato alle ultime revisioni e uniformazioni stilistiche pre-consegna, mantenendo un'elevata leggibilità complessiva.

Nella fase PB, i valori si mantengono stabili o in lieve miglioramento grazie alle revisioni finali e all'armonizzazione dello stile documentale. Il mantenimento sopra la soglia di leggibilità conferma l'efficacia delle pratiche redazionali adottate, senza regressioni qualitative nei testi.

V - 7. Correttezza Ortografica (MPC-CO)

Metriche che misurano la presenza di errori ortografici nei documenti, valutando la qualità formale del contenuto.

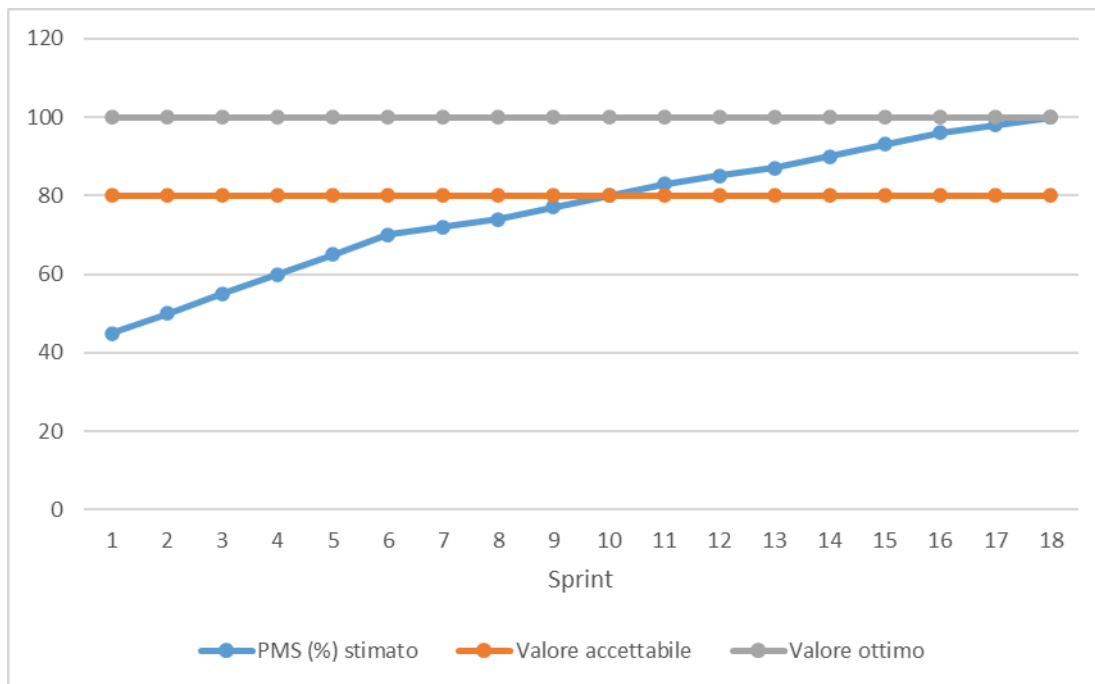


Il grafico evidenzia un progressivo miglioramento della qualità ortografica durante tutto il progetto. I documenti pre-codice (Analisi dei Requisiti e Norme di Progetto) mostrano il picco di errori nei primi periodi, seguiti da una rapida diminuzione verso l'obiettivo zero grazie alle revisioni sistematiche. Il Piano di Progetto presenta una decrescita costante riflettendo il continuo processo di correzione durante gli aggiornamenti, mentre il Piano di Qualifica mostra un miglioramento rapido dalla sua introduzione. Il Glossario mantiene sempre valori bassi data la semplicità dei contenuti, e il lieve aumento finale è dovuto alle intensive revisioni pre-consegna RTB.

Nella fase PB, la correttezza ortografica si stabilizza su valori ottimali, con la quasi totale assenza di errori. Gli interventi si limitano a micro-correzioni e uniformazioni stilistiche, indice di un processo di revisione ormai maturo e consolidato.

V - 8. Percentuale di Metriche Soddisfatte (MPC-PMS)

Percentuale di metriche di qualità definite per il progetto che sono state soddisfatte rispetto agli obiettivi prefissati.

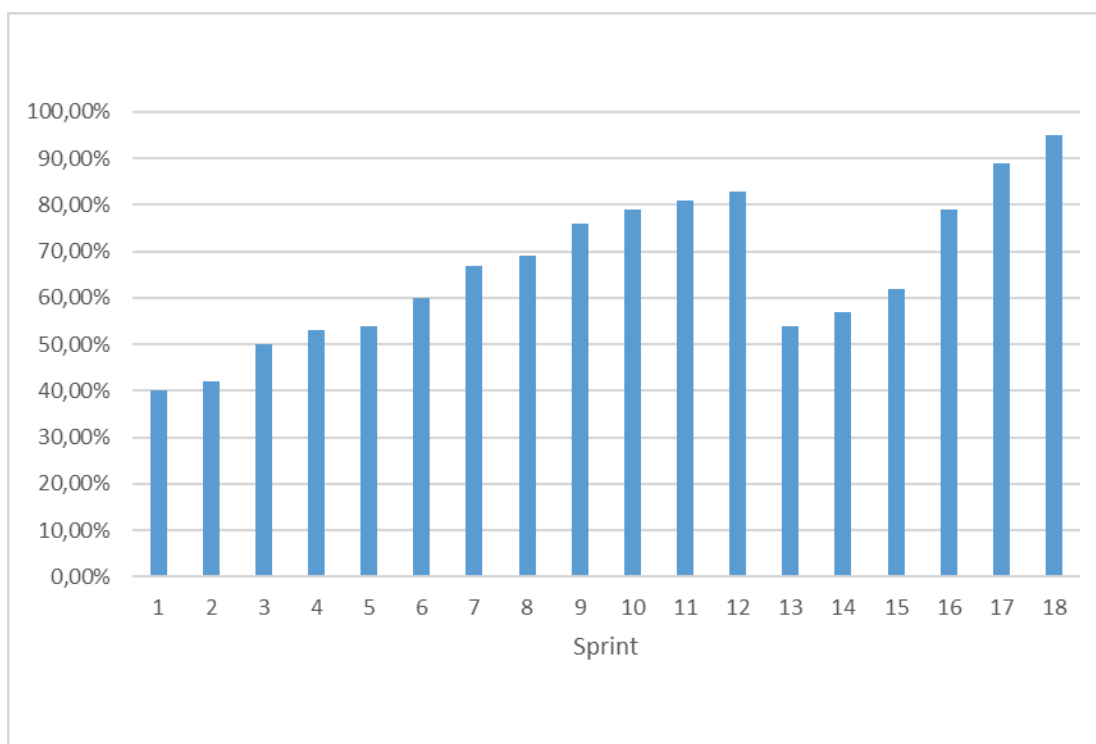


Nella fase RTB, il grafico evidenzia una crescita costante della qualità complessiva del progetto, con il superamento della soglia accettabile a metà progetto. Il picco finale coincide con l'approvazione del P.o.C., testimoniando il raggiungimento degli standard qualitativi prefissati. Il lieve calo nell'ultimo tratto della RTB è attribuibile all'introduzione dello stream processor, che ha aggiunto nuove metriche da soddisfare senza compromettere la qualità già raggiunta.

Durante la fase PB, la percentuale di metriche soddisfatte torna a crescere fino a raggiungere nuovamente il 100% a ridosso della chiusura del progetto. Questo recupero evidenzia il completamento con successo delle metriche aggiunte in precedenza, assicurando il pieno rispetto degli obiettivi qualitativi stabiliti.

V - 9. Efficienza Temporale (MPC-ET)

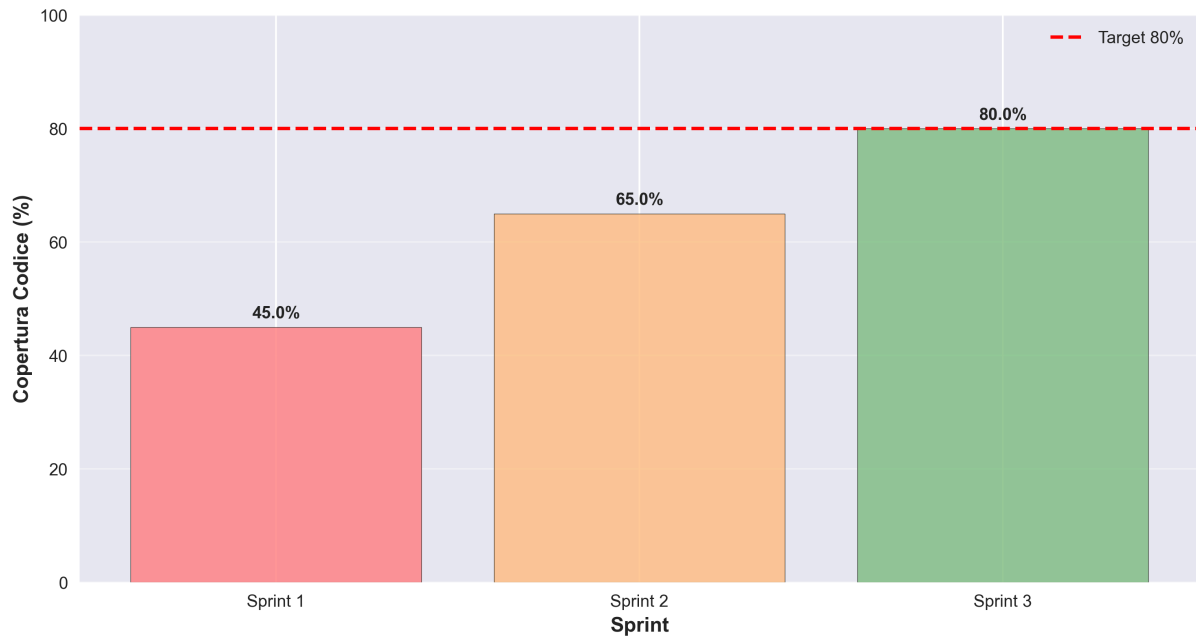
Misura l'efficacia nell'utilizzo del tempo per completare le attività, confrontando il tempo effettivamente impiegato con quello previsto.



Il grafico evidenzia un andamento variabile dell'efficienza temporale durante il progetto. I primi periodi mostrano buona efficienza, seguiti da superamenti delle ore previste dovuti alla sottostima della complessità delle attività tecniche. La criticità maggiore si concentra nei Periodi 10-12 dovuto alla sessione estiva universitaria che ha drasticamente ridotto le ore dedicate al progetto. Il Periodo 13 conferma il ritorno a pieno regime del team con un intenso sforzo finale per completare la fase RTB, dimostrando capacità di recupero nonostante le criticità temporanee legate agli impegni accademici. Durante i periodo 14 e 15 si registra un calo dovuto all'impegno del gruppo dirottato verso la preparazione alle fasi di revisione.

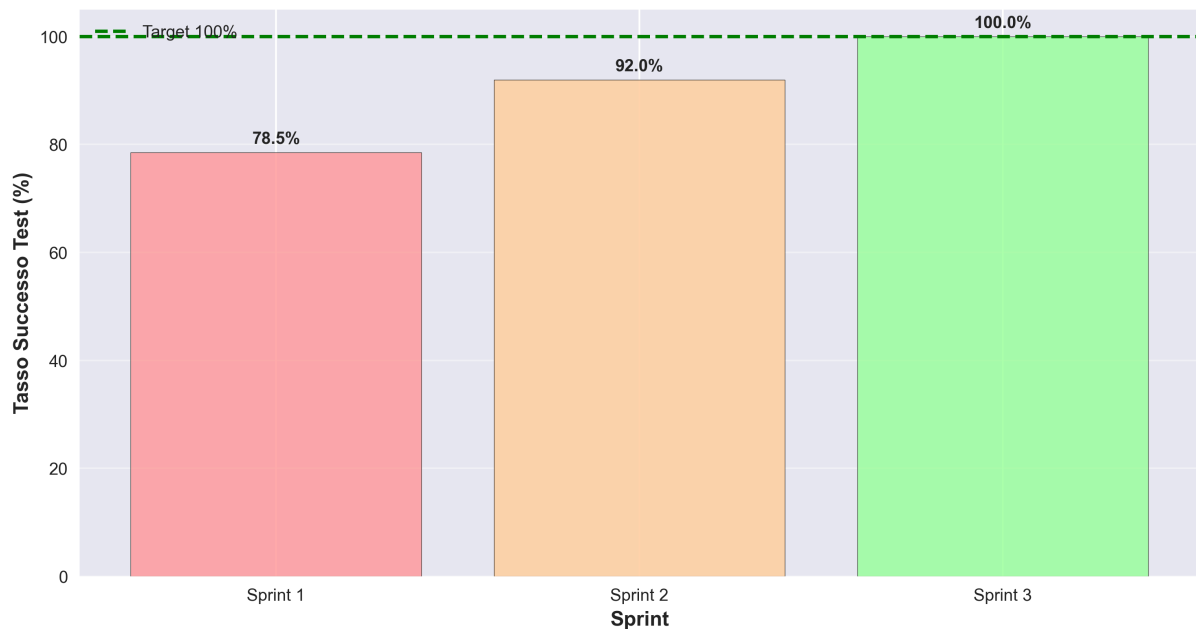
Durante la fase PB l'impegno e la produttività del gruppo tornano a crescere in modo intensivo per garantire la consegna del progetto entro i termini stabiliti.

V - 10. Copertura Codice (MPC-CC)



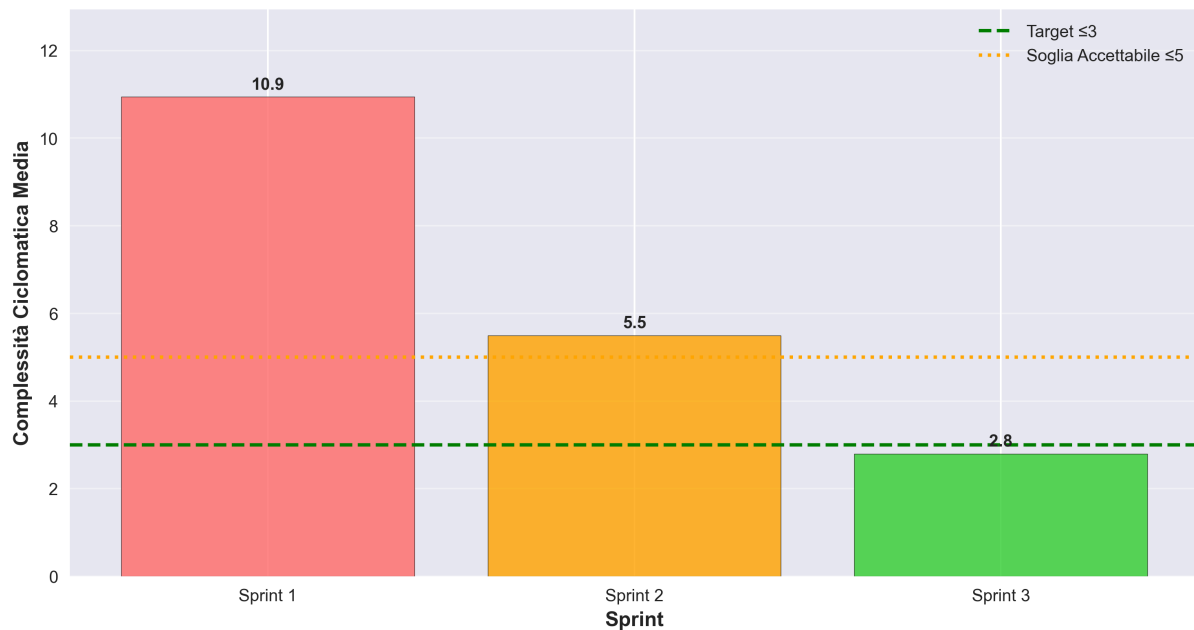
Dal grafico si evince che la copertura del codice, al partire della fase di PB e dunque dall'inizio dello sviluppo dei test, ha mostrato un miglioramento costante nel tempo, raggiungendo valori ottimali nella fase finale del progetto.

V - 11. Passed test cases percentage (MDP-PTCP)



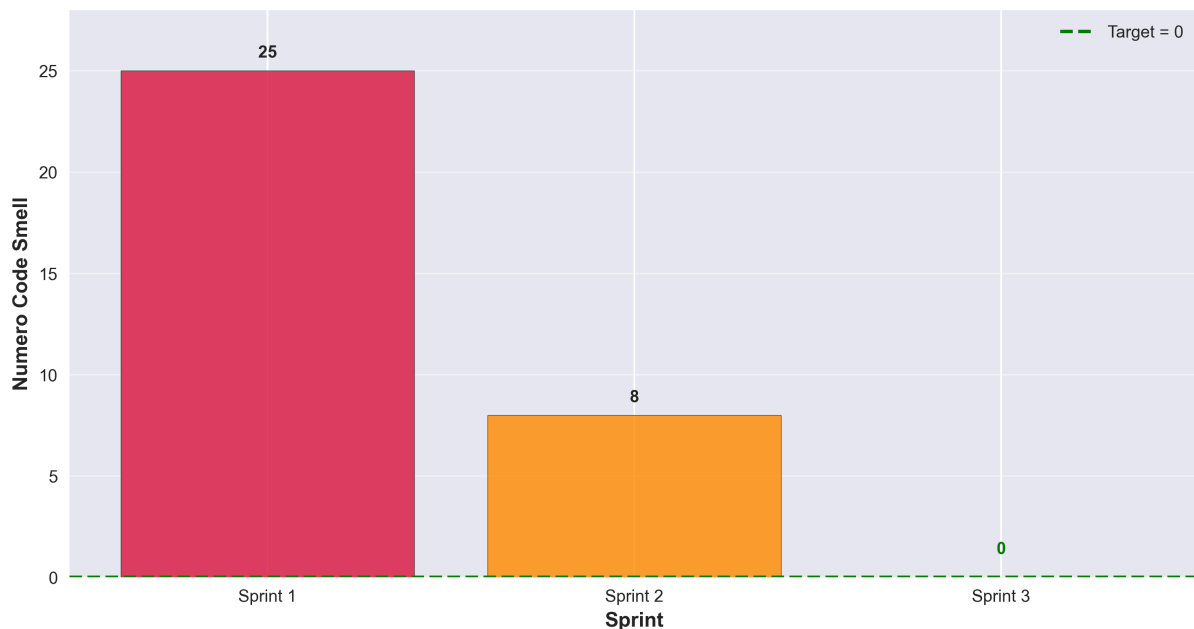
La percentuale di test superati rispetto al totale eseguito è anch'essa cresciuta regolarmente nel tempo man mano che il progetto si avvicinava alla sua conclusione. Il picco finale, che raggiunge il 100%, è stato raggiunto grazie all'esecuzione di tutti i test di accettazione in presenza di *SyncLab* e dall'aver passato tutti i test di unità, integrazione e sistema.

V - 12. Complessità Ciclomatica Media (MDP-CCM)



La complessità ciclomatica media del progetto, seppur inizialmente elevata, ha mostrato un trend di riduzione costante nel tempo, grazie all'adozione di pratiche di refactoring e all'evoluzione del design.

V - 13. Code Smell (MDP-CS)



I code smell sono stati monitorati nel tempo a partire dalla fase di PB e, sebbene in principio presenti in numero elevato, sono stati progressivamente ridotti grazie all'implementazione di pratiche di refactoring e alla revisione del codice.

VI. Processi di automiglioramento

Il miglioramento continuo costituisce un elemento essenziale per assicurare la qualità complessiva del progetto software. Nella sezione seguente vengono illustrati i principali problemi emersi durante lo sviluppo e le strategie adottate per affrontarli, con l'obiettivo di eliminarne le cause o mitigarne gli effetti.

VI - 1. Valutazione sull'organizzazione

Descrizione problema	Misure adottate
La difficoltà maggiore è stata riuscire a trovare dei momenti dedicati agli incontri di gruppo a cui potessero partecipare tutti i membri contemporaneamente. La conseguenza erano tempi più lunghi di attesa per una riunione plenaria e una maggiore difficoltà a restare aggiornati sui progressi effettuati.	Sono state previste più strade per risolvere questa problematica. La più efficace è stata prediligere degli incontri in modalità online, seguita da sondaggi per la scelta della data migliore e un utilizzo intensivo delle applicazioni di messaggistica istantanea per mantenere tutti i membri aggiornati.
In seguito alle verifiche con i docenti, è emerso che il metodo di sviluppo incrementale adottato fino a quel momento non era considerato adeguato alle esigenze e alle tempistiche delle correzioni a cui sottoporre il progetto.	Abbiamo abbandonato il modello incrementale, adottando il metodo agile. Questo ha permesso di aumentare la flessibilità nella gestione delle attività, ridurre i rischi legati a consegne troppo distanziate e migliorare la reattività del gruppo di fronte ai cambiamenti di priorità.

VI - 2. Valutazione sui ruoli

Descrizione problema	Misure adottate
Inizialmente non era stato definito con chiarezza il ruolo adibito al caricamento della documentazione online, causando discrepanza tra la documentazione caricata online e gli effettivi progressi effettuati.	Durante una delle riunioni è stato attribuito questo compito al responsabile _G , mantenendo una chiara divisione dei compiti tra redattore e verificatore _G .
Abbiamo riscontrato che alcuni ruoli avevano un carico di lavoro sbilanciato rispetto agli altri, in particolare nelle fasi avanzate del progetto dove il ruolo di analista _G e amministratore _G trovavano un'utilità ridotta.	Si è scelto di ridurre al minimo le ore dedicate a questi ruoli, ma lasciandoli comunque attivi per avere un punto di riferimento in caso di problemi o integrazioni da apportare.

Nella pianificazione iniziale della fase PB erano state previste troppo poche ore per i ruoli di progettista e verificatore, generando un sovraccarico in corso d'opera e rallentamenti nelle consegne intermedie.	Abbiamo incrementato il monte ore per questi ruoli e riorganizzato la distribuzione dei compiti, garantendo così una copertura adeguata delle attività di progettazione e verifica fino alla conclusione del progetto.
--	--

VI - 3. Valutazione sugli strumenti

Descrizione problema	Misure adottate
Inizialmente non era stato previsto uno stream processor all'interno del progetto. La sua utilità è risultata sempre più rilevante nel corso dello sviluppo.	Vista la sua crescente necessità procedendo nelle fasi di sviluppo del P.o.C. si è scelto di aggiungere una fase di incremento _G a quelle inizialmente previste per integrarlo con la baseline _G .
Trovare uno strumento comune per redigere la documentazione è stato uno degli obiettivi iniziali del gruppo. Era necessario trovare uno strumento che coniugasse semplicità di utilizzo ad una sufficiente versatilità _G .	La scelta è ricaduta su Typst _G . Vista la sua relativa semplicità di apprendimento e di utilizzo si è ritenuto che tutti i membri del gruppo sarebbero riusciti ad apprendere l'utilizzo in tempistiche brevi. LateX _G è stato scartato vista la maggiore complessità di utilizzo e di apprendimento. Si è ritenuto che le sue funzionalità aggiuntive non fossero necessarie, rendendo Typst un compromesso decisamente migliore.
Inizialmente non tutti i membri avevano una sufficiente esperienza nell'utilizzo di Git _G e GitHub _G da consentire a tutti di caricare la documentazione e risolvere eventuali problemi di caricamento dei documenti nel sito.	Per evitare disallineamenti eccessivi, ogni membro ha deciso di approfondire autonomamente le tecnologie necessarie per lavorare in modo efficace. Questo veniva spesso facilitato dalla disponibilità dei componenti più esperti, che hanno preparato dei tutorial _G scritti su specifici argomenti, mettendoli a disposizione di tutti, in modo da avere procedure _G chiare e uguali per tutti.