

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
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	4
I - 1. Scopo del documento	4
I - 2. Scopo del prodotto	4
I - 3. Riferimenti	4
I - 3.1. Riferimenti informativi	4
I - 3.2. Riferimenti normativi	4
II. Qualità di processo	5
II - 1. Fornitura	5
II - 2. Sviluppo	6
II - 3. Documentazione	6
II - 4. Verifica	6
II - 5. Gestione della qualità	6
III. Qualità di prodotto	8
III - 1. Funzionalità	8
III - 2. Affidabilità	8
III - 3. Usabilità	9
III - 4. Efficienza	9
III - 5. Manutenibilità	9
IV. Specifiche dei test	11
IV - 1. Convenzioni di nomenclatura	11
IV - 2. Test di unità	11
IV - 3. Test di integrazione	11
IV - 4. Test di sistema	11
IV - 5. Test di accettazione	11
V. Cruscotto valutazione della qualità	12
V - 1. Estimated At Completion (MPC-EAC)	12
V - 2. Earned Value (MPC-EV) e Planned Value (MPC-PV)	13
V - 3. Actual Cost (MPC-AC) e Estimate To Completion (MPC-ETC)	14
V - 4. Budget Variance (MPC-BV) e Schedule Variance (MPC-SV)	15
V - 5. Indice di Stabilità dei Requisiti (MPC-RSI)	16
V - 6. Indice Gulpease (MPC-GP)	17
V - 7. Correttezza Ortografica (MPC-CO)	18
V - 8. Percentuale di Metriche Soddisfatte (MPC-PMS)	19
V - 9. Efficienza Temporale (MPC-ET)	20
VI. Processi di automiglioramento	20
VI - 1. Valutazione sull'organizzazione	20
VI - 2. Valutazione sui ruoli	21
VI - 3. Valutazione sugli strumenti	21

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 (v1.1.0)
- 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 (v1.1.0)

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.

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.

IV - 4. Test di sistema

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

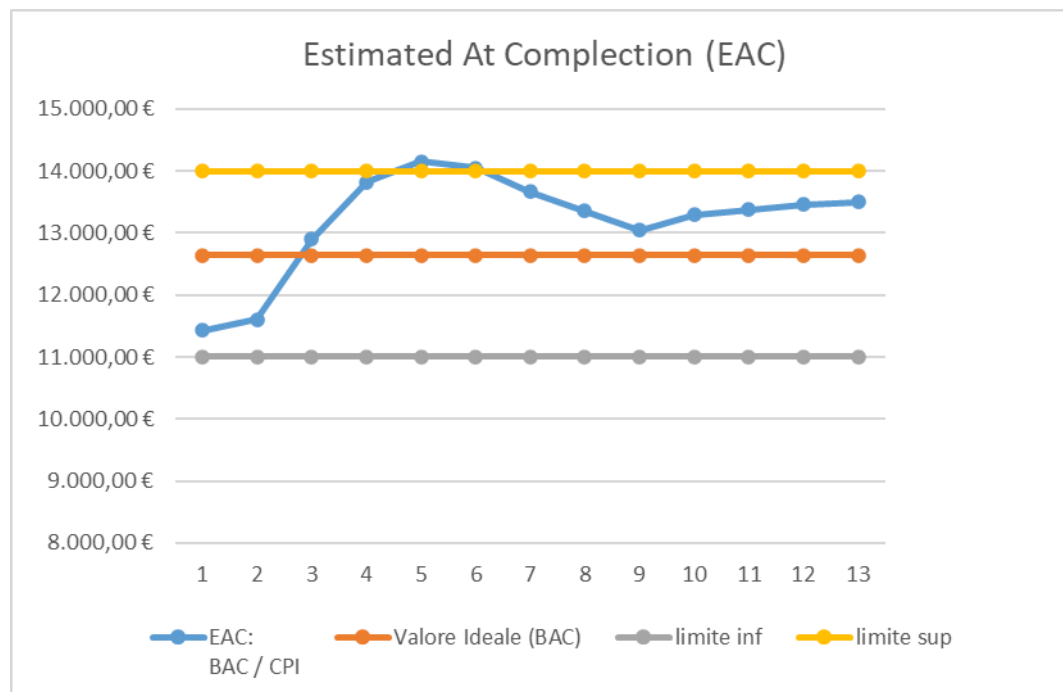
IV - 5. Test di accettazione

I test di accettazione verificano che i requisiti business siano soddisfatti dal punto di vista dell'utente finale.

V. Cruscotto valutazione della qualità

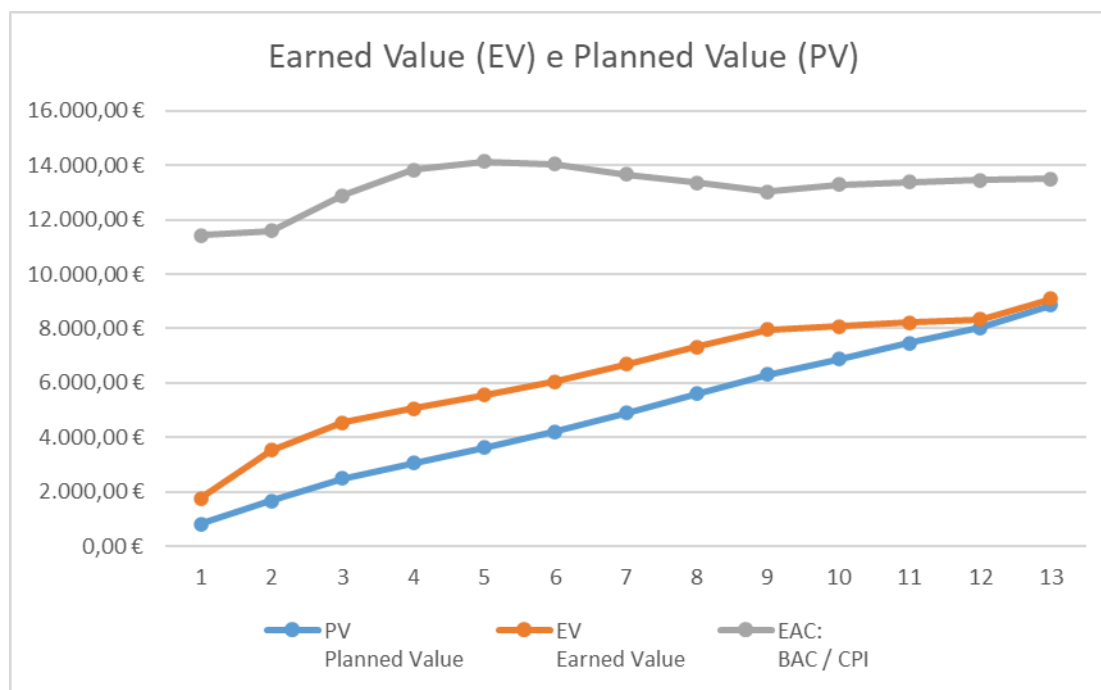
Di seguito sono riportate le misurazioni effettuate nel periodo compreso tra l'assegnazione del capitolato e il raggiungimento della Requirements and Technology Baseline (RTB_G). Le misure mancanti saranno rilevate nel corso delle attività previste per la Product Baseline (PB_G).

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



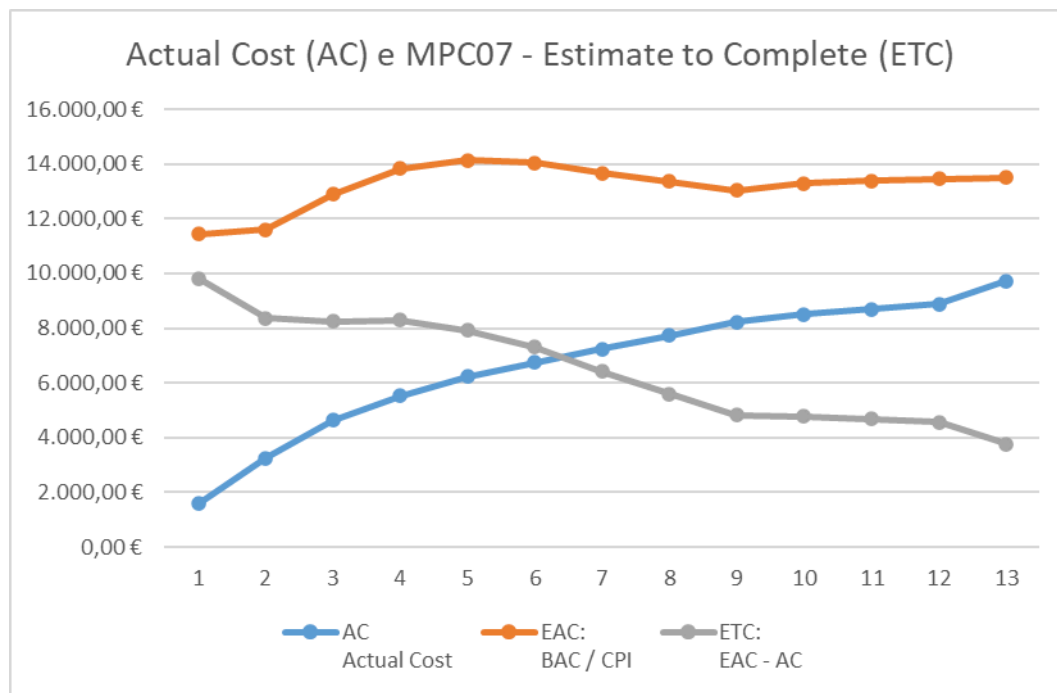
Il grafico sull'EAC evidenzia un andamento complessivamente sotto controllo, con una prima fase in cui le stime di completamento si mantengono vicine o addirittura inferiori al budget previsto. Nei periodi centrali si registra un aumento dell'EAC, con un picco massimo al quinto periodo che supera temporaneamente 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.

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

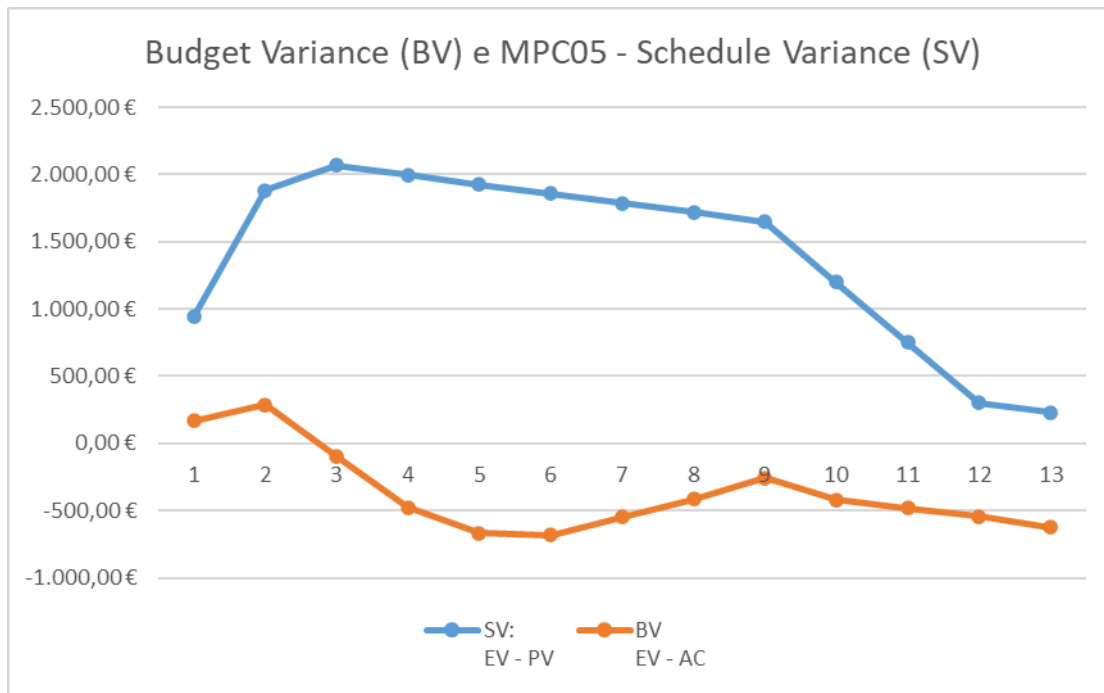


Il grafico mostra un andamento molto solido della fase RTB, con un Earned Value (EV) costantemente superiore al Planned Value (PV) fino al dodicesimo periodo. Questo indica che il progetto ha mantenuto un buon ritmo di avanzamento, completando attività in anticipo rispetto al piano. Il margine tra EV 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.

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.

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.

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

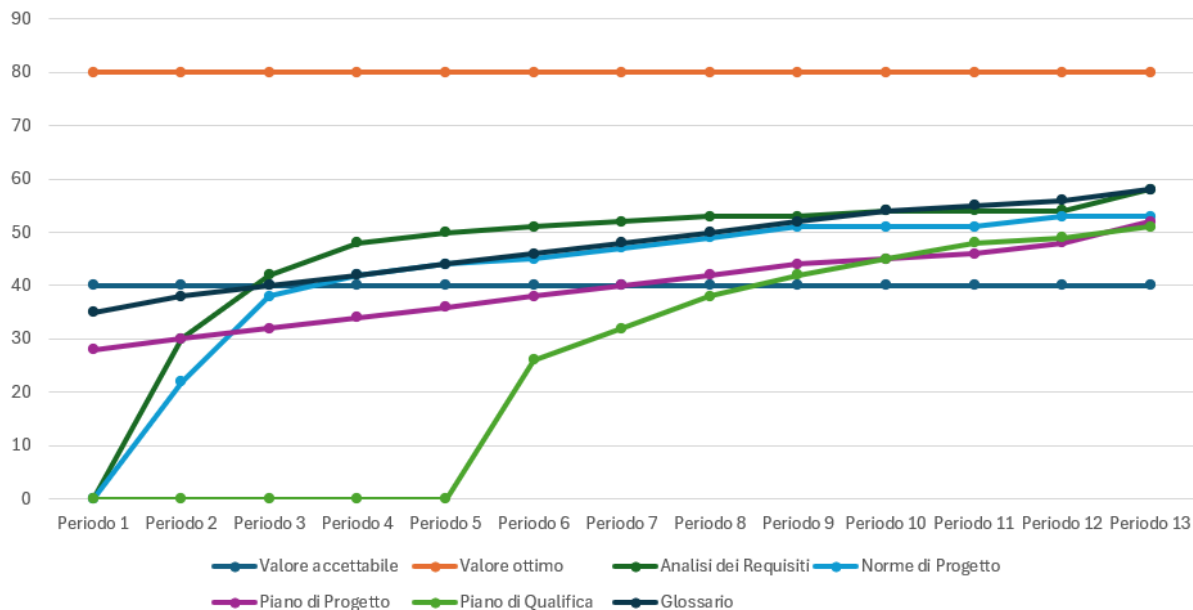
L'ISR misura la stabilità dei requisiti del progetto nel tempo, valutando quanto siano stati modificati o aggiornati durante il ciclo di vita.



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.

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.

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.

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.



Il grafico evidenzia una crescita costante della qualità complessiva del progetto attraverso il modello incrementale adottato. Il superamento della soglia accettabile avviene a metà progetto, dimostrando l'efficacia dei cicli di sviluppo e verifica sistematici. Il picco finale coincide con l'approvazione del P.o.C. da parte dell'azienda, testimoniando il raggiungimento degli standard qualitativi prefissati. Il lieve calo nell'ultimo periodo è dovuto all'introduzione dello stream processor, che ha aggiunto nuove metriche da soddisfare senza compromettere la qualità già raggiunta.

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.

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	Sono state previste più strade per risolvere questa problematica. La più efficace è stata prediligere degli incontri

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.	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.

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 .

<p>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.</p>	<p>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 apprenderne 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.</p>
<p>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.</p>	<p>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.</p>