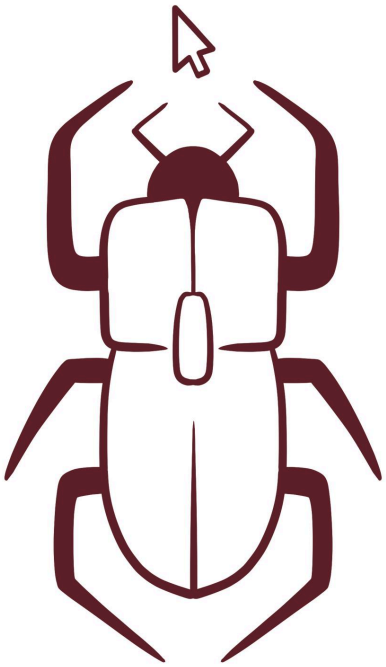


Piano di Qualifica

v1.0.0

Registro delle Modifiche

Data	Versione	Descrizione	Redattore	Verificatore
2026/03/02	1.0.0	Ufficializzazione per RTB		Suar Alberto
2026/02/26	0.11.0	Aggiunta grafici mancanti	Zago Alice	Suar Alberto
2026/02/25	0.10.0	Aggiunta Test di Unità mancanti	Zago Alice	Suar Alberto
2026/02/06	0.9.0	Test di Accettazione e rielaborazione Test di Sistema	Zago Alice	Suar Alberto
2026/01/27	0.8.1	Aggiunta grafico BV-SV, revisione Metodi di Testing e aggiunta test, modifiche minori al documento	Zago Alice	Suar Alberto
2026/01/22	0.7.0	Grafici CPI-SPI, EAC, RSI e SGA	Zago Alice	Suar Alberto
2026/01/21	0.6.0	Cruscotto di valutazione, grafico PV-AC-EV	Zago Alice	Suar Alberto
2026/01/17	0.5.2	Revisione Automiglioramento	Suar Alberto	Zago Alice
2026/01/13	0.5.1	Rielaborazione introduzione documento e qualità di processo	Suar Alberto	Zago Alice
2025/12/02	0.5.0	Modifica tabelle qualità di processo, inserimento tabelle qualità di prodotto	Zago Alice	Suar Alberto
2025/12/30	0.4.0	Iniziati metodi di testing, inserimento tabelle test	Berengan Riccardo	Suar Alberto
2025/12/28	0.3.0	Processi secondari e processi organizzativi con tabelle soglie metriche, iniziata sezione	Zago Alice	Suar Alberto

Data	Versione	Descrizione	Redattore	Verificatore
		automiglioramento e qualità di prodotto		
2025/12/27	0.2.0	Qualità di processo, processi primari	Zago Alice	Suar Alberto
2025/12/26	0.1.0	Inizio stesura documento, introduzione, scopo e riferimenti	Zago Alice	Suar Alberto
2025/12/23	0.0.0	Creazione documento	Zago Alice	Suar Alberto

Indice

Indice tabelle

1 Introduzione

1.1 Contesto del Progetto

Il presente documento descrive il Piano di Qualifica [G] relativo al progetto Code Guardian [G], commissionato dall'azienda Var Group [G] e realizzato dal team di sviluppo Skarab Group [G] nell'ambito del corso di Ingegneria del Software presso l'Università degli Studi di Padova.

Il progetto ha come obiettivo la realizzazione di un sistema per l'automazione dei processi di audit [G] e remediation [G] delle vulnerabilità del software. L'architettura si basa sul paradigma degli agenti [G] software intelligenti, operanti su repository di codice sorgente. La conformità del sistema è vincolata ai requisiti definiti nel Capitolato **C2**.

La piattaforma supporta attività di analisi statica del codice sorgente e di individuazione delle principali criticità di sicurezza, fornendo suggerimenti di correzione attraverso meccanismi automatizzati basati su modelli di linguaggio di grandi dimensioni (LLM [G]).

1.2 Finalità del Documento

Il Piano di Qualifica definisce l'impostazione metodologica per la gestione della qualità, specificando come il gruppo intenda prevenire, rilevare e correggere i difetti.

Il documento costituisce il riferimento primario per il Responsabile [G] e per i Verificatori [G], strutturando gli obiettivi nelle seguenti macro-aree:

- **Piano della Qualità (Quality Assurance)**: definizione della strategia di gestione della qualità, identificando gli standard di riferimento (in particolare ISO/IEC 25010 [G]), le metriche di misurazione e le relative soglie di accettazione/ottimalità.
- **Controllo di Qualità (Quality Control)**: pianificazione operativa delle attività di Verifica [G] (analisi statica, test dinamici) per garantire la correttezza tecnica degli artefatti prodotti.
- **Validazione di Prodotto**: definizione delle procedure necessarie per accertare che il sistema soddisfi i bisogni effettivi degli Stakeholder [G] e i requisiti del capitolato.
- **Miglioramento Continuo**: applicazione di meccanismi retroattivi (basati sul ciclo Plan-Do-Check-Act [G]) che utilizzano i risultati delle misurazioni per ottimizzare i processi e il **Way of Working** [G] in corso d'opera.

1.3 Traguardi Qualitativi

L'assicurazione della qualità segue l'approccio incrementale del progetto, fissando obiettivi specifici per le due principali milestone:

1.3.1 Revisione dei Requisiti e della Tecnologia (RTB)

Per la milestone RTB (**25/02/2026**), le attività di qualità si concentrano sulla correttezza formale e sulla fattibilità tecnica:

- **Qualità dei Documenti**: Verifica approfondita della documentazione (Analisi dei Requisiti, PdP, NdP) tramite analisi statica e walkthrough, per garantire assenza di ambiguità e coerenza interna (Indice di Gulpease).
- **Qualità del Prototipo (PoC [G])**: L'attività di verifica è focalizzata esclusivamente sulla **dimostrazione della fattibilità tecnica** (Technology Baseline), con particolare attenzione all'interazione Agenti-LLM. Il testing in questa fase ha valore *sperimentale e propedeutico*: esso funge da caso di studio per calibrare le metriche e validare le strategie di verifica che saranno poi applicate in modo sistematico ed estensivo sul MVP [G].

1.3.2 Revisione di Accettazione (Product Baseline – PB)

Per il rilascio finale (**15/04/2026**), il focus si sposta sulla robustezza, sulla copertura e sulla soddisfazione dei requisiti:

- **Qualità del Prodotto (MVP)**: Esecuzione completa dei test di unità, integrazione e sistema. Validazione finale rispetto ai requisiti funzionali e prestazionali del capitolato.
- **Qualità del Codice**: Rispetto dei vincoli di stile, assenza di **code smells** [G] e raggiungimento delle soglie di copertura del codice Code Coverage [G] definite nel presente piano.
- **Validazione Utente**: Verifica dell'usabilità tramite test di accettazione (UAT) basati sui casi d'uso principali.

1.4 Glossario

Al fine di prevenire ambiguità interpretative, è stato redatto un glossario che definisce in modo univoco la terminologia tecnica, gli acronimi e i concetti di dominio utilizzati all'interno della documentazione.

Nel testo, **ogni termine evidenziato tramite una G come apice**, rimanda alla voce corrispondente del Glossario pubblicato sul sito ufficiale del gruppo, consentendo al lettore di accedere direttamente alla definizione associata.

La versione più recente del Glossario è disponibile al seguente link: Link al Glossario (v1.0.0).

1.5 Riferimenti

1.5.1 Riferimenti Normativi

I seguenti documenti hanno valore vincolante per la definizione delle strategie di qualità e per le attività di verifica:

- **Capitolato C2**: Piattaforma ad agenti per l'audit e la remediation dei repository software.
  https://www.math.unipd.it/~tullio/IS-1/2025/Progetto/C2.pdf
  (ultimo accesso: **24/02/2026**)
- **Norme di Progetto**: Il documento definisce il "Way of Working", stabilendo gli strumenti e le procedure che questo Piano si occupa di misurare.
  https://skarabgroup.github.io/DocumentazioneProgetto/RTB/NdP.pdf
  (versione: **v1.0.0**)

1.5.2 Riferimenti Informativi

- **ISO/IEC 25010:2011**: Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE).
  https://iso25000.com/index.php/en/iso-25000-standards/iso-25010
  (ultimo accesso: **24/02/2026**)
- **ISO/IEC 12207:2008**: Systems and software engineering — Software life cycle processes.
  https://ieeexplore.ieee.org/document/4475826
  (ultimo accesso: **24/02/2026**)
- **Dispense del corso di Ingegneria del Software – Qualità del software**
  https://www.math.unipd.it/~tullio/IS-1/2025/Dispense/T06.pdf
  (ultimo accesso: **24/02/2026**)

2 Qualità di Processo

La garanzia della qualità del prodotto finale è intrinsecamente legata alla qualità dei processi produttivi che lo generano. Per il progetto *Code Guardian*, la gestione dei processi mira a rendere il Way of Working [G] sostenibile, tracciabile e soggetto a miglioramento continuo attraverso l'applicazione del ciclo PDCA.

2.1 Centralizzazione delle Metriche e Obiettivi

Il presente documento costituisce il riferimento unico e autoritativo per la gestione della qualità del progetto Code Guardian. Il Piano di Qualifica centralizza la 'scienza della misurazione' del gruppo, definendo rigorosamente i criteri analitici, le metriche e le soglie necessarie per garantire la conformità degli artefatti agli standard prefissati.

In particolare, ogni metrica qui esposta è corredata da:

- **Identificativo univoco:** (MPC per il processo, MPD per il prodotto);
- **Formulazione matematica:** Per garantire l'oggettività del calcolo;
- **Soglie di Valutazione:** Distinte in "Accettabilità" (requisito minimo per la validazione) e "Ottimalità" (target di eccellenza desiderato).

Ogni scostamento rilevato tra i valori misurati e le soglie qui definite viene analizzato durante le retrospettive di fine Sprint [G]. Tali evidenze costituiscono la base oggettiva per l'attivazione di azioni correttive o per la ricalibrazione delle soglie stesse, garantendo che il processo di qualità evolva insieme alla maturità del team.

2.1.1 Processi Primari: Fornitura e Sviluppo

Questi processi definiscono le attività core per la realizzazione del software. Il monitoraggio si focalizza sul rispetto dei vincoli di tempo e budget (tramite la metodologia EVM) e sulla gestione rigorosa dell'ambito di progetto.

ID	Nome	Formula	V. Accettabile	V. Ottimo
MPC01	Budget at Completion (BAC)	Preventivo	Preventivo	Preventivo
MPC02	Planned Value (PV)	PV	≥ 0	Da Piano
MPC03	Actual Cost (AC)	AC	$\leq EAC$	$\leq EV$
MPC04	Earned Value (EV)	EV	$\geq 90\% PV$	$\geq PV$
MPC05	Budget Variance (BV)	$BV = BAC - EAC$	≥ 0	> 0
MPC06	Schedule Variance (SV)	$SV = EV - PV$	$> -10\%$ BAC	≥ 0
MPC07	Cost Performance Index (CPI)	$CPI = \frac{EV}{AC}$	$0.90 \leq v \leq 1.10$	1.00
MPC08	Schedule Performance Index (SPI)	$SPI = \frac{EV}{PV}$	$0.90 \leq v \leq 1.10$	1.00
MPC09	Estimate at Completion (EAC)	$EAC = \frac{BAC}{CPI}$	$\leq BAC + 5\%$	$\leq BAC$

Table 1: Soglie metriche per il processo di Fornitura (EVM)

Il monitoraggio della stabilità dei requisiti è cruciale per prevenire lo **scope creep**, specialmente a seguito delle revisioni correttive post-S2.

ID	Nome	Formula	V. Accettabile	V. Ottimo
MPC10	Requirements Stability Index	$RSI = \frac{R_{\text{tot}} - \Delta R}{R_{\text{tot}}} \times 100$	$\geq 75\%$	100%

Table 2: Soglie metriche per il processo di Sviluppo

2.1.2 Processi di Supporto

I processi di supporto garantiscono l'integrità e la verificabilità degli artefatti. La leggibilità della documentazione (Indice di Gulpease) e la copertura dei test sono i parametri cardine per assicurare la manutenibilità futura.

ID	Nome	Formula	V. Accettabile	V. Ottimo
MPC11	Gulpease Index	$89 + \frac{300(L_f) - 10(L_p)}{F_p}$	≥ 40	≥ 60
MPC12	Correttezza Ortografica	Errori segnalati	0	0
MPC13	Code Coverage	$\frac{\text{Linee coperte}}{\text{Linee totali}} \times 100$	$\geq 70\%$	$\geq 80\%$
MPC14	Test Success Rate	$\frac{\text{Passati}}{\text{Eseguiti}} \times 100$	100%	100%

Table 3: Soglie metriche Documentazione e Verifica

2.1.3 Processi Organizzativi

Misurano l'efficienza interna del team Skarab Group nell'auto-organizzarsi e nel rispettare gli impegni presi durante gli Sprint.

ID	Nome	Formula	V. Accettabile	V. Ottimo
MPC15	Metrics Satisfaction	$\frac{\text{Metriche OK}}{\text{Metriche Tot}} \times 100$	$\geq 90\%$	100%
MPC16	Sprint Goal Achievement	$\frac{\text{Completati}}{\text{Pianificati}} \times 100$	$\geq 80\%$	100%

Table 4: Soglie metriche Organizzative

3 Qualità di Prodotto

La qualità di prodotto valuta il software consegnato rispetto ai requisiti e alle caratteristiche intrinseche definite dallo standard ISO/IEC 25010.

3.1 Adeguatezza Funzionale e Affidabilità

Si misura la capacità del sistema di svolgere i compiti richiesti e di rimanere operativo senza guasti critici, parametro fondamentale per un tool di audit.

ID	Nome	Formula	V. Accettabile	V. Ottimo
MPD01	Copertura Req. Obbligatori	$\frac{\text{Soddisfatti}}{\text{Totale Obbl.}} \times 100$	100%	100%
MPD02	Failure Density	$\frac{\text{N. guasti}}{\text{KLOC}}$	≤ 0.5	0
MPD03	Availability	$\frac{\text{Tempo Up}}{\text{Tempo Tot}} \times 100$	$\geq 98\%$	$\geq 99.9\%$

Table 5: Metriche Adeguatezza e Affidabilità

3.2 Manutenibilità e Sicurezza

Data la natura del progetto Code Guardian, queste metriche rappresentano il valore distintivo del prodotto. Un codice manutenibile e privo di vulnerabilità è condizione necessaria per l'accettazione.

ID	Nome	Formula	V. Accettabile	V. Ottimo
MPD04	Comment Density	$\frac{\text{Linee commento}}{\text{Linee codice}} \times 100$	$\geq 15\%$	$20\% - 25\%$
MPD05	Cyclomatic Complexity	$V(G)$	≤ 15	≤ 10
MPD06	Coupling (Fan-out)	Dipendenze esterne	≤ 6	≤ 3
MPD07	Vulnerability Detection	N. vulnerabilità critiche	0	0

Table 6: Metriche Manutenibilità e Sicurezza

4 Strategie di Testing

Il processo di testing rappresenta una fase cruciale nello sviluppo del prodotto *CodeGuardian*.

Skarab Group ha adottato un approccio di testing multilivello che copre:

- **Test di Sistema**.
- **Test di Unità**.
- **Test di Accettazione**.
- **Test di Regressione**.
- **Test di Integrazione**.

La definizione dei test e la nomenclatura utilizzata sono presenti all'interno delle **Norme di Progetto**, alla sezione **2.1.9.3.5**. I Test di Regressione e i Test di Integrazione, qui non presenti, verranno identificati durante lo svolgimento delle attività per la *Product Baseline* (PB).

4.1 Test di Sistema

ID Test	Descrizione	UC Riferimento
TS-1	Verificare la corretta creazione di un account CodeGuardian a seguito dell'inserimento di dati validi.	UC1
TS-1.1	Verificare la validazione dello username rispetto ai vincoli di formato (alfanumerico, 4-20 caratteri).	UC1.1.1
TS-1.2	Verificare l'inibizione della registrazione in caso di username già associato a un account esistente.	UC1.1.2
TS-1.3	Verificare la validazione sintattica dell'indirizzo email secondo gli standard previsti.	UC1.2.1
TS-1.4	Verificare l'inibizione della registrazione in caso di email già associata a un account esistente.	UC1.2.2
TS-1.5	Verificare la validazione della password rispetto ai criteri di complessità (sicurezza).	UC1.3.1
TS-1.6	Verificare la segnalazione di errore in caso di invio del modulo con campi obbligatori vuoti.	UC1.0.1
TS-2	Verificare l'accesso alle funzionalità riservate tramite inserimento di credenziali corrette.	UC2
TS-2.1	Verificare la validazione del formato delle credenziali in fase di login.	UC2.1.1, UC2.2.1
TS-2.2	Verificare la segnalazione di errore per identificativo (username) non presente a sistema.	UC2.1.2

ID Test	Descrizione	UC Riferimento
TS-2.3	Verificare la segnalazione di errore in caso di password non corrispondente all'identificativo fornito.	UC2.2.2
TS-2.4	Verificare l'inibizione dell'accesso in caso di modulo di login incompleto.	UC2.0.1
TS-3	Verificare il corretto completamento del flusso di autorizzazione OAuth con GitHub.	UC3
TS-3.1	Verificare l'annullamento della procedura a seguito del rifiuto del reindirizzamento esterno.	UC3.1.1
TS-3.2	Verificare la gestione degli errori tecnici durante lo scambio dei parametri di autorizzazione.	UC3.2.1
TS-3.3	Verificare l'inibizione del collegamento se il profilo GitHub risulta già associato a un altro utente.	UC3.2.2
TS-3.4	Verificare la gestione del mancato consenso dell'utente sulla piattaforma GitHub.	UC3.2.3
TS-4	Verificare la corretta presa in carico di una richiesta di analisi per un repository GitHub.	UC4
TS-4.1	Verificare la segnalazione dell'informativa se l'analisi risulta già aggiornata rispetto ai dati remoti.	UC4.0.1
TS-4.2	Verificare la segnalazione dell'informativa se esiste un'analisi già pendente per lo stesso repository.	UC4.0.2
TS-4.3	Verificare l'inibizione della richiesta se non viene selezionata almeno un'area di interesse.	UC4.1.1
TS-5	Verificare la navigazione e la corretta visualizzazione dell'elenco dei repository analizzati.	UC5, UC5.1
TS-5.1	Verificare la visualizzazione dell'informativa "lista vuota" in assenza di repository analizzati.	UC5.0.1
TS-6	Verificare il caricamento della dashboard di dettaglio a seguito della selezione di un report.	UC6
TS-6.1	Verificare l'aggiornamento dinamico delle sezioni visibili tramite i filtri (Codice, Sicurezza, Doc.).	UC6.1

ID Test	Descrizione	UC Riferimento
TS-6.2	Verificare l'inibizione del rendering e la notifica se non viene selezionata alcuna area.	UC6.1.1
TS-6.3	Verificare l'esposizione corretta dei metadati di audit (timestamp, hash commit, richiedente).	UC6.2.1, UC6.2.2, UC6.2.3
TS-6.4	Verificare la visualizzazione del messaggio di assenza di criticità se non vi sono remediation.	UC6.3.1.1
TS-7	Verificare la generazione della vista comparativa previo inserimento di un intervallo valido.	UC7
TS-7.1	Verificare la segnalazione di errore in caso di invio con campi temporali incompleti.	UC7.0.1
TS-7.2	Verificare la segnalazione di assenza dati se non vi sono report nel periodo scelto.	UC7.0.2
TS-7.3	Verificare la segnalazione di errore in caso di data inizio successiva alla data fine.	UC7.0.3
TS-7.4	Verificare l'inibizione della richiesta se l'intervallo supera l'ampiezza massima (12 mesi).	UC7.0.4
TS-8	Verificare la corretta generazione dei grafici di andamento e della tabella comparativa.	UC8
TS-8.1	Verificare l'esposizione dei dati puntuali all'interazione (click/hover) con il grafico.	UC8
TS-8.2	Verificare il calcolo e la visualizzazione degli indicatori di trend in tabella.	UC8
TS-9.1	Verificare l'esposizione dei rilievi di analisi statica (bug, smell, vulnerabilità).	UC9.1
TS-9.2	Verificare la visualizzazione delle metriche di copertura dei test di unità.	UC9.2
TS-9.3	Verificare la visualizzazione dell'informativa di esito positivo per l'area codice.	UC9.3.1
TS-10.1	Verificare l'esposizione delle vulnerabilità delle librerie e conformità OWASP.	UC10.1, UC10.2
TS-10.2	Verificare la visualizzazione dell'informativa di assenza criticità di sicurezza.	UC10.3.1

ID Test	Descrizione	UC Riferimento
TS-11.1	Verificare la visualizzazione degli errori sintattici e della completezza documentale.	UC11.1, UC11.2
TS-11.2	Verificare la visualizzazione dell'informativa di assenza criticità documentali.	UC11.3.1
TS-12	Verificare la generazione della graduatoria ordinata per punteggio di qualità globale.	UC12
TS-12.1	Verificare la segnalazione di assenza dati se l'utente non ha mai effettuato analisi.	UC12.1
TS-13	Verificare l'effettiva revoca dell'associazione e dei token dell'account GitHub.	UC13
TS-14	Verificare il corretto download del report nel formato selezionato (PDF/JSON).	UC14, UC14.2
TS-14.1	Verificare la segnalazione di errore se l'utente non seleziona alcun formato.	UC14.1.1
TS-15.1	Verificare la segnalazione di errore in caso di password corrente omessa.	UC15.1.1
TS-15.2	Verificare la segnalazione di errore in caso di password corrente errata.	UC15.1.2
TS-15.3	Verificare la segnalazione di errore se la nuova password è assente o non conforme.	UC15.2.1, UC15.2.2
TS-15.4	Verificare la segnalazione di errore se la nuova password coincide con la precedente.	UC15.2.3
TS-15.5	Verificare la corretta persistenza e la notifica di successo post-modifica.	UC15.3, UC15.4
TS-16	Verificare la corretta visualizzazione dei dettagli di una remediation selezionata.	UC16
TS-17	Verificare che l'Orchestratore verifichi con successo l'accessibilità di un repository pubblico tramite le API GitHub.	UC17
TS-17.1	Verificare la gestione dell'errore di comunicazione con GitHub.	UC17.1.1

ID Test	Descrizione	UC Riferimento
TS-17.2	Verificare che l'Orchestratore tenti l'accesso tramite credenziali o token OAuth in caso di repository privato non raggiungibile pubblicamente.	UC17.2.1
TS-17.3	Verificare che l'Orchestratore annulli l'audit se tutti i metodi di accesso falliscono.	UC17.2.1.1
TS-18	Verificare che l'Utente Avanzato possa accettare una singola remediation.	UC18
TS-19	Verificare che l'Utente Avanzato possa rifiutare una singola remediation.	UC19
TS-20	Verificare la corretta creazione di una raccolta di report a seguito dell'inserimento di nome e URL repository validi.	UC20
TS-20.1	Verificare la segnalazione di errore in caso di tentativo di conferma con campi obbligatori non popolati.	UC20.0.1
TS-20.2	Verificare la corretta acquisizione del nome identificativo della raccolta nel campo dedicato.	UC20.1
TS-20.3	Verificare la segnalazione di errore in caso di nome raccolta non conforme ai vincoli alfanumerici o di lunghezza.	UC20.1.1
TS-20.4	Verificare la corretta acquisizione dell'URL del repository GitHub nel campo dedicato.	UC20.2
TS-20.5	Verificare la segnalazione di errore in caso di URL sintatticamente non valido (protocollo o dominio errati).	UC20.2.1
TS-20.6	Verificare la segnalazione di errore in caso di repository non esistente o non accessibile tramite le API GitHub.	UC20.2.2
TS-20.7	Verificare la segnalazione di errore in caso di campo URL non popolato al momento della conferma.	UC20.2.3
TS-20.8	Verificare la corretta acquisizione della descrizione della raccolta.	UC20.3

ID Test	Descrizione	UC Riferimento
TS-21	Verificare che l'Orchestratore avvii le richieste di analisi verso tutti gli strumenti esterni (Codice, Sicurezza, Doc.).	UC21
TS-21.1	Verificare la corretta clonazione del repository nell'ambiente AWS e la disponibilità del codice sorgente per l'elaborazione.	UC21.1
TS-21.2	Verificare che l'Orchestratore interrompa il processo e liberi le risorse allocate in caso di errore durante la clonazione.	UC21.1.1
TS-21.3	Verificare che l'Orchestratore inoltri i file sorgente allo strumento di analisi del codice al completamento della clonazione.	UC21.2
TS-21.4	Verificare che l'Orchestratore inoltri i file di documentazione allo strumento di analisi documentale.	UC21.3
TS-21.5	Verificare che l'Orchestratore inoltri la codebase allo strumento di analisi della sicurezza.	UC21.4
TS-22	Verificare che lo stato dell'analisi venga registrato correttamente come "pending" nella persistenza a seguito dell'inizializzazione degli strumenti.	UC22
TS-22.1	Verificare che, in caso di errore critico nella scrittura dello stato, l'Orchestratore notifichi l'utente.	UC22.0.1
TS-23	Verificare che l'Orchestratore recuperi correttamente i risultati al completamento delle analisi degli strumenti esterni.	UC23
TS-23.1	Verificare che il sistema proceda con i soli dati disponibili e aggiorni il database in caso di risultati parziali per timeout o errore di uno strumento.	UC23.0.1
TS-23.2	Verificare il corretto controllo periodico dello stato delle attività degli strumenti e il rilevamento del completamento.	UC23.1
TS-23.3	Verificare che i file dei risultati vengano acquisiti e validati prima dell'utilizzo.	UC23.2

ID Test	Descrizione	UC Riferimento
TS-24	Verificare la corretta aggregazione dei dati provenienti dai diversi strumenti (Codice, Sicurezza, Doc.) in un unico report strutturato.	UC24
TS-24.1	Verificare che il report venga validato prima del salvataggio.	UC24
TS-25	Verificare che il report finale venga archiviato permanentemente e lo stato dell'analisi venga aggiornato a "completed".	UC25
TS-25.1	Verificare la notifica di errore all'utente in caso di fallimento del salvataggio del report.	UC25.0.1
TS-26	Verificare l'invio della notifica di completamento dell'analisi del repository.	UC26
TS-26.1	Verificare che il fallimento della notifica venga registrato nei log interni anche in caso di errore di invio.	UC26.0.1
TS-27	Verificare la corretta ricezione della notifica di fine analisi attraverso i canali configurati dall'utente.	UC27
TS-27.1	Verificare che i risultati dell'analisi siano consultabili nella dashboard anche in assenza di ricezione della notifica.	UC27.0.1
TS-28	Verificare l'invio dell'avviso di errore critico all'utente in caso di interruzione imprevista del processo.	UC28
TS-28.1	Verificare che lo stato di fallimento sia visibile nella dashboard indipendentemente dalla ricezione dell'avviso di errore.	UC28.0.1
TS-29	Verificare lo scambio corretto del codice OAuth temporaneo con un token di accesso persistente e cifrato associato al profilo utente.	UC29
TS-29.1	Verificare l'annullamento del collegamento e la notifica all'utente in caso di codice OAuth scaduto o non valido.	UC29.0.1
TS-30	Verificare la corretta visualizzazione del dettaglio di una singola remediation dell'area codice.	UC30

ID Test	Descrizione	UC Riferimento
TS-31	Verificare la corretta visualizzazione del dettaglio di una singola remediation dell'area sicurezza.	UC31
TS-32	Verificare la corretta visualizzazione del dettaglio di una singola remediation dell'area documentazione.	UC32
TS-33	Verificare l'applicazione della remediation del codice alla codebase e l'aggiornamento dello stato a "eseguita" nella dashboard.	UC33
TS-33.1	Verificare la notifica di fallimento all'utente e l'invarianza della codebase in caso di errore durante l'applicazione.	UC33.0.1
TS-34	Verificare che il rifiuto di una remediation del codice aggiorni lo stato a "rifiutata" senza apportare modifiche al repository.	UC34
TS-35	Verificare l'applicazione delle patch di sicurezza previste dalla remediation accettata e l'aggiornamento dello stato a "eseguita" nella dashboard.	UC35
TS-35.1	Verificare la notifica di fallimento all'utente in caso di errore durante l'applicazione della remediation di sicurezza.	UC35.0.1
TS-36	Verificare che il rifiuto di una remediation di sicurezza aggiorni lo stato a "rifiutata" senza modificare il repository.	UC36
TS-37	Verificare l'applicazione delle modifiche documentali previste dalla remediation accettata e l'aggiornamento dello stato a "eseguita" nella dashboard.	UC37
TS-37.1	Verificare la notifica di fallimento all'utente e l'invarianza della documentazione in caso di errore durante l'applicazione.	UC37.0.1
TS-38	Verificare che il rifiuto di una remediation documentale aggiorni lo stato a "rifiutata" e non apporti modifiche ai file del repository.	UC38

ID Test	Descrizione	UC Riferimento
TS-39	Verificare la corretta presa in carico di una richiesta di analisi per un repository privato da parte di un Utente Avanzato.	UC39
TS-40	Verificare il corretto inserimento di un repository privato nel catalogo personale dell'Utente Avanzato.	UC40
TS-40.1	Verificare la segnalazione di duplicazione in caso di inserimento di un URL già presente nel catalogo personale.	UC40.0.1
TS-41	Verificare la corretta visualizzazione del catalogo dei repository privati inseriti dall'Utente Avanzato.	UC41
TS-41.1	Verificare la visualizzazione dell'informativa specifica quando il catalogo privato risulta vuoto.	UC41.0.1
TS-42	Verificare la corretta rimozione di un repository dal catalogo privato previa conferma esplicita dell'utente.	UC42, UC42.1
TS-42.1	Verificare che l'annullamento della rimozione mantenga intatto il catalogo privato.	UC42.1.1
TS-43	Verificare la corretta visualizzazione dell'elenco dei profili autorizzati alla consultazione dei report per un repository privato selezionato.	UC43
TS-43.1	Verificare la visualizzazione dell'informativa di assenza utenti autorizzati quando la lista è vuota.	UC43.0.1
TS-44	Verificare la corretta aggiunta di un utente autorizzato tramite username o email con validazione del profilo nel sistema.	UC44, UC44.1
TS-44.1	Verificare la segnalazione di errore per formato identificativo non valido, utente inesistente, utente già autorizzato e campo vuoto.	UC44.1.1, UC44.1.2, UC44.1.3, UC44.1.4
TS-45	Verificare la corretta revoca dei permessi di consultazione per un utente precedentemente autorizzato, previa conferma del proprietario.	UC45, UC45.1
TS-46	Verificare la corretta rimozione di una raccolta di report senza che i singoli report in essa contenuti vengano eliminati.	UC46, UC46.1

ID Test	Descrizione	UC Riferimento
TS-46.1	Verificare che l'annullamento dell'operazione mantenga intatta la raccolta nel profilo utente.	UC46.1.1
TS-47	Verificare la corretta cancellazione del profilo CodeGuardian a seguito della verifica dell'identità tramite password e della conferma definitiva dell'utente.	UC47, UC47.1
TS-47.1	Verificare che a seguito della cancellazione vengano rimossi i dati personali e le associazioni OAuth e che le credenziali precedenti risultino invalide.	UC47

Table 7: Tabella dei Test di Sistema

4.2 Test di Unità

ID Test	Requisito	Descrizione Test	Risultato Atteso
TU-1.1	FRObl	Verifica rendering del componente di creazione account.	Caricamento corretto del modulo di registrazione.
TU-1.2	FRObl2	Verifica predisposizione comando di conferma nel modulo di registrazione.	Pulsante di conferma presente e funzionante.
TU-1.3	FRObl3	Verifica esecuzione della validazione completa di tutti i campi obbligatori all'invio.	Tutti i controlli vengono eseguiti prima di procedere.
TU-1.4	FRObl4	Verifica che la finalizzazione sia consentita solo dopo validazione positiva di tutti i campi.	Blocco della procedura in caso di parametri non validi.
TU-1.5	FRObl5	Verifica logica di persistenza dei dati utente nel DB.	Le credenziali vengono scritte correttamente nel database.
TU-1.6	FRObl6	Verifica della funzione di hashing: algoritmo sicuro + salt univoco.	La password non è leggibile; l'hash prodotto è coerente con gli standard.
TU-1.7	FRObl7	Verifica atomicità della registrazione.	In caso di errore nessun record parziale viene mantenuto nel database.
TU-1.8	FRObl8	Verifica visualizzazione messaggio di conferma avvenuta creazione account.	Messaggio di conferma mostrato a seguito di registrazione completata.
TU-1.9	FRObl9	Controllo rilevamento campi obbligatori vuoti (null check).	Rilevamento campi vuoti nel modulo di registrazione.
TU-1.10	FRObl10	Verifica logica di inibizione e notifica per campi mancanti.	Impossibilità di procedere; messaggio specifico per ogni campo assente.
TU-1.11	FRObl11	Verifica input username: vincoli alfanumerici e lunghezza (4-20 caratteri).	Rifiuto di stringhe < 4 o > 20 caratteri o con caratteri non alfanumerici.
TU-1.12	FRObl12	Query di verifica unicità dello username nel database.	Identificazione di collisioni con account già esistenti.

ID Test	Requisito	Descrizione Test	Risultato Atteso
TU-1.13	FRObi3	Verifica vincolo di unicità lato persistenza su username.	Il sistema rileva la duplicazione e annulla la registrazione.
TU-1.14	FRObi4	Verifica notifica username già in uso a seguito di violazione unicità.	Feedback visivo immediato per username non disponibile.
TU-1.15	FRObi5	Verifica inibizione e trigger notifica errore formato username non conforme.	Comparsa del messaggio di errore; procedura inibita.
TU-1.16	FRObi6	Verifica input email e validazione sintattica secondo standard RFC.	Accettazione di formati standard (user@domain.ext).
TU-1.17	FRObi7	Verifica rifiuto di email con spazi o prive del carattere “@”.	Email malformate rifiutate con messaggio di errore.
TU-1.18	FRObi8	Query di verifica unicità email nel database.	Identificazione di email già associate ad altri profili.
TU-1.19	FRObi9	Verifica vincolo di unicità lato persistenza su email.	Il sistema impedisce registrazioni duplicate anche con richieste concorrenti.
TU-1.20	FRObi20	Verifica trigger notifica errore email non valida o già registrata.	Messaggio di errore per email duplicata o malformata; procedura inibita.
TU-1.21	FRObi21	Verifica requisito lunghezza password (minimo 8 caratteri).	Password con meno di 8 caratteri rifiutate.
TU-1.22	FRObi22	Verifica requisiti complessità password.	Validazione positiva solo se tutti i criteri di complessità sono soddisfatti.
TU-1.23	FRObi23	Verifica rifiuto password coincidente o contenente lo username.	Password che contengono lo username come sottostringa rifiutate.
TU-1.24	FRObi24	Verifica trigger notifica errore password non conforme ai requisiti di sicurezza.	Elenco puntuale dei criteri non rispettati.
TU-2.1	FRObi25	Verifica rendering pagina di Login.	Visualizzazione corretta del form di autenticazione.

ID Test	Requisito	Descrizione Test	Risultato Atteso
TU-2.2	FRObb26	Verifica predisposizione comando di conferma per il login.	Pulsante di conferma presente e funzionante.
TU-2.3	FRObb27	Verifica validazione completa credenziali all'invio.	Procedura inibita se uno dei controlli fallisce.
TU-2.4	FRObb28	Verifica autorizzazione post-validazione credenziali.	Rilascio della sessione solo con dati corretti; accesso alle funzionalità riservate.
TU-2.5	FRObb29	Verifica reindirizzamento verso dashboard a seguito di autenticazione avvenuta.	L'utente viene reindirizzato correttamente dopo il login con successo.
TU-2.6	FRObb30	Verifica protocollo di trasmissione credenziali (HTTPS).	Dati cifrati durante il transito verso il server.
TU-2.7	FRObb31	Verifica utilizzo username per fetch del record account dalla persistenza.	Lo username viene utilizzato per recuperare il record account nel DB.
TU-2.8	FRObb32	Verifica confronto hash password fornita con hash memorizzato (stessa funzione e salt).	Accesso concesso solo se coincidono.
TU-2.9	FRObb33	Test del meccanismo di rate limiting / lockout temporaneo.	Blocco dell'account dopo N tentativi falliti consecutivi.
TU-2.10	FRObb34	Verifica visualizzazione spinner durante validazione credenziali.	Indicatore di caricamento mostrato; bottone disabilitato per prevenire invii multipli.
TU-2.11	FRObb35	Verifica rilevamento campi mancanti e inibizione del login.	Trigger errore per username o password non inseriti; accesso negato.
TU-2.12	FRObb36	Verifica notifica formato username non conforme in fase di login.	Messaggio di errore su formato username errato; procedura inibita.
TU-2.13	FRObb37	Verifica notifica username non esistente nel sistema.	Feedback specifico per username non censito.
TU-2.14	FRObb38	Verifica notifica formato password errato in login.	Feedback su errore sintattico password; procedura inibita.

ID Test	Requisito	Descrizione Test	Risultato Atteso
TU-2.15	FRObb39	Verifica notifica password errata (hash non corrisponde).	Feedback specifico per credenziali non corrispondenti.
TU-3.1	FRObb40	Verifica accesso sezione collegamento del profilo GitHub.	Sezione raggiungibile.
TU-3.2	FRObb41	Verifica impedimento accesso se GitHub già associato.	Procedura bloccata se account GitHub già collegato.
TU-3.3	FRObb42	Verifica utilizzo parametro “state” per prevenzione attacchi CSRF nel flusso OAuth2.	Stato CSRF generato e verificato; richieste prive di state valido rifiutate.
TU-3.4	FRObb43	Verifica cifratura Access Token GitHub nel database (AES-256 o equivalente).	I token GitHub non sono salvati in chiaro; algoritmo di cifratura forte applicato.
TU-3.5	FRObb44	Verifica assenza di persistenza token in caso di procedura fallita.	Nessun token o associazione parziale mantenuta in caso di errore.
TU-3.6	FRObb45	Verifica rendering avviso informativo obbligatorio pre-reindirizzamento su dominio esterno.	Visualizzazione informativa di sicurezza prima del redirect verso GitHub.
TU-3.7	FRObb46	Verifica gestione comando annulla redirect con ripristino stato.	Esecuzione o blocco del redirect; in caso di annullamento la sezione integrazioni viene ripristinata.
TU-3.8	FRObb47	Verifica elaborazione esito e visualizzazione messaggio al ritorno da GitHub.	Esito della procedura di associazione mostrato all’utente al rientro su CodeGuardian.
TU-3.9	FRObb48	Verifica gestione errore sincronizzazione.	Feedback su fallimento tecnico della procedura; possibilità di ripetere.
TU-3.10	FRObb49	Verifica blocco associazione per profilo GitHub già in uso su altro account.	Inibizione collegamento se account GitHub già censito su diverso profilo.

ID Test	Requisito	Descrizione Test	Risultato Atteso
TU-3.11	FRObo50	Verifica gestione errore “Consenso Negato” su GitHub.	Feedback specifico per interruzione volontaria dell’utente su piattaforma esterna.
TU-4.1	FRObo51	Verifica predisposizione campo URL nel modulo di richiesta analisi.	Campo URL presente e funzionante nel modulo.
TU-4.2	FRObo52	Verifica vincoli sintattici URL: protocollo “https://” e dominio “github.com”.	Validazione positiva solo per URL conformi.
TU-4.3	FRObo53	Verifica controllo dimensione repository tramite API GitHub.	Analisi inibita se il repository supera i limiti tecnici prestabiliti.
TU-4.4	FRObo54	Verifica blocco invio se report già aggiornato.	Informativa su analisi già aggiornata; nuova analisi impedita.
TU-4.5	FRObo55	Verifica disabilitazione comando di conferma dopo prima pressione.	Bottone disabilitato dopo il click per prevenire richieste duplicate.
TU-4.6	FRObo56	Verifica rilevamento processi analisi attivi e impedimento invio duplicato.	Feedback su audit già pendente per il medesimo repository.
TU-4.7	FRObo57	Verifica inibizione richiesta analisi in assenza di selezione aree di interesse.	Messaggio di errore se nessuna area è selezionata.
TU-5.1	FRObo58	Verifica ordinamento lista repository per data ultima analisi (decrescente).	Repository visualizzati in ordine dall’analisi più recente.
TU-5.2	FRObo59	Controllo mapping metadati: Nome, URL, Data ultima analisi per ogni elemento.	Oggetti lista popolati correttamente con tutti i campi richiesti.
TU-5.3	FRObo60	Validazione logica per visualizzazione informativa lista vuota.	Messaggio informativo mostrato.

ID Test	Requisito	Descrizione Test	Risultato Atteso
TU-5.4	FRObi61	Verifica inibizione rendering lista e notifica errore in caso di persistenza irraggiungibile.	Messaggio di errore tecnico mostrato; lista non renderizzata.
TU-5.5	FRObi62	Verifica presenza e funzionamento comando di aggiornamento (Refresh) in caso di errore.	Nuovo tentativo di caricamento avviato al click.
TU-6.1	FRObi63	Verifica selezione e caricamento report da lista.	Caricamento riuscito dei dati del report selezionato.
TU-6.2	FRObi64	Verifica validazione lato server: report appartiene al repository dell'utente.	Rendering inibito con errore di autorizzazione per report non associati al profilo.
TU-6.3	FRObi65	Verifica inibizione rendering per report non autorizzati.	Errore di autorizzazione per report non associati al profilo.
TU-6.4	FRObi66	Verifica gestione timeout nel recupero dati analitici con notifica utente.	Notifica di indisponibilità temporanea mostrata in caso di timeout.
TU-6.5	FRObi67	Verifica stato dei filtri aree (toggle on/off).	Le aree analitiche (Codice, Sicurezza, Doc.) sono correttamente filtrate.
TU-6.6	FRObi68	Verifica aggiornamento dinamico contenuto in base ai filtri senza ricaricamento pagina.	Il report si aggiorna alla variazione dei filtri.
TU-6.7	FRObi69	Controllo validazione: almeno un'area attiva nei filtri.	Blocco visualizzazione con avviso informativo se nessuna sezione selezionata.
TU-6.8	FRObi70	Verifica esposizione metadati identificativi del report.	Metadati caricati correttamente.
TU-6.9	FRObi71	Verifica correttezza timestamp generazione audit (formato ISO 8601).	Data e ora corrispondono al record del database.
TU-6.10	FRObi72	Verifica corrispondenza SHA commit GitHub con link diretto al commit.	Link verso GitHub funzionante.

ID Test	Requisito	Descrizione Test	Risultato Atteso
TU-6.11	FRObi73	Controllo visualizzazione username richiedente report.	Lo username corrisponde all'autore della richiesta di analisi.
TU-6.12	FRObi74	Verifica integrità metriche tecniche aggregate per aree attive.	Dati numerici (punteggi, numero bug, vulnerabilità) visualizzati correttamente.
TU-6.13	FRObi75	Verifica caricamento e visualizzazione lista azioni correttive (remediation).	Lista remediation caricata correttamente e associata alle criticità rilevate.
TU-6.14	FRObi77	Verifica espansione dettaglio singola remediation.	Dettaglio tecnico della proposta di risoluzione visibile.
TU-6.15	FRObi78	Controllo messaggio esito positivo in assenza di criticità.	Badge di conformità mostrato per aree senza criticità rilevate.
TU-7.1	FRObi79	Verifica selezione intervallo temporale (data inizio e fine) tramite input di data.	Intervallo acquisito correttamente dai campi di input.
TU-7.2	FRObi80	Verifica predisposizione e invio comando aggiornamento confronto.	Trigger di ricalcolo comparativo attivato correttamente.
TU-7.3	FRObi81	Controllo campi obbligatori temporali: inibizione se non popolati.	Avviso mostrato e confronto inibito per input di data nulli.
TU-7.4	FRObi82	Validazione coerenza: data inizio cronologicamente precedente alla data fine.	Blocco e segnalazione errore se la data d'inizio è successiva alla fine.
TU-7.5	FRObi83	Controllo ampiezza massima intervallo.	Errore restituito e richiesta inibita.
TU-7.6	FRObi84	Verifica query di ricerca report in intervallo: informativa assenza dati.	Messaggio "Nessun report trovato" restituito se il range è privo di analisi.
TU-8.1	FRDe1	Verifica logica di generazione dataset per grafici dinamici (linee/ istogrammi).	Dati trasformati correttamente in serie storiche visualizzabili.

ID Test	Requisito	Descrizione Test	Risultato Atteso
TU-8.2	FRDe2	Verifica tooltip informativi all'hover sui punti dati del grafico.	Valore esatto e hash commit mostrati al passaggio del cursore.
TU-8.3	FRDb85	Verifica allineamento dati tra vista grafica e tabellare.	Dati coerenti tra grafico e tabella.
TU-8.4	FRDb86	Verifica popolamento righe tabella comparativa in ordine cronologico crescente.	Ordinamento cronologico corretto.
TU-8.5	FRDb87	Validazione algoritmo calcolo indicatori di variazione (trend incrementali/decrementali).	Calcolo variazione percentuale tra analisi consecutive eseguito correttamente.
TU-8.6	FRDb88	Verifica fallback visualizzazione dati grezzi in tabella in caso di errore rendering grafico.	In assenza di librerie grafiche i dati vengono mostrati in formato tabellare.
TU-9.1	FRDb89	Verifica caricamento sezione "Codice" solo se area attiva nei filtri.	Modulo renderizzato con parametri corretti; non caricato se area non attiva.
TU-9.2	FRDb90	Verifica esposizione risultati analisi statica con gravità e posizione nel file sorgente.	Bug, code smell e vulnerabilità mostrati con livello di gravità e riferimento al file e riga.
TU-9.3	FRDb92	Controllo calcolo percentuale copertura test (Code Coverage) e rapporto test superati/falliti.	Valore normalizzato 0-100% e conteggio test calcolati correttamente.
TU-9.4	FRDb93	Verifica presentazione lista remediation codice con navigazione al dettaglio.	Soluzioni visualizzate con link al dettaglio della singola azione correttiva.
TU-9.5	FRDb94	Controllo informativa "Codice Conforme" in assenza di bug o violazioni qualitative.	Esito positivo mostrato.
TU-10.1	FRDb95	Verifica caricamento asincrono sezione sicurezza rispetto alle altre sezioni.	I dati di sicurezza vengono caricati in modo indipendente senza bloccare le altre aree.

ID Test	Requisito	Descrizione Test	Risultato Atteso
TU-10.2	FRObb96	Verifica esposizione dipendenze vulnerabili con codice CVE, severità CVSS e versione sicura.	Librerie obsolete o vulnerabili elencate con tutti i campi informativi richiesti.
TU-10.3	FRObb98	Verifica del mappatore di conformità OWASP Top 10.	Associazione corretta tra vulnerabilità e categoria OWASP.
TU-10.4	FRObb99	Verifica presentazione remediation di sicurezza ordinate per criticità.	Remediation esposte in ordine decrescente di severità della vulnerabilità associata.
TU-10.5	FRObb100	Validazione logica “Repository Sicuro” in assenza di vulnerabilità note.	Restituisce stato “Safe” se il contatore CVE e vulnerabilità è pari a zero.
TU-11.1	FRObb101	Verifica caricamento e visualizzazione sezione documentazione.	Sezione renderizzata correttamente con i dati dei file documentali analizzati.
TU-11.2	FRObb102	Verifica rilevamento errori sintattici e link interrotti nella documentazione.	Identifica URL malformati o non raggiungibili e errori formali nei file di testo.
TU-11.3	FRObb104	Calcolo dell’indice di completezza documentale basato sulla copertura delle interfacce pubbliche.	Rapporto coerente tra interfacce pubbliche del codice e blocchi di documentazione presenti.
TU-11.4	FRObb105	Verifica esposizione suggerimenti testuali per integrazione documentazione mancante.	Suggerimenti di miglioramento visualizzati correttamente per ogni lacuna rilevata.
TU-11.5	FRObb106	Controllo informativa “Documentazione Completa” in assenza di criticità documentali.	Esito positivo mostrato se non ci sono errori o mancanze documentali rilevate.
TU-12.1	FRObb107	Verifica del calcolo del punteggio di qualità globale (0-100) basato su metriche pesate.	Il punteggio finale è la media pesata dei punteggi di codice, sicurezza e documentazione.
TU-12.2	FRObb108	Algoritmo di generazione graduatoria con ordinamento decrescente per punteggio.	La lista viene ordinata correttamente dal punteggio più alto al più basso.

ID Test	Requisito	Descrizione Test	Risultato Atteso
TU-12.3	FRObi09	Verifica esposizione dati per ogni riga: posizione, nome repository, punteggio, trend mensile.	Tutti i campi previsti per ogni riga del ranking vengono popolati correttamente.
TU-12.4	FRObi10	Verifica inibizione rendering ranking e informativa in assenza di analisi completate.	Messaggio che suggerisce la prima analisi mostrato se il ranking è vuoto.
TU-13.1	FRObi11	Verifica richiesta conferma esplicita prima della rimozione integrazione GitHub.	Operazione bloccata in assenza di conferma; dialog di conferma presente.
TU-13.2	FRObi12	Verifica della funzione di revoca token OAuth verso le API di GitHub.	Richiesta di revoca inviata correttamente alle API GitHub al momento della conferma.
TU-13.3	FRObi13	Verifica eliminazione definitiva di token (access/refresh) e ID GitHub dal database.	Token e ID utente GitHub rimossi dal DB dopo la disconnessione.
TU-13.4	FRObi14	Verifica gestione errore comunicazione con GitHub durante revoca: pulizia locale garantita.	Anche in caso di errore API GitHub i dati locali sensibili vengono eliminati.
TU-14.1	FRDe3	Verifica disponibilità link di download del file generato.	Link di download presente e funzionante dopo la generazione del file.
TU-14.2	FRDe4	Verifica supporto formati di esportazione PDF e JSON.	Accetta esclusivamente PDF o JSON come formati validi.
TU-14.3	FRDe5	Verifica inibizione invio richiesta in assenza di formato selezionato.	Messaggio di errore e blocco se l'utente tenta di esportare senza scegliere il formato.
TU-14.4	FRDe6	Verifica del modulo di generazione file: mapping dati e inclusione metadati.	I dati del report, inclusi metadati e sezioni analizzate, vengono mappati senza perdite.
TU-14.5	FRDe7	Verifica processo di generazione file asincrono senza blocco dell'interfaccia.	L'UI rimane responsiva durante il parsing di report voluminosi.

ID Test	Requisito	Descrizione Test	Risultato Atteso
TU-15.1	FRObi15	Verifica accesso alla sezione dedicata alla modifica password.	Rendering corretto del modulo di modifica password nell'area profilo.
TU-15.2	FRObi16	Confronto hash tra password corrente inserita e hash memorizzato nel database.	Restituisce true solo se i due hash corrispondono.
TU-15.3	FRObi17	Verifica inibizione procedura e notifica errore per password corrente mancante o errata.	Messaggio specifico mostrato sia per campo vuoto sia per password non corrispondente.
TU-15.4	FRObi18	Validatore criteri complessità nuova password.	Rigetto di password che non rispettano i requisiti di lunghezza e complessità.
TU-15.5	FRObi19	Controllo eterogeneità: nuova password diversa da quella attuale.	Restituisce errore se l'hash della nuova password coincide con quello attuale.
TU-15.6	FRObi20	Verifica aggiornamento password nella persistenza tramite nuovo hashing sicuro e salt univoco.	La nuova password viene salvata cifrata con algoritmo sicuro e salt rigenerato.
TU-15.7	FRObi21	Verifica invio notifica email automatica a seguito di modifica password avvenuta.	Email di notifica inviata all'indirizzo associato al profilo dopo il cambio.
TU-15.8	FRObi22	Verifica invalidazione di tutte le sessioni attive post-cambio password.	Le sessioni parallele risultano invalidate; sessione corrente mantenuta attiva.
TU-16.1	FRObi23	Verifica visualizzazione dettaglio tecnico singola remediation selezionata.	Dettaglio correttamente caricato e visualizzato per la remediation selezionata.
TU-16.2	FRObi24	Verifica esposizione campi obbligatori remediation: descrizione difetto, snippet, severità, proposta.	Tutti i campi previsti presenti.
TU-16.3	FRObi25	Verifica inclusione riferimenti esterni per vulnerabilità di sicurezza note.	Link a documentazione esterna mostrati correttamente per remediation di tipo sicurezza.

ID Test	Requisito	Descrizione Test	Risultato Atteso
TU-17.1	FRObl26	Verifica gestione ciclo di vita verifica accessibilità tramite chiamate asincrone API GitHub.	Chiamate asincrone eseguite correttamente; stato della verifica tracciato.
TU-17.2	FRObl27	Verifica meccanismo “Exponential Backoff” per tentativi di riconnessione su errori di rete.	Ritardi crescenti tra tentativi; interruzione dopo il numero massimo configurato.
TU-17.3	FRObl28	Verifica validazione raggiungibilità endpoint GitHub tramite richiesta “Heartbeat”.	Heartbeat inviato prima del fetch; verifica operatività del servizio remoto.
TU-17.4	FRObl29	Verifica tentativo accesso pubblico al repository prima dell’uso di credenziali.	Prima richiesta senza intestazioni di autorizzazione.
TU-17.5	FRObl30	Verifica accesso privato via token OAuth/PAT su errore HTTP 403/404 della risorsa pubblica.	Seconda richiesta con token utente iniettato a seguito di errore 403/404.
TU-17.6	FRObl31	Verifica controllo “scopes” del token: permessi minimi di lettura (repo / public_repo).	Token con scopes insufficienti viene rifiutato con errore specifico.
TU-17.7	FRObl32	Verifica aggiornamento stato analisi a “FAILED_ACCESS” e segnale interruzione in caso di fallimento definitivo.	Stato impostato a “FAILED_ACCESS”; modulo notifica informato del fallimento.
TU-18.1	FRDe8	Verifica applicazione automatica modifiche al repository tramite integrazione GitHub.	Commit con le modifiche inviato correttamente al repository remoto.
TU-18.2	FRDe9	Verifica validazione di integrità della proposta correttiva prima del commit.	Proposta validata prima dell’invio; commit non eseguito se la validazione fallisce.
TU-18.3	FRDe10	Verifica aggiornamento stato remediation in “Applied” o “Dismissed” nel database.	Stato correttamente aggiornato a seguito dell’azione dell’utente.
TU-18.4	FRDe11	Verifica notifica all’utente in caso di fallimento del commit sul repository remoto.	Messaggio di errore mostrato; codebase rimane invariata.

ID Test	Requisito	Descrizione Test	Risultato Atteso
TU-20.1	FRObi33	Verifica definizione nome univoco raccolta.	Nomi non conformi ai vincoli rifiutati con messaggio specifico.
TU-20.2	FRObi34	Verifica validazione sintattica URL GitHub: HTTPS + struttura path user/repo.	URL non conformi rifiutati con avviso circa il protocollo o il formato richiesto.
TU-20.3	FRObi35	Verifica interrogazione API GitHub per conferma esistenza e raggiungibilità repository.	Repository inesistenti o irraggiungibili bloccano la creazione della raccolta.
TU-20.4	FRObi36	Verifica gestione repository inaccessibile (privato senza permessi) con notifica utente.	Avviso specifico per repository privato non accessibile dall'utente.
TU-20.5	FRObi37	Verifica impedimento creazione raccolta duplicata per stesso repository e stesso utente.	Errore di duplicazione se l'URL è già associato a una raccolta dell'utente.
TU-20.6	FRObi38	Verifica memorizzazione descrizione facoltativa con supporto codifica UTF-8.	Caratteri speciali e simboli memorizzati correttamente.
TU-21.1	FRObi39	Verifica parallelizzazione richieste di analisi verso i diversi strumenti esterni.	Le richieste agli strumenti vengono inviate in parallelo.
TU-21.2	FRObi40	Verifica inclusione parametri di configurazione utente nelle richieste agli strumenti.	Parametri definiti in fase di richiesta correttamente trasmessi agli strumenti esterni.
TU-21.3	FRObi41	Verifica trasmissione sicura delle credenziali al servizio AWS per la clonazione.	Credenziali non esposte in chiaro; passate tramite secret manager.
TU-21.4	FRObi42	Verifica monitoraggio completamento clonazione e gestione timeout / disco insufficiente.	Timeout e errori di spazio rilevati; procedura interrotta con segnalazione.
TU-21.5	FRObi43	Verifica inibizione invio richieste agli strumenti e liberazione risorse in caso di errore clonazione.	Nessuna richiesta inoltrata agli strumenti.

ID Test	Requisito	Descrizione Test	Risultato Atteso
TU-21.6	FRObl44	Verifica inoltro codebase agli strumenti di analisi via protocolli sicuri.	File trasmessi agli strumenti tramite canali cifrati.
TU-22.1	FRObl45	Verifica registrazione stato analisi come “PENDING” a seguito di inizializzazione strumenti.	Stato “PENDING” correttamente scritto nel database dopo l’avvio degli strumenti.
TU-22.2	FRObl46	Verifica associazione univoca ID analisi a repository e utente richiedente.	ID analisi associato correttamente; impossibile creare duplicati per stessa coppia repository-utente.
TU-22.3	FRObl47	Verifica persistenza metadati di avvio: hash commit analizzato e timestamp di sistema.	Hash commit e timestamp registrati correttamente al momento dell’avvio.
TU-22.4	FRObl48	Verifica procedura di Rollback e segnalazione agli strumenti in caso di errore scrittura stato.	In caso di fallimento persistenza gli strumenti vengono informati di annullare l’analisi.
TU-22.5	FRObl49	Verifica registrazione nei log di audit di ogni fallimento di persistenza con stack trace.	Log di errore scritti con dettaglio diagnostico completo.
TU-23.1	FRObl50	Verifica polling periodico o ricezione segnale di completamento dagli strumenti esterni.	L’orchestratore rileva correttamente la disponibilità dei risultati.
TU-23.2	FRObl51	Verifica download dei risultati non appena disponibili dagli strumenti.	File dei risultati scaricati correttamente all’atto della disponibilità.
TU-23.3	FRObl52	Verifica controllo integrità e leggibilità dei file ricevuti.	File corrotti o incompleti rilevati; segnalazione prima dell’utilizzo.
TU-23.4	FRObl53	Verifica prosecuzione creazione report con dati parziali in caso di fallimento di uno strumento.	Report generato con i dati disponibili; sezione mancante segnalata come assente.

ID Test	Requisito	Descrizione Test	Risultato Atteso
TU-23.5	FRObi54	Verifica impostazione timeout massimo di attesa per strumento ritardatario.	Attesa interrotta dopo il timeout configurato; prosecuzione con dati disponibili.
TU-23.6	FRObi55	Verifica segnalazione nel database di report contenente dati parziali.	Flag “parziale” impostato correttamente nel record del report nel DB.
TU-24.1	FRObi56	Verifica unificazione dati provenienti dai diversi strumenti in documento di sintesi unico.	Output di codice, sicurezza e documentazione aggregati correttamente in un unico report.
TU-24.2	FRObi57	Verifica conversione formati eterogenei degli strumenti in modello standard comune.	Tutti i formati raw vengono normalizzati correttamente senza perdita di informazioni.
TU-24.3	FRObi58	Verifica validazione completezza report prima del salvataggio (risultati, data, versione codice).	Report privo di informazioni essenziali non viene inoltrato alla fase di salvataggio.
TU-24.4	FRObi59	Verifica calcolo punteggi di riepilogo basati sui risultati delle singole aree analizzate.	Punteggi aggregati calcolati correttamente e coerenti con i risultati delle singole sezioni.
TU-25.1	FRObi60	Verifica archiviazione permanente report con collegamento a repository e utente.	Report salvato correttamente e associato al repository e al profilo richiedente.
TU-25.2	FRObi61	Verifica aggiornamento stato analisi a “Completato” solo dopo conferma salvataggio riuscito.	Stato impostato a “Completato” esclusivamente se la scrittura nel DB ha avuto esito positivo.
TU-25.3	FRObi62	Verifica notifica utente con messaggio di errore in caso di impossibilità di salvataggio.	Messaggio di errore generato e reso disponibile per la notifica all’utente.
TU-25.4	FRObi63	Verifica tracciamento interno dei motivi di fallimento del salvataggio per audit tecnico.	Log di errore dettagliato scritto con causa del fallimento.
TU-25.5	FRObi64	Verifica mantenimento copia temporanea del report in caso di errore salvataggio definitivo.	Copia temporanea disponibile per recupero manuale o retry automatico.

ID Test	Requisito	Descrizione Test	Risultato Atteso
TU-26.1	FRObi65	Verifica generazione automatica notifica utente a report pronto e salvato.	Notifica generata automaticamente non appena il report risulta archiviato.
TU-26.2	FRObi66	Verifica inclusione link/pulsante accesso diretto al report nella notifica.	Notifica contiene riferimento diretto al report consultabile.
TU-26.3	FRObi67	Verifica inclusione informazioni identificative nella notifica (nome repository, data).	Nome repository e data di esecuzione presenti nel corpo della notifica.
TU-26.4	FRObi68	Verifica indipendenza stato report dall'esito dell'invio notifica.	Report rimane disponibile anche se la notifica fallisce.
TU-26.5	FRObi69	Verifica registrazione anomalia nei log interni in caso di errore invio notifica.	Log di errore scritto con dettaglio del canale e del tipo di anomalia.
TU-26.6	FRObi70	Verifica meccanismo di retry invio notifica per problemi temporanei di rete.	Numero configurato di tentativi effettuati prima di rinunciare; log aggiornato.
TU-27.1	FRObi71	Verifica consegna notifica di fine analisi tramite canali scelti dall'utente.	Notifica recapitata sul canale configurato dall'utente.
TU-27.2	FRObi72	Verifica inclusione dettagli analisi (nome progetto, ora) nell'avviso ricevuto.	Messaggio contiene nome progetto e timestamp correttamente formattati.
TU-27.3	FRObi73	Verifica consultabilità risultati nell'area personale anche in assenza di notifica email.	Il report è accessibile dalla dashboard indipendentemente dalla ricezione della notifica.
TU-28.1	FRObi74	Verifica invio avviso immediato con causa errore in caso di interruzione analisi.	Notifica di errore con breve spiegazione generata tempestivamente.
TU-28.2	FRObi75	Verifica contrassegno analisi come "Fallita" nella lista progetti utente.	Stato "Fallita" visibile nella dashboard dell'utente.
TU-28.3	FRObi76	Verifica visibilità cause del fallimento nella dashboard indipendentemente dalla notifica.	Dettaglio errore consultabile nella dashboard a prescindere dall'invio dell'avviso.

ID Test	Requisito	Descrizione Test	Risultato Atteso
TU-29.1	FRObi177	Verifica scambio codice temporaneo OAuth con token di accesso permanente.	Richiesta di scambio correttamente inoltrata a GitHub; token ricevuto e disponibile.
TU-29.2	FRObi178	Verifica cifratura token GitHub prima del salvataggio nel database.	Token cifrato con algoritmo forte; non leggibile in chiaro nel DB.
TU-29.3	FRObi179	Verifica collegamento esclusivo token GitHub al profilo dell'utente autorizzante.	Token associato univocamente all'account CodeGuardian corretto.
TU-29.4	FRObi180	Verifica annullamento collegamento e richiesta ripetizione procedura per codice scaduto/invalido.	Errore di protocollo gestito; utente informato della necessità di riautenticarsi.
TU-30.1	FRObi181	Verifica caricamento e visualizzazione dettaglio singola remediation area codice.	Dettaglio correttamente caricato per remediation selezionata dall'area codice.
TU-30.2	FRObi182	Verifica presenza campi obbligatori nella remediation codice.	Tutti i campi previsti presenti nel pannello di dettaglio.
TU-31.1	FRObi183	Verifica caricamento e visualizzazione dettaglio singola remediation area sicurezza.	Dettaglio correttamente caricato per remediation selezionata dall'area sicurezza.
TU-31.2	FRObi184	Verifica presenza campi obbligatori nella remediation sicurezza.	Tutti i campi previsti presenti nel pannello di dettaglio.
TU-32.1	FRObi185	Verifica caricamento e visualizzazione dettaglio singola remediation area documentazione.	Dettaglio correttamente caricato per remediation selezionata dall'area documentazione.
TU-32.2	FRObi186	Verifica presenza campi obbligatori nella remediation documentale.	Tutti i campi previsti presenti nel pannello di dettaglio.

ID Test	Requisito	Descrizione Test	Risultato Atteso
TU-33.1	FRObi187	Verifica abilitazione comando di accettazione remediation codice per Utente Avanzato.	Controllo accesso: solo Utente Avanzato vede il comando di accettazione.
TU-33.2	FRObi188	Verifica applicazione automatica modifiche alla codebase a seguito di accettazione.	Commit con le modifiche suggerite inviato correttamente al repository.
TU-33.3	FRObi189	Verifica aggiornamento stato remediation codice a “eseguita” nel database.	Stato aggiornato correttamente nella dashboard dopo applicazione riuscita.
TU-33.4	FRObi190	Verifica notifica fallimento e invarianza codebase in caso di errore applicazione remediation codice.	Messaggio di errore mostrato; nessuna modifica al repository in caso di fallimento.
TU-34.1	FRObi191	Verifica abilitazione comando di rifiuto remediation codice.	Comando di rifiuto disponibile nell’interfaccia di dettaglio della remediation.
TU-34.2	FRObi192	Verifica aggiornamento stato remediation codice a “rifiutata” senza modifiche al repository.	Stato “rifiutata” aggiornato; codebase rimane invariata.
TU-35.1	FRObi193	Verifica abilitazione comando di accettazione remediation sicurezza.	Comando disponibile per Utente Avanzato nell’area sicurezza.
TU-35.2	FRObi194	Verifica applicazione patch/ configurazioni di sicurezza a seguito di accettazione.	Modifiche di sicurezza applicate correttamente al repository.
TU-35.3	FRObi195	Verifica aggiornamento stato remediation sicurezza a “eseguita”.	Stato aggiornato nella dashboard dopo applicazione riuscita.
TU-35.4	FRObi196	Verifica notifica fallimento applicazione remediation sicurezza.	Messaggio di errore e vulnerabilità non mitigata in caso di insuccesso.
TU-36.1	FRObi197	Verifica abilitazione comando di rifiuto remediation sicurezza.	Comando di rifiuto disponibile nell’area sicurezza.

ID Test	Requisito	Descrizione Test	Risultato Atteso
TU-36.2	FRObi198	Verifica aggiornamento stato remediation sicurezza a “rifiutata” senza modifiche al repository.	Stato aggiornato; repository invariato rispetto alla vulnerabilità segnalata.
TU-37.1	FRObi199	Verifica abilitazione comando di accettazione remediation documentale.	Comando disponibile per Utente Avanzato nell’area documentazione.
TU-37.2	FRObi200	Verifica applicazione automatica modifiche ai file documentali a seguito di accettazione.	File di documentazione aggiornati correttamente nel repository.
TU-37.3	FRObi201	Verifica aggiornamento stato remediation documentale a “eseguita”.	Stato aggiornato nella dashboard dopo applicazione riuscita.
TU-37.4	FRObi202	Verifica notifica errore e invarianza documentazione in caso di fallimento I/O o permessi.	Messaggio di errore mostrato; file di documentazione rimangono invariati.
TU-38.1	FRObi203	Verifica accesso al dettaglio singola remediation documentale per la procedura di rifiuto.	Dettaglio remediation visualizzato correttamente prima del rifiuto.
TU-38.2	FRObi204	Verifica abilitazione comando di rifiuto remediation documentale.	Comando di rifiuto disponibile nell’area documentazione.
TU-38.3	FRObi205	Verifica aggiornamento stato remediation documentale a “rifiutata” nel database.	Stato aggiornato correttamente a seguito del rifiuto confermato.
TU-38.4	FRObi206	Verifica che il rifiuto non comporti modifiche ai file sorgente o documentali.	Nessuna modifica al repository dopo il rifiuto.
TU-38.5	FRObi207	Verifica rimozione o marcatura visiva remediation rifiutata dalla lista azioni pendenti.	Remediation non più presente come pendente o visivamente segnata come scartata.
TU-38.6	FRObi208	Verifica visualizzazione conferma visiva avvenuto rifiuto della proposta correttiva.	Messaggio o badge di conferma rifiuto mostrato all’utente.
TU-39.1	FRObi209	Verifica abilitazione richiesta analisi repository privato solo per	Funzionalità disponibile solo se GitHub è collegato; inibita per utenti senza integrazione.

ID Test	Requisito	Descrizione Test	Risultato Atteso
		Utente Avanzato con integrazione GitHub attiva.	
TU-39.2	FRObi210	Verifica validazione presenza integrazione GitHub valida prima di accettare richiesta per risorsa privata.	Richiesta bloccata se il token GitHub risulta assente o scaduto.
TU-39.3	FRObi211	Verifica inibizione richiesta analisi privata in assenza di selezione aree di interesse.	Errore mostrato se nessuna area selezionata.
TU-40.1	FRObi212	Verifica inserimento URL repository privato di proprietà nel catalogo personale.	Repository aggiunto correttamente al catalogo; disponibile per analisi future.
TU-40.2	FRObi213	Verifica impedimento inserimento URL duplicato nel catalogo personale con notifica.	Avviso di duplicazione mostrato; catalogo rimane invariato.
TU-41.1	FRObi214	Verifica esposizione lista repository privati registrati con nome e URL.	Lista popolata correttamente con tutti i repository del catalogo personale.
TU-41.2	FRObi215	Verifica visualizzazione informativa catalogo vuoto con suggerimento inserimento.	Messaggio che suggerisce l'inserimento del primo repository mostrato se catalogo vuoto.
TU-42.1	FRObi216	Verifica rimozione repository dal catalogo privato previa conferma esplicita utente.	Repository rimosso solo dopo conferma; dialog di conferma presente.
TU-42.2	FRObi217	Verifica integrità catalogo in caso di annullamento procedura di rimozione.	Repository mantenuto nel catalogo se l'utente annulla l'operazione.
TU-43.1	FRObi218	Verifica visualizzazione elenco profili autorizzati per repository privato selezionato.	Lista degli utenti autorizzati caricata correttamente.
TU-43.2	FRObi219	Verifica informativa per accesso limitato esclusivamente al proprietario.	Messaggio mostrato se lista utenti autorizzati è vuota.

ID Test	Requisito	Descrizione Test	Risultato Atteso
TU-44.1	FRObb220	Verifica aggiunta utente autorizzato tramite username o email con validazione del profilo.	Profilo esistente aggiunto; identificativo non valido o inesistente rifiutato con errore specifico.
TU-44.2	FRObb221	Verifica validazione corrispondenza identificativo a profilo registrato in piattaforma.	Identificativo non trovato genera avviso di inesistenza.
TU-44.3	FRObb222	Verifica impedimento autorizzazione multipla del medesimo profilo per stesso repository.	Avviso di duplicazione mostrato; lista permessi invariata.
TU-45.1	FRObb223	Verifica revoca permessi utente autorizzato previa conferma proprietario.	Profilo rimosso dalla lista autorizzati dopo conferma; accesso al report revocato.
TU-46.1	FRObb224	Verifica rimozione raccolta report senza eliminazione dei singoli report contenuti.	Raccolta rimossa; report precedentemente contenuti ancora consultabili tramite altri percorsi.
TU-47.1	FRObb225	Verifica richiesta inserimento password attuale come verifica identità obbligatoria pre-cancellazione.	Procedura di cancellazione bloccata senza verifica password corretta.
TU-47.2	FRObb226	Verifica visualizzazione avviso di irreversibilità con possibilità di annullamento.	Dialog di avviso irreversibilità mostrato; annullamento disponibile.
TU-47.3	FRObb227	Verifica rimozione dati personali, associazioni OAuth e invalidazione credenziali post-cancellazione.	Accesso con credenziali precedenti impossibile.

Table 8: Tabella dei Test di Unità

4.3 Test di Accettazione

ID Test	Descrizione
TA-1	Verificare che l'utente possa completare con successo la procedura di registrazione e la successiva autenticazione al sistema.
TA-2	Verificare che le chiavi di accesso non siano mai salvate o trasmesse in chiaro, garantendo l'integrità del sistema di hashing.
TA-3	Verificare che il sistema gestisca correttamente il reindirizzamento e il ritorno dalla piattaforma esterna GitHub, associando correttamente l'identificativo OAuth e cifrando il token ottenuto.
TA-4	Verificare che l'utente possa configurare e avviare una richiesta di analisi fornendo un URL valido e selezionando le aree di interesse.
TA-5	Verificare che il sistema inibisca l'avvio di analisi ridondanti qualora il repository non abbia subito modifiche dall'ultimo report.
TA-6	Verificare che il sistema impedisca l'avvio di analisi concorrenti sul medesimo repository, notificando correttamente lo stato di "Analisi in corso".
TA-7	Verificare che l'accesso all'analisi di repository privati sia interdetto agli utenti che non hanno completato l'integrazione con GitHub, e che la richiesta di analisi (pubblica o privata) sia inibita in assenza di selezione di almeno un'area di interesse.
TA-8	Verificare che il sistema protegga i dati di sessione e i token GitHub tramite cifratura e protocolli di comunicazione sicuri (HTTPS).
TA-9	Verificare che l'utente possa visualizzare correttamente lo storico globale dei repository analizzati, identificando chiaramente i progetti tramite i metadati esposti.
TA-10	Verificare che l'utente possa navigare nel dettaglio di un singolo report, filtrando le sezioni di interesse (Codice, Sicurezza, Documentazione) e visualizzando i relativi metadati di audit.
TA-11	Verificare che il sistema presenti chiaramente le criticità rilevate e le relative remediation suggerite, distinguendo i casi di conformità (esito positivo).
TA-12	Verificare che l'utente possa impostare un intervallo temporale valido per generare un confronto storico tra le metriche di diversi report.
TA-13	Verificare che il sistema generi visualizzazioni grafiche e tabelle comparative coerenti, evidenziando i trend di miglioramento o peggioramento delle metriche del codice.

ID Test	Descrizione
TA-14	Verificare che l'analisi della qualità del codice esponga correttamente i dati di analisi statica (bug/smell) e le percentuali di copertura dei test di unità.
TA-15	Verificare che l'analisi della sicurezza esponga correttamente le vulnerabilità delle librerie (CVE) e i rilievi di conformità agli standard OWASP.
TA-16	Verificare che l'analisi della documentazione identifichi correttamente gli errori di sintassi e il grado di completezza rispetto al codice sorgente.
TA-17	Verificare che l'utente possa consultare il ranking dei repository ordinati per punteggio di qualità globale, ricevendo un'informativa corretta in assenza di dati.
TA-18	Verificare che l'utente possa disconnettere l'account GitHub dal profilo CodeGuardian, con conseguente revoca delle autorizzazioni e dei token.
TA-19	Verificare che l'utente possa esportare i report di analisi in formati standard (PDF/JSON), garantendo la selezione obbligatoria del formato.
TA-20	Verificare che l'utente possa modificare la propria password di accesso previa validazione della credenziale attuale e rispetto dei criteri di sicurezza.
TA-21	Verificare che l'utente possa creare una raccolta di report associata a un repository GitHub, fornendo nome e URL validi, con eventuale descrizione facoltativa.
TA-22	Verificare che l'Orchestratore verifichi correttamente l'accessibilità del repository prima di avviare l'analisi, distinguendo tra risorse pubbliche e private e gestendo i fallimenti di accesso.
TA-23	Verificare che l'Orchestratore avvii correttamente il processo di analisi, clonando il repository e distribuendo la codebase agli strumenti di analisi per le aree selezionate.
TA-24	Verificare che il sistema aggreghi i risultati degli strumenti di analisi in un report strutturato, lo archivi correttamente nel sistema di persistenza e aggiorni lo stato dell'analisi a "completato".
TA-25	Verificare che l'utente riceva una notifica al completamento dell'analisi e che il report risulti consultabile nella propria area personale anche in assenza di ricezione della notifica.

ID Test	Descrizione
TA-26	Verificare che l'utente venga notificato in caso di errore critico durante l'analisi e che lo stato di fallimento sia visibile nella dashboard indipendentemente dalla ricezione della notifica.
TA-27	Verificare che l'Utente Avanzato possa accettare una remediation proposta, con conseguente applicazione delle modifiche al repository e aggiornamento dello stato nella dashboard.
TA-28	Verificare che l'Utente Avanzato possa rifiutare una remediation proposta, con conseguente scarto della proposta e invarianza del repository.
TA-29	Verificare che l'Utente Avanzato con integrazione GitHub attiva possa avviare con successo l'analisi di un repository GitHub privato presente nel proprio catalogo, selezionando le aree di interesse.
TA-30	Verificare che l'Utente Avanzato possa gestire il proprio catalogo di repository privati, inserendo, visualizzando e rimuovendo risorse, con corretta gestione dei duplicati.
TA-31	Verificare che il proprietario di un repository privato possa gestire i permessi di accesso ai report, aggiungendo e revocando le autorizzazioni per altri utenti della piattaforma.
TA-32	Verificare che l'utente possa rimuovere una raccolta di report dal proprio profilo senza che i singoli report in essa contenuti vengano eliminati.
TA-33	Verificare che l'utente possa cancellare definitivamente il proprio profilo CodeGuardian, con conseguente rimozione dei dati personali, delle associazioni OAuth e invalidazione delle credenziali precedenti.

Table 9: Tabella dei Test di Accettazione

5 Cruscotto di Valutazione

Il presente cruscotto costituisce il sistema di monitoraggio attraverso il quale Skarab Group valuta oggettivamente l'andamento del progetto. Le metriche qui raccolte rappresentano l'evidenza empirica necessaria per attivare il ciclo *PDCA* (Plan-Do-Check-Act), trasformando i dati grezzi in informazioni per il miglioramento continuo.

In questa sezione vengono presentati i risultati delle misurazioni effettuate nel periodo di riferimento. L'analisi dei dati non è fine a se stessa, ma è orientata a fornire una visione oggettiva ("Data-Driven") dello stato di salute del progetto e della qualità del software rilasciato. I dati sono organizzati per area di processo e per qualità di prodotto, permettendo una rapida identificazione delle aree critiche e il confronto con le soglie di accettabilità definite nel Piano di Qualifica.

5.1 Processi Primari: Fornitura (EVM)

Questa sezione monitora l'andamento economico e temporale del progetto utilizzando lo standard **Earned Value Management**. L'obiettivo è evidenziare scostamenti tra quanto pianificato (Baseline) e quanto effettivamente realizzato.

5.1.1 Trend di Progetto (PV, AC, EV)

Metriche: MPC02, MPC03, MPC04

Viene visualizzato l'andamento cumulativo del valore pianificato (**Planned Value**), del costo reale sostenuto (**Actual Cost**) e del valore guadagnato (**Earned Value**). La sovrapposizione delle curve indica un progetto in linea con le aspettative; divergenze significative segnalano la necessità di interventi correttivi su budget o scadenze.

5.1.2 Indici di Efficienza (CPI, SPI)

Metriche: MPC07, MPC08

Vengono riportati gli indici di performance puntuali per ogni Sprint. Questi valori normalizzati permettono di capire immediatamente l'efficienza di costo (**CPI**) e di schedulazione (**SPI**), dove un valore pari o superiore a 1.00 rappresenta lo stato ottimale.

5.1.3 Varianze e Previsioni (CV, SV, EAC)

Metriche: MPC05, MPC06, MPC09

Questa sezione quantifica in termini monetari l'eventuale risparmio o perdita (**Cost Variance**) e l'anticipo o ritardo (**Schedule Variance**). Viene inoltre proiettata la stima del costo finale a finire (**Estimate At Completion**) confrontandola con il budget iniziale.

5.2 Processi Primari: Sviluppo

Questa sezione analizza la stabilità dell'ambito tecnico del progetto.

5.2.1 Requirements Stability Index (RSI)

Metrica: MPC10

Grafico che traccia la volatilità dei requisiti nel tempo. Un indice stabile e alto garantisce che il team stia lavorando su obiettivi consolidati, mentre fluttuazioni frequenti possono indicare incertezze nell'analisi o richieste di modifica eccessive (**Scope Creep**).

5.3 Processi di Supporto: Documentazione

Monitoraggio della qualità formale e della fruibilità della documentazione prodotta.

5.3.1 Indice di Gulpease e Correttezza

Metriche: MPC11, MPC12

Viene riportato il livello di leggibilità linguistica (**Gulpease Index**) calcolato sui documenti

principali, unitamente al numero di errori ortografici rilevati, per garantire che la documentazione sia accessibile e professionale.

5.4 Processi di Supporto: Verifica

Monitoraggio dell'efficacia delle attività di testing dinamico.

5.4.1 Code Coverage e Test Success

Metriche: MPC13, MPC14

Cruscotto tecnico che visualizza la copertura del codice raggiunta dai test automatizzati e il tasso di successo dei test eseguiti. Questi indicatori sono fondamentali per valutare la robustezza del codice prima del rilascio.

5.5 Processi di Supporto: Gestione della Qualità

Visione d'insieme sull'efficacia del Piano di Qualifica stesso.

5.5.1 Soddisfazione delle Metriche

Metrica: MPC15

Indicatore sintetico (KPI) che mostra la percentuale totale delle metriche di progetto che rispettano le soglie di accettabilità definite. Fornisce un'indicazione immediata sulla conformità complessiva dei processi.

5.6 Processi Organizzativi: Gestione dei Processi

Analisi dell'efficienza del metodo di lavoro Agile adottato dal team.

5.6.1 Sprint Goal Achievement

Metrica: MPC16

Viene illustrata la capacità del team di completare gli obiettivi pianificati all'inizio di ogni iterazione (Sprint Planning). Questo dato è essenziale per calibrare la **Velocity** del team e migliorare la precisione delle pianificazioni future.

5.7 Qualità di Prodotto

In questa sezione si verifica la conformità del software rilasciato rispetto ai requisiti e agli standard di qualità ISO/IEC 25010.

5.7.1 Copertura Funzionale

Metriche: MPD01, MPD02, MPD03

Visualizzazione dello stato di implementazione dei requisiti, suddivisi per priorità (Obbligatori, Desiderabili, Opzionali), per confermare l'adeguatezza funzionale del rilascio corrente.

5.7.2 Affidabilità e Manutenibilità

Metriche: MPD04, MPD05, MPD08, MPD09, MPD10

Analisi tecnica che combina indicatori di affidabilità (densità guasti, disponibilità) e metriche statiche del codice (complessità ciclomatica, densità commenti) per valutare la salute tecnica del prodotto.

5.7.3 Usabilità e Sicurezza

Metriche: MPD06, MPD07, MPD11

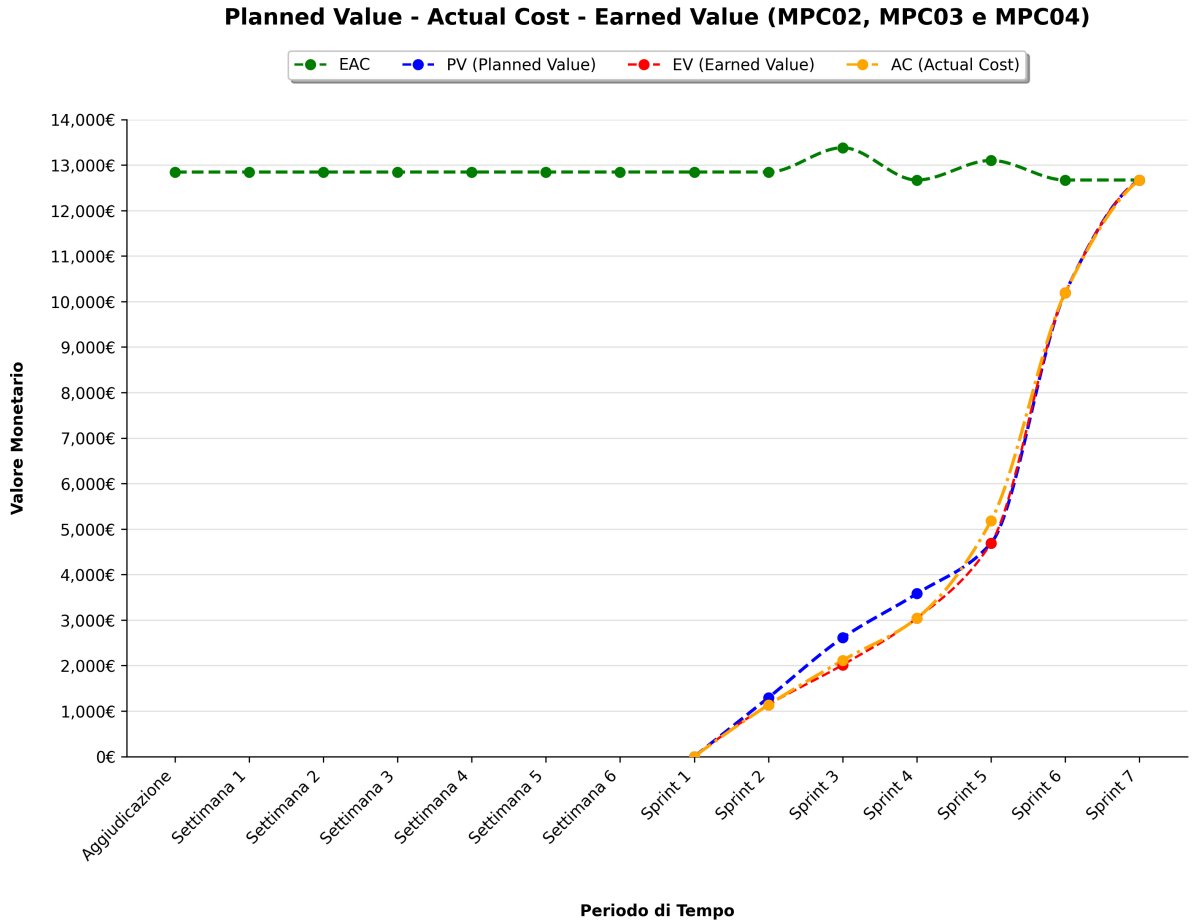
Report sugli esiti delle verifiche di usabilità (comprensibilità, prevenzione errori) e scansione delle vulnerabilità di sicurezza, garantendo che il prodotto sia sicuro e utilizzabile dall'utente finale.

È importante evidenziare che il periodo iniziale, dall'aggiudicazione fino all'avvio formale delle attività di progetto (*Sprint 1*), ha rappresentato una fase di "palestra" durante la quale il gruppo si è

dedicato allo studio approfondito delle tecnologie necessarie, partecipando anche a sessioni di formazione organizzate dall'azienda proponente Var Group.

5.8 Processi Primari: Fornitura e Sviluppo

5.8.1 Planned Value - Actual Cost - Earned Value (MPC02, MPC03 e MPC04)

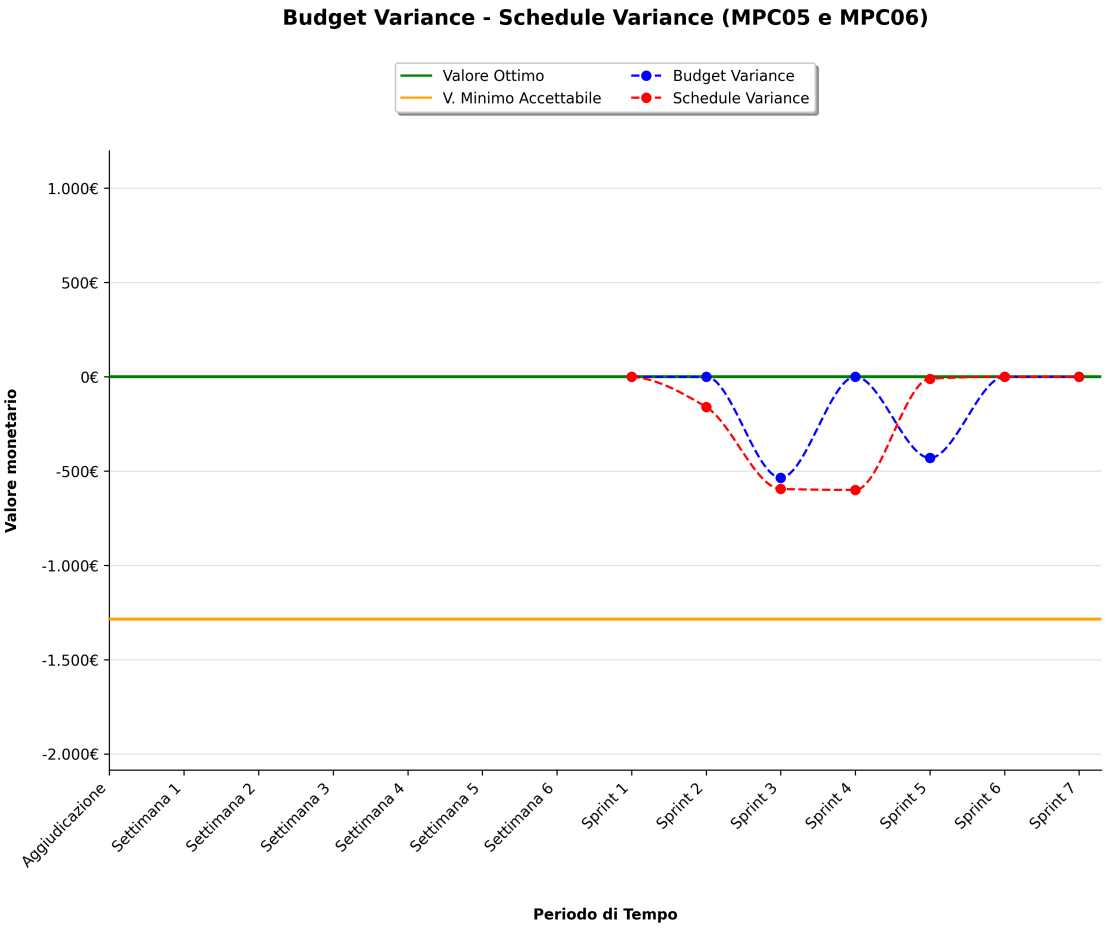


Dopo la fase iniziale, in cui le attività di formazione e setup sono state gestite come investimento interno senza gravare sul budget, il progetto è entrato nella fase operativa con l'avvio dello *Sprint 1*. In questa prima iterazione Skarab Group ha mostrato un buon equilibrio economico, completando il lavoro con un dispendio di risorse coerente con il valore prodotto, pur registrando un lieve ritardo rispetto alla pianificazione ideale.

Tuttavia, la situazione ha subito una variazione significativa durante lo *Sprint 2*: a fronte di un incremento del *Planned Value* (PV) e dell'*Actual Cost* (AC), l'*Earned Value* (EV) ha subito una flessione. Questo testimonia l'insorgere di inefficienze produttive e debito tecnico, legati alla necessità di ricalibrare task qualitativamente insufficienti che hanno rallentato la produzione.

Nello *Sprint 3*, nonostante una parziale ripresa, l'*Earned Value* rimane ancora al di sotto del *Planned Value*, segnalando che il ritardo accumulato non è ancora stato recuperato. Nello *Sprint 4* EV e PV iniziano ad allinearsi, a fronte però di un aumento dell'AC.

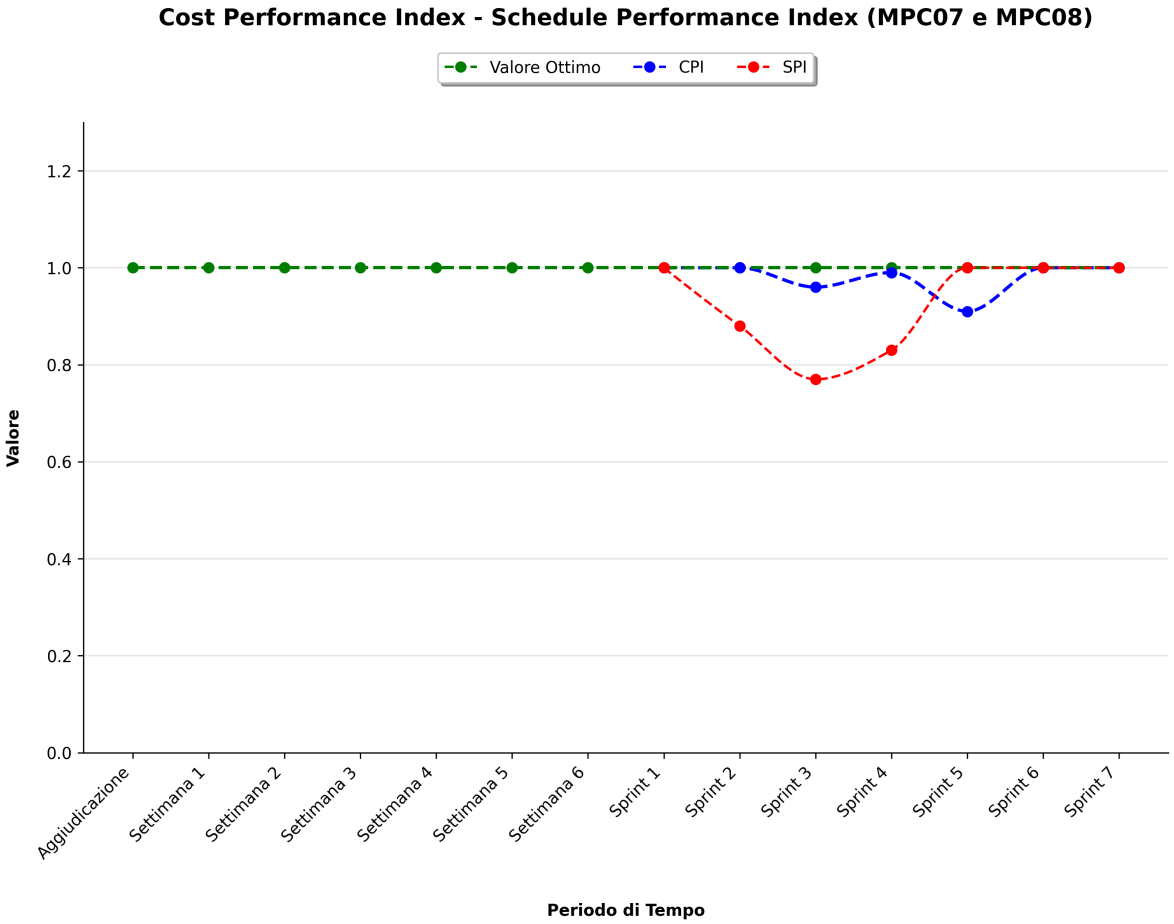
5.8.2 Budget Variance - Schedule Variance (MPC05 e MPC06)



Il grafico monitora la salute economica e temporale del progetto a partire dallo *Sprint 1* durante il quale la *Schedule Variance* (SV) mostra una leggera flessione. Quest'ultima si è accentuata nello *Sprint 2*, riflettendosi anche sulla *Budget Variance* (BV).

Successivamente, il ritardo accumulato negli sprint precedenti ha continuato a pesare sulla metrica. Nel corso dello *Sprint 4*, però, la *Schedule Variance* torna quasi a zero.

5.8.3 Cost Performance Index - Schedule Performance Index (MPC07 e MPC08)

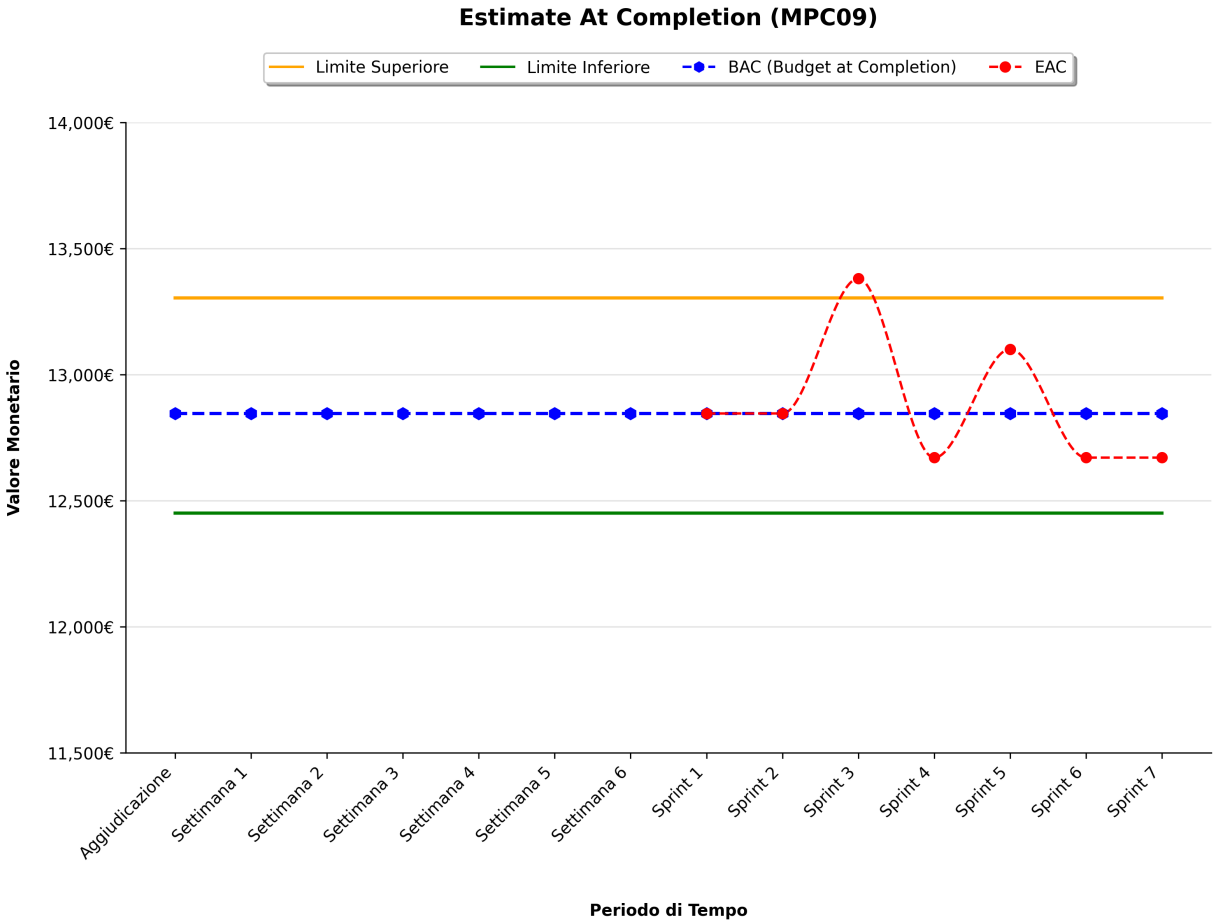


Dal grafico è possibile notare come, inizialmente, lo *Schedule Performance Index* (SPI) sia inferiore a 1, indicando un leggero ritardo fisiologico. La buona gestione dei costi è invece documentata dal *Cost Performance Index* (CPI).

La situazione è peggiorata nel corso dello *Sprint 2*, durante il quale si è verificato un crollo dello *Schedule Performance Index* (SPI) che segnala un ritardo critico rispetto alla pianificazione.

A partire dallo *Sprint 3* lo SPI inizia a recuperare progressivamente. Nello Sprint 4 lo SPI raggiunge il valore ottimale, mentre il CPI scende a causa dello sforamento orario.

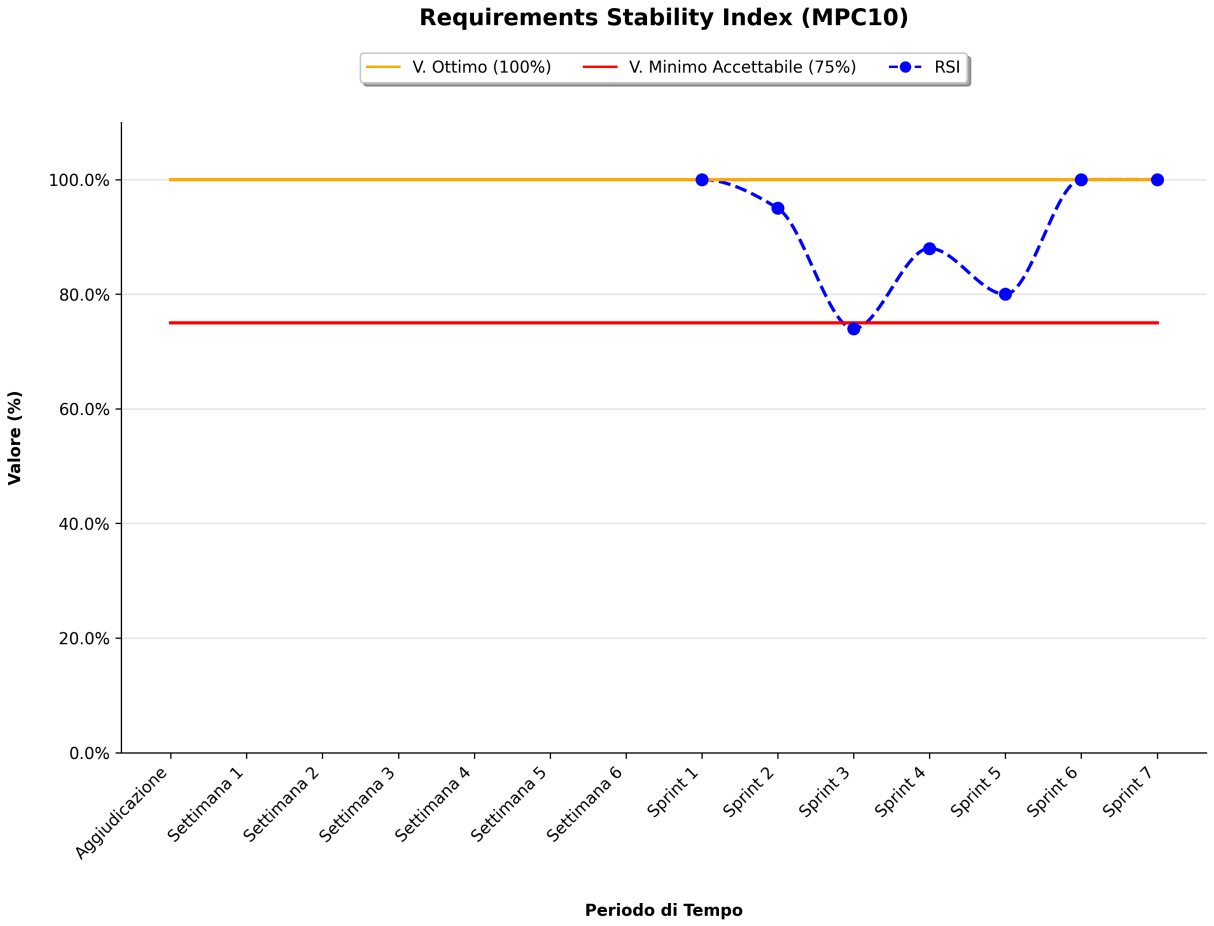
5.8.4 Estimate at Completion (MPC09)



Dopo una fase iniziale di stabilità coincidente con il budget originale, è possibile notare come una gestione inefficiente delle risorse abbia spinto la previsione di spesa verso il limite massimo.

Nello *Sprint 3* la situazione è migliorata, con l'EAC che è rientrato entro i limiti accettabili. Tuttavia, nello *Sprint 4*, lo sforamento orario dovuto alle revisioni dell'Analisi dei Requisiti ha spinto nuovamente l'EAC a 13.100€, superando il BAC ma rimanendo al di sotto del limite superiore. Il team si impegna ad adottare azioni correttive nella fase successiva per ricondurre la previsione di spesa entro i parametri ottimali.

5.8.5 Requirements Stability Index (MPC10)

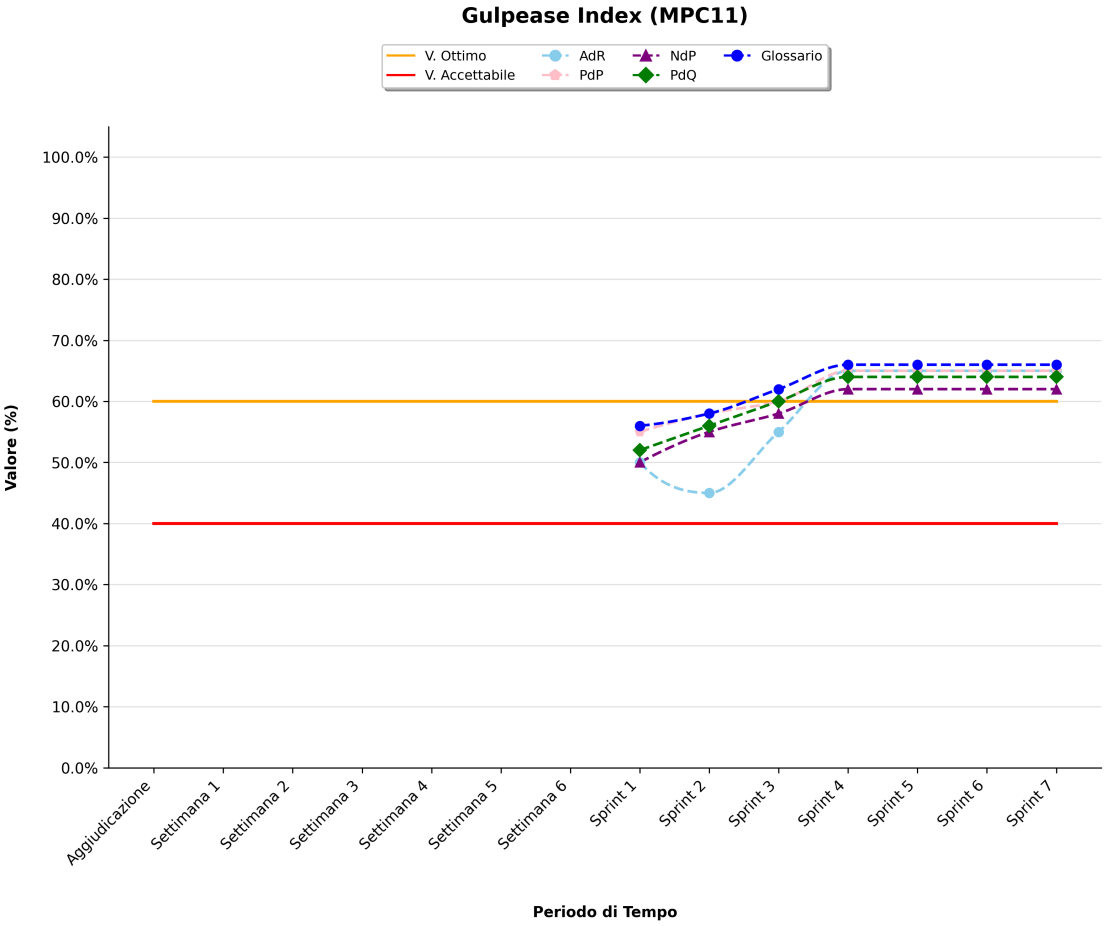


Il *Requirements Stability Index* (RSI) registra un peggioramento nel corso dello *Sprint 2*. Tale flessione è riconducibile a una sottostima iniziale dei requisiti impliciti e all'emersione di ulteriori requisiti in seguito al colloquio con il Prof. Cardin: il team ha dovuto apportare modifiche significative per aggiungere i requisiti non tracciati in precedenza dagli Analisti.

Nello *Sprint 4*, la riscrittura di casi d'uso malposti e l'inserimento di requisiti mancanti hanno causato un ulteriore calo dell'indice, mantenendosi comunque al di sopra della soglia minima accettabile.

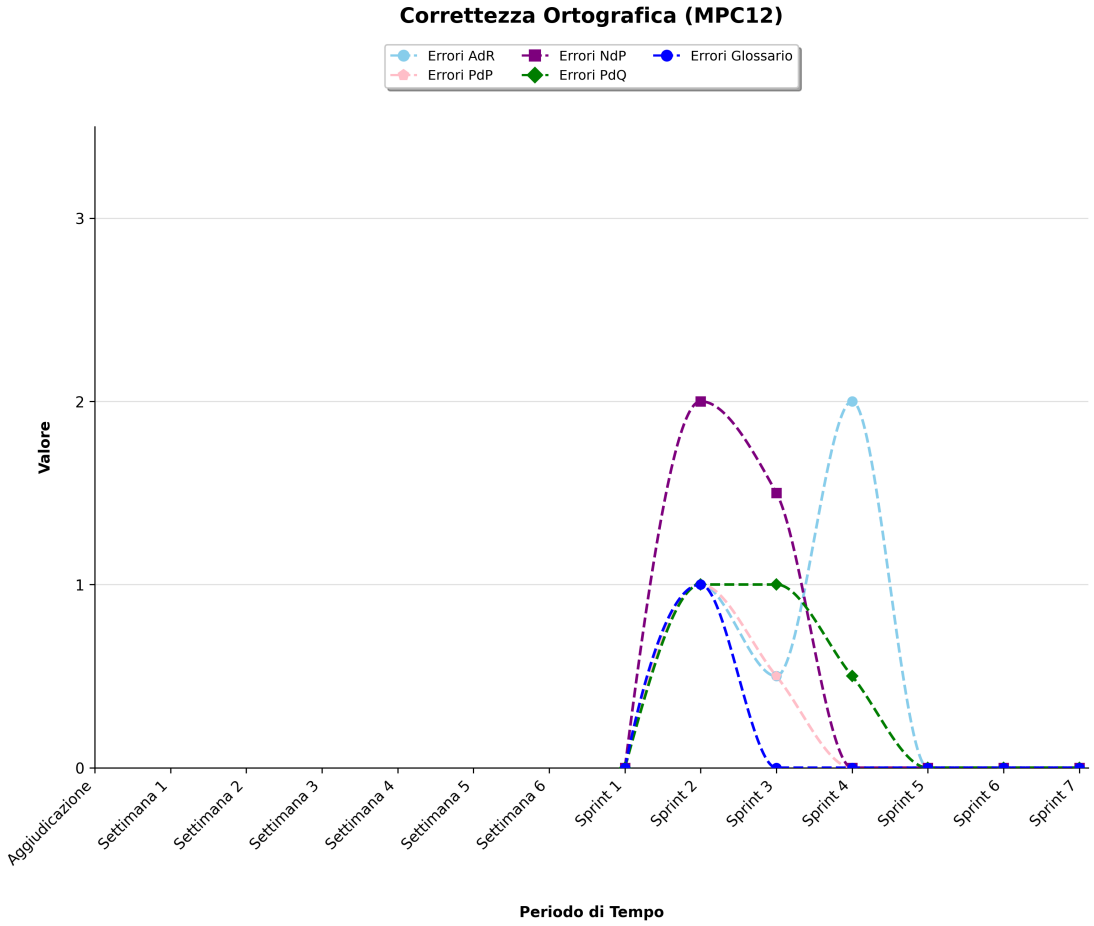
5.9 Processi di Supporto

5.9.1 Gulpease Index (MPC11)



Il grafico mostra un andamento complessivamente positivo dei documenti monitorati: a partire dallo *Sprint 1*, i valori si attestano al di sopra della soglia minima accettabile. L'*Analisi dei Requisiti* presenta una lieve flessione nello *Sprint 2*, riconducibile alla necessità di introdurre terminologia tecnica più densa e alla fase di revisione intensiva dei requisiti. In generale, Skarab Group si impegna a mantenere nel tempo una buona leggibilità dei documenti.

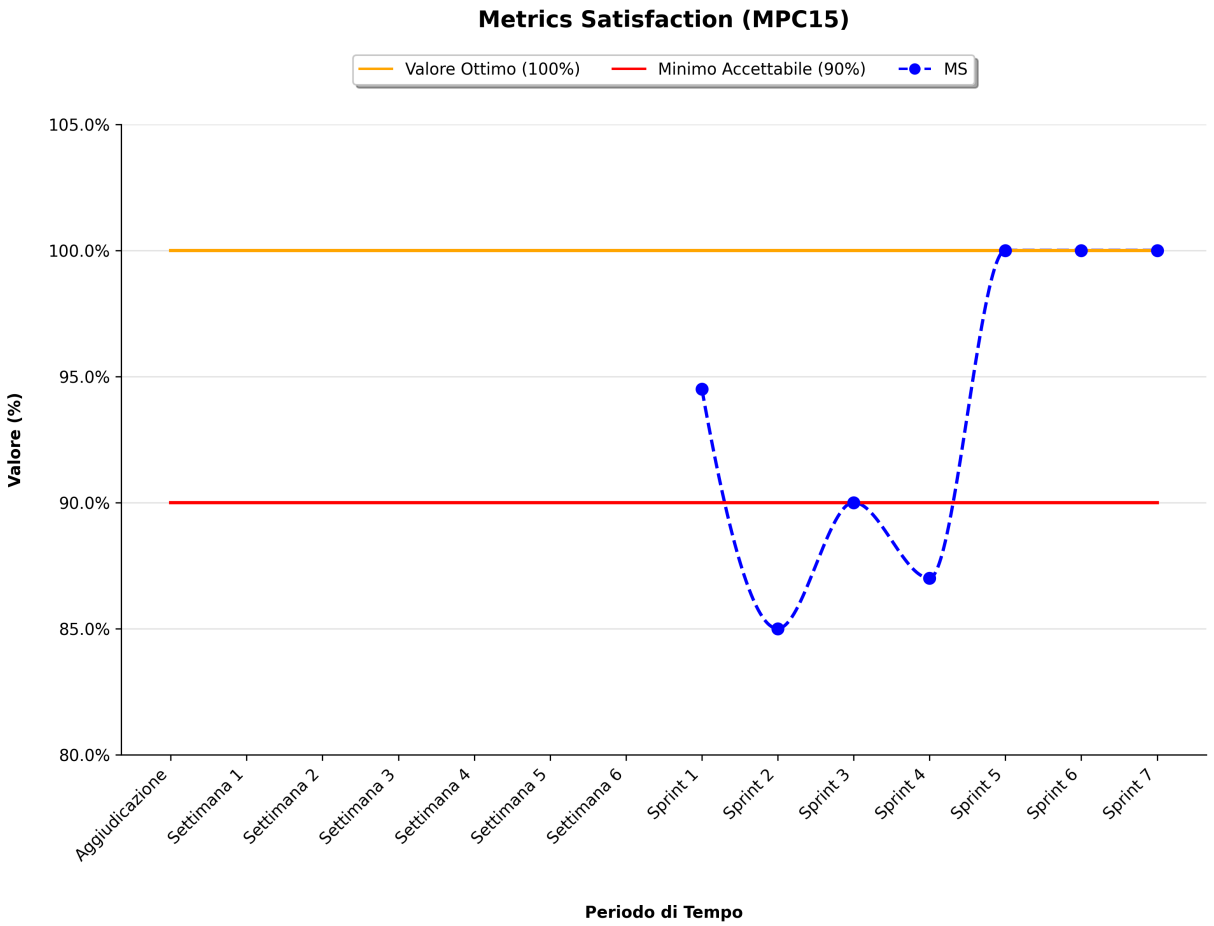
5.9.2 Correttezza Ortografica (MPC12)



Durante i primi sprint, il grafico evidenzia la presenza di alcuni errori ortografici nella documentazione, con un picco registrato nello *Sprint 2*. Il team ha prontamente identificato la criticità e adottato misure correttive, portando il conteggio degli errori a zero entro lo *Sprint 3* per la maggior parte dei documenti.

5.10 Processi Organizzativi

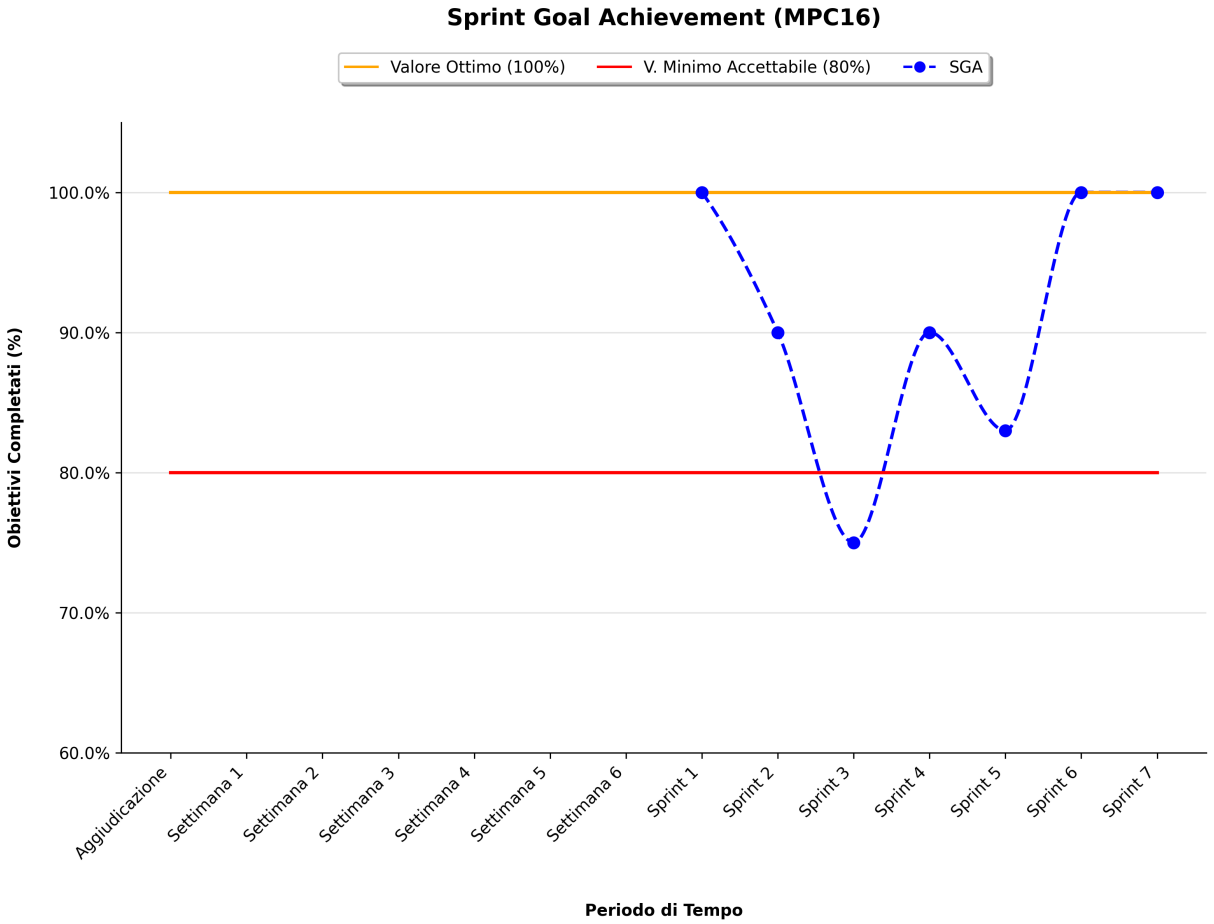
5.10.1 Metrics Satisfaction (MPC15)



Durante lo *Sprint 1* il valore si è attestato al di sopra della soglia minima accettabile, indicando una buona ma non ancora ottimale conformità ai criteri di qualità definiti. Nello *Sprint 2* si è registrata una flessione al di sotto della soglia minima accettabile.

Nello *Sprint 4*, lo sforamento orario e le difficoltà incontrate hanno causato un nuovo calo. Il team prende atto della criticità e si impegna ad adottare misure correttive nella fase successiva per garantire un maggiore rispetto delle metriche definite.

5.10.2 Sprint Goal Achievement (MPC16)



Dal grafico è possibile osservare la buona efficacia operativa dimostrata durante lo *Sprint 1* dal team, che è riuscito a completare gli obiettivi prefissati quasi nella loro interezza. Durante lo *Sprint 2*, invece, la metrica ha subito una flessione poiché gli obiettivi prefissati non sono stati pienamente raggiunti.

6 Miglioramento Continuo

Il processo di miglioramento continuo rappresenta il motore evolutivo del **Way of Working** del gruppo Skarab Group. Non ci si limita a correggere gli errori nel codice, ma si punta a ottimizzare sistematicamente i processi organizzativi e di supporto per prevenire la ricorrenza delle anomalie.

La strategia adottata implementa rigorosamente il ciclo di Deming (PDCA), integrandosi con le iterazioni previste dalla metodologia Agile [G]:

- **Plan (Pianificazione):** Definizione degli obiettivi di qualità e delle metriche e pianificazione delle attività.
- **Do (Esecuzione):** Svolgimento delle attività di sviluppo e gestione durante lo Sprint.
- **Check (Controllo):** Al termine di ogni iterazione si misurano i valori delle metriche e si confrontano con le soglie attese.
- **Act (Azione):** Qualora si rilevino scostamenti negativi, vengono definite **Azioni Correttive** che modificano le Norme di Progetto, diventando operative dallo Sprint successivo.

6.1 Azioni di Miglioramento Intraprese

Di seguito sono riportate le criticità emerse e le relative azioni correttive, suddivise per ambito di intervento.

ID	Problema / Causa	Azione Correttiva
Area: Comunicazione		
AM01	**Inefficienza nella Comunicazione Interna** Si sono verificati rallentamenti operativi causati da una comunicazione asincrona poco reattiva. Il team ha preso piena consapevolezza della scarsa efficacia delle modalità iniziali, che generavano colli di bottiglia.	**Ristrutturazione dei Flussi Informativi** I membri coinvolti hanno analizzato le cause (Root Cause Analysis) e compreso la lezione. Sono stati istituiti canali dedicati alle urgenze e aumentata la frequenza dei micro-allineamenti per sbloccare i task pendenti.
Area: Ruoli e Pianificazione		
AM02	**Pressione sulle Scadenze (Time-to-Result)** La necessità di produrre risultati tangibili (PoC) in tempi brevi per la revisione RTB rischiava di non essere soddisfatta con la pianificazione lineare iniziale.	**Ridistribuzione del Budget Orario** È stata effettuata una rimodulazione delle ore pianificate, allocando maggiori risorse sulle attività critiche di sviluppo e riducendo temporaneamente quelle a basso valore aggiunto, per garantire il rilascio puntuale.
Area: Strumenti e Tecnologie		
AM03	**Disomogeneità nella Documentazione** La stesura parallela dei documenti da parte di più persone ha	**Adozione di Template e Funzioni Comuni** Per garantire coerenza, sono state ingegnerizzate le funzioni di typesetting (in

ID	Problema / Causa	Azione Correttiva
	inizialmente generato incoerenze stilistiche e ripetizioni ridondanti o formattate diversamente.	Typst) e creati template condivisi. Questo forza l'uniformità visiva e strutturale indipendentemente dall'autore della sezione.
AM04	**Overhead Nuovi Strumenti** L'adozione contemporanea di nuovi strumenti (Jira, GitHub, Typst) ha comportato un rallentamento iniziale dovuto alla curva di apprendimento.	**Consolidamento della Toolchain** Dopo la fase di rodaggio, l'uso degli strumenti è stato standardizzato nelle Norme di Progetto. La corretta rendicontazione è ora integrata nel flusso di lavoro quotidiano, trasformando l'overhead iniziale in un guadagno di efficienza.

Table 10: Storico delle azioni di miglioramento (Periodo RTB)

7 Conclusioni

L'attività di miglioramento continuo per il progetto *Code Guardian* si è rivelata non solo una pratica formale, ma una necessità operativa. L'analisi delle metriche e le retrospettive hanno evidenziato come l'avvio del progetto abbia scontato l'inevitabile "prezzo d'ingresso" dovuto alla curva di apprendimento dei nuovi strumenti (come Jira e Typst) e al necessario assestamento delle dinamiche comunicative interne.

Le azioni correttive intraprese, in particolare la ristrutturazione dei canali informativi (**AM01**) e la standardizzazione documentale (**AM03**), hanno permesso di superare l'iniziale frammentazione operativa. Sebbene la ridistribuzione del budget orario (**AM02**) abbia garantito il raggiungimento degli obiettivi RTB, ha reso evidente l'importanza di una pianificazione più granulare per il futuro.

Consapevoli che l'assetto attuale non è un traguardo definitivo ma uno stato da preservare, il team si impegna a mantenere alta la vigilanza. L'obiettivo per le prossime fasi non è l'assenza di problemi, ma la capacità di identificarli tempestivamente tramite il monitoraggio dei dati e risolverli con la stessa reattività dimostrata in questo primo periodo.