



NoTIP

NO TESTS IN PRODUCTION

Piano di Qualifica

Versione	1.1.1
Data Modifica	2026-03-09
Utilizzo	Esterno

Abstract dei contenuti

Il documento descrive le metodologie di controllo qualità applicate al progetto, includendo l'utilizzo di metriche di monitoraggio e pianificazione di test. L'obiettivo è assicurare la conformità del prodotto finale alle specifiche di sicurezza ed efficienza richieste per il sistema distribuito e l'architettura Cloud.

Changelog

Versione	Data	Autori	Verificatore _G	Descrizione
1.1.1	2026-03-09	Leonardo Preo	Valerio Solito	Uniformazione struttura Riferimenti del documento
1.1.0	2026-02-25	Francesco Marcon, Matteo Mantoan	Valerio Solito	Aggiustamento test a seguito a seguito dell'aggiornamento dell'AdR a v1.1.0
1.0.0	2026-02-16	Valerio Solito (responsabile)		Approvazione per ingresso in <i>baseline_G</i> <i>RTB_G</i>
0.9.0	2026-02-11	Alessandro Contarini	Alessandro Mazzariol	Sistemazione del Cruscotto di Valutazione
0.8.3	2026-02-08	Alessandro Contarini	Alessandro Mazzariol	Fix valutazione tecnologia <i>Angular_G</i>
0.8.2	2026-02-06	Alessandro Mazzariol	Alessandro Contarini	Fix numeri requisiti nei test
0.8.1	2026-02-06	Mario De Pasquale	Alessandro Mazzariol	Correzione errori di battitura
0.8.0	2026-02-05	Alessandro Contarini	Alessandro Mazzariol	Definizione delle valutazioni per l'automiglioramento (per quanto riscontrato fino in data odierna)
0.7.0	2026-02-04	Alessandro Contarini	Alessandro Mazzariol	Definizione delle ultime sezioni del documento (con contenuti riguardanti i risultati finora riscontrati)
0.6.0	2026-02-04	Alessandro Contarini	Leonardo Preo	Finita la sezione di Testing
0.5.1	2026-02-03	Alessandro Contarini, Mario De Pasquale	Leonardo Preo	Fix test di Integrazione e di Unità e aggiunta bozza Test di Sistema
0.5.0	2026-02-01	Mario De Pasquale	Leonardo Preo	Aggiunti Test di integrazione
0.4.0	2026-01-27	Alessandro Contarini	Alessandro Mazzariol	Aggiunti Test di unità
0.3.0	2025-12-13	Francesco Marcon	Valerio Solito	Aggiornamento e aggiunta MQ04, MQ08
0.2.0	2025-12-12	Francesco Marcon	Valerio Solito	Metriche qualità di prodotto
0.1.0	2025-12-12	Francesco Marcon, Valerio Solito	Matteo Mantoan	Metriche qualità di processo
0.0.1	2025-12-05	Francesco Marcon	Valerio Solito	Prima bozza struttura del documento

Indice

1. Introduzione	7
1.1. Scopo del documento	7
1.2. Scopo del prodotto	7
1.3. Glossario	8
1.4. Riferimenti	8
1.4.1. Riferimenti normativi	8
1.4.2. Riferimenti informativi	8
2. Qualità di processo	8
2.1. Processi primari	9
2.1.1. Fornitura	9
2.1.2. Sviluppo	11
2.2. Processi di supporto	12
2.2.1. Documentazione	12
2.2.2. Verifica	13
2.2.3. Gestione della Configurazione	14
2.2.4. Gestione della Qualità	14
2.3. Processi organizzativi	15
2.3.1. Gestione dei Processi	15
3. Qualità di prodotto	17
3.1. Funzionalità	17
3.2. Affidabilità	18
3.3. Usabilità	20
3.4. Efficienza	20
3.5. Manutenibilità	21
3.6. Portabilità	22
3.7. Sicurezza	23
4. Metodi di testing	24
4.1. Test di Unità	24
4.2. Test di Integrazione	26
4.3. Test di Sistema	27
5. Cruscotto di valutazione	31
5.1. MP01 e MP02: Earned Value e Planned Value	32
5.2. MP03 e MP07: Actual Cost e Estimate To Complete	32
5.3. MP04 e MP05: Cost Performance Index e Schedule Performance Index	33
5.4. MP06: Estimate At Completion	33
5.5. MP08: Time Estimate At Completion	34
5.6. MP09: Budget Burn Rate	34
5.7. MP11: Indice di Gulpease	35
5.8. MP12: Correttezza Ortografica	36
5.9. MP18: Quality Metrics Satisfied	36
5.10. MP20: Time Efficiency	37
5.11. MP21: <i>Sprint_G</i> Velocity Stability	37
5.12. MP22: Meeting Efficiency Index	38
5.13. MP23: <i>PR_G</i> Resolution Time	39
6. Valutazioni per l'automiglioramento	39



6.1. Valutazione sulle tecnologie adottate	40
6.2. Valutazione sull'organizzazione del team	40
6.3. Valutazione sui singoli componenti del gruppo	41

Indice delle figure

Figura 1	Grafico per <i>Sprint_G</i> di MP01 e MP02	32
Figura 2	Grafico per <i>Sprint_G</i> di MP03 e MP07	32
Figura 3	Grafico per <i>Sprint_G</i> di MP04 e MP05	33
Figura 4	Grafico per <i>Sprint_G</i> di MP06	33
Figura 5	Grafico per <i>Sprint_G</i> di MP08	34
Figura 6	Grafico per <i>Sprint_G</i> di MP09	34
Figura 7	Grafico per <i>Sprint_G</i> di MP11	35
Figura 8	Grafico per <i>Sprint_G</i> di MP12	36
Figura 9	Grafico per <i>Sprint_G</i> di MP18	36
Figura 10	Grafico per <i>Sprint_G</i> di MP20	37
Figura 11	Grafico per <i>Sprint_G</i> di MP21	37
Figura 12	Grafico per <i>Sprint_G</i> di MP22	38
Figura 13	Grafico per <i>Sprint_G</i> di MP23	39

Indice delle tabelle

Tabella 1	Soglie metriche Fornitura di processo	9
Tabella 2	Soglie metriche Sviluppo di processo	11
Tabella 3	Soglie metriche Documentazione di processo	12
Tabella 4	Soglie metriche Verifica di processo	13
Tabella 5	Soglie metriche Gestione della Configurazione di processo	14
Tabella 6	Soglie metriche Gestione della Qualità	14
Tabella 7	Soglie metriche Gestione dei Processi	15
Tabella 8	Soglie metriche Funzionalità del prodotto	17
Tabella 9	Soglie metriche Affidabilità del prodotto	18
Tabella 10	Soglie metriche Usabilità del prodotto	20
Tabella 11	Soglie metriche Efficienza del prodotto	20
Tabella 12	Soglie metriche Manutenibilità del prodotto	21
Tabella 13	Soglie metriche Portabilità del prodotto	22
Tabella 14	Soglie metriche Sicurezza del prodotto	23

1. Introduzione

1.1. Scopo del documento

Il presente Piano di Qualifica definisce le strategie, le metodologie e le metriche adottate dal gruppo per garantire la qualità del prodotto software e l'efficienza dei processi del progetto. Esso costituisce il riferimento principale per le attività di:

- **Verifica:** accertarsi che il prodotto stia venendo costruito a regola d'arte.
- **Validazione:** accertarsi che si stia costruendo il prodotto giusto rispetto alle attese del proponente.

Il documento nasce all'inizio del progetto, ma è concepito come uno strumento dinamico che accompagna l'intero ciclo di vita dello stesso, evolvendosi attraverso revisioni migliorative continue. La strategia di gestione della qualità si ispira al ciclo di $PDCA_G$ (Plan, Do, Check, Act), articolandosi nelle seguenti fasi:

- **Pianificazione (Plan):** Definizione degli obiettivi qualitativi e quantitativi. Questa fase stabilisce le metriche di riferimento e le soglie di accettazione da applicare sia ai processi che al prodotto finale.
- **Esecuzione (Do):** Attuazione delle procedure pianificate. In questa fase vengono eseguiti i processi di sviluppo e di test secondo le metodologie stabilite, raccogliendo i dati necessari per le misurazioni successive.
- **Controllo (Check):** Applicazione di misurazioni oggettive durante lo svolgimento delle attività. L'obiettivo è rilevare scostamenti rispetto a quanto pianificato e verificare la conformità ai requisiti stabiliti.
- **Miglioramento (Act):** Analisi critica dei risultati ottenuti (consuntivo). Questa fase mira a identificare difficoltà organizzative o tecniche e ad attuare azioni correttive tempestive per ottimizzare i processi negli $sprint_G$ successivi.

L'obiettivo finale è assicurare che il rilascio soddisfi pienamente i requisiti del $capitolato_G$, minimizzando i difetti e garantendo alti livelli di manutenibilità e usabilità.

1.2. Scopo del prodotto

Il progetto ha l'obiettivo di realizzare una piattaforma Cloud multi- $tenant_G$ e scalabile, dedicata all'acquisizione e alla gestione sicura di dati provenienti da sensori Bluetooth Low Energy (BLE_G).

L'architettura del sistema si fonda su due componenti principali:

- **Infrastruttura Cloud Centralizzata:**
 - Costituisce il cuore del sistema per l'ingestione, segregazione ed esposizione dei dati.
 - Gestisce il $provisioning_G$ sicuro dei $gateway_G$ e il controllo degli accessi in un contesto multi- $tenant_G$.
 - Garantisce la persistenza dei dati e la loro accessibilità.
- **Simulatore Software di $Gateway_G$:**
 - Replica il comportamento di un $Gateway_G BLE_G$ -WiFi fisico in un contesto reale.
 - Genera flussi di dati verosimili per validare la robustezza del sistema.
 - Gestisce la comunicazione sicura verso il livello Cloud.

L'obiettivo finale è quello di fornire uno strumento completo per l'amministrazione e la visualizzazione dei dati, garantendo requisiti non funzionali critici quali sicurezza, affidabilità e scalabilità.

1.3. Glossario

Al fine di garantire un linguaggio univoco e prevenire possibili ambiguità, i termini tecnici, gli acronimi e i vocaboli con accezione specifica nel contesto del progetto sono stati definiti puntualmente nel documento [Glossario v2.0.0](#). Si invita il lettore a fare riferimento a tale documento per chiarire qualsiasi dubbio. Le parole che possiedono un riferimento nel Glossario saranno indicate nel modo che segue:

parola_G

1.4. Riferimenti

1.4.1. Riferimenti normativi

- [Norme di Progetto v1.1.0](#)
- [Capitolato_G d'appalto_G C7 - Sistema di acquisizione dati da sensori](#)
Ultimo accesso: 2026-03-09

1.4.2. Riferimenti informativi

- [Standard ISO/IEC 12207:1995_G](#)
Ultimo accesso: 2026-03-09
- [Standard ISO/IEC 9126](#)
Ultimo accesso: 2026-03-09
- [T07 - Qualità di prodotto](#)
Ultimo accesso: 2026-03-09
- [T08 - Qualità di processo](#)
Ultimo accesso: 2026-03-09
- [T09 - Verifica e Validazione](#)
Ultimo accesso: 2026-03-09
- [Glossario v2.0.0](#)

2. Qualità di processo

Garantire la qualità del processo significa assicurarsi che il nostro metodo di lavoro sia solido, e non solo che il codice funzioni. Un processo ben strutturato è l'unico modo per evitare sprechi di risorse, ridurre il rischio di errori a cascata e mantenere il controllo sulle tempistiche del progetto.

Per ottenere questo risultato, non ci affidiamo al caso ma a una strategia basata su:

- Riferimenti normativi: abbiamo preso come riferimento il modello *CMMI_G* e lo standard ISO/IEC 12207;
- Approccio Quantitativo: per evitare valutazioni soggettive, utilizziamo metriche oggettive. Ogni metrica ha due soglie di riferimento: un valore accettabile e un valore ottimale;
- Iterazione e Miglioramento: sfruttiamo le retrospettive di fine *sprint_G* per analizzare i dati raccolti e, se i numeri si discostano dagli obiettivi, allora applicando il ciclo *PDCA_G*, cerchiamo di correggere nel successivo *sprint_G*.

2.1. Processi primari

2.1.1. Fornitura

Codice	Metrica	Valore Accettabile	Valore Ottimale
MP01	Earned Value	≥ 0	$= PV$
MP02	Planned Value	≥ 0	$\leq BAC$
MP03	Actual Cost	≥ 0	$\leq PV$
MP04	Cost Performance Index	≥ 0.6	≥ 1.0
MP05	Schedule Performance Index	≥ 0.6	≥ 1.0
MP06	Estimate At Completion	≥ 0	$\leq BAC$
MP07	Estimate To Complete	≥ 0	$\leq$ Budget residuo
MP08	Time Estimate At Completion	≥ 0	$\leq$ Durata pianificata
MP09	Budget Burn Rate	$\leq BAC/\text{Giorni totali}$	Costante

Tabella 1: Soglie metriche Fornitura di processo

2.1.1.1. Motivazioni delle soglie

MP01 - Earned Value

Motivazione valore accettabile: Il valore accettabile minimo è zero poiché l'EV rappresenta il valore del lavoro completato. Non può essere negativo per definizione e valori molto bassi rispetto al pianificato indicano ritardi nell'esecuzione, tuttavia non sono invalidanti dal punto di vista metrico.

Motivazione valore ottimo: Il valore ottimo corrisponde esattamente al Planned Value (EV = PV), indicando che il lavoro completato coincide o eccede addirittura quello pianificato. Questo significa che il progetto procede esattamente nei tempi previsti, senza ritardi.

Note di utilizzo: L'EV deve essere calcolato a intervalli regolari (es. fine *sprint*_G) per tracciare l'andamento del progetto. Variazioni significative richiedono analisi delle cause e potenziali azioni correttive.

MP02 - Planned Value

Motivazione valore accettabile: Il PV non può essere negativo poiché rappresenta il budget allocato per il lavoro pianificato fino a un dato momento. Il valore minimo zero si verifica all'inizio del progetto quando nessun lavoro è ancora pianificato.

Motivazione valore ottimo: Il valore ottimo è che il PV rimanga sempre minore o uguale al Budget At Completion (BAC), ovvero che la pianificazione non ecceda il budget totale disponibile. Un PV che cresce linearmente nel tempo indica una pianificazione equilibrata.

Note di utilizzo: Il PV funge da *baseline*_G per tutti i calcoli di performance. Deve essere definito accuratamente in fase di pianificazione basandosi su stime realistiche delle attività.

MP03 - Actual Cost

Motivazione valore accettabile: L'AC non può essere negativo in quanto rappresenta costi realmente sostenuti. Il valore minimo zero si ha all'inizio del progetto prima di iniziare qualsiasi attività.

Motivazione valore ottimo: Il valore ottimo è che l'AC rimanga sempre minore o uguale al PV, indicando che il progetto sta spendendo meno o uguale a quanto pianificato rispetto al periodo corrente. Valori di AC eccedenti PV segnalano potenziali problemi di budget.

Note di utilizzo: L'AC deve includere tutti i costi diretti. La sua accuratezza è critica per calcolare correttamente CPI ed EAC.

MP04 - Cost Performance Index

Motivazione valore accettabile: A seguito di ricerche, solitamente $CPI \geq 0.85$ è considerato accettabile secondo standard di project management. Per l'inesperienza dei membri del gruppo abbiamo deciso di abbassarlo a valori $CPI \geq 0.6$, simboleggianti la sufficienza. Valori inferiori segnalano seri problemi di efficienza tecnico-economica che richiedono interventi immediati, in modo da iniziare a produrre in modo maggiormente efficiente.

Motivazione valore ottimo: Un $CPI = 1.0$ è ottimale, indicando che si sta ottenendo esattamente quello che ci aspetta per la spesa fatta. $CPI > 1$ significa efficienza superiore alle previsioni, ottimale di definizione.

MP05 - Schedule Performance Index

Motivazione valore accettabile: Un $SPI \geq 0.6$ è accettabile, per i motivi presentati nella metrica precedente indicando che il lavoro completato è almeno l'60% di quello pianificato. $SPI < 0.6$ implica, con tutta probabilità, ritardi significativi nel planning del progetto.

Motivazione valore ottimo: Un $SPI \geq 1.0$ è ottimale, indicando che il lavoro completato coincide o eccede quello pianificato. $SPI > 1$ significa che si è in anticipo sulla tabella di marcia. Il valore 1.0 rappresenta l'aderenza perfetta alla pianificazione.

Note di utilizzo: L'SPI è particolarmente importante nelle fasi iniziali per identificare problemi di stima. A differenza del CPI, l'SPI può essere recuperato in *sprint* successivi aumentando la velocity del team.

MP06 - Estimate At Completion

Motivazione valore accettabile: L'EAC non può essere negativo essendo una proiezione di costo ed il valore zero è puramente teorico, non si verifica in progetti reali.

Motivazione valore ottimo: Il valore ottimo è $EAC \leq BAC$, ovvero che la stima finale dei costi non superi il budget inizialmente allocato. Questo indica che il progetto si concluderà entro il budget previsto. Nel caso in cui $EAC = BAC$ significa perfetta aderenza al budget.

Note di utilizzo: L'EAC dovrebbe essere ricalcolato ad ogni *sprint* utilizzando il CPI aggiornato, fornendo quindi una prospettiva concreta sul modo in cui il lavoro sta procedendo. Avere una tendenza crescente nell'EAC è considerabile un warning di potenziale sfioramento budget.

MP07 - Estimate To Complete

Motivazione valore accettabile: L'ETC non può essere negativo poiché rappresenta il costo stimato per completare il lavoro rimanente. Risulta essere zero se e solo se il progetto è completato.

Motivazione valore ottimo: Il valore ottimo è che l'ETC rimanga entro il budget residuo originariamente pianificato. ETC superiore al budget residuo indica necessità di fondi aggiuntivi.

Note di utilizzo: Se ETC supera significativamente il budget disponibile, potrebbe essere necessario intraprendere trattative a ribasso con la proponente riguardo il proseguimento del progetto.

MP08 - Time Estimate At Completion

Motivazione valore accettabile: Il TEAC non può essere negativo essendo una stima temporale sul futuro. Risulterà essere zero solamente al termine del progetto.

Motivazione valore ottimo: Il valore ottimo è $TEAC \leq \text{Durata pianificata}$, indicando che il progetto si concluderà entro le tempistiche previste, dove $TEAC = \text{Durata pianificata}$ indica perfetta aderenza alla schedulazione.

Note di utilizzo: Il TEAC è critico per gestire aspettative degli stakeholder e per decisioni su *milestone_G* contrattuali. Un TEAC significativamente superiore alla durata pianificata può richiedere rinegoziazione delle scadenze o delle richieste.

MP09 - Budget Burn Rate

Motivazione valore accettabile: Il burn rate accettabile è che rimanga minore o uguale al burn rate pianificato. Questo assicura che il budget non si esaurisca prima del completamento del progetto. Un burn rate leggermente superiore è sostenibile solamente se temporaneo.

Motivazione valore ottimo: Il valore ottimo è un burn rate costante e prevedibile nel tempo, idealmente coincidente con quello pianificato. Fluttuazioni minime sono assolutamente normali, ma avere un burn rate costante facilita la comprensione del modo in cui sta venendo speso il budget.

Note di utilizzo: Il burn rate dovrebbe essere visualizzato graficamente (burn-down chart) per identificare visivamente trend problematici.

2.1.2. Sviluppo

Codice	Metrica	Valore Accettabile	Valore Ottimale
MP10	Requirements Stability Index	$\geq 70\%$	100%

Tabella 2: Soglie metriche Sviluppo di processo

2.1.2.1. Motivazioni delle soglie

MP10 - Requirements Stability Index

Motivazione valore accettabile: Questo valore riconosce un margine di cambiamento accettabile, in quanto in progetti con gestione *agile_G*, sono inevitabili cambiamenti per rispondere a feedback o nuove esigenze. $RSI < 70\%$ segnala analisi iniziale inadeguata.

Motivazione valore ottimo: Un $RSI = 100\%$ è ottimale. Questo livello di stabilità riflette un'eccellente analisi iniziale e requisiti ben compresi. Permette pianificazione affidabile e riduce drasticamente i costi di rework.

Note di utilizzo: L'*RSI* diventa significativo solo dopo aver stabilito una *baseline_G* dei requisiti.

2.2. Processi di supporto

2.2.1. Documentazione

Codice	Metrica	Valore Accettabile	Valore Ottimale
MP11	Indice di Gulpease	≥ 60	≥ 80
MP12	Correttezza Ortografica	0 errori	0 errori

Tabella 3: Soglie metriche Documentazione di processo

2.2.1.1. Motivazioni delle soglie

MP11 - Indice di Gulpease

Motivazione valore accettabile: Un indice ≥ 60 garantisce che la documentazione sia comprensibile per lettori con istruzione media, appropriato per documentazione tecnica rivolta a professionisti del settore. Questo livello bilancia precisione terminologica e chiarezza espositiva.

Motivazione valore ottimo: Un indice ≥ 80 indica documentazione molto accessibile, comprensibile anche per lettori con istruzione base.

Note di utilizzo: Non è sempre necessario massimizzare tale indice, ma va interpretato considerando il tipo di documento in esame: manuali utente dovrebbero cercare di raggiungere valori più alti (≥ 75), mentre documentazione tecnica tenderà ad avere valori più bassi (≥ 55) data la natura tecnica del contenuto.

MP12 - Correttezza Ortografica

Motivazione valore accettabile: Zero errori ortografici è l'unico valore accettabile per documentazione professionale.

Motivazione valore ottimo: Il valore ottimo coincide con l'accettabile: zero errori, in quanto la documentazione formale destinata agli stakeholder deve essere priva errori di battitura o refusi da versioni precedenti, in modo da riflettere la professionalità del gruppo e dei processi dallo stesso impiegati.

Note di utilizzo: La presenza di errori indica processi di quality assurance inadeguati che devono essere rafforzati.

2.2.2. Verifica

Codice	Metrica	Valore Accettabile	Valore Ottimale
MP13	Code Coverage	$\geq 80\%$	$\geq 90\%$
MP14	Test Success Rate	100%	100%
MP15	Test Automation Percentage	$\geq 70\%$	$\geq 85\%$
MP16	Defect Discovery Rate	Crescente → Decrescente	Picco in System Test, 0 rilascio

Tabella 4: Soglie metriche Verifica di processo

2.2.2.1. Motivazioni delle soglie

MP13 - Code Coverage

Motivazione valore accettabile: In ogni progetto è importante mantenere una percentuale di code coverage alta, $\geq 80\%$ è una convenzione spesso usata, affinché si possa garantire che il codice scritto sia stato effettivamente testato per la sua maggioranza.

Motivazione valore ottimo: L'ottimo non è stato posto al 100% sia per avere una stima più realistica e sia per evitare che il team sia troppo focalizzato sul raggiungere il 100% di code coverage, creando così test che non controllano il codice nella maniera corretta.

Note di utilizzo: L'obiettivo è avere un code coverage quanto più vicino al 100% nelle sezioni critiche del codice e dell'80% per quanto riguarda il restante codice.

MP14 - Test Success Rate

Motivazione valore accettabile: Trattandosi di un sistema che gestisce dati sensibili ci si prefissa di avere una percentuale del 100%, perché ogni bug potrebbe essere pericoloso in un sistema di questo tipo.

Motivazione valore ottimo: Il valore ottimo coincide con l'accettabile, in quanto non sono tollerati test falliti in *branch_G* principali.

Note di utilizzo: Qualsiasi test fallito indica regressioni o bug che devono essere risolti immediatamente prima di procedere con ulteriore sviluppo

MP15 - Test Automation Percentage

Motivazione valore accettabile: È importante che una parte considerevole dei test da eseguire sul sistema siano automatizzati, così da garantire automaticamente che la parte più importante di tutto il codice sia stata testata in maniera veloce ed efficace.

Motivazione valore ottimo: L'85% è un valore ottimo ragionevole, permettendo esecuzioni frequenti senza costi significativi. Il restante 15% per test di usabilità per l'eventuale interfaccia grafica e per altre verifiche che potrebbero essere necessarie.

Note di utilizzo: Il 100% di *automazione_G* non è considerato un obiettivo perché potrebbe anche introdurre dei test instabili o avere una fase di test eccessivamente lunga.

MP16 - Defect Discovery Rate

Motivazione valore accettabile: Nelle fasi iniziali e centrali dello sviluppo (Unit e Integration Test), il trend deve essere crescente: un numero basso di bug trovati in questa fase indicherebbe test inefficaci o poco aggressivi. Il trend deve diventare decrescente solo nella fase di stabilizzazione finale.

Motivazione valore ottimo: Il valore ottimo prevede che il picco massimo di bug trovati coincida con la fase di Testing di Sistema e Integrazione, per poi crollare drasticamente verso lo zero nell'imminenza del rilascio.

Note di utilizzo: Se la curva rimane piatta (pochi bug trovati) durante lo sviluppo intenso, è necessario rivedere la strategia di test.

2.2.3. Gestione della Configurazione

Codice	Metrica	Valore Accettabile	Valore Ottimale
MP17	<i>Commit_G</i> Message Quality Score	7/10	9/10

Tabella 5: Soglie metriche Gestione della Configurazione di processo

MP17 - *Commit_G* Message Quality Score

Motivazione valore accettabile: Abbiamo optato per un valore accettabile 7/10, perché la qualità del *commit_G* deve essere almeno sopra la sufficienza per garantire chiarezza del messaggio, oltre a permettere in alcuni casi di risalire ad errori introdotti con recenti *commit_G*.

Motivazione valore ottimo: Un punteggio $\geq 9/10$ è ottimale, indicando *commit_G* messages esemplari che documentano completamente sia il «cosa» che il «perché» delle modifiche. Questo facilita enormemente debugging futuro, *code review_G*, e inserimento di nuovi sviluppatori.

Note di utilizzo: *Template_G* di *commit_G* messages e *linee guida_G* chiare nelle Norme di Progetto sono essenziali per raggiungere punteggi alti.

2.2.4. Gestione della Qualità

Codice	Metrica	Valore Accettabile	Valore Ottimale
MP18	Quality Metrics Satisfied	$\geq 80\%$	$\geq 100\%$
MP19	<i>Quality Gate_G</i> Pass Rate	$\geq 85\%$	$\geq 95\%$

Tabella 6: Soglie metriche Gestione della Qualità

2.2.4.1. Motivazioni delle soglie

MP18 - Quality Metrics Satisfied

Motivazione valore accettabile: Un valore $\geq 80\%$ di metriche soddisfatte rappresenta un buon livello qualitativo generale, permettendo un margine del 20% per metriche particolarmente difficili da soddisfare completamente. Sotto questa soglia il progetto ha problemi di qualità.

Motivazione valore ottimo: Avere soddisfazione del 100% significa che tutte le metriche definite sono soddisfatte. Questo livello dimostra eccellenza complessiva nella gestione della qualità.

Note di utilizzo: Questa metrica dovrebbe essere calcolata pesando correttamente le varie metriche, dove quelle critiche dovrebbero avere peso maggiore di metriche meno critiche.

MP19 - *Quality Gate_G* Pass Rate

Motivazione valore accettabile: Indica che la maggior parte del codice proposto rispetta gli standard minimi di sicurezza e manutenibilità. Un 15% di fallimenti è concesso per permettere iterazioni veloci e feedback loop rapidi dalla CI senza imporre un perfezionismo prematuro.

Motivazione valore ottimo: Un tasso $\geq 95\%$ testimonia un'elevata maturità del team nell'eseguire verifiche locali prima del push sul *repository_G*. Indica un processo di integrazione fluido, dove la CI fallisce solamente raramente.

Note di utilizzo: Il valore va interpretato con attenzione: un pass rate costantemente al 100% potrebbe indicare controlli troppo laschi, mentre un valore troppo basso suggerisce la necessità di rivedere la configurazione dei gate o di migliorare la formazione del team sui test locali.

2.3. Processi organizzativi

2.3.1. Gestione dei Processi

Codice	Metrica	Valore Accettabile	Valore Ottimale
MP20	Time Efficiency	$\geq 60\%$	100%
MP21	<i>Sprint_G</i> Velocity Stability	$\leq 30\%$	$\leq 10\%$
MP22	Meeting Efficiency Index	≥ 2 decisioni/ora	≥ 5 decisioni/ora
MP23	<i>PR_G</i> Resolution Time	≤ 3 giorni	≤ 1 giorni

Tabella 7: Soglie metriche Gestione dei Processi

2.3.1.1. Motivazioni delle soglie

MP20 - Time Efficiency

Motivazione valore accettabile: Il minimo accettabile per garantire un avanzamento accettabile del progetto. Scendere sotto questa soglia indica una dispersione eccessiva di risorse in attività non produttive.

Motivazione valore ottimo: Indica uno scenario di efficienza assoluta in cui sono stati eliminati tutti i tempi morti, le riunioni superflue e le attività a non-valore aggiunto.

Note di utilizzo: Per calcolare la metrica, è necessario distinguere nel tracciamento le ore «produttive» da quelle «organizzative/di palestra».

MP21 - *Sprint_G* Velocity Stability

Motivazione valore accettabile: Una deviazione standard $\leq 30\%$ della media è accettabile, permettendo una certa variabilità naturale tra *sprint_G* dovuta alla complessità variabile delle *task_G*, impegni dei membri del gruppo o eventi imprevisti.

Motivazione valore ottimo: $\leq 10\%$ di deviazione è ottimale, indicando velocity altamente prevedibile. Questo livello si raggiunge con: stime accurate, team stabile, scope ben definito, e impedimenti gestiti efficacemente. Facilita enormemente commitment e pianificazione di release.

Note di utilizzo: Variabilità $> 20\%$ rende la pianificazione inaffidabile. Dovrebbe però migliorare naturalmente con il passare del tempo, stabilizzandosi su valori tendenti all'ottimo.

MP22 - Meeting Efficiency Index

Motivazione valore accettabile: ≥ 2 decisioni/ora è accettabile per meeting produttivi. Riconosce che alcune discussioni richiedono tempo per esplorare alternative prima di decidere. Meeting sotto questa soglia tendono ad essere troppo poco produttivi.

Motivazione valore ottimo: ≥ 5 decisioni/ora è ottimale, indicando meeting altamente focalizzati con agenda chiara e partecipanti preparati. Questo livello massimizza il valore del tempo speso.

Note di utilizzo: Preparazione dell'agenda con decisioni da prendere, condividere materiale da discutere in precedenza e documentare decisioni prese.

MP23 - *PR_G* Resolution Time

Motivazione valore accettabile: Un tempo di *merge_G* ≤ 3 giorni è accettabile. Garantisce che il codice o un documento non rimanga bloccato troppo a lungo, riducendo il rischio di conflitti di *merge_G* complessi.

Motivazione valore ottimo: Un tempo ≤ 1 giorno è ottimale. Indica un processo di *Code Review_G* estremamente efficiente e una cultura di team che dà priorità al processo di revisione rispetto all'inizio di nuove *task_G*, favorendo il Continuous Integration.

Note di utilizzo: Un tempo di risoluzione breve garantisce che la correzione o l'aggiunta avvenga mentre la memoria dell'autore è ancora fresca. Inoltre è raccomandata l'apertura di *PR_G* più piccole e focalizzate per migliorare l'efficacia e la velocità della verifica

3. Qualità di prodotto

La qualità del prodotto costituisce il traguardo fondamentale del progetto. Questa la possiamo intendere come la capacità del software rilasciato di rispondere puntualmente alle necessità degli stakeholder e di operare con robustezza in ambiente di produzione. Un livello qualitativo elevato non si raggiunge in maniera casuale, ma è la diretta conseguenza del rigore metodologico applicato ai processi.

Per garantire una valutazione oggettiva e strutturata, il gruppo ha adottato come framework di riferimento lo standard ISO/IEC 9126. Secondo tale modello, l'eccellenza del software si misura attraverso il soddisfacimento delle seguenti caratteristiche:

- **Funzionalità**, per coprire integralmente i requisiti espressi;
- **Affidabilità**, per assicurare continuità di servizio senza malfunzionamenti;
- **Usabilità**, per garantire un'interazione intuitiva;
- **Efficienza**, per ottimizzare l'uso delle risorse;
- **Manutenibilità**, per facilitare l'evoluzione futura del codice;
- **Portabilità**, per permettere l'esecuzione in ambienti diversi

3.1. Funzionalità

Codice	Metrica	Valore Accettabile	Valore Ottimale
MQ01	Requisiti Obbligatori Soddisfatti	100%	100%
MQ02	Requisiti Desiderabili Soddisfatti	0%	100%
MQ03	Requisiti Opzionali Soddisfatti	0%	100%
MQ04	Requirements Test Coverage	100% (R. Obbligatori)	100% (R. Obbligatori + Accettabili)

Tabella 8: Soglie metriche Funzionalità del prodotto

Motivazioni delle soglie

MQ01 - Requisiti Obbligatori Soddisfatti

Motivazione valore accettabile: L'unico valore accettabile è 100%. I requisiti obbligatori rappresentano funzionalità contrattualmente vincolanti e essenziali per l'utilizzo base del sistema, richieste esplicitamente dallo stakeholder. Non implementare anche un solo requisito obbligatorio rende il prodotto non conforme e non accettabile dal committente.

Motivazione valore ottimo: Il valore ottimo coincide con l'accettabile.

Note di utilizzo: Problemi nell'implementazione di requisiti obbligatori dovranno avere priorità massima e potranno richiedere riallocazione urgente di ore e ruoli.

MQ02 - Requisiti Desiderabili Soddisfatti

Motivazione valore accettabile: 0% è tecnicamente accettabile poiché questi requisiti non sono essenziali per l'accettazione del prodotto. Tuttavia, anche a seguito degli incontri tenuti con la proponente, l'implementazione di una parte di essi è altamente consigliata e aggiunge valore al prodotto.

Motivazione valore ottimo: 100% rappresenta un prodotto completo e ricco di funzionalità che supera le aspettative minime. Questo livello dovrebbe essere perseguito se il progetto ha risorse e tempo sufficienti dopo aver completato tutti i requisiti obbligatori.

Note di utilizzo: La priorità esecutiva assegnata ai requisiti desiderabili dovrebbe basarsi su: feedback del committente, rapporto valore/costo di implementazione e dipendenze con altri requisiti.

MQ03 - Requisiti Opzionali Soddisfatti

Motivazione valore accettabile: 0% risulta essere accettabile, in quanto i requisiti opzionali rappresentano dei «nice-to-have» da considerare solamente dopo aver completato obbligatori e desiderabili. Non implementarli non rende il prodotto incompleto rispetto alle richieste.

Motivazione valore ottimo: 100% rappresenta un prodotto estremamente completo, ma questo livello difficilmente raggiungibile senza un notevole sforzo economicamente.

Note di utilizzo: La loro implementazione non dovrebbe mai sottrarre risorse a requisiti di priorità superiore durante lo sviluppo della versione corrente.

MQ04 - Requirements Test Coverage

Motivazione valore accettabile: Non è sufficiente che un requisito sia implementato; deve essere verificato. Il 100% dei requisiti obbligatori deve essere tracciato da almeno un caso di test che termini con successo.

Motivazione valore ottimo: Estendere la copertura dei test automatizzati anche ai requisiti desiderabili e opzionali garantisce che l'intero valore del prodotto sia testato.

3.2. Affidabilità

Codice	Metrica	Valore Accettabile	Valore Ottimale
MQ05	Branch _G Coverage	≥ 60%	≥ 80%
MQ06	Statement Coverage	≥ 70%	≥ 90%
MQ07	Failure Density	≤ 0.5	≤ 0.1
MQ08	Modified Condition/Decision Coverage (MC/DC)	100% (Critical Only)	100%

Tabella 9: Soglie metriche Affidabilità del prodotto

Motivazioni delle soglie

MQ05 - *Branch*_G Coverage

Motivazione valore accettabile: $\geq 60\%$ è accettabile in quanto testare tutti i *branch*_G possibili risulta complesso. Questo livello garantisce che un livello sufficiente delle decisioni logiche sia verificata.

Motivazione valore ottimo: Un valore $\geq 80\%$ è ottimale e rappresenta un'eccellente copertura, riducendo drasticamente i bug di logica condizionale e migliora la confidenza nel comportamento del codice in tutte le circostanze.

Note di utilizzo: Focus particolare dovrebbe essere dato a: business logic critica, gestione errori, e validazione input.

MQ06 - Statement Coverage

Motivazione valore accettabile: Un valore $\geq 70\%$ garantisce che la maggioranza delle istruzioni sia stata eseguita almeno una volta durante i test, riducendo il rischio di bug nascosti in codice non verificato.

Motivazione valore ottimo: $\geq 90\%$ è ottimale e dimostra una test suite molto completa. Il 10% non coperto può giustificarsi per: codice di gestione errori estremi, codice legacy non critico, o sezioni in attesa di rimozione.

Note di utilizzo: Deve essere affiancata da *branch*_G coverage e test di qualità che verifichino comportamenti attesi, non solo l'esecuzione di linee.

MQ07 - Failure Density

Motivazione valore accettabile: Un valore ≤ 0.5 failure/KLOC è accettabile per software in fase di stabilizzazione. Corrisponde mediamente a 1 bug ogni 2000 linee di codice.

Motivazione valore ottimo: ≤ 0.1 failure/KLOC è ottimale e rappresenta software di altissima qualità, tipico di software a seguito di estensive sessioni di testing e bug-fixing.

Note di utilizzo: È cruciale distinguere failure per severity: un bug critico (crash, data loss) ha impatto molto maggiore di uno cosmetico (typo in UI).

MQ08 - Modified Condition/Decision Coverage (MC/DC)

Motivazione valore accettabile: Dato l'alto costo di questa metrica, è accettabile applicarla rigorosamente (100%) solo ai moduli identificati come «Safety-Critical» o «Business-Critical». Garantisce che ogni singola condizione in una decisione complessa influenzi indipendentemente il risultato.

Motivazione valore ottimo: L'ottimo prevede il 100% di copertura sui moduli critici senza eccezioni e l'estensione ai moduli ad alta complessità ciclomatica. Questo livello offre la certezza matematica che non esistano condizioni superflue o logiche errate (mascheramento) nelle decisioni complesse.

Note di utilizzo: L'adozione della metrica MC/DC è limitata ai soli componenti critici (Core Domain) a causa dell'elevato overhead di implementazione e della necessità di strumentazione di testing avanzata. Per i moduli standard (UI, DTO_G , Controller) la $Branch_G$ Coverage è ritenuta sufficiente per garantire la qualità senza impattare eccessivamente sulla velocity del team.

3.3. Usabilità

Codice	Metrica	Valore Accettabile	Valore Ottimale
MQ09	Time on $Task_G$	≤ 60 sec	≤ 30 sec
MQ10	Error Rate	$\leq 5\%$	$\leq 2\%$

Tabella 10: Soglie metriche Usabilità del prodotto

Motivazioni delle soglie

MQ09 - Time on $Task_G$

Motivazione valore accettabile: Dal momento in cui l'utente impiega un tempo ≤ 60 secondi per eseguire $task_G$ standard, viene considerato accettabile. Oltre questo tempo, l'utente può percepire il sistema come lento o inefficiente.

Motivazione valore ottimo: ≤ 30 secondi è ottimale e indica un'interfaccia altamente efficiente e intuitiva.

Note di utilizzo: Misurare con utenti reali o rappresentativi del target. Definire $task_G$ specifici e misurabili (es. «creare un nuovo $gateway_G$ e associare 3 sensori»). $Task_G$ complesse possono giustificare tempi maggiori. Questa metrica diventa rilevante solo quando l'interfaccia è sufficientemente sviluppata.

MQ10 - Error Rate

Motivazione valore accettabile: $\leq 5\%$ è accettabile, riconoscendo che alcuni errori utente sono inevitabili. Questo margine tollera una certa curva di apprendimento e situazioni ambigue nell'interfaccia.

Motivazione valore ottimo: $\leq 2\%$ è ottimale e indica un'interfaccia estremamente intuitiva dove gli errori sono rari. Si raggiunge con: design patterns consolidati, validazione input efficace, feedback chiari e prevenzione degli errori.

Note di utilizzo: Errori frequenti in $task_G$ specifici indicano problemi di design localizzati. È quindi utile andare ad analizzare quali $task_G$ generano più errori e perché.

3.4. Efficienza

Codice	Metrica	Valore Accettabile	Valore Ottimale
MQ11	Response Time	≤ 2 sec	≤ 1 sec

Tabella 11: Soglie metriche Efficienza del prodotto

Motivazioni delle soglie

MQ11 - Response Time

Motivazione valore accettabile: ≤ 2 secondi è il limite accettabile secondo standard di usabilità (Nielsen). Oltre questo tempo, l'utente percepisce attesa significativa.

Motivazione valore ottimo: Un valore ≤ 1 secondo è ottimale e fornisce un'esperienza fluida senza attese percepite. Mantiene il flusso di lavoro dell'utente ininterrotto.

Note di utilizzo: Operazioni complesse possono giustificare tempi $> 2s$. Potrebbe aiutare aggiungere degli indicatori di progresso, per dare un feedback diretto all'utilizzatore.

3.5. Manutenibilità

Codice	Metrica	Valore Accettabile	Valore Ottimale
MQ12	Code Smells	≤ 10	≤ 5
MQ13	Coefficient of Coupling	≤ 0.4	≤ 0.2
MQ14	Cyclomatic Complexity	≤ 10	≤ 5
MQ15	Code Duplication Percentage	$\leq 5\%$	$\leq 3\%$

Tabella 12: Soglie metriche Manutenibilità del prodotto

Motivazioni delle soglie

MQ12 - Code Smells

Motivazione valore accettabile: ≤ 10 code smell/KLOC consente la presenza di alcuni problemi di design non critici, senza compromettere significativamente la manutenibilità complessiva.

Motivazione valore ottimo: ≤ 5 code smell/KLOC indica codice ben strutturato e facilmente manutenibile; richiede revisione costante e refactoring mirato per mantenere livelli elevati di qualità.

Note di utilizzo: Strumenti come *SonarQube_G* individuano automaticamente code smell. Prioritizzare la correzione nelle aree critiche o soggette a frequenti modifiche. Gli smell temporanei devono essere documentati e risolti nelle iterazioni successive.

MQ13 - Coefficient of Coupling

Motivazione valore accettabile: ≤ 0.4 indica accoppiamento moderato e gestibile. Permette interdipendenze necessarie per funzionalità del sistema mantenendo architettura sufficientemente modulare per modifiche localizzate senza effetti a cascata eccessivi.

Motivazione valore ottimo: Un valore ≤ 0.2 rappresenta sistema disaccoppiato, coeso internamente e con poche dipendenze esterne. Ideale per manutenibilità, in quanto modifiche a un componente hanno impatto minimo su altri.

Note di utilizzo: Accoppiamento alto rende il sistema fragile e difficile da testare.

MQ14 - Cyclomatic Complexity

Motivazione valore accettabile: ≤ 10 per metodo/funzione indica codice complesso ma ancora comprensibile e testabile. Oltre questa soglia, codice diventa difficile da capire e richiede numero eccessivo di test per coprire tutti i percorsi logici.

Motivazione valore ottimo: ≤ 5 indica codice semplice, lineare, facile da leggere, testare e mantenere. Questo dovrebbe essere l'obiettivo per la maggior parte delle funzioni. Metodi più complessi dovrebbero essere refactored in funzioni più piccole e coese.

Note di utilizzo: Complessità > 10 richiede refactoring urgente: estrarre metodi, usare polimorfismo invece di condizionali, semplificare logica booleana. La complessità correla fortemente con probabilità di bug e effort di manutenzione.

MQ15 - Code Duplication Percentage

Motivazione valore accettabile: $\leq 5\%$ è accettabile, riconoscendo che piccole duplicazioni possono essere tollerabili se ben giustificate (es. performance critiche, isolamento di contesti). Duplicazioni limitate non compromettono gravemente la manutenibilità.

Motivazione valore ottimo: $\leq 3\%$ è ottimale e indica codice ben fattorizzato seguendo principio DRY. Questo livello minimizza effort di manutenzione e rischio di inconsistenze quando si correggono bug in codice duplicato.

Note di utilizzo: Tool come *SonarQube_G* rilevano duplicazioni automaticamente. Soluzione: estrarre codice comune in funzioni riutilizzabili, usare *Template_G Method* o *Strategy pattern* per gestire variazioni. Duplicazioni legittime (es. test fixtures) dovrebbero essere escluse dal calcolo.

3.6. Portabilità

Codice	Metrica	Valore Accettabile	Valore Ottimale
MQ16	Container Image Size	≤ 500 MB	≤ 200 MB

Tabella 13: Soglie metriche Portabilità del prodotto

Motivazioni delle soglie

MQ16 - Container Image Size

Motivazione valore accettabile: Un valore ≤ 500 MB è accettabile, permettendo l'inclusione di runtime e librerie necessarie ma mantenendo deployment relativamente rapidi e storage gestibili.

Motivazione valore ottimo: ≤ 200 MB è ottimale e tipicamente raggiungibile con base images leggere, garantendo deployment veloci e *cold start_G* rapidi in ambiente cloud.

3.7. Sicurezza

Codice	Metrica	Valore Accettabile	Valore Ottimale
MQ17	Authentication Success Rate	$\geq 98\%$	$\geq 99.5\%$
MQ18	Encryption Coverage	100%	100%

Tabella 14: Soglie metriche Sicurezza del prodotto

Motivazioni delle soglie

MQ17 - Authentication Success Rate

Motivazione valore accettabile: $\geq 98\%$ è accettabile per il sistema di *provisioning_G gateway_G*, considerando che 2% di failure può derivare da errori utente legittimi. Success rate inferiore può indicare problemi nel meccanismo di autenticazione.

Motivazione valore ottimo: $\geq 99.5\%$ è ottimale e indica sistema robusto e affidabile dove quasi tutti i tentativi legittimi hanno successo. Failure residui coprono edge case.

Note di utilizzo: Fondamentale distinguere tra failure legittimi (credenziali errate) e tecnici (servizio down, timeout).

MQ18 - Encryption Coverage

Motivazione valore accettabile: Tutti i dati sensibili devono essere crittografati sia in storage che durante le comunicazioni, quindi 100% è l'unico valore accettabile.

Motivazione valore ottimo: Il valore ottimo coincide con l'accettabile. Trattando potenziali dati sensibili, la mancata crittografia espone a rischi non solo legali ma anche di reputazione.

4. Metodi di testing

I seguenti metodi di testing verranno adottati per garantire la qualità minima del prodotto:

- **Unit Testing_G**: test automatici per singole unità di codice, garantendo correttezza funzionale isolata.
- **Integration Testing_G**: verifica delle interazioni tra unità, assicurando che collaborino correttamente.
- **System Testing_G**: test end-to-end del Sistema completo in ambiente simulato.

4.1. Test di Unità

Codice	Descrizione (unitaria)	Requisiti di riferimento	Stato
T-U-1	Verificare la validazione del campo email nella richiesta di login (presenza e formato)	R-117-F	NI
T-U-2	Verificare la validazione del campo password nella richiesta di login	R-2-F	NI
T-U-3	Verificare il mapping dell'errore "credenziali errate"	R-3-F	NI
T-U-4	Verificare la generazione del segreto $TOTP_G$ (formato e lunghezza)	R-4-F	NI
T-U-5	Verificare la serializzazione del segreto $TOTP_G$	R-4-F	NI
T-U-6	Verificare la valutazione della condizione "2FA richiesto"	R-5-F	NI
T-U-7	Verificare la validazione dell'input OTP (solo cifre, lunghezza)	R-6-F	NI
T-U-8	Verificare il mapping dell'errore "OTP errato"	R-7-F	NI
T-U-9	Verificare la validazione della richiesta di recupero password	R-8-F	NI
T-U-10	Verificare il mapping dell'errore "account non esistente"	R-10-F	NI
T-U-11	Verificare la validazione della nuova password	R-11-F	NI
T-U-12	Verificare il mapping dell'errore "password non corrispondenti"	R-12-F	NI
T-U-13	Verificare il rispetto della policy di complessità password	R-13-F	NI
T-U-14	Verificare il mapping dello stato $gateway_G$ verso label testuale	R-21-F	NI
T-U-15	Verificare la formattazione del timestamp ultimo invio $gateway_G$	R-22-F	NI
T-U-16	Verificare la formattazione del timestamp ultimo invio sensore	R-23-F	NI

Codice	Descrizione (unitaria)	Requisiti di riferimento	Stato
T-U-17	Verificare la normalizzazione e serializzazione dello stream dati	R-24-F	NI
T-U-18	Verificare la trasformazione dello stream in struttura tabellare	R-25-F	NI
T-U-19	Verificare la trasformazione dello stream in serie per grafico	R-26-F	NI
T-U-20	Verificare la validazione dell'intervallo temporale ($\text{start} \leq \text{end}$)	R-29-F	NI
T-U-21	Verificare il mapping dello stato "dati non disponibili"	R-30-F	NI
T-U-22	Verificare la serializzazione dei dati per export	R-31-F	NI
T-U-23	Verificare la formattazione del valore fuori range	R-33-F	NI
T-U-24	Verificare la formattazione del range accettato	R-33-F	NI
T-U-25	Verificare la formattazione del timestamp di un dato irregolare	R-33-F	NI
T-U-26	Verificare il mapping del tipo di alert verso descrizione testuale	R-35-F	NI
T-U-27	Verificare il mapping dell'hardware interessato da un alert	R-35-F	NI
T-U-28	Verificare la formattazione del timestamp di emissione alert	R-35-F	NI
T-U-29	Verificare la validazione delle impostazioni di notifica	R-39-F, R-40-F	NI
T-U-30	Verificare la validazione del nome utente $tenant_G$	R-58-F	NI
T-U-31	Verificare la validazione dei ruoli assegnabili	R-60-F	NI
T-U-32	Verificare l'assenza di duplicati nella selezione utenti	R-64-F	NI
T-U-33	Verificare la generazione del Client ID	R-66-F	NI
T-U-34	Verificare la generazione del Client Secret	R-67-F	NI
T-U-35	Verificare la costruzione di una entry di $audit_G$ (campi obbligatori)	R-74-F	NI
T-U-36	Verificare il mapping dell'operazione di $audit_G$ verso testo	R-74-F	NI
T-U-37	Verificare la validazione di un token JWT_G (scadenza e formato)	R-86-F	NI

Codice	Descrizione (unitaria)	Requisiti di riferimento	Stato
T-U-38	Verificare la validazione dei permessi secondo policy	R-6-S	NI
T-U-39	Verificare il mapping della lista <i>gateway_G</i> simulati	R-S-1-F	NI
T-U-40	Verificare il mapping del <i>gateway_G</i> simulato	R-S-2-F	NI
T-U-41	Verificare la validazione della configurazione <i>gateway_G</i> simulato	R-S-11-F	NI
T-U-42	Verificare il mapping dell'errore di <i>deploy_G gateway_G</i> simulato	R-S-12-F	NI
T-U-43	Verificare la validazione della configurazione sensore simulato	R-S-14-F	NI
T-U-44	Verificare il mapping dell'errore range invalido sensore	R-S-15-F	NI

4.2. Test di Integrazione

Codice	Descrizione	Requisiti di riferimento	Stato
T-I-1	Verificare la corretta ricezione e validazione dei pacchetti <i>JSON_G</i> inviati dal Simulatore al Cloud Ingestion Service	R-113-F	NI
T-I-2	Verificare il flusso di controllo bidirezionale: invio e corretta applicazione dei comandi di configurazione dal Cloud al Simulatore	R-S-22-F	NI
T-I-3	Verificare la gestione del buffer locale del Simulatore e della successiva integrazione con il Cloud dopo un ripristino di rete	R-S-20-F	NI
T-I-4	Verificare la persistenza dei dati nel database e dell'isolamento dei dati per ciascun <i>Tenant_G</i>	R-113-F, R-2-S	NI
T-I-5	Verificare l'attivazione dei trigger di allerta tra il modulo di analisi dati e il servizio notifiche al superamento delle soglie	R-33-F	NI
T-I-6	Verificare l'integrazione del Sistema di autenticazione (<i>JWT_G</i>) tra l' <i>API Gateway_G</i> e i <i>microservizi_G</i> interni	R-6-S	NI
T-I-7	Verificare la gestione degli errori di configurazione del <i>Gateway_G</i> simulato in caso di payload invalido o frequenza fuori range	R-S-23-F, R-S-24-F	NI

Codice	Descrizione	Requisiti di riferimento	Stato
T-I-8	Verificare la propagazione e gestione di misure anomale (outlier) dal Simulatore al Cloud	R-S-21-F	NI
T-I-9	Verificare l'integrazione del Sistema con più <i>Gateway_G</i> simulati attivi in parallelo	R-S-17-F	NI
T-I-10	Verificare che i dati scambiati tra le diverse componenti del Sistema distribuito siano cifrati e trasmessi tramite protocollo <i>TLS_G</i> .	R-1-S	NI
T-I-11	Verificare l'isolamento dei dati tra <i>Tenant_G</i> tramite API e persistenza su database	R-2-S	NI

4.3. Test di Sistema

Codice	Descrizione	Requisiti di riferimento	Stato
T-S-001	Verificare che un utente non autenticato possa accedere al Sistema inserendo credenziali valide tramite interfaccia UI.	R-1-F, R-2-F, R-117-F	NI
T-S-002	Verificare che il Sistema neghi l'accesso e mostri un messaggio di errore in caso di credenziali errate.	R-3-F	NI
T-S-003	Verificare che un utente possa configurare correttamente il meccanismo 2FA durante il primo accesso.	R-4-F, R-3-S	NI
T-S-004	Verificare che un utente possa effettuare il login tramite codice OTP valido.	R-5-F, R-6-F, R-3-S	NI
T-S-005	Verificare che il Sistema notifichi un errore in caso di inserimento di codice OTP errato.	R-7-F	NI
T-S-006	Verificare che tutti gli endpoint UI e API rifiutino richieste provenienti da utenti non autenticati.	R-6-S	NI
T-S-007	Verificare che un utente non autenticato possa avviare la procedura di recupero password inserendo una mail valida.	R-8-F	NI
T-S-008	Verificare che il Sistema permetta la modifica della password tramite procedura di recupero.	R-9-F, R-11-F	NI
T-S-009	Verificare che il Sistema notifichi un errore in caso di email non associata ad alcun account.	R-10-F	NI

Codice	Descrizione	Requisiti di riferimento	Stato
T-S-010	Verificare che il sistema notifichi un errore in caso di password inserite non corrispondenti o non valide.	R-12-F, R-13-F	NI
T-S-011	Verificare che un utente autenticato possa modificare la propria email, con corretta validazione del formato, segnalazione di duplicati e richiesta di conferma.	R-14-F, R-15-F, R-16-F, R-17-F	NI
T-S-012	Verificare che un utente autenticato possa modificare la password del proprio account.	R-18-F	NI
T-S-013	Verificare che un utente autenticato possa effettuare il logout dal Sistema.	R-19-F	NI
T-S-014	Verificare che un utente autenticato possa visualizzare la lista dei <i>Gateway_G</i> associati al proprio <i>Tenant_G</i> .	R-20-F	NI
T-S-015	Verificare che il sistema mostri correttamente nome, stato, timestamp dell'ultimo invio dati e lista sensori di un <i>Gateway_G</i> selezionato.	R-21-F, R-22-F	NI
T-S-016	Verificare che il sistema mostri la lista dei sensori associati a un <i>Gateway_G</i> .	R-23-F	NI
T-S-017	Verificare che il sistema mostri correttamente l'ID e l'ultimo timestamp di invio dati di ciascun sensore.	R-23-F	NI
T-S-018	Verificare che il sistema visualizzi i dati di stream in formato tabellare.	R-24-F, R-25-F	NI
T-S-019	Verificare che il sistema visualizzi i dati di stream in formato grafico.	R-24-F, R-26-F	NI
T-S-020	Verificare che l'utente possa filtrare i dati per <i>Gateway_G</i> , sensore e intervallo temporale.	R-27-F, R-28-F, R-29-F	NI
T-S-021	Verificare che il sistema notifichi l'utente quando non sono disponibili dati per i filtri selezionati.	R-30-F	NI
T-S-022	Verificare che l'utente possa esportare i dati visualizzati in formato supportato.	R-31-F	NI
T-S-023	Verificare che il sistema generi un alert in caso di <i>Gateway_G</i> non raggiungibile.	R-32-F	NI
T-S-024	Verificare che il sistema generi un alert in caso di valore sensore fuori range.	R-33-F	NI
T-S-025	Verificare che l'utente possa visualizzare lo storico degli alert generati.	R-34-F	NI

Codice	Descrizione	Requisiti di riferimento	Stato
T-S-026	Verificare che il sistema mostri i dettagli completi di un alert selezionato.	R-35-F, R-36-F, R-37-F, R-38-F	NI
T-S-027	Verificare che l'utente possa attivare e disattivare la ricezione di notifiche alert via email e <i>dashboard_G</i> .	R-39-F, R-40-F	NI
T-S-028	Verificare che il <i>Tenant_G</i> Admin possa modificare il nome di un <i>Gateway_G</i> e ricevere errore in caso di nome duplicato.	R-41-F, R-42-F	NI
T-S-029	Verificare che il <i>Tenant_G</i> Admin possa modificare lo stato operativo di un <i>Gateway_G</i> selezionato.	R-43-F, R-44-F	NI
T-S-030	Verificare che il <i>Tenant_G</i> Admin possa modificare il range di allarme per un sensore specifico e per un tipo di sensore, ricevendo errore in caso di range invalido.	R-45-F, R-46-F, R-47-F, R-48-F, R-49-F, R-50-F, R-116-F	NI
T-S-031	Verificare che il <i>Tenant_G</i> Admin possa modificare il timeout che determina l'irraggiungibilità di un <i>Gateway_G</i> .	R-51-F	NI
T-S-032	Verificare che il <i>Tenant_G</i> Admin possa modificare la frequenza di invio dati di un <i>Gateway_G</i> .	R-81-F	NI
T-S-033	Verificare che il <i>Tenant_G</i> Admin possa visualizzare i costi stimati del <i>Tenant_G</i> , inclusi costi di storage e banda.	R-52-F, R-53-F, R-54-F	NI
T-S-034	Verificare che il <i>Tenant_G</i> Admin possa visualizzare la lista degli utenti del <i>Tenant_G</i> e i dettagli di un singolo utente.	R-55-F, R-56-F	NI
T-S-035	Verificare che il <i>Tenant_G</i> Admin possa creare, modificare ed eliminare uno o più utenti del <i>Tenant_G</i> .	R-57-F, R-58-F, R-59-F, R-60-F, R-61-F, R-62-F, R-63-F, R-64-F	NI
T-S-036	Verificare che il <i>Tenant_G</i> Admin possa creare, visualizzare e revocare credenziali API per il <i>Tenant_G</i> .	R-65-F, R-66-F, R-67-F, R-68-F, R-69-F, R-70-F, R-71-F	NI
T-S-037	Verificare che il <i>Tenant_G</i> Admin possa modificare le impostazioni del login 2FA per il <i>Tenant_G</i> .	R-72-F	NI
T-S-039	Verificare che un utente autorizzato possa consultare, filtrare per intervallo temporale,	R-73-F, R-74-F, R-75-F, R-76-F, R-77-F	NI

Codice	Descrizione	Requisiti di riferimento	Stato
	esportare e scaricare i log di <i>Audit_G</i> del <i>Tenant_G</i> .		
T-S-038	Verificare che il <i>Tenant_G</i> Admin e l'Amministratore di Sistema possano configurare le impostazioni di <i>impersonificazione_G</i> del <i>Tenant_G</i> .	R-78-F	NI
T-S-040	Verificare che il sistema registri correttamente tutte le operazioni rilevanti nei log di <i>Audit_G</i> .	R-4-S	NI
T-S-041	Verificare che l'Amministratore di Sistema possa visualizzare metriche di latenza, volume di traffico e utilizzo storage.	R-110-F	NI
T-S-042	Verificare che il <i>Tenant_G</i> Admin e l'Amministratore di Sistema possano selezionare uno o più <i>Gateway_G</i> e installare un aggiornamento firmware.	R-79-F, R-80-F	NI
T-S-043	Verificare che l'Amministratore di Sistema possa visualizzare la lista dei <i>Tenant_G</i> e i dettagli di un singolo <i>Tenant_G</i> , inclusi nome, stato, ID e intervallo di sospensione.	R-91-F, R-92-F, R-93-F, R-94-F, R-95-F, R-96-F, R-97-F	NI
T-S-044	Verificare che l'Amministratore di Sistema possa creare, sospendere, riattivare, modificare ed eliminare un <i>Tenant_G</i> , con corretta gestione di errori interni durante la creazione.	R-98-F, R-99-F, R-100-F, R-101-F, R-102-F, R-103-F, R-104-F, R-105-F, R-106-F	NI
T-S-045	Verificare che l'Amministratore di Sistema possa avviare una sessione di <i>impersonificazione_G</i> di un utente <i>Tenant_G</i> .	R-107-F	NI
T-S-046	Verificare che l'Amministratore di Sistema possa registrare un <i>Gateway_G</i> e associarlo a un <i>Tenant_G</i> di interesse.	R-108-F	NI
T-S-047	Verificare che l'Amministratore di Sistema possa creare un utente con ruolo Amministratore <i>Tenant_G</i> per un <i>Tenant_G</i> di interesse.	R-109-F	NI
T-S-048	Verificare che un Client API possa autenticarsi tramite le credenziali ottenute e ricevere notifica di errore in caso di credenziali invalide o server non disponibile.	R-82-F, R-83-F, R-84-F	NI

Codice	Descrizione	Requisiti di riferimento	Stato
T-S-049	Verificare che un Client API autenticato possa recuperare dati storici on-demand e ricevere notifiche di errore per token, ID $Gateway_G$, ID sensore o intervallo temporale non validi.	R-85-F, R-86-F, R-87-F, R-88-F, R-89-F	NI
T-S-050	Verificare che un Client API autenticato possa ricevere dati real-time tramite endpoint di streaming.	R-90-F	NI
T-S-051	Verificare che un $Gateway_G$ non provisioned possa completare correttamente il processo di onboarding e connettersi al Sistema.	R-111-F	NI
T-S-052	Verificare che il sistema notifichi errori di autenticazione per $Gateway_G$ con credenziali di fabbrica non valide o con processo di autenticazione fallito.	R-112-F, R-115-F	NI
T-S-053	Verificare che il $Gateway_G$ invii dati al Cloud tramite canale sicuro cifrato e che il Cloud li accetti correttamente.	R-113-F, R-114-F, R-1-S	NI
T-S-054	Verificare che l'utente del simulatore possa visualizzare la lista dei $Gateway_G$ simulati, i dettagli di ciascuno e gestire creazione, $deploy_G$ ed eliminazione.	R-S-1-F, R-S-2-F, R-S-3-F, R-S-4-F, R-S-8-F, R-S-10-F, R-S-11-F, R-S-12-F	NI
T-S-055	Verificare che l'utente del simulatore possa configurare e simulare sensori associati a un $Gateway_G$ simulato, inclusa la gestione degli errori di range e creazione.	R-S-5-F, R-S-6-F, R-S-7-F, R-S-9-F, R-S-13-F, R-S-14-F, R-S-15-F, R-S-16-F	NI
T-S-056	Verificare che il simulatore permetta di creare più istanze $Gateway_G$ in parallelo e di iniettare anomalie di rete e valori sensore anomali.	R-S-17-F, R-S-18-F, R-S-19-F, R-S-20-F, R-S-21-F	NI
T-S-057	Verificare che il Cloud possa inviare comandi di configurazione al $Gateway_G$ simulato e ricevere notifiche di errore per payload o frequenza non validi.	R-S-22-F, R-S-23-F, R-S-24-F	NI
T-S-058	Verificare la corretta trasmissione del flusso di dati tra sensore e relativo $gateway_G$	R-113-F	NI

5. Cruscotto di valutazione

Di seguito vengono presentate le misurazioni raccolte nel periodo compreso tra l'aggiudicazione del $capitolato_G$ e la definizione della Requirements and Technology *Baseline_G* (RTB_G).

5.1. MP01 e MP02: Earned Value e Planned Value

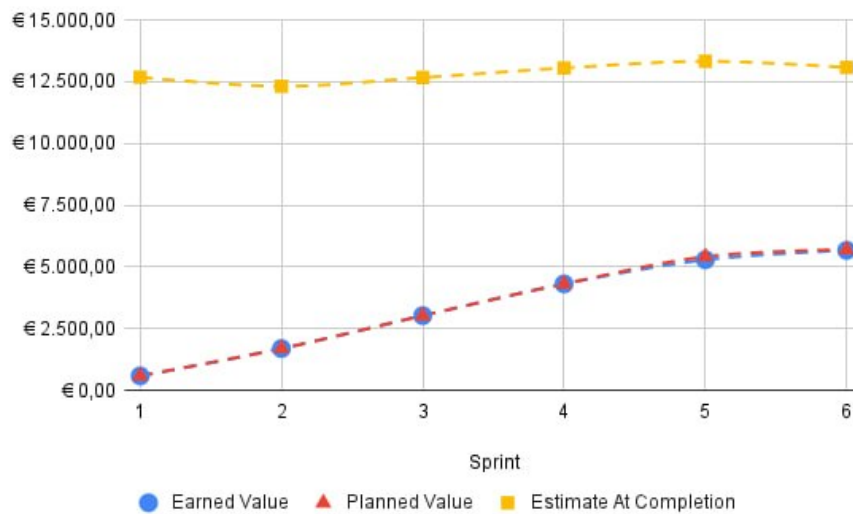


Figura 1: Grafico per *Sprint_G* di MP01 e MP02

Dal grafico non emergono scostamenti rilevanti tra andamento pianificato e andamento reale. Sia l'Earned Value (EV) sia la Planned Value (PV) mostrano un incremento progressivo lungo i sei *sprint_G* considerati. La quasi sovrapposizione delle due curve indica che il valore effettivamente maturato segue con buona precisione quanto previsto in fase di pianificazione. Gli scostamenti risultano minimi e non evidenziano ritardi o anticipi significativi rispetto al piano iniziale.

5.2. MP03 e MP07: Actual Cost e Estimate To Complete

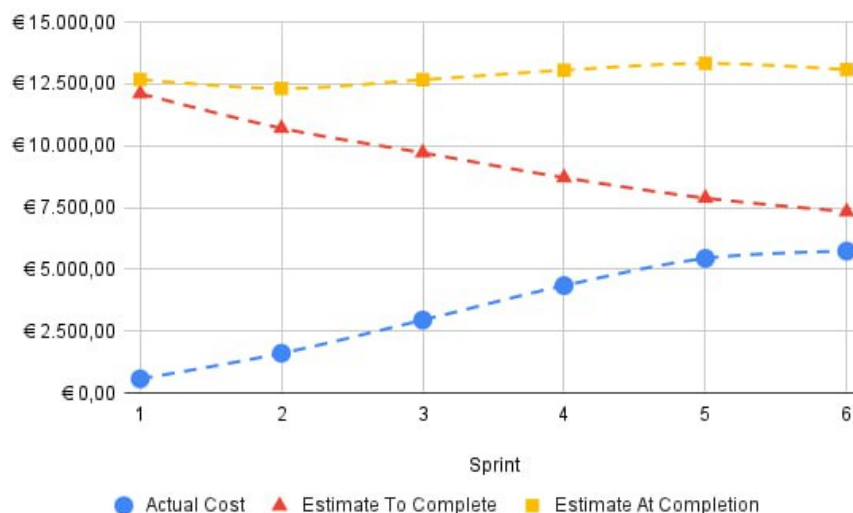


Figura 2: Grafico per *Sprint_G* di MP03 e MP07

Dal grafico non emergono particolari criticità nell'andamento dei costi. L'Actual Cost (AC) cresce progressivamente lungo i sei *sprint_G*, mostrando un incremento regolare e coerente con l'avanzamento delle attività. Questo andamento è spiegato dal naturale sviluppo del progetto: man mano che il lavoro viene completato, i costi effettivi aumentano. Parallelamente, l'Estimate To Complete (ETC) diminuisce in modo costante, indicando una progressiva riduzione del lavoro ancora da svolgere.

5.3. MP04 e MP05: Cost Performance Index e Schedule Performance Index

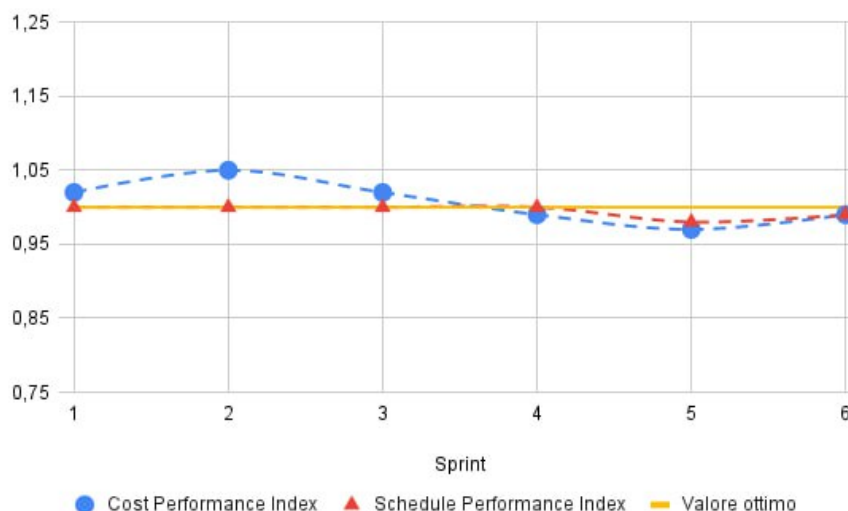


Figura 3: Grafico per $Sprint_G$ di MP04 e MP05

Dal grafico si osserva che nei primi $sprint_G$ il Cost Performance Index (CPI) risulta superiore a 1 (indicando un utilizzo efficiente del budget) e lo Schedule Performance Index (SPI) è in linea con la pianificazione. Tuttavia, in seguito il diagramma mostra una lieve flessione; evidenzia infatti che nelle fasi conclusive del periodo analizzato, le attività hanno richiesto un impegno leggermente maggiore rispetto a quanto previsto. Nonostante questa diminuzione, la situazione rimane complessivamente sotto controllo: gli scostamenti sono contenuti e non evidenziano criticità rilevanti.

5.4. MP06: Estimate At Completion



Figura 4: Grafico per $Sprint_G$ di MP06

Dal grafico si osserva che l'Estimate At Completion (EAC) tende progressivamente ad aumentare dallo $Sprint_G$ 2 in poi, arrivando a superare il Budget At Completion (BAC) nella parte centrale e finale del periodo considerato. Tale incremento è riconducibile a una temporanea riduzione della produttività del gruppo, dovuta alla concomitanza con il periodo degli esami universitari. Il rallentamento delle attività ha comportato un maggiore impiego di risorse

rispetto a quanto inizialmente pianificato, incidendo così sulla stima finale dei costi. Tuttavia, nell'ultimo *sprint_G* l'andamento dell'EAC mostra un'attenuazione della crescita e una tendenza alla stabilizzazione. Questo rappresenta un segnale positivo, in quanto indica una maggiore attenzione nella gestione delle risorse e un impegno concreto da parte del gruppo nel contenimento dei costi.

5.5. MP08: Time Estimate At Completion

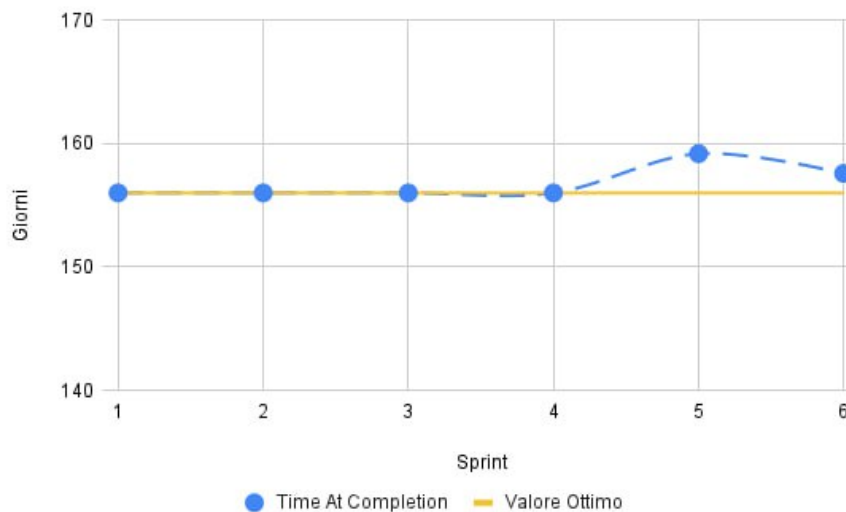


Figura 5: Grafico per *Sprint_G* di MP08

Dal grafico si osserva che il Time At Completion si mantiene sostanzialmente allineato al valore ottimale per i primi quattro *sprint_G*. Solo a partire dallo *sprint_G* 5 si registra un lieve aumento rispetto alla durata prevista, seguito da una leggera riduzione nello *sprint_G* 6, pur rimanendo leggermente sopra il valore ottimale. Lo scostamento registrato nella parte centrale può essere ricondotto a un rallentamento temporaneo delle attività, che ha inciso sulla previsione complessiva della durata del progetto. Tuttavia, tale variazione risulta contenuta e non evidenzia un allungamento significativo dei tempi.

5.6. MP09: Budget Burn Rate

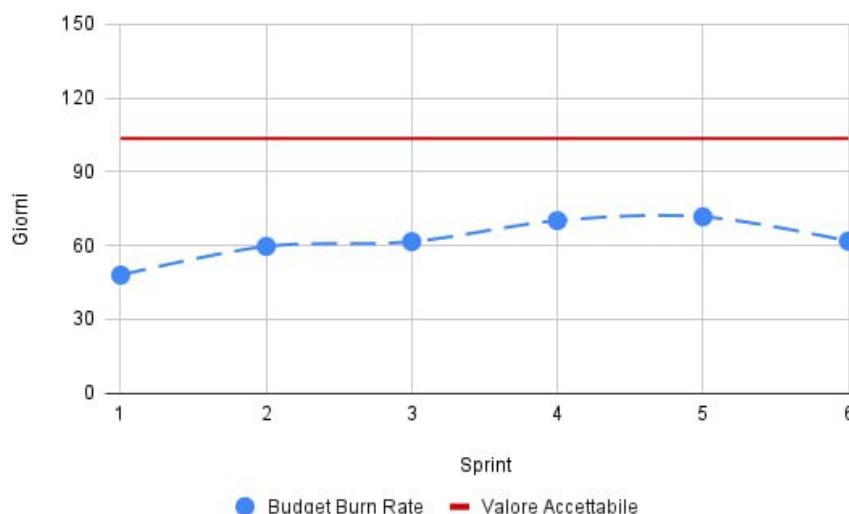


Figura 6: Grafico per *Sprint_G* di MP09

Dal grafico si osserva che il consumo medio di budget per *sprint_G* mostra un andamento inizialmente crescente. In particolare, dal primo al quarto *sprint_G* il valore aumenta progressivamente, passando da circa 50 giorni/uomo a un picco intorno ai 70 giorni/uomo. Nel quinto *sprint_G* il valore si mantiene su livelli simili al precedente, evidenziando una fase di stabilizzazione del ritmo di spesa. Nell'ultimo *sprint_G*, invece, si registra una lieve diminuzione del burn rate, che torna intorno ai 60 giorni/uomo. È comunque importante sottolineare che per l'intera durata del periodo considerato, il Budget Burn Rate rimane sempre al di sotto del valore soglia accettabile (pari a circa 100 giorni/uomo), rappresentato dalla linea rossa nel grafico. Questo aspetto evidenzia che il progetto ha mantenuto un livello di consumo delle risorse coerente con quanto pianificato, senza superare i limiti stabiliti.

5.7. MP11: Indice di Gulpease

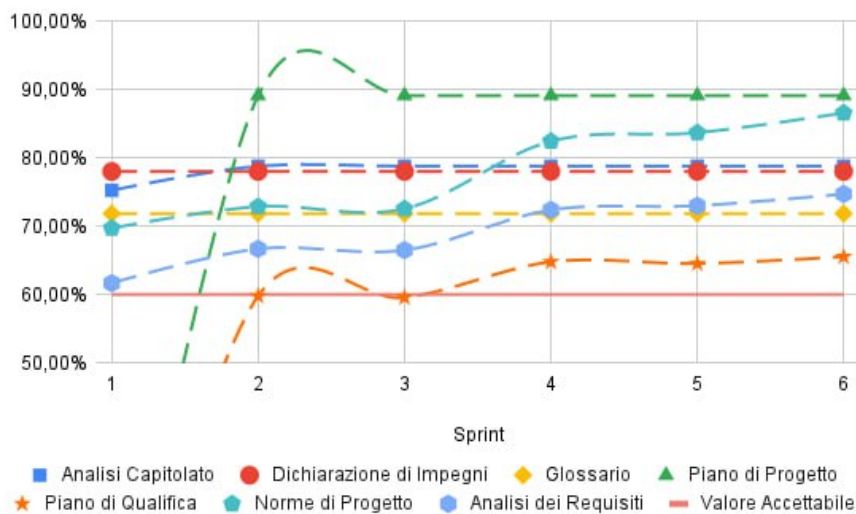


Figura 7: Grafico per *Sprint_G* di MP11

Dal grafico emerge che tutti i documenti mantengono un livello di avanzamento superiore alla soglia minima del 60% definita internamente dal gruppo. Nel complesso, l'andamento conferma una gestione della documentazione stabile e coerente con quanto fissato, senza scostamenti al di sotto del livello minimo previsto.

5.8. MP12: Correttezza Ortografica

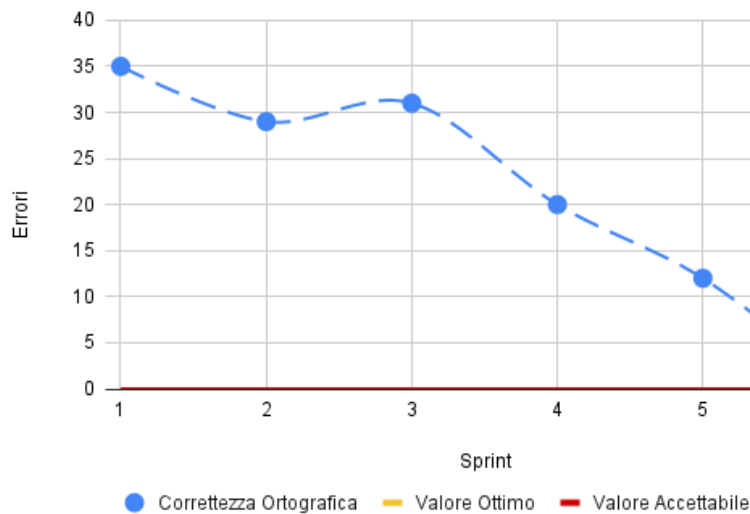


Figura 8: Grafico per $Sprint_G$ di MP12

Dal grafico si osserva un andamento complessivamente decrescente del numero di errori ortografici rilevati nei documenti prodotti durante gli $sprint_G$. Nel primo $sprint_G$ il valore iniziale risulta piuttosto elevato (circa 35 errori), segno di una fase iniziale in cui la revisione formale non era ancora pienamente strutturata. Nel secondo $sprint_G$ si registra una lieve diminuzione, seguita però da un piccolo incremento nel terzo $sprint_G$. A partire dal quarto $sprint_G$, tuttavia, il numero di errori cala in modo significativo, passando da circa 20 nel quarto $sprint_G$ a 12 nel quinto, fino ad azzerarsi completamente nel sesto $sprint_G$. Il progressivo avvicinamento e infine il raggiungimento del valore ottimo (e accettabile) dimostrano un miglioramento continuo nella qualità redazionale e una maggiore attenzione ai processi di revisione.

5.9. MP18: Quality Metrics Satisfied

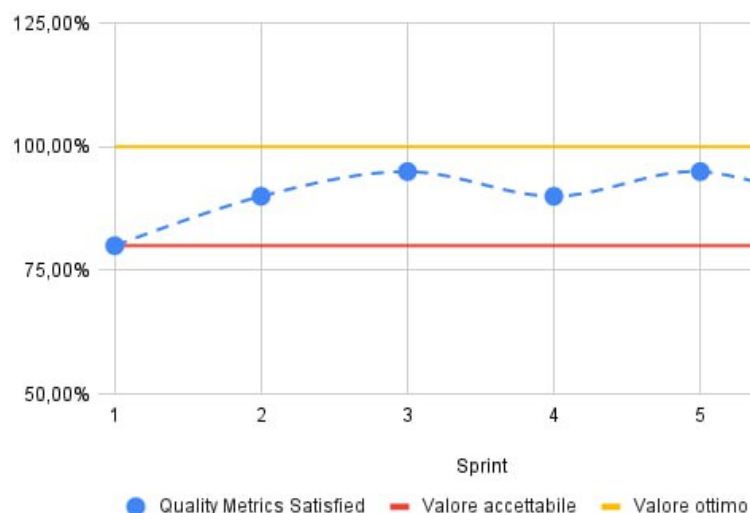


Figura 9: Grafico per $Sprint_G$ di MP18

Il grafico evidenzia come la percentuale di Quality Metrics Satisfied si sia mantenuta costantemente al di sopra della soglia di minimo accettabile (80%) per tutti gli $sprint_G$ considerati. Sebbene non sia stato raggiunto il target ottimale (100%), le lievi oscillazioni registrate non

hanno compromesso gli standard qualitativi prefissati. Tali risultati riflettono una gestione efficace della qualità e una pronta capacità di reazione del Team nel monitorare e mantenere il controllo sulle metriche prestabilite.

5.10. MP20: Time Efficiency

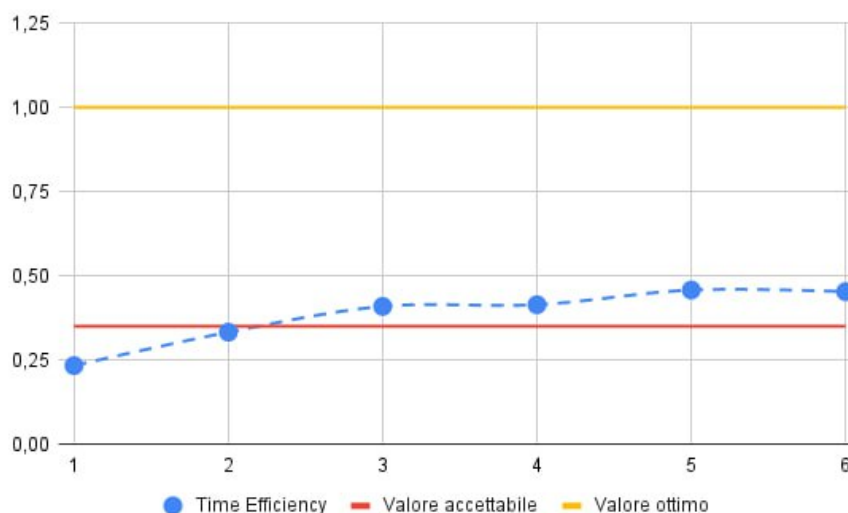


Figura 10: Grafico per *Sprint_G* di MP20

Dal grafico si osserva un progressivo incremento della Time Efficiency (TE) dallo *sprint_G* 1 allo *sprint_G* 6. Tuttavia, l'indicatore rimane costantemente al di sotto del valore ottimale, evidenziando un livello di efficienza temporale ancora migliorabile. Questo andamento è legato alla significativa incidenza delle “ore di palestra”, indispensabili al team per lo svolgimento dei *task_G* in analisi. Nonostante ciò, la crescita costante dell'indicatore dimostra un miglioramento graduale nella gestione del tempo. Inoltre, si prevede che, dopo la *RTB_G*, le ore di palestra tenderanno a diminuire, poiché il gruppo avrà ormai consolidato competenze e modalità operative. Questo comporterà un aumento delle ore effettivamente produttive e, di conseguenza, un incremento della Time Efficiency nei prossimi *sprint_G*.

5.11. MP21: *Sprint_G* Velocity Stability

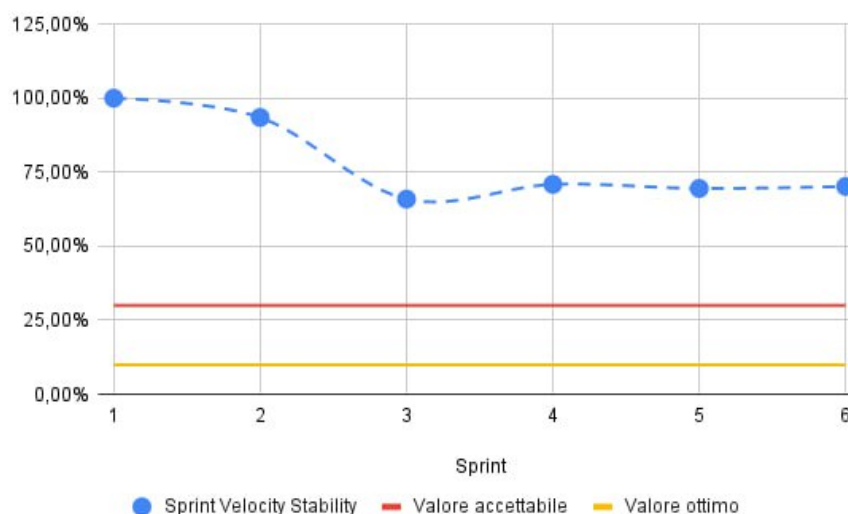


Figura 11: Grafico per *Sprint_G* di MP21

Dal grafico si osserva un andamento progressivamente migliorativo nel corso degli *sprint_G*. Poiché la metrica misura lo scostamento percentuale tra velocity pianificata e velocity effettiva, valori più elevati indicano una maggiore instabilità, mentre valori più bassi rappresentano una migliore capacità di previsione e quindi una maggiore stabilità. Nel primo *sprint_G* il valore si attesta intorno al 100%, evidenziando uno scostamento molto elevato e quindi una situazione iniziale critica. Nel secondo *sprint_G* si registra una lieve riduzione, ma il miglioramento più significativo si osserva nel terzo *sprint_G*, dove lo scostamento scende intorno al 65%. Negli *sprint_G* successivi il valore continua a ridursi gradualmente e tende a stabilizzarsi attorno al 70%, mostrando comunque una distanza dalla soglia di accettabilità (linea rossa). Sebbene il team non abbia ancora raggiunto il valore accettabile, il valore decrescente rappresenta un segnale positivo: lo scostamento si sta progressivamente riducendo e la prevedibilità della velocity sta migliorando.

5.12. MP22: Meeting Efficiency Index

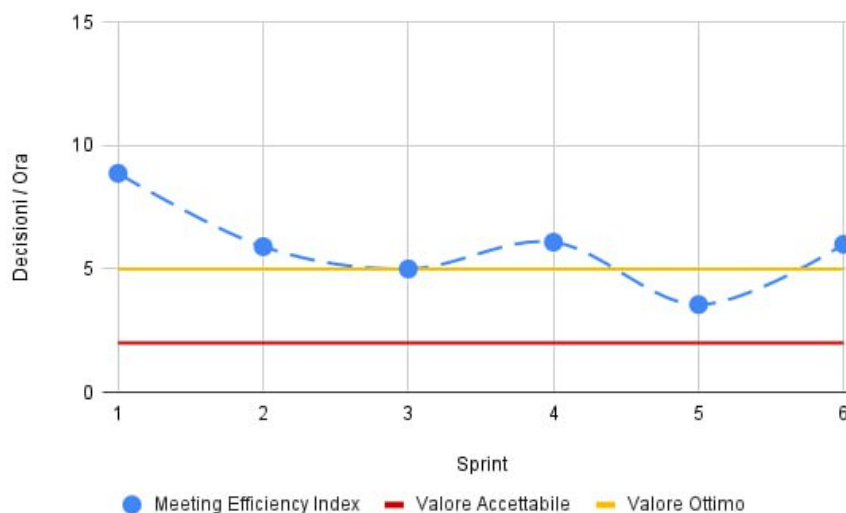


Figura 12: Grafico per *Sprint_G* di MP22

Dal grafico si osserva un andamento variabile nel corso degli *sprint_G*, con valori che oscillano attorno al valore ottimo fissato a 5 decisioni/ora (linea gialla). Poiché la metrica misura il numero medio di decisioni prese per ora di riunione, valori più elevati indicano una maggiore efficienza, mentre valori più bassi riflettono un *minor_G* rendimento degli incontri. Nel primo *sprint_G* l'indice si attesta intorno a 9 decisioni/ora, valore significativamente superiore a quello ottimo. Negli *sprint_G* successivi si osserva una progressiva riduzione: nel secondo *sprint_G* il valore scende a circa 6, mentre nel terzo si allinea al valore ottimo di 5 decisioni/ora, evidenziando un buon equilibrio tra produttività e qualità delle decisioni. Nel quarto *sprint_G* si registra un lieve incremento, seguito però da un calo più marcato nel quinto *sprint_G* (circa 3–4 decisioni/ora), valore che si avvicina alla soglia di accettabilità (linea rossa), indice di riunioni meno focalizzate e di fasi in cui le decisioni effettive risultano meno numerose. Nell'ultimo *sprint_G* si osserva una ripresa, con un ritorno a circa 6 decisioni/ora, riportando l'indicatore sopra il valore ottimo.

5.13. MP23: PR_G Resolution Time

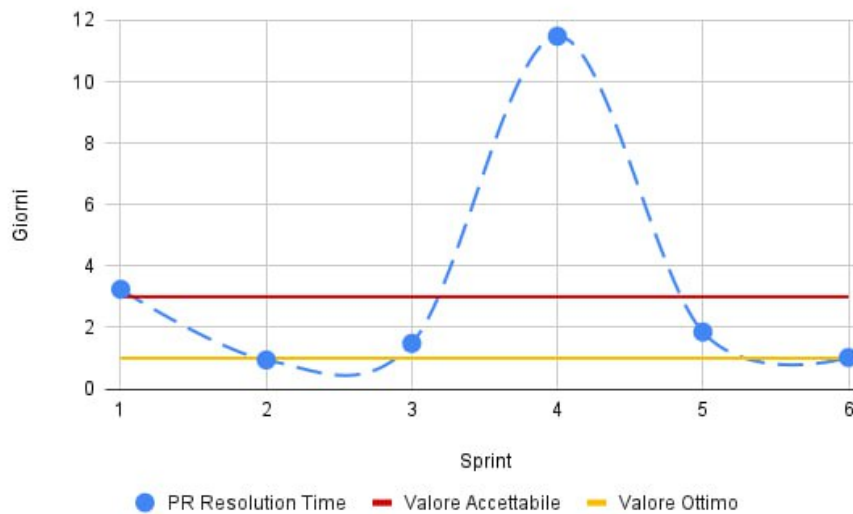


Figura 13: Grafico per $Sprint_G$ di MP23

Dal grafico si osserva un andamento complessivamente variabile nel tempo, con un picco significativo nella parte centrale del periodo considerato. La metrica misura il numero medio di giorni necessari per la risoluzione delle Pull Request: trattandosi di un indicatore temporale, valori più bassi sono preferibili, poiché indicano maggiore reattività e fluidità nel processo di revisione. Nel primo $sprint_G$ il tempo medio di risoluzione si attesta intorno ai 3 giorni, valore in prossimità della soglia di accettabilità (linea rossa). Nel secondo $sprint_G$ si registra un miglioramento significativo, con una riduzione a circa 1 giorno, in linea con il valore ottimo (linea gialla). Anche il terzo $sprint_G$ mantiene un tempo contenuto, sebbene leggermente superiore rispetto al precedente. Nel quarto $sprint_G$ si verifica un marcato peggioramento, con un picco di circa 11–12 giorni, ampiamente superiore alla soglia accettabile. A partire dal quinto $sprint_G$ si osserva un recupero significativo: il tempo medio scende nuovamente sotto la soglia accettabile (circa 2 giorni) e nell'ultimo $sprint_G$ si riporta in prossimità del valore ottimo. Questo andamento evidenzia la capacità del team di reagire a una criticità temporanea e di ristabilire un processo di revisione più efficiente.

6. Valutazioni per l'automiglioramento

Al fine di perseguire un miglioramento continuo durante lo svolgimento del progetto, è opportuno effettuare valutazioni periodiche. Tali valutazioni hanno l'obiettivo di individuare le criticità emerse e le relative soluzioni adottate per affrontarle, consentendo al gruppo di acquisire una maggiore consapevolezza e di ridurre il rischio di ripetere gli stessi errori in futuro. Le analisi effettuate si basano sulle tre categorie di rischio definite nel *Piano di Progetto v1.0.0*, ovvero:

- Rischi connessi alle tecnologie adottate (**RT1**);
- Rischi connessi all'organizzazione del team (**RO1**);
- Rischi connessi ai singoli componenti del gruppo (**RP1**).

6.1. Valutazione sulle tecnologie adottate

Problema	Soluzione adottata	Tecnologia coinvolta
La gestione del codice condiviso può generare conflitti e perdita di tracciabilità.	Utilizzo di <i>repository_G</i> <i>GitHub_G</i> con <i>branch_G</i> protection, pull request e Sistema di <i>issue_G</i> tracking.	<i>GitHub_G</i>
La pianificazione e il monitoraggio delle attività possono risultare disorganizzati senza uno strumento dedicato.	Utilizzo di <i>Jira_G</i> per la gestione del <i>backlog_G</i> , la pianificazione degli <i>sprint_G</i> e il monitoraggio dello stato delle attività.	<i>Jira_G</i>
Alcuni membri del gruppo non avevano familiarità con il linguaggio <i>Go_G</i> , causando una curva di apprendimento iniziale.	Ogni membro ha intrapreso un percorso di autoapprendimento mirato, supportato da documentazione ufficiale ed esempi pratici.	Linguaggio <i>Go_G</i>
La maggior parte del team non aveva esperienza con tecnologie di messaging come <i>NATS_G</i> .	Studio della documentazione ufficiale e realizzazione di prototipi incrementali per validarne l'utilizzo nel <i>PoC_G</i> .	<i>NATS_G</i> <i>JetStream_G</i>
L'utilizzo di un database time-series basato su estensione <i>PostgreSQL_G</i> non era noto a tutti i membri.	Approfondimento delle funzionalità principali (Hypertables, partizionamento) e utilizzo guidato nelle prime fasi di sviluppo.	<i>PostgreSQL_G</i> + <i>TimescaleDB_G</i>
La struttura modulare di <i>NestJS_G</i> può risultare inizialmente complessa.	Adozione delle convenzioni ufficiali del framework e suddivisione chiara dei moduli per facilitare la comprensione.	<i>TypeScript_G</i> / <i>NestJS_G</i>
Il framework <i>Angular_G</i> richiede una buona organizzazione dei componenti e dei servizi.	Suddivisione dell'interfaccia in componenti con responsabilità ben definite e organizzazione della struttura del progetto seguendo le <i>linee guida_G</i> della documentazione.	<i>Angular_G</i>
La gestione della decifratura lato client introduce complessità applicativa.	Implementazione semplificata con chiavi statiche per dimostrare la fattibilità dell'approccio nel <i>PoC_G</i> .	Cifratura end-to-end (Zero Knowledge).

6.2. Valutazione sull'organizzazione del team

Problema	Soluzione adottata
La difficoltà nel coordinamento del lavoro asincrono tra i membri del team ha rallentato il progresso del progetto.	Pianificazione di riunioni regolari per allineare gli obiettivi e le attività del team, oltre all'adozione di strumenti di gestione del progetto per migliorare la comunicazione.

Problema	Soluzione adottata
La comunicazione della rendicontazione delle ore lavorative durante la retrospettiva di <i>sprint_G</i> ha inizialmente causato un rallentamento del flusso organizzativo.	Utilizzo di <i>Jira_G</i> per tracciare le ore lavorative in modo più efficiente e trasparente, riducendo il tempo dedicato alla rendicontazione manuale.
La gestione non strutturata delle priorità delle attività ha portato a ritardi nell'implementazione di funzionalità chiave.	Definizione delle priorità a inizio <i>sprint_G</i> e loro revisione durante la pianificazione.
Rallentamento delle attività di progetto nel periodo della sessione d'esami invernale.	Pianificazione delle attività in modo da recuperare il ritardo una volta terminati gli esami.

6.3. Valutazione sui singoli componenti del gruppo

Problema	Soluzione adottata
Impegni esterni di alcuni componenti del gruppo hanno causato temporanee indisponibilità.	Comunicazione immediata al team e riorganizzazione delle attività per garantire la continuità del lavoro.
Differenti livelli di esperienza tra i membri del gruppo.	Collaborazione attiva e supporto reciproco per garantire un avanzamento uniforme delle attività.