



UNIVERSITÀ DEGLI STUDI DI PADOVA
CORSO DI LAUREA IN INFORMATICA (L-31)
Corso di Ingegneria del Software
Anno Accademico 2025/2026

Piano di Qualifica

Gruppo: NightPRO
swe.nightpro@gmail.com

Data: 2026-01-30

Versione: 0.3

Tabella delle Versioni

Versione	Data	Autore/i	Descrizione	Verificatore
0.3	2026-01-30	Davide Biasuzzi	Inserimento metriche di processo e di qualità sw dettagliate con codici, formule e soglie di accettazione, grafici, strategie di test	Francesco Zanella
0.2	2025-12-18	Davide Biasuzzi	Aggiunto paragrafo relativo alla qualità della documentazione	Samuele Perozzo
0.1	2025-12-05	Mihaela-Mariana Roma-scu	Stesura iniziale del documento, definizione metriche di processo e prodotto.	Giovanni Ponso

Indice

Tabella delle Versioni	2
Informazioni Generali	5
1 Introduzione	6
1.1 Scopo del documento	6
1.2 Glossario	6
2 Riferimenti	6
2.1 Riferimenti normativi	6
2.2 Riferimenti informativi	6
3 Qualità di Processo	7
3.1 Scopo e Obiettivi	7
3.2 Parametri del Progetto	7
3.3 Metriche di Processo	7
3.3.1 PM-1 - Planned Value	7
3.3.2 PM-2 - Actual Cost	8
3.3.3 PM-3 - Earned Value	8
3.3.4 PM-4 - Cost Variance	8
3.3.5 PM-5 - Schedule Variance	9
3.3.6 PM-6 - Varianza di Budget	9
3.3.7 PM-7 - Varianza dell'Impegno Orario	10
3.3.8 PM-8 - Cost Performance Index	10
3.3.9 PM-9 - Schedule Performance Index	10
3.3.10 PM-10 - Estimate to Complete	11
3.3.11 PM-11 - Estimate at Completion	11
3.3.12 PM-12 - Budget at Completion	11
3.3.13 PM-13 - Copertura Unit Test (CI/CD)	12
3.3.14 PM-14 - Misure di Mitigazione Insufficienti	12
3.3.15 PM-15 - Rischi Inattesi	12
3.3.16 Tabella Riepilogativa delle Metriche di Processo	13
4 Qualità di Prodotto	14
4.1 Metriche di Qualità del Software (Statica)	14
4.1.1 QM-1 - Complessità Ciclomantica	14
4.1.2 QM-2 - Profondità di Ereditarietà	14
4.1.3 QM-3 - Duplicazione del Codice	15
4.1.4 QM-4 - Code Churn	15
4.1.5 QM-5 - Debito Tecnico	15
4.2 Metriche di Qualità del Software (Dinamica)	16
4.2.1 QM-6 - Copertura Complessiva (Integration/System Test)	16
4.2.2 QM-7 - Copertura dei Branch	16
4.3 Strategie di Test per Requisito	17
4.3.1 Legenda Tipologie di Test	17
4.3.2 Requisiti Funzionali	17
4.3.3 Requisiti di Qualità	19
4.3.4 Requisiti di Vincolo	19
4.3.5 Requisiti Prestazionali	20

4.3.6	Requisiti di Sicurezza	20
5	Qualità della Documentazione	22
5.1	Metriche di Leggibilità e Correttezza Linguistica	22
5.1.1	Indice Gulpease	22
5.1.2	Controllo Grammaticale e Linguistico (LanguageTool)	22
5.1.3	Controllo Sintattico L ^A T _E X (ChkTeX)	22
5.2	Soglie di Accettazione e Target	23
5.3	Checklist di Qualità Interna	23
5.3.1	Coerenza Strutturale	23
5.3.2	Correttezza Ortografica e Grammaticale	24
5.3.3	Conformità alle Norme Grafiche	24
5.3.4	Qualità dei Requisiti	25
5.4	Automazione e Verifica Continua	26
6	Cruscotto di Valutazione della Qualità	27
6.1	Andamento del Valore e dei Costi	27
6.2	Indici di Performance (CPI e SPI)	28
6.3	Proiezioni di Costo: ETC, EAC e BAC	29
6.4	Varianza dell’Impegno Orario	30
6.5	Varianza di Budget	31
6.6	Requisiti Soddisfatti	31
6.7	Gestione dei Rischi	32
6.8	Indice di Gulpease della Documentazione	33
6.9	Errori di Qualità Linguistica	35

Informazioni Generali

Componenti del Gruppo

Cognome	Nome	Matricola
Biasuzzi	Davide	2111000
Bilato	Leonardo	2071084
Zanella	Francesco	2116442
Romascu	Mihaela-Mariana	2079726
Ogniben	Michele	2042325
Perozzo	Samuele	2110989
Ponso	Giovanni	2000558

Tabella 1: Componenti del gruppo NightPRO.

1 Introduzione

Il *Piano di Qualifica* rappresenta il documento strategico per la gestione della qualità all'interno del progetto **SmartOrder**. Esso stabilisce gli standard, le metodologie di verifica e le metriche quantitative necessarie per valutare sia la qualità del processo di sviluppo che la qualità del prodotto software finale.

1.1 Scopo del documento

Questo documento ha lo scopo di definire:

- Gli **obiettivi di qualità** che il gruppo NightPRO si impegna a raggiungere.
- Le **metriche** utilizzate per monitorare l'andamento del progetto (costi, tempi) e la qualità del codice.
- Le **procedure di verifica e validazione** da applicare ai documenti e al software.
- I criteri di accettazione per le fasi di avanzamento.

L'adozione di questo piano garantisce che il prodotto finale sia conforme ai requisiti espressi nel Capitolato C8 e rispetti gli standard accademici richiesti.

1.2 Glossario

Al fine di evitare ambiguità, i termini tecnici o specifici del dominio sono raccolti nel documento *Glossario*. Nel presente testo, tali termini sono contrassegnati con una "G" in pedice alla loro prima occorrenza (es. Agile_G).

2 Riferimenti

2.1 Riferimenti normativi

- [Norme di Progetto V1.0](#)
- [Capitolato d'appalto C8 - Smart Order](#)

2.2 Riferimenti informativi

- [Glossario](#)
- [Analisi dei Requisiti](#)
- [Verbali esterni](#)
- [Verbali interni](#)
- [Piano di progetto](#)

3 Qualità di Processo

3.1 Scopo e Obiettivi

La qualità di processo riguarda la capacità del gruppo di gestire le attività pianificate rispettando vincoli di budget e tempistiche. Il gruppo NightPRO adotta il ciclo di Deming_G, noto come PDCA_G:

- **Plan (Pianificare):** stabilire gli obiettivi e i processi necessari per fornire risultati in accordo con i risultati attesi.
- **Do (Fare):** attuare il piano, eseguire i processi, realizzare il prodotto.
- **Check (Verificare):** monitorare e misurare i processi e il prodotto a fronte delle politiche, degli obiettivi e dei requisiti.
- **Act (Agire):** adottare azioni per migliorare continuamente le prestazioni dei processi.

3.2 Parametri del Progetto

I seguenti valori costituiscono la baseline_G per il calcolo delle metriche:

Parametro	Descrizione	Valore
BAC (Budget At Completion)	Costo totale preventivato del progetto (v0.4)	€ 12.850,00
Ore Totali	Monte ore complessivo disponibile	630 ore
Componenti	Numero di membri del gruppo	7
Ore medie per componente	Media ore di lavoro pro capite (630/7)	90 ore
Scadenza Progetto	Data prevista per la consegna finale (PB _G)	21/03/2026

Tabella 2: Parametri fondamentali del progetto

3.3 Metriche di Processo

Per il monitoraggio dell'avanzamento, il gruppo utilizza la metodologia Earned Value Management_G integrata con metriche di gestione del rischio e della qualità.

3.3.1 PM-1 - Planned Value

Codice: PM-1

Processo: Fornitura

Formula:

Dove:

- AC_{k-1} = Actual Cost cumulativo fino allo sprint_G precedente
- BP_k = Budget Preventivato per lo sprint corrente

$$PV = AC_{k-1} + BP_k$$

Descrizione: Indica il valore cumulativo pianificato del lavoro che dovrebbe essere completato fino allo sprint corrente. Viene calcolato sommando i costi effettivi pregressi al budget pianificato dello sprint in corso.

Valori di accettazione:**Preferibile:** -**Accettabile:** -**3.3.2 PM-2 - Actual Cost****Codice:** PM-2**Processo:** Fornitura**Formula:***Dove:*

- BC_i = Budget Consuntivato dello sprint i -esimo
- n = Numero totale di sprint completati fino alla data corrente

$$AC = \sum_{i=1}^n BC_i$$

Descrizione: Rappresenta la somma cumulativa dei costi realmente sostenuti dall'inizio del progetto fino al momento della misurazione. Include tutti i costi effettivi degli sprint completati.

Valori di accettazione:**Preferibile:** -**Accettabile:** -**3.3.3 PM-3 - Earned Value****Codice:** PM-3**Processo:** Fornitura**Formula:***Dove:*

- BP = Budget Preventivato per lo sprint
- $\%Comp$ = Percentuale di completamento delle attività pianificate

$$EV = BP \cdot \frac{\%Comp}{100}$$

Descrizione: Quantifica il valore monetario del lavoro effettivamente completato al termine dello sprint. Rappresenta quanto del budget pianificato è stato "guadagnato" attraverso il completamento delle attività. Viene confrontato con l'Actual Cost per valutare l'efficienza economica.

Valori di accettazione:**Preferibile:** -**Accettabile:** \geq PM-2**3.3.4 PM-4 - Cost Variance****Codice:** PM-4**Processo:** Fornitura**Formula:***Dove:*

- EV = Earned Value (valore del lavoro completato)
- AC = Actual Cost (costo effettivamente sostenuto)

$$CV = EV - AC$$

Descrizione: Misura la differenza tra il valore prodotto e il costo sostenuto. Un valore positivo segnala efficienza economica (si è speso meno del valore prodotto), mentre un valore negativo indica inefficienza (si è speso più del valore generato).

Valori di accettazione:

Preferibile: 0

Accettabile: ± 150

3.3.5 PM-5 - Schedule Variance

Codice: PM-5

Processo: Fornitura

Formula:

Dove:

- EV = Earned Value (valore del lavoro completato)
- PV = Planned Value (valore del lavoro pianificato)

$$SV = EV - PV$$

Descrizione: Valuta lo scostamento temporale confrontando il lavoro effettivamente completato con quello pianificato. Valori positivi indicano anticipo sulla pianificazione, valori negativi segnalano ritardi rispetto alla schedulazione prevista.

Valori di accettazione:

Preferibile: 0

Accettabile: ± 150

3.3.6 PM-6 - Varianza di Budget

Codice: PM-6

Processo: Fornitura

Formula:

Dove:

- BC = Budget Consuntivato (costo effettivo dello sprint)
- BP = Budget Preventivato (costo pianificato dello sprint)

$$\text{Varianza Budget (\%)} = 100 \cdot \frac{BC - BP}{BP}$$

Descrizione: Misura lo scostamento percentuale tra il costo effettivamente sostenuto e quello pianificato per uno sprint. Valori positivi indicano uno sfioramento del budget, mentre valori negativi segnalano risparmi rispetto alla pianificazione iniziale.

Valori di accettazione:

Preferibile: $\pm 0\%$

Accettabile: $\pm 5\%$

3.3.7 PM-7 - Varianza dell'Impegno Orario

Codice: PM-7
Processo: Fornitura

Formula:

Dove:

- OC = Ore Consuntivate (ore effettivamente lavorate)
- OP = Ore Preventivate (ore pianificate per lo sprint)

$$\text{Varianza Oraria (\%)} = 100 \cdot \frac{OC - OP}{OP}$$

Descrizione: Calcola la deviazione percentuale tra le ore di lavoro effettivamente impiegate e quelle stimate per uno sprint. Un valore positivo indica un impiego di tempo superiore al previsto, mentre un valore negativo indica maggiore efficienza.

Valori di accettazione:

Preferibile: $\pm 0\%$
Accettabile: $\pm 10\%$

3.3.8 PM-8 - Cost Performance Index

Codice: PM-8
Processo: Fornitura

Formula:

Dove:

- EV = Earned Value (valore del lavoro completato)
- AC = Actual Cost (costo effettivamente sostenuto)

$$CPI = \frac{EV}{AC}$$

Descrizione: Indice di performance economica che esprime l'efficienza nell'utilizzo delle risorse finanziarie. Un valore pari a 1 indica perfetto allineamento, valori maggiori di 1 indicano efficienza superiore al previsto, valori inferiori a 1 segnalano inefficienza.

Valori di accettazione:

Preferibile: 1
Accettabile: 1 ± 0.15

3.3.9 PM-9 - Schedule Performance Index

Codice: PM-9
Processo: Fornitura

Formula:

Dove:

- EV = Earned Value (valore del lavoro completato)
- PV = Planned Value (valore del lavoro pianificato)

$$SPI = \frac{EV}{PV}$$

Descrizione: Indice di performance temporale che misura l'efficienza rispetto alla pianificazione. Valori pari a 1 indicano rispetto della pianificazione, valori maggiori di 1 segnalano avanzamento in anticipo, valori inferiori a 1 indicano ritardi.

Valori di accettazione:**Preferibile:** 1**Accettabile:** 1 ± 0.15 **3.3.10 PM-10 - Estimate to Complete****Codice:** PM-10**Processo:** Fornitura**Formula:***Dove:*

- BAC = Budget at Completion (budget totale del progetto)
- EV = Earned Value (valore del lavoro già completato)
- CPI = Cost Performance Index (indice di efficienza economica)

$$ETC = \frac{BAC - EV}{CPI}$$

Descrizione: Stima il costo necessario per completare le attività rimanenti del progetto, considerando l'efficienza economica dimostrata fino al momento corrente. Tiene conto delle performance passate per proiettare i costi futuri.

Valori di accettazione:**Preferibile:** -**Accettabile:** -**3.3.11 PM-11 - Estimate at Completion****Codice:** PM-11**Processo:** Fornitura**Formula:***Dove:*

- AC = Actual Cost (costo già sostenuto)
- ETC = Estimate to Complete (stima del costo rimanente)

$$EAC = AC + ETC$$

Descrizione: Proiezione del costo totale finale del progetto basata sull'andamento corrente. Combina i costi già sostenuti con la stima dei costi futuri per fornire una previsione realistica del budget complessivo necessario.

Valori di accettazione:**Preferibile:** PM-12**Accettabile:** $\pm 5\%$ di PM-12**3.3.12 PM-12 - Budget at Completion****Codice:** PM-12**Processo:** Fornitura

Descrizione: Rappresenta il budget complessivo originariamente allocato per l'intero progetto. Costituisce il valore di riferimento rispetto al quale vengono valutate tutte le metriche di costo e le proiezioni di spesa finale.

Valori di accettazione:

Preferibile: -

Accettabile: -

3.3.13 PM-13 - Copertura Unit Test (CI/CD)

Codice: PM-13

Processo: Sviluppo

Descrizione: Misura la copertura del codice raggiunta esclusivamente dagli unit test_G eseguiti automaticamente nella pipeline_G CI/CD_G. Questa metrica monitora la qualità del processo di sviluppo e garantisce che ogni modifica al codice sia adeguatamente coperta da test unitari prima del merge. Le soglie sono più stringenti rispetto alla copertura generale poiché gli unit test rappresentano la prima linea di difesa contro i difetti.

Valori di accettazione:

Preferibile: $\geq 90\%$

Accettabile: $\geq 80\%$

3.3.14 PM-14 - Misure di Mitigazione Insufficienti

Codice: PM-14

Processo: Risoluzione dei problemi

Descrizione: Conteggia le strategie di mitigazione_G dei rischi che, una volta implementate, non hanno prodotto l'effetto preventivo atteso. Monitora l'efficacia della gestione proattiva dei rischi identificati.

Valori di accettazione:

Preferibile: 0

Accettabile: 3

3.3.15 PM-15 - Rischi Inattesi

Codice: PM-15

Processo: Risoluzione dei problemi

Descrizione: Traccia il numero di eventi problematici non previsti durante la fase di analisi dei rischi. Un valore elevato segnala la necessità di migliorare il processo di identificazione e valutazione dei rischi.

Valori di accettazione:

Preferibile: 0

Accettabile: 3

3.3.16 Tabella Riepilogativa delle Metriche di Processo

Codice	Nome	Accettabile	Preferibile
PM-1	Planned Value	-	-
PM-2	Actual Cost	-	-
PM-3	Earned Value	\geq PM-2	-
PM-4	Cost Variance	± 150	0
PM-5	Schedule Variance	± 150	0
PM-6	Varianza di Budget	$\pm 5\%$	$\pm 0\%$
PM-7	Varianza dell'impegno orario	$\pm 10\%$	$\pm 0\%$
PM-8	Cost Performance Index	1 ± 0.15	1
PM-9	Schedule Performance Index	1 ± 0.15	1
PM-10	Estimate to Complete	-	-
PM-11	Estimate at Completion	$\pm 5\%$ di PM-12	PM-12
PM-12	Budget at Completion	-	-
PM-13	Copertura Unit Test (CI/CD)	$\geq 80\%$	$\geq 90\%$
PM-14	Misure di mitigazione insufficienti	3	0
PM-15	Rischi inattesi	3	0

Tabella 3: Metriche di qualità di processo

4 Qualità di Prodotto

Il gruppo NightPRO fa riferimento allo standard **ISO/IEC 25010_G** per definire le caratteristiche di qualità del prodotto software (es. Manutenibilità, Usabilità, Efficienza, Sicurezza).

***Nota:** Le metriche di qualità di prodotto presentate in questa sezione fanno riferimento allo sviluppo del **Minimum Viable Product (MVP)_G** e non al **Proof of Concept (PoC)_G**. Tali metriche sono state definite in previsione del futuro sviluppo dell'MVP, al fine di stabilire fin da subito criteri oggettivi e misurabili per garantire la qualità del software finale.*

4.1 Metriche di Qualità del Software (Statica)

Queste metriche mirano a garantire la manutenibilità e la qualità del codice sorgente prodotto.

4.1.1 QM-1 - Complessità Ciclomatica

Codice: QM-1
Processo: Sviluppo

Formula:

Dove:

- E = Numero di archi nel grafo di controllo di flusso
- N = Numero di nodi nel grafo di controllo di flusso
- P = Numero di componenti connesse (tipicamente 1)

$$CX = E - N + 2P$$

Descrizione: La complessità ciclomatica misura il numero di cammini linearmente indipendenti attraverso il codice sorgente di un programma. Un valore elevato indica codice complesso e difficile da testare, mantenere e comprendere. Questa metrica è calcolata per ogni singolo metodo/funzione.

Valori di accettazione:

Preferibile: ≤ 10

Accettabile: ≤ 15

4.1.2 QM-2 - Profondità di Ereditarietà

Codice: QM-2
Processo: Sviluppo

Formula:

$$DI = \max(\text{livelli di ereditarietà dalla classe radice})$$

Descrizione: Indica la profondità massima dell'albero di ereditarietà per una classe. Una gerarchia troppo profonda rende il codice difficile da comprendere e mantenere, poiché il comportamento di una classe dipende da molte classi antenate. Questa metrica aiuta a identificare gerarchie di classi eccessivamente complesse.

Valori di accettazione:

Preferibile: ≤ 4

Accettabile: ≤ 6

4.1.3 QM-3 - Duplicazione del Codice

Codice: QM-3
Processo: Sviluppo

Formula:

Dove:

- L_d = Linee di codice duplicate
- L_t = Linee totali di codice

$$CD = \frac{L_d}{L_t} \cdot 100\%$$

Descrizione: Percentuale di codice duplicato rispetto al totale del codice sorgente. La presenza di codice duplicato aumenta i costi di manutenzione, poiché le modifiche devono essere replicate in più punti, e aumenta il rischio di inconsistenze. Indica la necessità di refactoring per estrarre funzionalità comuni.

Valori di accettazione:

Preferibile: 0%
Accettabile: $\leq 5\%$

4.1.4 QM-4 - Code Churn

Codice: QM-4
Processo: Sviluppo

Formula:

Dove:

- L_a = Linee di codice aggiunte nel periodo
- L_m = Linee di codice modificate nel periodo
- L_c = Linee di codice cancellate nel periodo

$$CCh = L_a + L_m + L_c$$

Descrizione: Misura la volatilità del codice, ovvero la frequenza e il volume delle modifiche apportate al codice sorgente in un determinato periodo (tipicamente una settimana o uno sprint). Un code churn elevato può indicare instabilità del codice, requisiti poco chiari o necessità di refactoring. Valori costantemente elevati possono segnalare problemi di progettazione.

Valori di accettazione:

Preferibile: Basso (< 200 linee/settimana)
Accettabile: Medio (< 500 linee/settimana)

4.1.5 QM-5 - Debito Tecnico

Codice: QM-5
Processo: Sviluppo

Formula:

$$TD = \sum_{i=1}^n T_i$$

Dove:

- T_i = Tempo stimato per risolvere l'issue i -esimo di qualità del codice

- n = Numero totale di issue di qualità rilevate

Descrizione: Quantifica il tempo stimato necessario per correggere tutti i problemi di qualità del codice identificati da strumenti di analisi statica (es. code smell_G, bug potenziali, vulnerabilità di sicurezza). Un debito tecnico elevato indica che il codice richiede interventi di refactoring significativi e può compromettere la velocità di sviluppo futura.

Valori di accettazione:

Preferibile: ≤ 2 giorni

Accettabile: ≤ 5 giorni

4.2 Metriche di Qualità del Software (Dinamica)

Queste metriche valutano l'efficacia dei test e la qualità del software durante l'esecuzione.

4.2.1 QM-6 - Copertura Complessiva (Integration/System Test)

Codice: QM-6

Processo: Validazione

Formula:

Dove:

- L_e = Linee di codice eseguite durante i test di integrazione e sistema
- L_t = Linee totali di codice testabile

$$CC = \frac{L_e}{L_t} \cdot 100\%$$

Descrizione: Percentuale di codice sorgente eseguita durante i test di integrazione e di sistema. Questa metrica valuta la copertura raggiunta dai test che verificano l'interazione tra componenti e il comportamento end-to-end dell'applicazione. Le soglie sono meno stringenti rispetto alla copertura degli unit test (PM-13) poiché i test di integrazione/sistema coprono scenari più complessi e alcuni percorsi di codice possono essere difficilmente raggiungibili.

Valori di accettazione:

Preferibile: $\geq 75\%$

Accettabile: $\geq 60\%$

4.2.2 QM-7 - Copertura dei Branch

Codice: QM-7

Processo: Validazione

Formula:

Dove:

- B_e = Numero di branch eseguiti durante i test
- B_t = Numero totale di branch nel codice

$$BC = \frac{B_e}{B_t} \cdot 100\%$$

Descrizione: Percentuale di branch decisionali (if, switch, loop) del codice che sono stati eseguiti almeno una volta durante i test. La copertura dei branch è più rigorosa della semplice copertura delle linee, poiché garantisce che tutte le possibili ramificazioni del flusso di controllo siano state testate, riducendo il rischio di bug in percorsi di codice raramente eseguiti.

Valori di accettazione:

Preferibile: $\geq 75\%$

Accettabile: $\geq 60\%$

4.3 Strategie di Test per Requisito

Questa sezione definisce la strategia di verifica e validazione per ciascun requisito specificato nell'*Analisi dei Requisiti*. Per ogni requisito viene indicato il tipo di test applicabile e la tecnica di verifica adottata, garantendo tracciabilità completa tra requisiti e attività di testing_G.

4.3.1 Legenda Tipologie di Test

- **UT** - Unit Test: test su singole unità di codice (funzioni, metodi, classi).
- **IT** - Integration Test: test di integrazione tra componenti/moduli.
- **ST** - System Test: test dell'intero sistema end-to-end.
- **UAT** - User Acceptance Test: test di accettazione con utenti reali.
- **PT** - Performance Test: test di carico e prestazioni.
- **SEC** - Security Test: test di sicurezza e vulnerabilità.
- **REV** - Code Review: revisione del codice e della configurazione.

4.3.2 Requisiti Funzionali

ID Req.	Requisito	Tipo Test	Strategia di Verifica
Gestione Input Multimodale			
RF1.1	Acquisizione testo	UT, IT, ST	Verifica ricezione messaggi da interfaccia chat. Test di integrazione con il backend.
RF1.2	Acquisizione audio	UT, IT, ST	Test upload file con formati diversi (MP3, WAV, OGG) e sample rate variabili. Verifica gestione file corrotti.
RF1.3	Validazione format input	UT	Test con input validi/invalidi, dimensioni fuori range, formati non supportati. Verifica messaggi di errore.
RF1.4	Normalizzazione dati grezzi	UT	Test su testi con rumore, caratteri speciali, formattazioni diverse. Verifica output normalizzato.
Estrazione e Riconoscimento Entità			
RF2.1	NER (Named Entity Recognition)	UT, IT	Test con frasi contenenti entità note/sconosciute. Verifica accuratezza identificazione.
RF2.2	Speech-to-text	UT, IT, PT	Test con audio di qualità variabile, accenti regionali. Verifica accuracy $\geq 90\%$. Benchmark prestazioni.

ID Req.	Requisito	Tipo Test	Strategia di Verifica
RF2.3	Tokenizzazione e normalizzazione testuale	UT	Test con test diversi. Verifica tokenizzazione, lowercasing, rimozione simboli.
Elaborazione Semantica e Mapping			
RF3.1	Generazione embedding semantici	UT, IT	Test generazione embedding per testi campione. Verifica dimensionalità e consistenza.
RF3.2	Generazione embedding audio	UT, IT	Test su audio trascritti. Verifica coerenza embedding audio/testo.
RF3.3	Mapping catalogo prodotti	UT, IT, ST	Test con descrizioni esatte/parziali/ambigue. Verifica confidence score e correttezza mapping.
RF3.4	Completamento informazioni	UT, IT	Test con ordini incompleti. Verifica completamento basato su regole aziendali.
Validazione e Arricchimento Dati			
RF4.1	Verifica integrità ordine	UT, IT	Test con ordini completi/incompleti. Verifica identificazione campi obbligatori mancanti.
RF4.2	Applicazione regole aziendali	UT, IT	Test quantità minime, disponibilità magazzino. Verifica applicazione regole configurabili.
RF4.3	Validazione coerenza	UT	Test coerenza quantità/unità misura, prodotto/categoria. Verifica rilevamento incoerenze.
RF4.4	Arricchimento metadati	UT, IT	Verifica presenza di tutti i metadati richiesti (timestamp, canale, confidence score).
RF4.5	Handling ambiguità	UT, IT, ST	Test con input ambigui. Verifica segnalazione confidence basso e routing a validazione manuale.
Strutturazione Output e Integrazione Database			
RF5.1	Strutturazione JSON	UT, IT	Verifica output JSON valido e conforme allo schema definito. Test parser JSON.
RF5.2	Strutturazione XML	UT, IT	Verifica output XML valido e conforme allo schema XSD. Test compatibilità con sistemi ERP.
RF5.3	Tracciamento metadati	UT, IT, ST	Verifica tracciabilità completa metadati per ogni ordine. Test query su metadati.
Supervisione e Gestione Manuale			
RF6.1	Interfaccia supervisione	ST, UAT	Test funzionalità dashboard. Verifica accessibilità e usabilità interfaccia.
RF6.2	Visualizzazione dettagli ordine	ST, UAT	Verifica visualizzazione completa dei dati. Test con ordini diversi.

ID Req.	Requisito	Tipo Test	Strategia di Verifica
RF6.3	Correzione manuale	IT, ST, UAT	Test modifica campi ordine. Verifica salvataggio e propagazione correzioni.
RF6.4	Approvazione/Rifiuto	IT, ST, UAT	Test flusso approvazione/rifiuto. Verifica cambio stato ordine e notifiche.
RF6.5	Feedback per retraining	IT, ST	Verifica registrazione correzioni. Test recupero dati per retraining.
RF6.6	Visualizzazione storico	ST, UAT	Test filtri (data, canale, cliente). Verifica performance con grande volume dati.
Monitoraggio e Feedback Continuo			
RF7.1	Logging dettagliato	REV, IT	Verifica logging completo operazioni. Test formato log e persistenza.
RF7.2	Monitoraggio prestazioni	IT, PT	Verifica tracciamento metriche. Test dashboard monitoraggio in tempo reale.
RF7.3	Meccanismo feedback	IT, ST	Test ciclo feedback operatore-sistema. Verifica registrazione e processamento feedback.
RF7.4	Retraining periodico	IT, ST	Test pipeline retraining. Verifica aggiornamento modelli con nuovi dati.
RF7.5	Aggiornamento regole	IT, ST, REV	Test modifica regole runtime. Verifica applicazione nuove regole senza riavvio.

Tabella 4: Strategie di test per i requisiti funzionali

4.3.3 Requisiti di Qualità

ID Req.	Requisito	Tipo Test	Strategia di Verifica
RQ01	Messaggi di errore chiari	ST, UAT	Test con input invalidi diversi. Verifica chiarezza e completezza messaggi.
RQ02	Estensibilità nuovi canali	REV, IT	Code review architettura. Test aggiunta mock di nuovo canale input.

Tabella 5: Strategie di test per i requisiti di qualità

4.3.4 Requisiti di Vincolo

ID Req.	Requisito	Tipo Test	Strategia di Verifica
RV01	Database relazionale	REV, IT	Verifica configurazione DB. Test connettività e operazioni CRUD.

ID Req.	Requisito	Tipo Test	Strategia di Verifica
RV02	API REST standard	REV, IT, ST	Code review API. Test conformità standard REST. Verifica documentazione OpenAPI.
RV03	Webapp responsive	ST, UAT	Test su dispositivi con risoluzioni diverse (desktop, tablet, mobile).
RV04	Compatibilità browser	ST	Test funzionalità su Chrome, Firefox, Safari, Edge. Verifica compatibilità CSS/JS.
RV05	Addestramento locale/cloud	IT, ST	Test pipeline addestramento in entrambi gli ambienti. Verifica portabilità.

Tabella 6: Strategie di test per i requisiti di vincolo

4.3.5 Requisiti Prestazionali

ID Req.	Requisito	Tipo Test	Strategia di Verifica
RP01	Tempo caricamento dashboard	PT, ST	Test caricamento con cronometraggio. Verifica tempo ≤ 2 secondi.
RP02	Elaborazione ordine testuale	PT, ST	Test processing ordini. Verifica tempo medio ≤ 10 secondi.
RP03	Trascrizione audio 1 min	PT, ST	Benchmark trascrizione audio. Verifica tempo ≤ 20 secondi.
RP04	Throughput 50 ordini/ora	PT	Test carico con 50+ ordini/ora. Verifica assenza degrado prestazioni.

Tabella 7: Strategie di test per i requisiti prestazionali

4.3.6 Requisiti di Sicurezza

ID Req.	Requisito	Tipo Test	Strategia di Verifica
RS01	Autenticazione credenziali	SEC, ST	Test login con credenziali valide/invalidi. Verifica gestione sessioni.
RS02	RBAC (controllo accessi)	SEC, ST	Test accesso risorse con ruoli diversi. Verifica enforcement permessi.
RS03	Comunicazioni HTTPS	SEC, REV	Verifica certificato SSL/TLS. Test intercettazione traffico.
RS04	Audit log operazioni critiche	IT, ST	Verifica tracciamento operazioni. Test immutabilità log.

ID Req.	Requisito	Tipo Test	Strategia di Verifica
RS05	Complessità password e hashing	SEC, UT	Test policy password. Verifica algoritmo hashing (bcrypt/Argon2). Test dizionario password deboli.

Tabella 8: Strategie di test per i requisiti di sicurezza

5 Qualità della Documentazione

La documentazione accompagna il prodotto software e deve garantire accessibilità e comprensione a tutti gli stakeholder_G. La qualità della documentazione è verificata attraverso un insieme coordinato di strumenti automatizzati che controllano la leggibilità linguistica, la correttezza grammaticale e la sintassi \LaTeX .

5.1 Metriche di Leggibilità e Correttezza Linguistica

Per valutare la qualità dei documenti redatti in italiano, il gruppo utilizza un insieme di indici automatizzati, ognuno specifico per un aspetto della qualità documentale.

5.1.1 Indice Gulease

L'indice Gulease_G è calibrato specificamente per la lingua italiana e valuta la leggibilità di un testo basandosi sulla lunghezza delle parole e delle frasi. Un indice elevato indica un testo più facilmente comprensibile.

$$G = 89 + \frac{300 \cdot (\text{numero frasi}) - 10 \cdot (\text{numero lettere})}{\text{numero parole}}$$

L'indice viene calcolato automaticamente su tutti i file `.tex` presenti nel repository_G, escludendo da tale calcolo i comandi \LaTeX _G, le formule matematiche e i commenti. La verifica è eseguita mediante lo script `check_gulease.py`, che genera un report CSV contenente il valore Gulease per ogni documento.

5.1.2 Controllo Grammaticale e Linguistico (LanguageTool)

LanguageTool è uno strumento di verifica grammaticale che identifica errori ortografici, grammaticali e stilistici nei testi in italiano. La verifica automatica:

- Rileva errori di ortografia, concordanza verbale e nominale, punteggiatura;
- Segnala problemi di stile e ridondanze lessicali;
- Esclude dal controllo i nomi propri del gruppo, i termini tecnici presenti nel glossario e gli identificativi tecnici (branch, commit, tool name, ecc.).

Lo script `check_languagetool.py` esegue questa verifica e genera un report Markdown dettagliato segnalando ogni errore rilevante con il numero della regola violata, il tipo di errore, il messaggio esplicativo e i suggerimenti di correzione.

5.1.3 Controllo Sintattico \LaTeX (ChkTeX)

ChkTeX è un linter_G specifico per file \LaTeX che verifica la correttezza sintattica e stilistica del codice sorgente. Controlla:

- Correttezza dei comandi \LaTeX ;
- Bilanciamento delle parentesi e delle graffe;
- Uso corretto degli spazi prima e dopo i comandi;
- Altre convenzioni stilistiche di buona pratica \LaTeX .

Lo script `check_chktex.py` esegue ChkTeX su tutti i file `.tex` e filtra alcuni warning innocui (ad esempio riferimenti a comandi personalizzati o numeri di versione non seguiti da unità). Il report viene salvato in formato JSON.

5.2 Soglie di Accettazione e Target

Le seguenti tabelle definiscono i target di qualità e le soglie di accettazione per le metriche documentali:

Metrica	Descrizione	Ottimale	Accettabile
Gulpease	Indice di leggibilità per testi in italiano	≥ 60	≥ 45

Tabella 9: Soglie per l'indice Gulpease

Indicatore	Descrizione	Target (PB)
Errori LanguageTool	Numero massimo di segnalazioni rilevanti per file	≤ 10
Errori ChkTeX	Numero massimo di warning significativi per file	≤ 5

Tabella 10: Soglie per gli errori linguistici e \LaTeX

5.3 Checklist di Qualità Interna

Oltre alle verifiche automatizzate, i Verificatori del gruppo NightPRO applicano checklist manuali per garantire la coerenza formale e la qualità complessiva dei documenti. Queste checklist coprono aspetti che richiedono valutazione umana e sensibilità al contesto.

5.3.1 Coerenza Strutturale

La seguente checklist verifica la corretta organizzazione e struttura dei documenti \LaTeX :

Elemento	Criterio di Verifica
Suddivisione in paragrafi	Le frasi devono essere organizzate in paragrafi coerenti, evitando blocchi di testo troppo lunghi che compromettono la leggibilità.
Ordinamento alfabetico	Gli elementi negli elenchi devono seguire l'ordine alfabetico quando non esiste una sequenza logica o temporale specifica.
Didascalie complete	Ogni tabella e figura deve essere accompagnata da una didascalia esplicativa mediante il comando <code>\caption{}</code> .
Sezioni non vuote	Tutte le sezioni e sottosezioni devono contenere testo o elementi. Non sono ammesse intestazioni senza contenuto.
Unicità dei file	Ogni documento deve essere definito in un file <code>.tex</code> dedicato, collegato al documento principale tramite <code>\input</code> per favorire la modularità.

Tabella 11: Checklist coerenza strutturale

5.3.2 Correttezza Ortografica e Grammaticale

Questa checklist integra i controlli automatici di LanguageTool con la verifica manuale di errori ricorrenti:

Elemento	Criterio di Verifica
Accenti gravi e acuti	Utilizzare l'accento grave (`) per "è", "più", "già" e l'accento acuto (´) per "perché", "poiché". L'accento sbagliato costituisce un errore ortografico.
Articolo eufonica	L'articolo "d" eufonica (es. "ad", "od") si utilizza solo quando la parola successiva inizia con la stessa vocale dell'articolo, evitando usi non necessari.
Concordanza soggetto- verbo	Il verbo deve concordare in numero e persona con il soggetto della frase. Verificare con particolare attenzione le frasi complesse.
Errori di digitazione	La maggior parte degli errori ortografici deriva da sviste di battitura. Effettuare una rilettura attenta dopo ogni modifica significativa.
Forma verbale	Privilegiare il presente indicativo nella stesura tecnica. Altre forme verbali (condizionale, congiuntivo) devono essere valutate caso per caso.
Forme impersonali	Nelle descrizioni formali, il soggetto deve essere sempre esplicito nella frase, evitando costruzioni impersonali ambigue.

Tabella 12: Checklist correttezza ortografica e grammaticale

5.3.3 Conformità alle Norme Grafiche

La seguente checklist assicura il rispetto delle convenzioni grafiche e tipografiche del gruppo:

Elemento	Criterio di Verifica
Punteggiatura negli elenchi	Ogni elemento di un elenco deve terminare con punto e virgola ";", ad eccezione dell'ultimo che termina con punto ".".
Utilizzo del grassetto	Il grassetto si utilizza per evidenziare termini chiave alla loro prima introduzione o per enfatizzare concetti fondamentali. Evitare un uso eccessivo.
Maiuscole nei ruoli	I ruoli di progetto devono essere scritti con la lettera iniziale maiuscola quando si riferiscono a membri specifici del gruppo.
Lettera iniziale dei titoli	Solo la prima parola dei titoli di sezione deve avere l'iniziale maiuscola, salvo nomi propri o acronimi.
Segnalazione termini glossario	Quando un termine del Glossario viene introdotto per la prima volta in un documento, deve essere marcato con il pedice G.
Aggiornamento registro modifiche	Dopo ogni verifica, il registro delle modifiche (changelog) del documento deve essere aggiornato con la descrizione delle correzioni apportate.
Indicazione versione	Quando si cita un documento, è necessario specificarne la versione. Se non si indica alcuna versione, si intende la versione più recente disponibile.

Tabella 13: Checklist conformità norme grafiche

5.3.4 Qualità dei Requisiti

Per i documenti di analisi, viene applicata una checklist specifica per garantire la qualità dei requisiti:

Elemento	Criterio di Verifica
Tracciamento UC-Requisiti	Ogni caso d'uso deve essere collegato ad almeno un requisito specifico che ne deriva, garantendo la tracciabilità bidirezionale.
Codifica Use Case	La numerazione degli Use Case deve rispettare la gerarchia: gli UC derivati devono avere lo stesso livello gerarchico del caso principale da cui dipendono.
Formulazione requisiti	I requisiti devono essere espressi nella forma "[soggetto] deve [verbo all'infinito] [complemento]" per garantire chiarezza e verificabilità.
Diagrammi UML completi	Estensioni, inclusioni e specializzazioni di un caso d'uso devono essere rappresentate all'interno del medesimo diagramma UML del caso principale per facilitare la comprensione.

Tabella 14: Checklist qualità dei requisiti

5.4 Automazione e Verifica Continua

Le verifiche di qualità della documentazione sono automatizzate e integrate nel processo di CI/CD tramite il workflow GitHub `quality_checks.yml`. Questo workflow:

1. Si attiva automaticamente dopo il completamento della compilazione $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ (workflow *Build LaTeX documents*);
2. Può essere eseguito manualmente in qualsiasi momento tramite *workflow dispatch*;
3. Esegue in sequenza i tre script di verifica (Gulpease, LanguageTool, ChkTeX);
4. Genera un report Markdown consolidato accessibile come artifact;
5. Commenta automaticamente i pull request con i risultati della verifica.

La verifica è non-bloccante per i build, ovvero gli errori rilevati non impediscono il progresso della pipeline, tuttavia vengono tracciati e riportati nel report per consentire al gruppo di monitorare gli standard di qualità e di correggere sistematicamente i problemi identificati.

6 Cruscotto di Valutazione della Qualità

Questa sezione presenta un'analisi quantitativa dell'andamento del progetto attraverso grafici e metriche derivate dai dati raccolti durante gli otto sprint completati fino alla data corrente. L'obiettivo è fornire una visione d'insieme dello stato di salute del progetto in termini di costi, tempi e qualità della documentazione prodotta.

6.1 Andamento del Valore e dei Costi

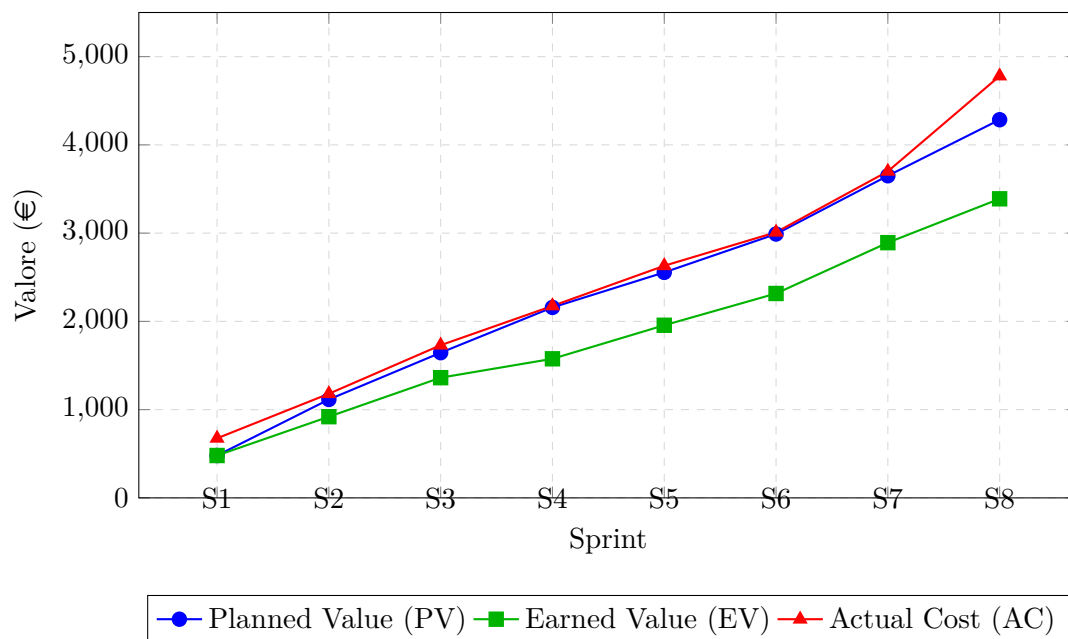


Figura 1: Andamento cumulativo di Planned Value, Earned Value e Actual Cost

Analisi

Il grafico mostra l'andamento cumulativo delle tre metriche fondamentali dell'Earned Value Management. Il **Planned Value (PV)** rappresenta il valore pianificato del lavoro che avrebbe dovuto essere completato, mentre l'**Earned Value (EV)** indica il valore effettivamente guadagnato attraverso il completamento delle attività. L'**Actual Cost (AC)** rappresenta i costi realmente sostenuti.

Si osserva che l'AC supera costantemente sia il PV che l'EV, indicando un impiego di risorse superiore al previsto. Questo andamento è riconducibile alla natura della fase RTB, caratterizzata da attività ad alto investimento formativo: il team ha dovuto acquisire competenze su tecnologie complesse (LLM, architetture RAG, containerizzazione Docker) e affrontare le sfide dello sviluppo del Proof of Concept. Lo sprint 8 mostra un incremento significativo della forbice tra AC e le altre metriche, corrispondente all'intensificazione delle attività di sviluppo del PoC.

Con il passaggio alla fase di Product Baseline, dove le conoscenze acquisite potranno essere applicate in modo più efficiente, il gruppo prevede un progressivo riallineamento delle curve, grazie anche alle politiche di ottimizzazione pianificate.

6.2 Indici di Performance (CPI e SPI)

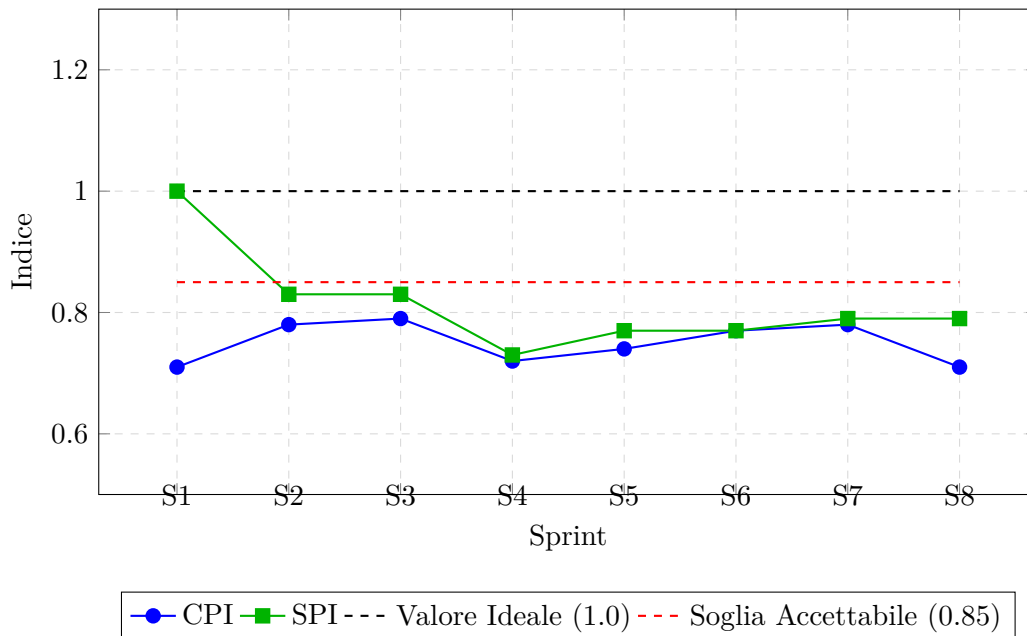


Figura 2: Andamento degli indici Cost Performance Index e Schedule Performance Index

Analisi

Il grafico evidenzia l'andamento degli indici di performance economica (CPI) e temporale (SPI). Il **CPI** si è mantenuto costantemente al di sotto della soglia di accettabilità (0.85), oscillando tra 0.71 e 0.79, indicando che per ogni euro di valore prodotto il team sta spendendo circa 1.30€. Lo **SPI** mostra un andamento variabile: partito da un valore ottimale di 1.0 nel primo sprint, è sceso progressivamente fino a stabilizzarsi intorno a 0.77-0.79.

È importante contestualizzare questi risultati: gli sprint analizzati corrispondono alla fase RTB_G del progetto, caratterizzata da un'intensa attività di apprendimento delle tecnologie richieste dal capitolato e dallo sviluppo esplorativo del Proof of Concept. La curva di apprendimento del team, unita alla complessità intrinseca delle tecnologie da padroneggiare (LLM_G, RAG_G, architetture a microservizi_G), ha comportato un impiego di risorse superiore a quanto inizialmente preventivato.

Per i prossimi sprint, il gruppo intende attuare le seguenti politiche correttive:

- Stime più conservative basate sull'esperienza acquisita durante la fase RTB;
- Allocazione più mirata delle risorse sui ruoli effettivamente necessari;
- Maggiore parallelizzazione delle attività per ottimizzare i tempi;
- Revisione periodica delle stime in corso d'opera.

6.3 Proiezioni di Costo: ETC, EAC e BAC

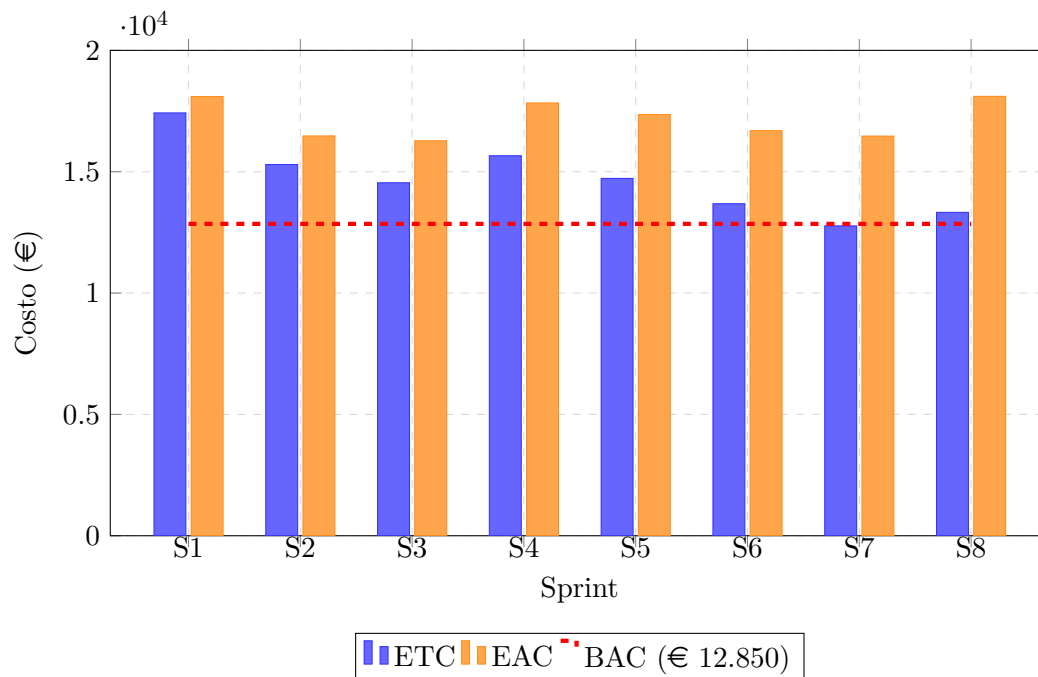


Figura 3: Proiezioni di costo: Estimate to Complete, Estimate at Completion e Budget at Completion

Analisi

Il grafico confronta le proiezioni di costo con il budget originale del progetto ($BAC = € 12.850$). L'**Estimate to Complete (ETC)** rappresenta la stima del costo necessario per completare le attività rimanenti, calcolata considerando l'efficienza economica dimostrata (CPI). L'**Estimate at Completion (EAC)** combina i costi già sostenuti (AC) con l'ETC per fornire una proiezione del costo finale totale.

I valori di EAC oscillano tra € 16.274 (S3) e € 18.105 (S8), superando il BAC di circa il 26-41%. Tuttavia, è fondamentale considerare che queste proiezioni si basano sull'efficienza della fase RTB, durante la quale il team ha affrontato una significativa curva di apprendimento: studio delle tecnologie del capitolato (LLM, architetture RAG, microservizi Docker_G), familiarizzazione con i processi di sviluppo agile e realizzazione del Proof of Concept.

Con il consolidamento delle competenze tecniche acquisite e l'avvio della fase di sviluppo effettivo (Product Baseline_G), il gruppo prevede un sensibile miglioramento dell'efficienza economica. Le politiche correttive pianificate includono:

- Riutilizzo delle conoscenze e dei componenti sviluppati durante il PoC;
- Preventivi più accurati basati sui dati storici raccolti;
- Ottimizzazione dell'allocazione oraria per ruolo;
- Monitoraggio settimanale degli scostamenti con interventi tempestivi.

L'obiettivo è riportare l'EAC entro margini accettabili rispetto al BAC nelle successive fasi del progetto.

6.4 Varianza dell'Impegno Orario

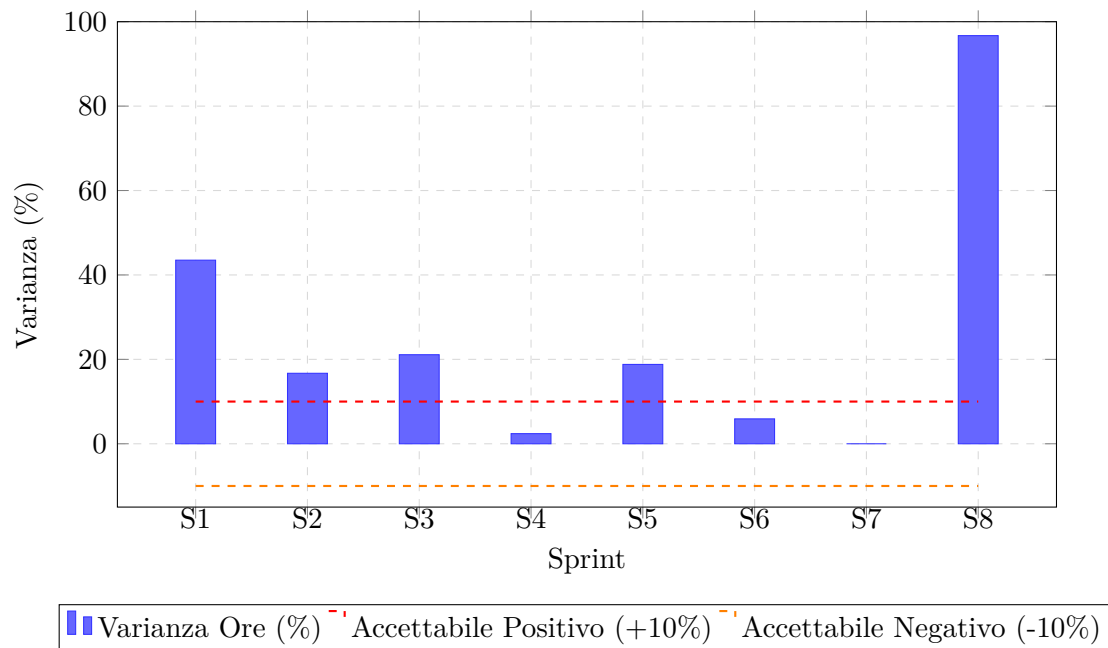


Figura 4: Varianza dell'impegno orario per sprint

Analisi

Analizzando i valori riportati nel grafico, è riscontrabile una difficoltà nel produrre preventivi orari vicini al consuntivato. Questo è dovuto alla scarsa esperienza del team nel preventivare il tempo necessario per lo svolgimento delle attività. Nel primo sprint la varianza è stata particolarmente elevata (+43,5%), riflettendo la fase di avvio del progetto e la necessità di familiarizzare con gli strumenti e i processi. Con il progredire degli sprint, il team ha acquisito maggiore esperienza nella preventivazione, riuscendo ad ottenere un valore perfettamente in linea nel settimo sprint (0,0%). Gli sprint successivi mostrano varianze contenute entro un range accettabile, indicando un miglioramento nella capacità di stima, ad eccezione dello sprint 8 che presenta una varianza critica del 96,7% dovuta alla complessità delle attività di sviluppo del PoC.

6.5 Varianza di Budget

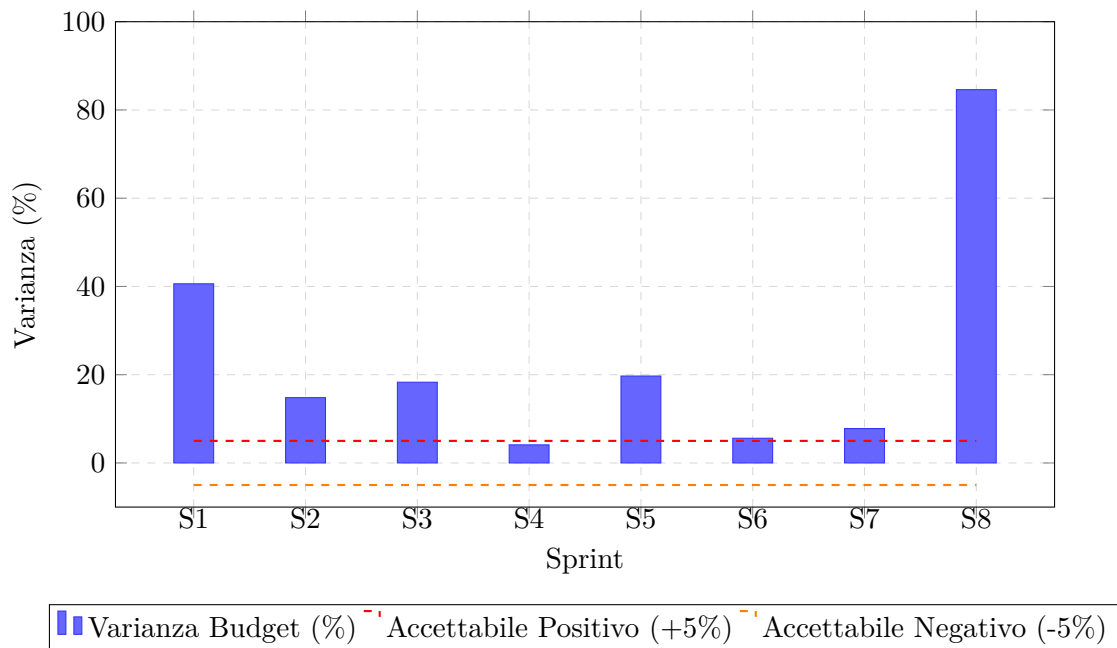


Figura 5: Varianza di budget per sprint

Analisi

Dall'analisi del grafico è evidente che i valori di varianza di budget sono sempre rimasti positivi, indicando che il costo effettivo è stato costantemente maggiore rispetto a quanto preventivato. Negli sprint 1, 2, 3 e 5 la varianza è risultata elevata (15-40%) ma comunque gestibile. Lo sprint 4 ha rappresentato un ottimo risultato con varianza minima (+4,1%), dimostrando quasi perfetta aderenza al budget pianificato. Gli sprint 6 e 7 hanno mostrato varianze contenute (+5,6% e +7,8%), indicando un temporaneo miglioramento nell'efficienza economica. Tuttavia, lo sprint 8 ha registrato un picco critico (+84,6%) attribuibile all'intensificazione delle attività di sviluppo del Proof of Concept e alla necessità di coinvolgere maggiormente ruoli ad alto costo orario (Responsabile, Analista). Il superamento frequente della soglia di accettabilità (+5%) evidenzia la necessità di rivedere le strategie di allocazione delle risorse e di migliorare l'accuratezza delle stime economiche per i prossimi sprint.

6.6 Requisiti Soddisfatti

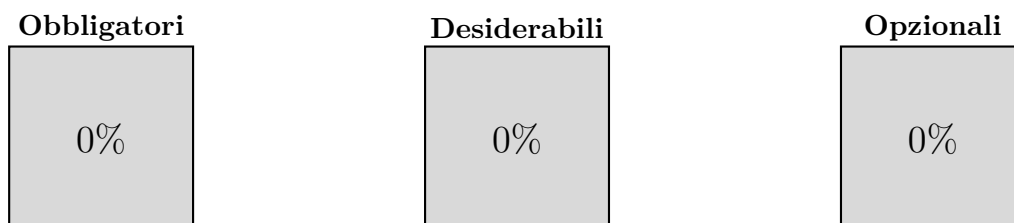


Figura 6: Requisiti obbligatori, desiderabili e opzionali soddisfatti

Analisi

Il progetto si trova attualmente nella fase RTB (Requirements and Technology Baseline), focalizzata sulla definizione dei requisiti e sullo sviluppo di un Proof of Concept. Di conseguenza, al momento non risultano soddisfatti requisiti funzionali né nelle categorie obbligatorie, desiderabili o opzionali. Questa situazione è coerente con lo stato di avanzamento del progetto: gli sprint completati sono stati dedicati principalmente ad attività di analisi, pianificazione, documentazione e sviluppo esplorativo del PoC. L'implementazione vera e propria dei requisiti avrà inizio nelle fasi successive, una volta consolidata la baseline tecnologica e approvato il PoC dal committente. È previsto che i primi requisiti obbligatori vengano soddisfatti a partire dalla fase di Product Baseline, quando il team procederà con lo sviluppo incrementale del prodotto finale seguendo le priorità definite nel Product Backlog.

6.7 Gestione dei Rischi

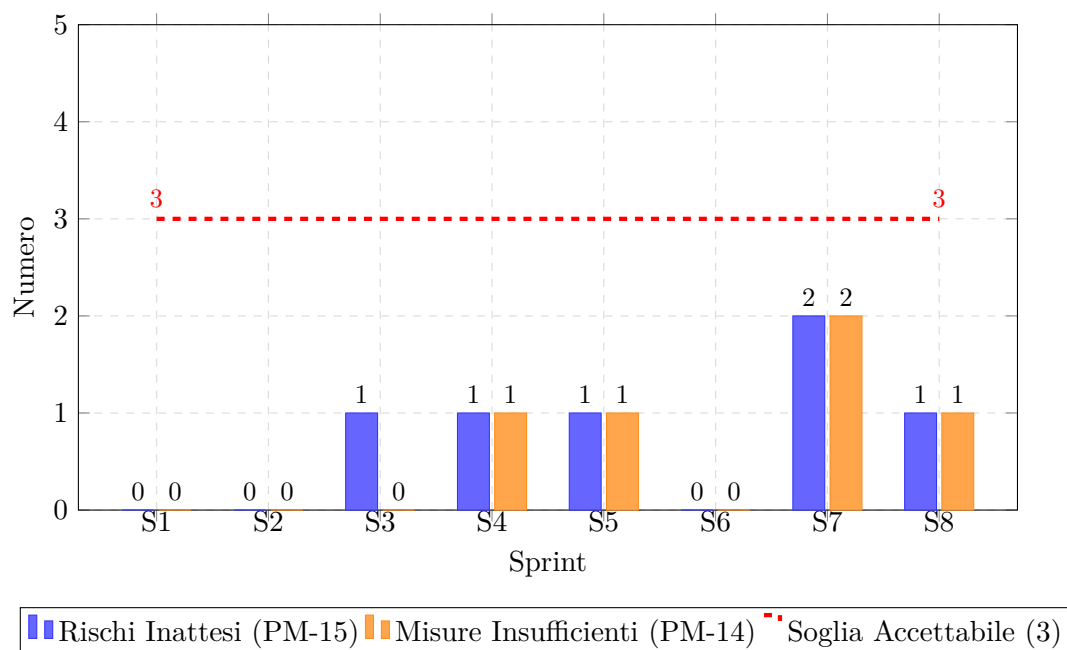


Figura 7: Rischi inattesi e misure di mitigazione insufficienti per sprint

Analisi

Il grafico mostra l'andamento dei rischi inattesi (PM-15) e delle misure di mitigazione insufficienti (PM-14) nel corso degli otto sprint. Complessivamente, entrambe le metriche si mantengono ben al di sotto della soglia di accettabilità fissata a 3.

Nei primi due sprint non si sono verificati rischi imprevisti né fallimenti nelle strategie di mitigazione, indicando una buona capacità iniziale di identificazione e gestione dei rischi. A partire dallo sprint 3 si osserva la comparsa di rischi inattesi, principalmente legati alla complessità tecnologica sottostimata e alle difficoltà di integrazione delle tecnologie del capitolato.

Lo sprint 7 ha registrato il picco massimo con 2 rischi inattesi e 2 misure insufficienti, coincidente con l'intensificazione delle attività di sviluppo del PoC e l'emergere di problematiche tecniche non previste in fase di analisi. Tuttavia, anche in questo caso i valori rimangono entro i limiti di accettabilità.

La correlazione tra rischi inattesi e misure insufficienti negli sprint 4, 5, 7 e 8 suggerisce che alcuni rischi emersi hanno trovato il team impreparato, evidenziando la necessità di rafforzare il

processo di identificazione proattiva dei rischi e di aggiornare periodicamente il registro dei rischi con le lezioni apprese.

6.8 Indice di Gulpease della Documentazione

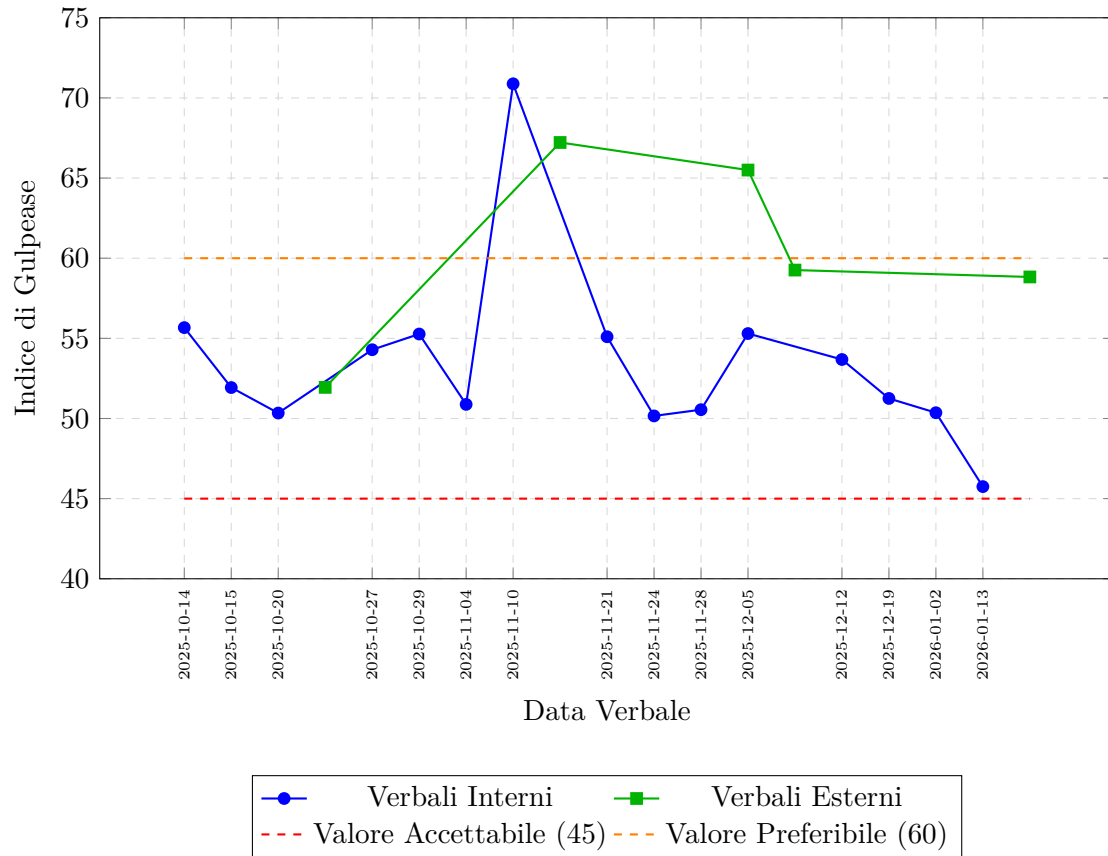


Figura 8: Andamento dell'indice di Gulpease nei verbali prodotti durante il progetto

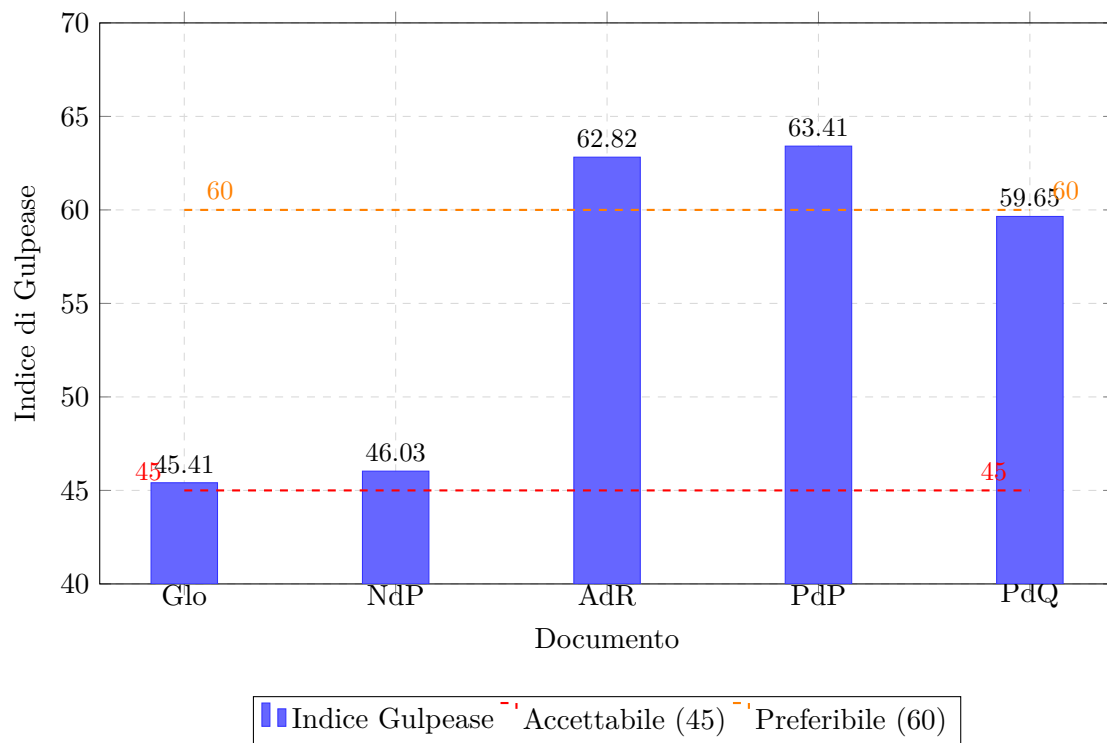


Figura 9: Indice di Gulpease per i documenti principali (RTB)

Legenda:

Glo = Glossario,

NdP = Norme di Progetto,

AdR = Analisi dei Requisiti,

PdP = Piano di Progetto,

PdQ = Piano di Qualifica

6.9 Errori di Qualità Linguistica

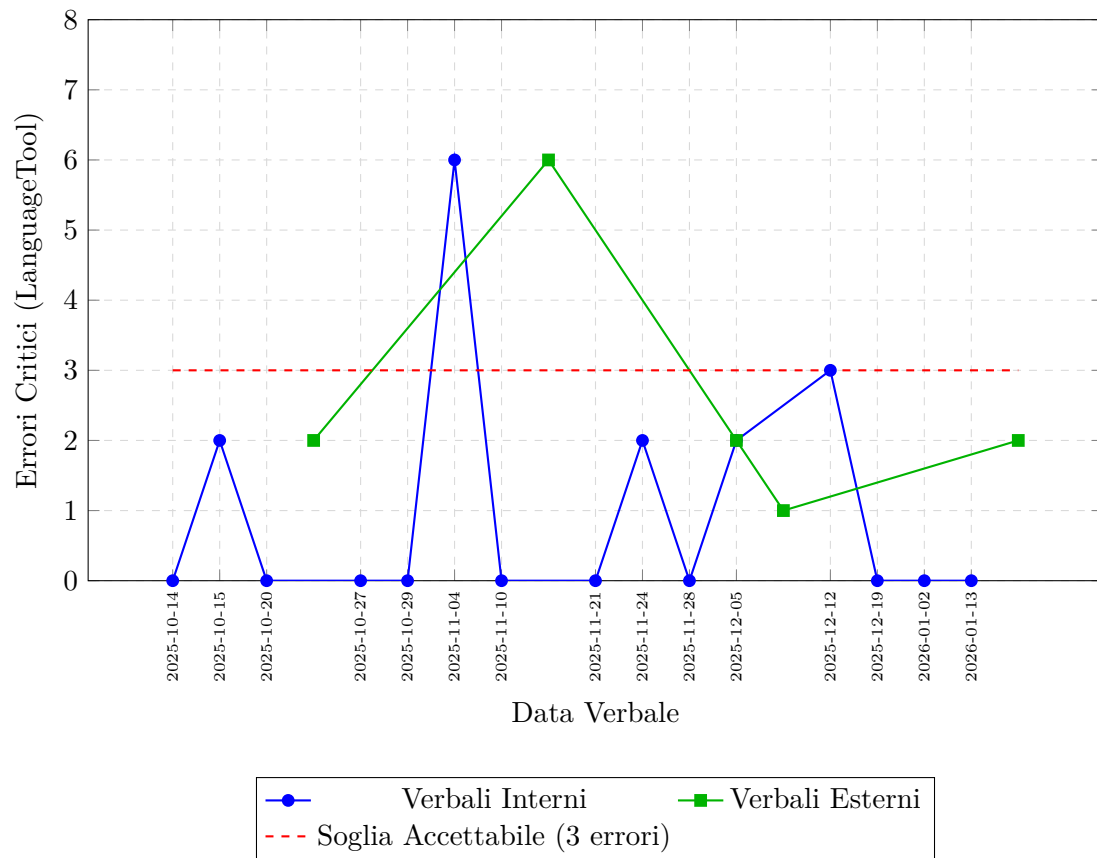


Figura 10: Errori critici rilevati da LanguageTool nei verbali

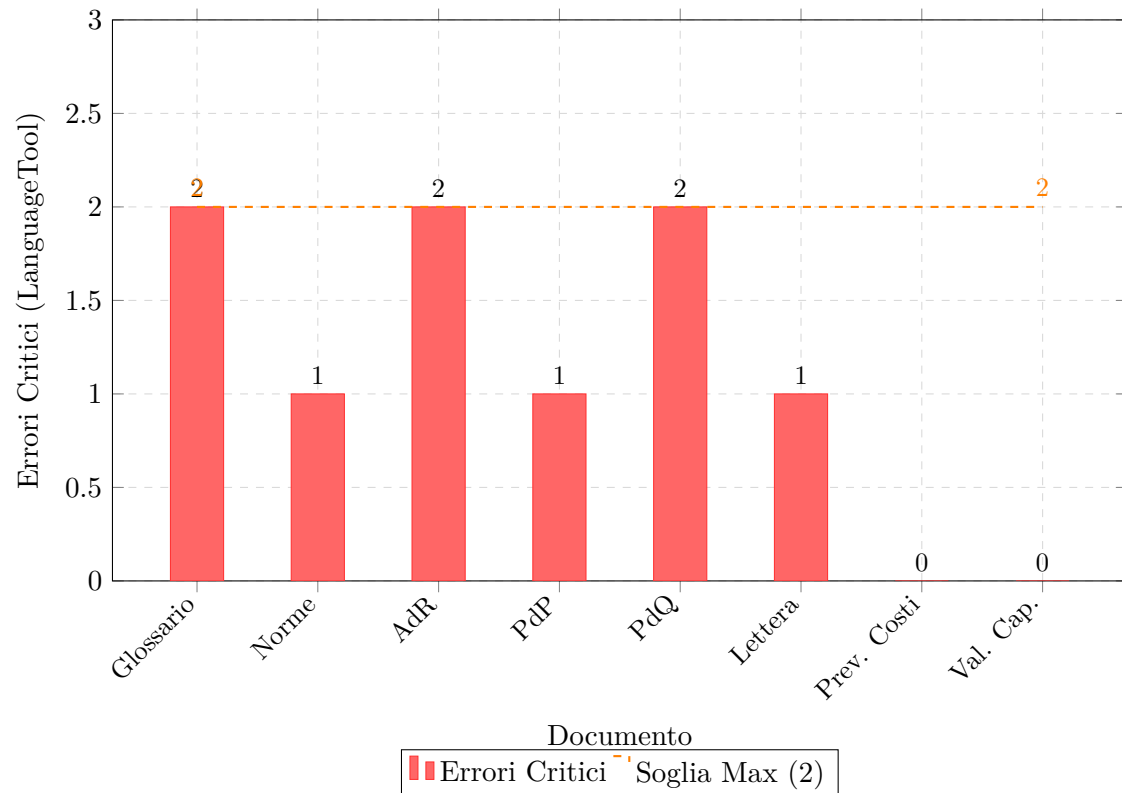


Figura 11: Errori critici rilevati da LanguageTool nei documenti principali

Analisi

L'analisi della qualità linguistica mostra risultati complessivamente positivi. Per i verbali, si osserva un andamento variabile con alcuni picchi critici: il verbale esterno del 2025-11-12 e il verbale interno del 2025-11-04 hanno registrato 6 errori critici ciascuno, superando la soglia accettabile di 3 errori. *Si precisa che questi errori sono risultati essere falsi positivi da parte di LanguageTool, dovuti a limitazioni dello strumento.* Tuttavia, la maggior parte dei verbali (70%) presenta 0-2 errori, rientrando nei parametri di qualità. I documenti principali mostrano una situazione eccellente: Preventivo Costi e Valutazione Capitolati non presentano errori critici, mentre Norme di Progetto e Piano di Progetto ne contengono solo 1 ciascuno. Glossario, Analisi dei Requisiti e Piano di Qualifica raggiungono la soglia massima di 2 errori, ma rimangono entro i limiti di accettabilità. Gli errori più comuni riguardano l'uso della d eufonica, anglicismi non tradotti ("input", "output") e occasionali errori di concordanza. Il gruppo ha implementato verifiche automatiche con LanguageTool nella pipeline CI/CD per garantire un monitoraggio continuo della qualità linguistica.

Per quanto riguarda l'indice di Gulpease, i verbali mostrano un andamento temporale con valori generalmente superiori a 50, con un picco eccezionale di 70,88 nel verbale interno del 2025-11-10. I verbali esterni tendono ad avere valori leggermente superiori (media 60,55) rispetto agli interni (media 53,05), probabilmente per la maggiore attenzione alla chiarezza comunicativa verso stakeholder esterni. Solo il verbale interno del 2026-01-13 scende sotto la soglia di 50 (45,75), richiedendo revisione. I documenti principali presentano una buona leggibilità: Analisi dei Requisiti (62,82) e Piano di Progetto (63,41) superano il valore preferibile di 60, mentre Piano di Qualifica (59,65) si avvicina. Glossario (45,41) e Norme di Progetto (46,03) rimangono sotto la soglia dei 50 punti, riflettendo la natura più tecnica e formale di questi documenti, che richiedono terminologia specialistica e costrutti sintattici complessi.