

Piano di Qualifica

v0.9.0

Registro delle Modifiche

Data	Versione	Descrizione	Redattore	Verificatore
2026/02/06	0.9.0	Test di Accettazione e rielaborazione Test di Sistema	Zago Alice	
2026/01/27	0.8.1	Aggiunta grafico BV-SV, revisione Metodi di Testing e aggiunta test, modifiche minori al documento	Zago Alice	
2026/01/22	0.7.0	Grafici CPI-SPI, EAC, RSI e SGA	Zago Alice	
2026/01/21	0.6.0	Cruscotto di valutazione, grafico PV-AC-EV	Zago Alice	
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 automiglioramento e qualità di prodotto	Zago Alice	Suar Alberto
2025/12/27	0.2.0	Qualità di processo, processi primari	Zago Alice	Suar Alberto
2025/12/26	0.1.0	Inizio stesura documento,	Zago Alice	Suar Alberto

Data	Versione	Descrizione	Redattore	Verificatore
		introduzione, scopo e riferimenti		
2025/12/23	0.0.0	Creazione documento	Zago Alice	Suar Alberto

Indice

1 Introduzione	7
1.1 Contesto del Progetto	7
1.2 Finalità del Documento	7
1.3 Traguardi Qualitativi	7
1.3.1 Revisione dei Requisiti e della Tecnologia (RTB)	7
1.3.2 Revisione di Accettazione (Product Baseline – PB)	7
1.4 Glossario	8
1.5 Riferimenti	8
1.5.1 Riferimenti Normativi	8
1.5.2 Riferimenti Informativi	8
1.6 Qualità di Processo	9
1.6.1 Centralizzazione delle Metriche e Obiettivi	9
1.6.2 Processi Primari: Fornitura e Sviluppo	10
1.6.3 Processi di Supporto	11
1.6.4 Processi Organizzativi	11
1.7 Qualità di Prodotto	12
1.7.1 Adeguatazza Funzionale e Affidabilità	12
1.7.2 Manutenibilità e Sicurezza	12
2 Metodi di Testing	13
2.1 Test di Sistema	13
2.2 Test di Unità	17
2.3 Test di Accettazione	24
2.4 Test di Accettazione	25
3 Cruscotto di Valutazione	29
3.1 Processi Primari: Fornitura (EVM)	29
3.1.1 Trend di Progetto (PV, AC, EV)	29
3.1.2 Indici di Efficienza (CPI, SPI)	29
3.1.3 Varianze e Previsioni (CV, SV, EAC)	29
3.2 Processi Primari: Sviluppo	29
3.2.1 Requirements Stability Index (RSI)	29
3.3 Processi di Supporto: Documentazione	29
3.3.1 Indice di Gulpease e Correttezza	30
3.4 Processi di Supporto: Verifica	30
3.4.1 Code Coverage e Test Success	30
3.5 Processi di Supporto: Gestione della Qualità	30
3.5.1 Soddisfazione delle Metriche	30
3.6 Processi Organizzativi: Gestione dei Processi	30
3.7 Finalità del Cruscotto	30
3.8 Processi Primari: Fornitura e Sviluppo	31
3.8.1 Planned Value - Actual Cost - Earned Value (MPC02, MPC03 e MPC04)	31
3.8.2 Budget Variance - Schedule Variance (MPC05 e MPC06)	32
3.8.3 Cost Performance Index - Schedule Performance Index (MPC07 e MPC08)	33
3.8.4 Estimate at Completion (MPC09)	34
3.8.5 Requirements Stability Index (MPC10)	35
3.9 Processi di Supporto	36
3.9.1 Gulpease Index (MPC11)	36

3.9.2	Correttezza Ortografica (MPC12)	36
3.10	Processi Organizzativi	37
3.10.1	Metrics Satisfaction (MPC15)	37
3.10.2	Sprint Goal Achievement (MPC16)	37
3.10.3	Sprint Goal Achievement	37
3.11	Qualità di Prodotto	37
3.11.1	Copertura Funzionale	37
3.11.2	Affidabilità e Manutenibilità	38
3.11.3	Usabilità e Sicurezza	38
4	Miglioramento Continuo	39
4.1	Azioni di Miglioramento Intraprese	39
5	Conclusioni	40
5.1	Automiglioramento	40
5.1.1	Introduzione	40
5.1.1.1	Valutazione Tecnologica	40
5.1.1.2	Valutazione Organizzativa	41
5.1.1.2.1	Valutazione delle Responsabilità	41
5.1.1.3	Conclusioni	42

Indice tabelle

Table 1 Soglie metriche per il processo di Fornitura (EVM)	10
Table 2 Soglie metriche per il processo di Sviluppo	10
Table 3 Soglie metriche Documentazione e Verifica	11
Table 4 Soglie metriche Organizzative	11
Table 5 Metriche Adeguatezza e Affidabilità	12
Table 6 Metriche Manutenibilità e Sicurezza	12
Table 7 Tabella dei Test di Sistema	13
Table 8 Tabella dei Test di Unità	17
Table 9	24
Table 10 Tabella dei Test di Accettazione	25
Table 11 Storico delle azioni di miglioramento (Periodo RTB)	39
Table 12 Ottimizzazione tecnologica	41
Table 13 Miglioramento dell'efficienza organizzativa	41
Table 14 Definizione e gestione delle responsabilità e risoluzione blocchi metodologici .	41

1 Introduzione

1.1 Contesto del Progetto

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

L'obiettivo del progetto è lo sviluppo di una piattaforma ad Agenti per l'audit e la remediation automatizzata delle vulnerabilità nei repository GitHub, in conformità con quanto definito dal capitolato C2.

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 e per i Verificatori, 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), le metriche di misurazione e le relative soglie di accettazione/ottimalità.
- **Controllo di Qualità (Quality Control):** pianificazione operativa delle attività di Verifica (analisi statica, test dinamici) per garantire la correttezza tecnica degli artefatti prodotti.
- **Validazione di Prodotto:** definizione delle procedure necessarie per accertare la conformità del sistema rispetto alle aspettative degli Stakeholder e ai requisiti del capitolato.
- **Miglioramento Continuo:** applicazione di meccanismi retroattivi (basati sul ciclo Plan-Do-Check-Act) che utilizzano i risultati delle misurazioni per ottimizzare i processi e il **Way of Working** 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 milestones:

1.3.1 Revisione dei Requisiti e della Tecnologia (RTB)

Per la milestone RTB (06/02/2026), 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):** 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.

1.3.2 Revisione di Accettazione (Product Baseline – PB)

Per il rilascio finale (21/03/2026), 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** e raggiungimento delle soglie di copertura del codice (Code Coverage) 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. Le occorrenze dei termini presenti nel Glossario sono evidenziate nel testo mediante apposita formattazione.

Versione aggiornata del Glossario: [Link al Glossario](#)

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>
- **Norme di Progetto:** Il documento definisce il "Way of Working", stabilendo gli strumenti e le procedure che questo Piano si occupa di misurare.
Documento interno

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>
- **ISO/IEC 12207:2008:** Systems and software engineering — Software life cycle processes.
<https://ieeexplore.ieee.org/document/4475826>
- **Dispense del corso di Ingegneria del Software – Qualità del software**
<https://www.math.unipd.it/~tullio/IS-1/2025/Dispense/T06.pdf>

1.6 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 sostenibile, tracciabile e soggetto a miglioramento continuo attraverso l'applicazione del ciclo PDCA.

1.6.1 Centralizzazione delle Metriche e Obiettivi

Il presente documento costituisce il riferimento unico, autoritativo e analitico per la gestione della qualità del progetto Code Guardian. Mentre le Norme di Progetto definiscono le procedure operative, gli strumenti e le responsabilità per l'estrazione dei dati, il Piano di Qualifica ha il compito di centralizzare la “scienza della misurazione” del gruppo.

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. 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.

1.6.2 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)

Riferimento: Norme di Progetto, Sezione [Stabilità dei Requisiti] 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	$RSI = \frac{R_{tot} - \Delta R}{R_{tot}} \times 100$	$\geq 75\%$	100%

Table 2: Soglie metriche per il processo di Sviluppo

1.6.3 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

1.6.4 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

1.7 Qualità di Prodotto

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

1.7.1 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

1.7.2 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

2 Metodi 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**. I Test di Regressione e i Test di Integrazione, qui non presenti, verranno identificati durante lo svolgimento delle attività per la *Product Baseline* (PB).

2.1 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 validazione sintattica dell'URL del repository (protocollo e dominio).	UC4.1.1
TS-4.2	Verificare la segnalazione di errore in caso di repository inesistente o non accessibile.	UC4.1.2
TS-4.3	Verificare l'inibizione della richiesta se il campo URL risulta vuoto.	UC4.1.3
TS-4.4	Verificare l'inibizione della richiesta se non viene selezionata almeno un'area di interesse.	UC4.2.1
TS-4.5	Verificare la visualizzazione dell'informativa se l'analisi risulta già aggiornata rispetto ai dati remoti.	UC4.0.1
TS-4.6	Verificare la visualizzazione dell'informativa se esiste un'analisi già pendente per lo stesso repository.	UC4.0.2
TS-5	Verificare la navigazione e la corretta visualizzazione dell'elenco dei repository analizzati.	UC5, UC5.1

ID Test	Descrizione	UC Riferimento
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
TS-6.2	Verificare l'inibizione del rendering e la notifica se non viene selezionata alcuna area.	UC6.0.1
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

ID Test	Descrizione	UC Riferimento
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
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

Table 7: Tabella dei Test di Sistema

2.2 Test di Unità

TODO: Verificare la numerazione e corrispondenza Requisito/Test

ID Test	Requisito	Descrizione Test	Risultato Atteso
TU-1.1	FROb1	Verifica rendering del componente di creazione account.	Caricamento corretto del modulo di registrazione.
TU-1.2	FROb2	Verifica del metodo di validazione globale pre-invio.	Blocco della procedura in caso di parametri non validi.
TU-1.3	FROb3	Verifica della logica di persistenza dei dati utente nel DB.	Le credenziali vengono scritte correttamente nel database.
TU-1.4	FROb4	Verifica della funzione di hashing: algoritmo sicuro + salt.	La password non è leggibile; l'hash prodotto è coerente con gli standard.
TU-1.5	FROb5	Controllo presenza campi obbligatori (null check).	Rilevamento campi vuoti nel modulo di registrazione.
TU-1.6	FROb6	Verifica logica di inibizione della registrazione.	Impossibilità di procedere con dati obbligatori mancanti.
TU-1.7	FROb7	Verifica attivazione notifiche di errore per campi vuoti.	Visualizzazione degli avvisi specifici per ogni campo mancante.
TU-1.8	FROb8	Verifica input username nel modulo.	Il valore digitato viene acquisito correttamente.
TU-1.9	FROb9	Test di validazione lunghezza username (limiti 4-20).	Restituisce errore per stringhe < 4 o > 20 caratteri.
TU-1.10	FROb10	Verifica regex per caratteri esclusivamente alfanumerici.	Esclusione di caratteri speciali e spazi.
TU-1.11	FROb11	Verifica trigger notifica errore formato username.	Comparsa del messaggio di errore in caso di violazione regex.
TU-1.12	FROb12	Query di verifica unicità dello username nel database.	Identificazione di collisioni con account già esistenti.
TU-1.13	FROb13	Verifica trigger notifica username già occupato.	Feedback visivo immediato per username non disponibile.
TU-1.14	FROb14	Verifica input email nel modulo.	Acquisizione corretta della stringa email.

ID Test	Requisito	Descrizione Test	Risultato Atteso
TU-1.15	FROb15	Validazione sintattica email secondo standard RFC.	Accettazione di formati standard (user@domain.ext).
TU-1.16	FROb16	Verifica trigger notifica errore sintassi email.	Feedback visivo in caso di email malformata.
TU-1.17	FROb17	Query di verifica unicità email nel database.	Identificazione di email già associate ad altri profili.
TU-1.18	FROb18	Verifica trigger notifica email già registrata.	Messaggio di errore per email duplicata.
TU-1.19	FROb19	Verifica input chiave di accesso.	Acquisizione sicura della stringa password.
TU-1.20	FROb20	Verifica requisiti complessità password (min 8, A, a, 1, !).	Validazione positiva solo se tutti i criteri sono soddisfatti.
TU-1.21	FROb21	Verifica trigger notifica password non sicura.	Elenco puntuale dei criteri di sicurezza non rispettati.
TU-2.1	FROb22	Verifica rendering pagina di Login.	Visualizzazione corretta del form di autenticazione.
TU-2.2	FROb23	Verifica autorizzazione post-validazione credenziali.	Rilascio della sessione solo con dati corretti.
TU-2.3	FROb24	Verifica protocollo di trasmissione credenziali (HTTPS).	Dati criptati durante il transito verso il server.
TU-2.4	FROb25	Test del meccanismo di rate limiting / logout.	Blocco dell'IP o account dopo N tentativi falliti.
TU-2.5	FROb26	Verifica rilevamento campi mancanti nel login.	Trigger errore per username o password non inseriti.
TU-2.6	FROb27	Verifica blocco autenticazione per form incompleto.	Negazione accesso in assenza di credenziali.
TU-2.7	FROb28	Verifica visualizzazione avviso campi obbligatori login.	Presenza di messaggio d'errore a schermo.
TU-2.8	FROb29	Verifica immissione username per autenticazione.	Stringa username correttamente catturata.
TU-2.9	FROb30	Verifica notifica formato username errato in login.	Messaggio informativo su errore sintattico.

ID Test	Requisito	Descrizione Test	Risultato Atteso
TU-2.10	FROb31	Verifica query esistenza account per username.	Controllo presenza utente prima del check password.
TU-2.11	FROb32	Verifica messaggio errore "Identificativo non trovato".	Feedback specifico per username inesistente.
TU-2.12	FROb33	Verifica immissione password per autenticazione.	Stringa password correttamente catturata.
TU-2.13	FROb34	Verifica notifica formato password errato in login.	Feedback su errore sintattico password.
TU-2.14	FROb35	Verifica corrispondenza hash password fornita vs DB.	Confronto sicuro tra input e dato persistito.
TU-2.15	FROb36	Verifica messaggio errore "Password errata".	Feedback specifico per credenziali non corrispondenti.
TU-3.1	FROb37	Verifica accesso area integrazioni GitHub.	Rendering sezione collegamento account.
TU-3.2	FROb38	Verifica gestione reindirizzamento OAuth GitHub.	Generazione URL corretto verso dominio esterno.
TU-3.3	FROb39	Verifica autorizzazione post-integrazione riuscita.	Abilitazione interazioni repository per l'utente.
TU-3.4	FROb40	Verifica cifratura Access Token nel database.	I token GitHub non sono salvati in chiaro.
TU-3.5	FROb41	Verifica associazione univoca Token-Sessione.	Il token non è accessibile da altre sessioni utente.
TU-3.6	FROb42	Verifica rendering avviso pre-reindirizzamento.	Visualizzazione informativa di sicurezza.
TU-3.7	FROb43	Verifica gestione comando conferma/annulla redirect.	Esecuzione o blocco del redirect in base all'input.
TU-3.8	FROb44	Verifica gestione errore sincronizzazione (timeout/dati).	Feedback su fallimento tecnico della procedura.
TU-3.9	FROb45	Verifica blocco associazione per profilo GitHub già in uso.	Inibizione collegamento se account già censito.

ID Test	Requisito	Descrizione Test	Risultato Atteso
TU-3.10	FROb46	Verifica gestione errore “Negato Consenso” GitHub.	Feedback su interruzione volontaria dell’utente.
TU-4.1	FROb47	Verifica accesso sezione configurazione analisi.	Rendering modulo di richiesta audit.
TU-4.2	FROb48	Verifica predisposizione campi URL e Aree Interesse.	Presenza degli input richiesti nel modulo.
TU-4.3	FROb49	Verifica validazione globale parametri pre-audit.	Inoltro consentito solo con tutti i dati validi.
TU-4.4	FROb50	Verifica titolarità repository (lato server).	Il sistema convalida il possesso del repo prima dell’invio.
TU-4.5	FROb51	Verifica generazione ID univoco per la richiesta.	Tracciabilità garantita tramite identificativo univoco.
TU-4.6	FROb52	Verifica immissione URL nel campo dedicato.	Cattura corretta dell’indirizzo repository.
TU-4.7	FROb53	Verifica vincoli sintattici URL (https + github.com).	Validazione positiva solo per domini GitHub sicuri.
TU-4.8	FROb54	Verifica notifica errore formato URL.	Comparsa messaggio “URL non conforme”.
TU-4.9	FROb55	Verifica controllo accessibilità repository remoto.	Test di raggiungibilità della risorsa GitHub.
TU-4.10	FROb56	Verifica notifica inaccessibilità repository.	Feedback su repo privato o non esistente.
TU-4.11	FROb57	Verifica inibizione invio per URL mancante.	Messaggio di errore se il campo URL è vuoto.
TU-4.12	FROb58	Verifica selezione multipla aree di interesse.	Memorizzazione corretta del set di aree selezionate.
TU-4.13	FROb59	Verifica inibizione invio per nessuna area scelta.	Messaggio di errore se il set di aree è vuoto.
TU-4.14	FROb60	Verifica coerenza hash report vs commit GitHub.	Confronto tra versioni del codice per skip analisi.

ID Test	Requisito	Descrizione Test	Risultato Atteso
TU-4.15	FROb61	Verifica blocco invio per report up-to-date.	Informativa su analisi già aggiornata.
TU-4.16	FROb62	Verifica rilevamento processi analisi attivi.	Controllo lock sul repository nel database.
TU-4.17	FROb63	Verifica impedimento invio se analisi in corso.	Feedback su audit già pendente.
TU-5.1	FROb64	Verifica navigazione verso sezione riepilogativa.	Accesso riuscito alla dashboard repository.
TU-5.2	FROb65	Controllo mapping metadati: Nome, URL, Data.	Oggetti lista popolati correttamente.
TU-5.3	FROb66	Validazione logica per visualizzazione informativa lista vuota.	Messaggio mostrato se l'array repository è vuoto.
TU-6.1	FROb67	Verifica selezione e caricamento report da lista.	Caricamento riuscito dei dati del report selezionato.
TU-6.2	FROb68	Verifica stato dei filtri aree (toggle on/off).	Le aree analitiche sono correttamente filtrate.
TU-6.3	FROb69	Controllo validazione: almeno un'area attiva.	Blocco visualizzazione se nessuna sezione è filtrata.
TU-6.4	FROb70	Verifica correttezza timestamp generazione audit.	Data e ora corrispondono al record del database.
TU-6.5	FROb71	Verifica corrispondenza hash commit GitHub.	L'identificativo mostrato è identico all'hash originale.
TU-6.6	FROb72	Controllo visualizzazione username richiedente.	Lo username corrisponde all'autore della richiesta.
TU-6.7	FROb73	Verifica integrità metriche tecniche per aree scelte.	Dati numerici visualizzati correttamente.
TU-6.8	FROb74	Verifica associazione remediation-criticità.	Suggerimenti estratti correttamente per ogni difetto.
TU-6.9	FROb75	Controllo messaggio esito positivo in assenza di criticità.	Informativa mostrata per aree conformi agli standard.

ID Test	Requisito	Descrizione Test	Risultato Atteso
TU-7.1	FROb76	Verifica selezione limiti Start/End nel componente calendario.	Intervallo acquisito correttamente.
TU-7.2	FROb77	Verifica invio comando aggiornamento confronto.	Trigger di ricalcolo comparativo attivato.
TU-7.3	FROb78	Controllo campi obbligatori temporali.	Errore restituito per input nulli.
TU-7.4	FROb79	Verifica query di ricerca report in intervallo.	Messaggio "No Data" restituito se il range è vuoto.
TU-7.5	FROb80	Validazione coerenza: Start Date < End Date.	Blocco se la data d'inizio è successiva alla fine.
TU-7.6	FROb81	Controllo ampiezza massima intervallo (12 mesi).	Errore restituito se l'intervallo è troppo vasto.
TU-8.1	FRDe1	Verifica logica di generazione dataset per grafici.	Dati trasformati correttamente in serie storiche.
TU-8.2	FRDe2	Verifica estrazione dati puntuali all'interazione con grafico.	Valore esatto restituito al click sull'elemento.
TU-8.3	FRDe3	Verifica popolamento righe tabella comparativa.	Corrispondenza 1:1 tra report e righe tabella.
TU-8.4	FRDe4	Validazione algoritmo trend di variazione percentuale.	Calcolo (Delta/Valore)*100 eseguito correttamente.
TU-9.1	FROb82	Verifica caricamento sottomodulo qualità codice.	Modulo renderizzato con parametri corretti.
TU-9.2	FROb83	Verifica parser risultati analisi statica.	Bug, vulnerabilità e code smell identificati.
TU-9.3	FROb84	Controllo calcolo percentuale copertura test.	Valore normalizzato 0-100% calcolato correttamente.
TU-9.4	FROb85	Verifica mappatura remediation per area codice.	Soluzioni visualizzate coerentemente con i bug.
TU-9.5	FROb86	Controllo informativa "Code Clean".	Esito positivo mostrato se non ci sono remediation.

ID Test	Requisito	Descrizione Test	Risultato Atteso
TU-10.1	FROb87	Verifica del parser per vulnerabilità CVE nelle dipendenze.	Identificazione corretta di librerie obsolete dal JSON di audit.
TU-10.2	FROb88	Verifica del mappatore di conformità OWASP Top 10.	Associazione corretta tra vulnerabilità e categoria OWASP.
TU-10.3	FROb89	Validazione logica "Security esito positivo".	Restituisce stato "Safe" se il contatore CVE è pari a zero.
TU-11.1	FROb90	Verifica rilevamento link interrotti (broken links).	Identifica URL malformati o non raggiungibili nella doc.
TU-11.2	FROb91	Calcolo della percentuale di copertura documentale.	Rapporto coerente tra entry del codice e blocchi di documentazione.
TU-11.3	FROb92	Verifica mappatura remediation per area documentazione.	Suggerimenti di miglioramento visualizzati correttamente.
TU-11.4	FROb93	Controllo informativa "Doc Clean".	Esito positivo mostrato in assenza di criticità documentali.
TU-12.1	FROb94	Algoritmo di ordinamento decrescente del ranking.	La lista viene ordinata correttamente dal punteggio più alto al più basso.
TU-12.2	FROb95	Verifica del calcolo del punteggio di qualità globale.	Il punteggio finale è la media pesata dei punteggi di sezione.
TU-13.1	FROb96	Verifica della funzione di revoca token OAuth.	Il token viene rimosso correttamente dallo stato di sessione.
TU-14.1	FRDe5	Validazione formato di esportazione supportato.	Accetta esclusivamente PDF o JSON come input validi.
TU-14.2	FRDe6	Verifica del modulo di generazione file (parsing).	I dati del report vengono mappati senza perdite nel file di output.
TU-15.1	FROb97	Confronto hash tra password inserita e password DB.	Restituisce true solo se i due hash corrispondono.

ID Test	Requisito	Descrizione Test	Risultato Atteso
TU-15.2	FROb98	Validatore criteri complessità nuova password.	Rigetto di password che non includono caratteri speciali o numeri.
TU-15.3	FROb99	Controllo eterogeneità (Vecchia Password != Nuova).	Restituisce errore se il nuovo hash è identico al precedente.

Table 8: Tabella dei Test di Unità

2.3 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.
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 alle funzionalità di audit sia interdetto agli utenti che non hanno completato con successo l'integrazione con GitHub.
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.

ID Test	Descrizione
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.
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.

2.4 Test di Accettazione

ID Test	Descrizione
TA-1	Verificare che il prodotto dia la possibilità all'Utente di registrarsi e autenticarsi a CodeGuardian

ID Test	Descrizione
TA-2	Verificare che il prodotto dia la possibilità all'Utente di collegare il proprio account GitHub tramite OAuth
TA-3	Verificare che il prodotto dia la possibilità di richiedere l'analisi di un repository GitHub
TA-4	Verificare che il prodotto dia la possibilità di visualizzare i report di analisi completi con tutte le sezioni
TA-5	Verificare che il prodotto dia la possibilità di filtrare i report passati per intervallo temporale
TA-6	Verificare che il prodotto gestisca correttamente il caso di utente senza report disponibili
TA-7	Verificare che il prodotto dia la possibilità di confrontare report tramite grafici comparativi interattivi
TA-8	Verificare che il prodotto dia la possibilità di confrontare report tramite tabelle con indicatori di variazione
TA-9	Verificare che il prodotto dia la possibilità di visualizzare dettagli delle vulnerabilità di sicurezza (analisi statica, dipendenze, OWASP)
TA-10	Verificare che il prodotto dia la possibilità di visualizzare la qualità della documentazione (errori spelling, completezza)
TA-11	Verificare che il prodotto dia la possibilità di disconnettere l'account GitHub con richiesta di conferma
TA-12	Verificare che il prodotto dia la possibilità di esportare report in formati configurabili
TA-13	Verificare che il prodotto dia la possibilità di modificare la password con validazione dei criteri di sicurezza
TA-14	Verificare che il prodotto dia la possibilità di visualizzare e gestire suggerimenti di remediation per vulnerabilità dipendenze
TA-15	Verificare che il prodotto dia la possibilità di rilevare e gestire segreti/token esposti con revoca automatica opzionale
TA-16	Verificare che il prodotto dia la possibilità di verificare conformità licenze con integrazione al processo di approvazione legale
TA-17	Verificare che il prodotto dia la possibilità di ricevere revisione automatizzata delle Pull Request con test e codemods

ID Test	Descrizione
TA-18	Verificare che il prodotto dia la possibilità di monitorare la qualità del codice con integrazione a tool esterni e KPI
TA-19	Verificare che il prodotto dia la possibilità di ricevere suggerimenti di refactoring con verifica impatto tramite test
TA-20	Verificare che il prodotto dia la possibilità di generare changelog automatici con rilevamento breaking changes
TA-21	Verificare che il prodotto dia la possibilità di applicare policy CI/CD dinamiche per branch differenti
TA-22	Verificare che il prodotto dia la possibilità di configurare report programmabili con alert automatici
TA-23	Verificare che il prodotto crei correttamente ambienti sandbox e gestisca errori di creazione
TA-24	Verificare che il prodotto integri correttamente tool esterni di analisi con gestione fallback
TA-25	Verificare che il prodotto generi e trasferisca correttamente i report finali al frontend
TA-26	Verificare che il prodotto notifichi correttamente l'utente al completamento dell'analisi
TA-27	Verificare che il prodotto gestisca correttamente retry di comunicazione in caso di errori di rete
TA-28	Verificare che il prodotto gestisca correttamente errori critici durante l'analisi con notifiche appropriate
TA-29	Verificare che il prodotto salvi correttamente metadati, report e metriche aggregate nel database
TA-30	Verificare che il prodotto gestisca correttamente il workflow completo OAuth per collegamento GitHub
TA-31	Verificare che il prodotto completi correttamente il workflow end-to-end
TA-34	Verificare che il prodotto mantenga la sincronizzazione corretta dei dati tra frontend e backend durante tutte le operazioni
TA-35	Verificare che il prodotto sia scalabile e gestisca correttamente richieste multiple concorrenti

Table 10: Tabella dei Test di Accettazione

3 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.

<<<<<< **HEAD** 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.

3.1 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.

3.1.1 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.

3.1.2 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.

3.1.3 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.

3.2 Processi Primari: Sviluppo

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

3.2.1 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**).

3.3 Processi di Supporto: Documentazione

Monitoraggio della qualità formale e della fruibilità della documentazione prodotta.

3.3.1 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.

3.4 Processi di Supporto: Verifica

Monitoraggio dell'efficacia delle attività di testing dinamico.

3.4.1 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.

3.5 Processi di Supporto: Gestione della Qualità

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

3.5.1 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.

3.6 Processi Organizzativi: Gestione dei Processi

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

3.6.0.0.0.1

3.7 Finalità del Cruscotto

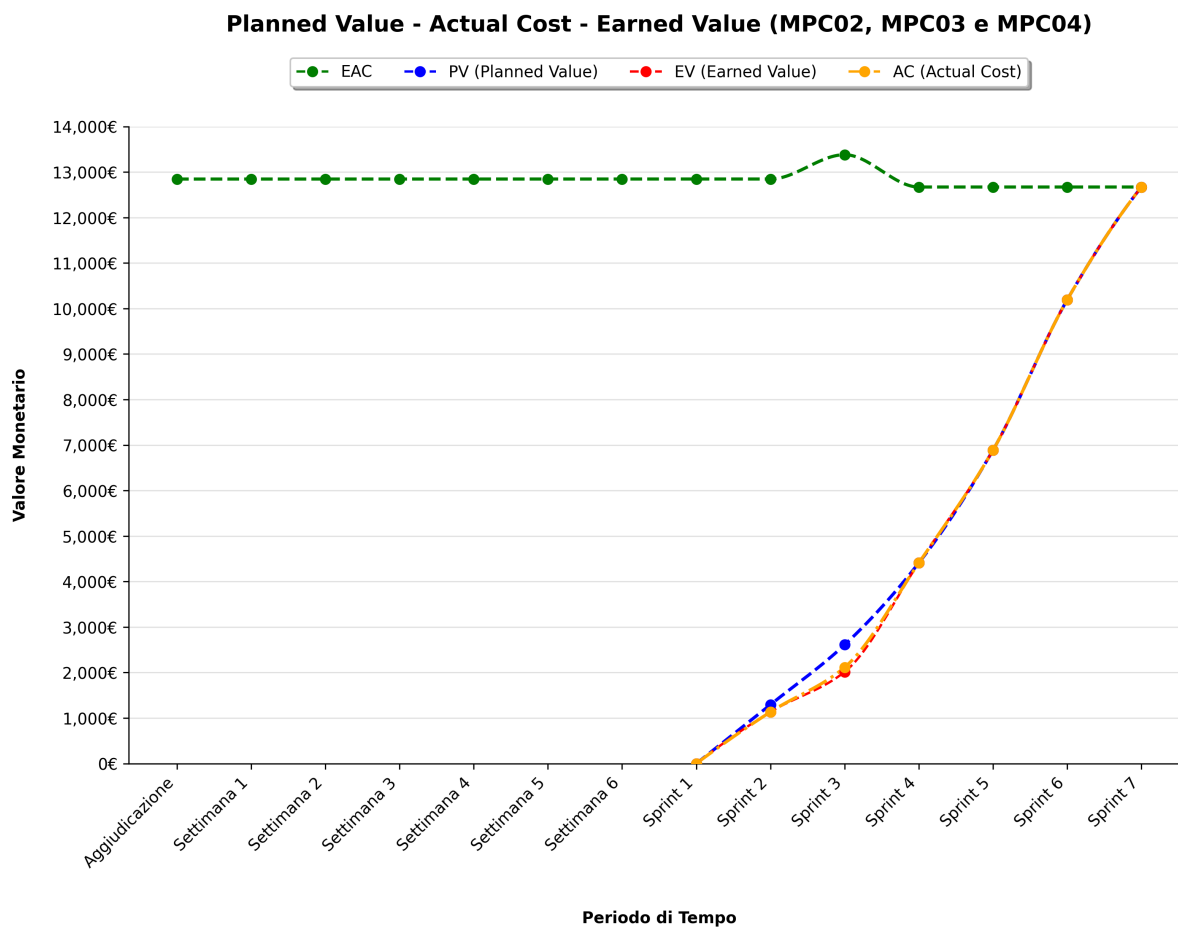
Il cruscotto di valutazione assolve a due funzioni fondamentali:

- **Monitoraggio Proattivo:** Consente di rilevare scostamenti rispetto alle soglie definite all'interno del *Piano di Qualifica* stesso, evitando che criticità latenti si trasformino in blocchi operativi.
- **Tracciabilità Storica:** Documenta l'evoluzione delle prestazioni del gruppo nel tempo, permettendo di identificare pattern ricorrenti e validare l'efficacia delle azioni correttive implementate.

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.

3.8 Processi Primari: Fornitura e Sviluppo

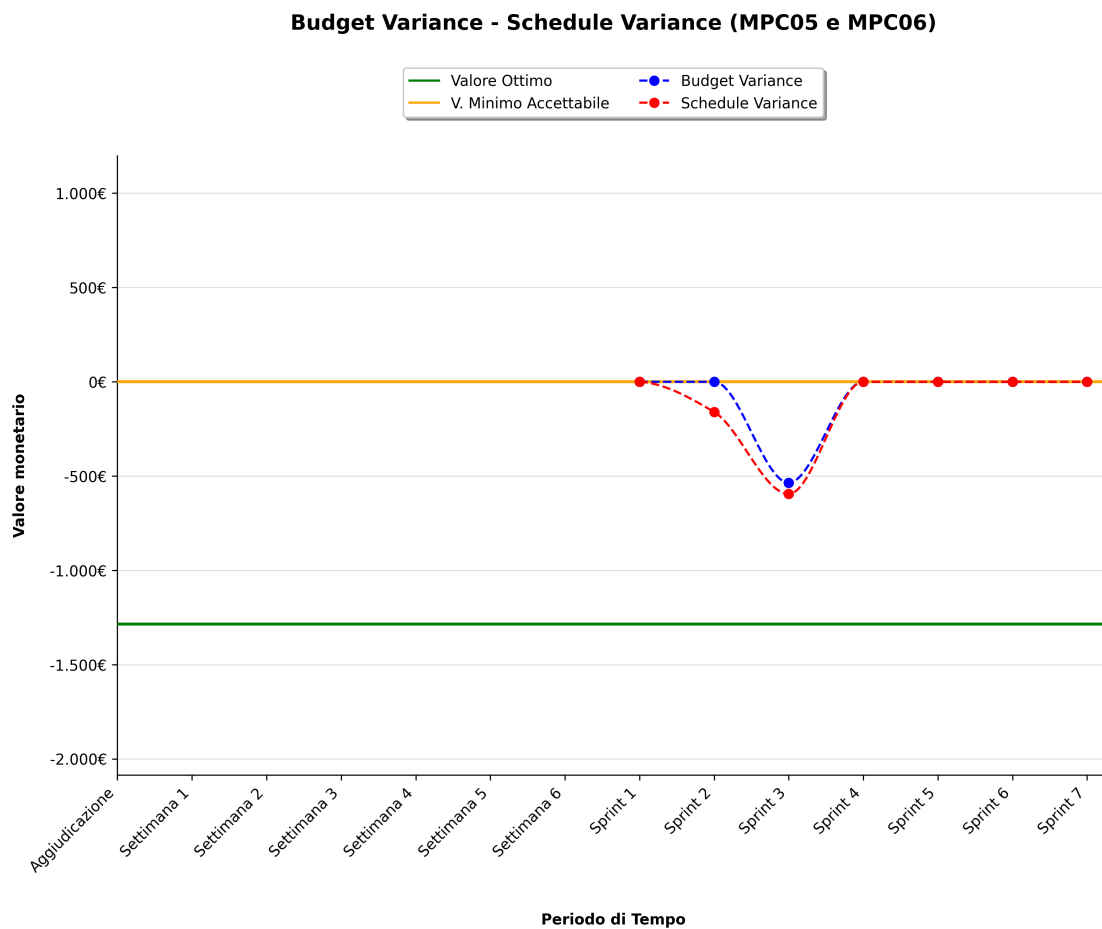
3.8.1 Planned Value - Actual Cost - Earned Value (MPC02, MPC03 e MPC04)



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.

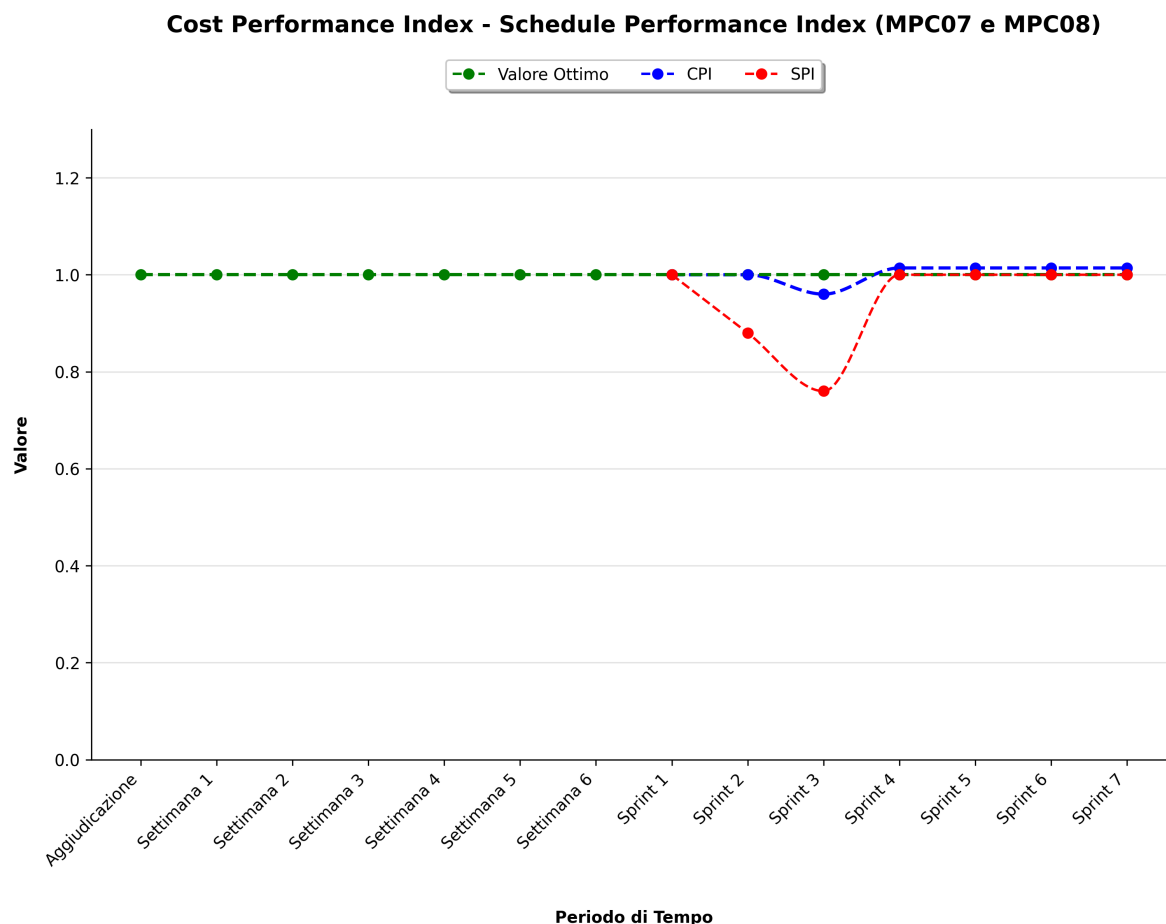
Tuttavia, la situazione ha subito una variazione significativa durante lo *Sprint 2*: a fronte di un incremento del *Planned Value* (PV) e dell'*Actual Cost* (AC), l'*Earned Value* (EV) ha subito una flessione. Questo testimonia l'insorgere di inefficienze produttive e debito tecnico, legati alla necessità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

3.8.2 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).

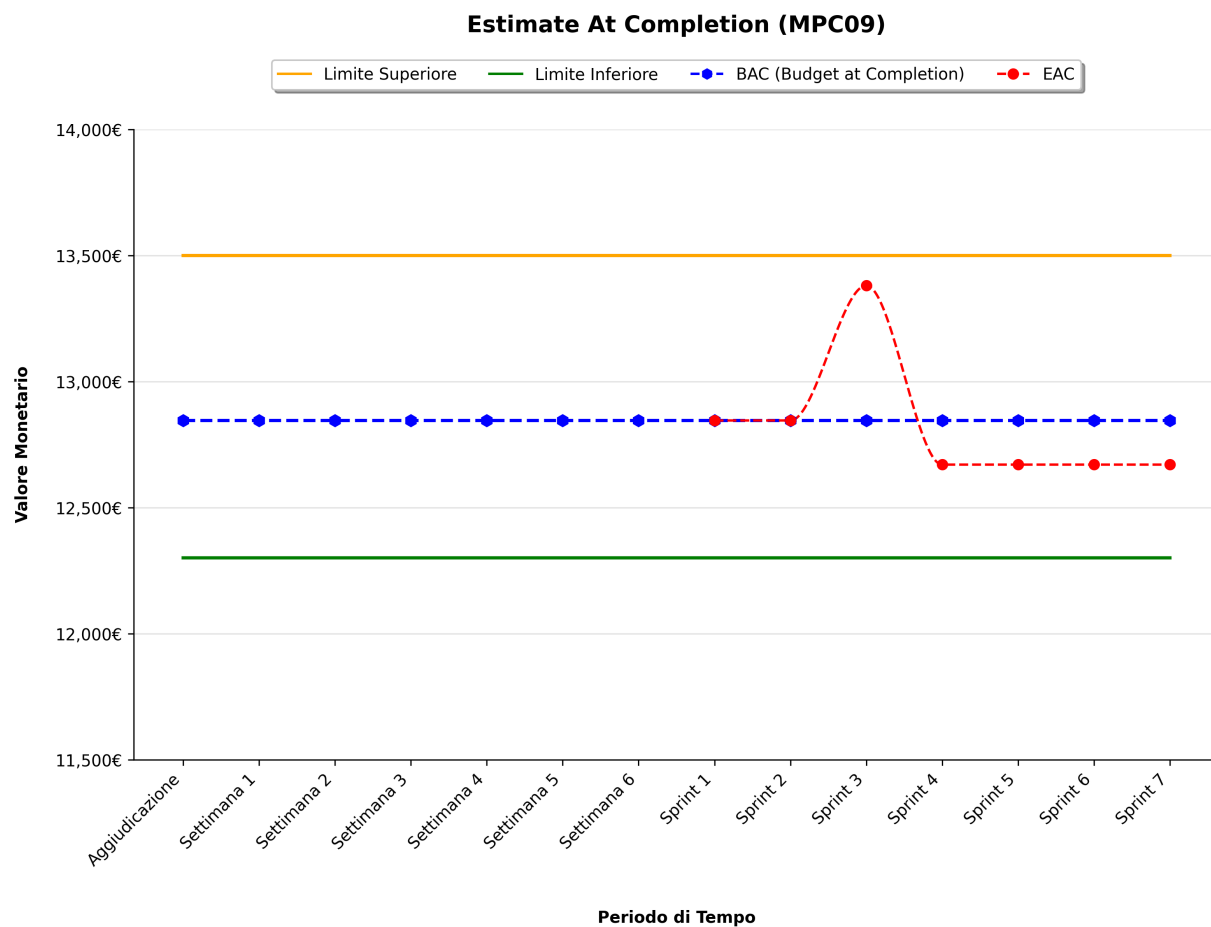
3.8.3 Cost Performance Index - Schedule Performance Index (MPC07 e MPC08)



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) che, essendo pari a 1, indica un ottimo utilizzo del budget.

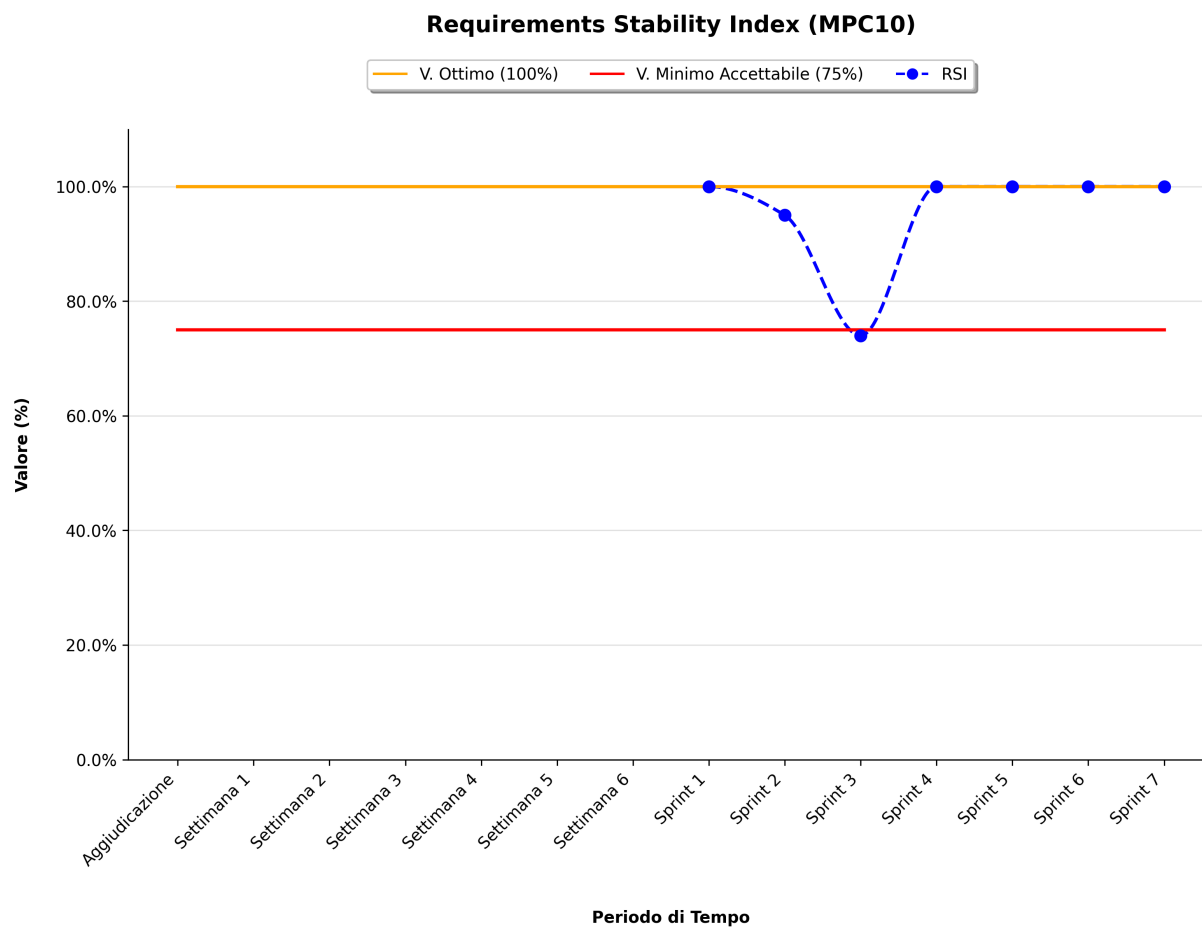
La situazione è peggiorata nel corso dello *Sprint 2*, durante il quale si è verificato un crollo dello *Schedule Performance Index* (SPI) che segnala un ritardo critico rispetto alla pianificazione. La cattiva gestione temporale è in contrasto con il *Cost Performance Index* (CPI) che si mantiene abbastanza stabile, confermando che il problema non è di natura economica ma organizzativa.

3.8.4 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.

3.8.5 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.

3.9 Processi di Supporto

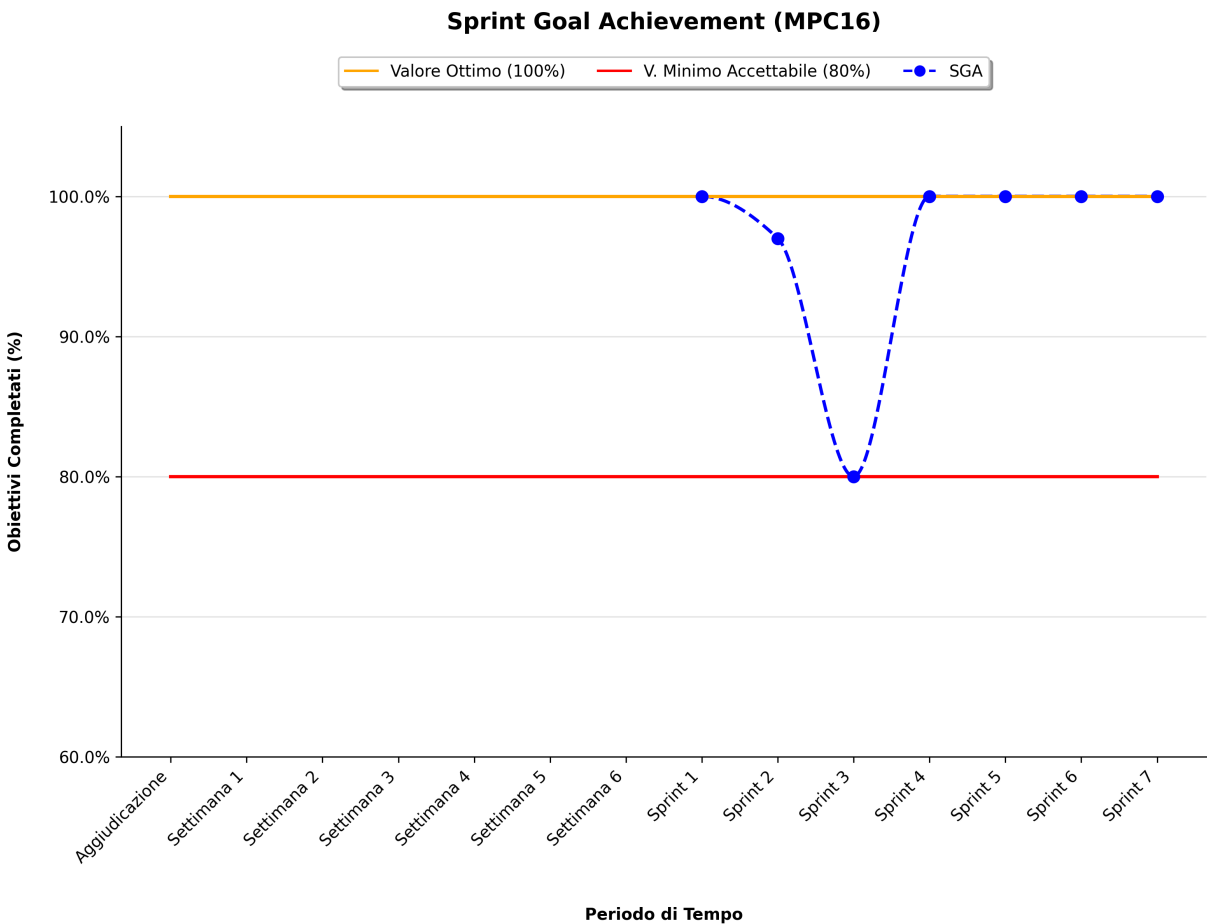
3.9.1 Gulpease Index (MPC11)

3.9.2 Correttezza Ortografica (MPC12)

3.10 Processi Organizzativi

3.10.1 Metrics Satisfaction (MPC15)

3.10.2 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. >>>>>> 50019de0eb6debfce9f7cbf985aae7591629f735

3.10.3 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.

3.11 Qualità di Prodotto

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

3.11.1 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.

3.11.2 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.

3.11.3 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.

<<<<<<< HEAD

4 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:

- **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.

4.1 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	Adozione di Template e Funzioni Comuni

ID	Problema / Causa	Azione Correttiva
	La stesura parallela dei documenti da parte di più persone ha inizialmente generato incoerenze stilistiche e ripetizioni ridondanti o formattate diversamente.	Per garantire coerenza, sono state ingegnerizzate le funzioni di typesetting (in 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 11: Storico delle azioni di miglioramento (Periodo RTB)

5 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 redistribuzione 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.

5.0.0.0.0.1

5.1 Automiglioramento

5.1.1 Introduzione

Il miglioramento continuo risulta fondamentale per garantire la qualità del progetto Code Guardian. Seguendo il Way of Working definito nelle **Norme di Progetto**, il team effettua retrospettive periodiche per identificare i colli di bottiglia operativi e implementare soluzioni correttive secondo il ciclo PDCA. Le valutazioni sono state suddivise in tre ambiti critici identificati durante lo sviluppo iniziale.

5.1.1.1 Valutazione Tecnologica

L'adozione di nuovi strumenti ha richiesto una fase di adattamento per garantire che l'infrastruttura tecnologica supportasse, e non ostacolasse, la produttività.

Strumento	Problema	Decisione presa
<u>Typst</u>	Curva di apprendimento ripida e rischio di disomogeneità stilistica nei documenti.	Studio autonomo obbligatorio e creazione di template condivisi per centralizzare la logica di formattazione.
<u>Issue Tracking System</u>	Frammentazione delle informazioni tecniche e incomprensioni sui requisiti.	Centralizzazione della comunicazione asincrona su <u>Jira</u> , con obbligo di risoluzione dei dubbi tramite commenti tracciabili sulle singole task.

Table 12: Ottimizzazione tecnologica

5.1.1.2 Valutazione Organizzativa

Il coordinamento di un gruppo numeroso ha richiesto un passaggio da una comunicazione informale a una struttura più gerarchica e definita.

Criticità	Soluzione Organizzativa
Difficoltà di allineamento immediato su decisioni logistiche e urgenze.	Definizione di canali gerarchici: <u>Telegram</u> per le urgenze, <u>Discord</u> per il lavoro sincrono e i meeting di allineamento.
Sovrapposizione di sforzi o “buchi” operativi dovuti alla dimensione del gruppo (7 persone).	Suddivisione in sotto-gruppi di lavoro tematici per ridurre il rumore comunicativo e aumentare la focalizzazione.

Table 13: Miglioramento dell'efficienza organizzativa

5.1.1.2.1 Valutazione delle Responsabilità

Per evitare lo stallo decisionale e risolvere ambiguità metodologiche, è stato necessario definire chiaramente i confini d'azione dei ruoli e attivare canali di supporto esterni.

Problema di Ruolo / Criticità	Azioni di Risposta
Mancanza di una visione d'insieme su documenti complessi come l'Analisi dei Requisiti.	Nomina di un referente responsabile per ogni macro-documento, incaricato di supervisionare la coerenza finale e il rispetto delle scadenze.
Forte difficoltà degli <u>Analisti</u> nel tracciare correttamente attori e sistemi in conformità con gli standard richiesti.	Attivazione di una strategia di chiarimento a più livelli: <ul style="list-style-type: none"> • Consultazione diretta con il Prof. Cardin per risolvere dubbi metodologici; • Confronto costruttivo con altri gruppi di progetto per allineamento sugli standard;

Problema di Ruolo / Criticità	Azioni di Risposta
	<ul style="list-style-type: none">Richiesta di intervento dell'azienda proponente (<u>Var Group</u>) per chiarire il perimetro operativo del sistema.
Incertezza sulla validazione degli incrementi prodotti e rischio di errori latenti.	Rafforzamento del ruolo dei <u>Verificatori</u> , con l'introduzione di una revisione obbligatoria "a quattro occhi" prima di ogni merge sul repository principale.

Table 14: Definizione e gestione delle responsabilità e risoluzione blocchi metodologici

5.1.1.3 Conclusioni

Il processo di automiglioramento ha permesso di trasformare le criticità iniziali — tipiche di un gruppo numeroso che opera su tecnologie nuove — in punti di forza procedurali. L'integrazione tra responsabilità chiare, strumenti di tracking e canali di comunicazione dedicati garantisce la sostenibilità del progetto verso la milestone PB. >>>>>>

50019de0eb6debfc9f7cbf985aae7591629f735