



---

# Piano di Qualifica

v0.5.1

---

## Registro delle Modifiche

Data	Versione	Descrizione	Redattore	Verificatore
2026/01/13	0.5.1	Rielaborazione introduzione documento e qualità di processo	Suar Alberto	
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, introduzione, scopo e riferimenti	Zago Alice	Suar Alberto
2025/12/23	0.0.0	Creazione documento	Zago Alice	Suar Alberto

# Indice

<b>Introduzione .....</b>	<b>7</b>
Contesto del Progetto .....	7
Finalità del Documento .....	7
Traguardi Qualitativi .....	7
Revisione dei Requisiti e della Tecnologia (RTB) .....	7
Revisione di Accettazione (Product Baseline – PB) .....	8
Glossario .....	8
Riferimenti .....	8
Riferimenti Normativi .....	8
Riferimenti Informativi .....	8
Nota sull'adozione dello Standard ISO/IEC 25010 e PDCA .....	8
<b>Qualità di Processo .....</b>	<b>9</b>
Processi Primari .....	9
Fornitura .....	9
Sviluppo .....	10
Processi di Supporto .....	10
Documentazione .....	10
Verifica .....	11
Gestione della Qualità .....	11
Processi Organizzativi .....	11
Gestione Organizzativa .....	11
<b>Qualità di Prodotto .....</b>	<b>12</b>
Adeguatezza Funzionale .....	12
Affidabilità (Reliability) .....	12
Usabilità (Usability) .....	12
Manutenibilità (Maintainability) .....	13
Sicurezza (Security) .....	13
<b>Metodi di Testing .....</b>	<b>14</b>
Convenzioni di Nomenclatura .....	14
Test di Sistema .....	14
Test di Unità .....	17
Test di Accettazione .....	19
<b>Cruscotto di Valutazione .....</b>	<b>21</b>
Processi Primari: Fornitura (EVM) .....	21
Trend di Progetto (PV, AC, EV) .....	21
Indici di Efficienza (CPI, SPI) .....	21
Varianze e Previsioni (CV, SV, EAC) .....	21
Processi Primari: Sviluppo .....	21
Requirements Stability Index (RSI) .....	21
Processi di Supporto: Documentazione .....	21
Indice di Gulpease e Correttezza .....	21
Processi di Supporto: Verifica .....	21
Code Coverage e Test Success .....	22
Processi di Supporto: Gestione della Qualità .....	22
Soddisfazione delle Metriche .....	22
Processi Organizzativi: Gestione dei Processi .....	22

---

Sprint Goal Achievement .....	22
Qualità di Prodotto .....	22
Copertura Funzionale .....	22
Affidabilità e Manutenibilità .....	22
Usabilità e Sicurezza .....	22
<b>Miglioramento Continuo .....</b>	<b>23</b>
Azioni di Miglioramento Intraprese .....	23
<b>Conclusioni .....</b>	<b>24</b>

## **Indice immagini**

## Indice tabelle

<b>Table 1 Soglie metriche per il processo di Fornitura (EVM) .....</b>	<b>10</b>
<b>Table 2 Soglie metriche per il processo di Sviluppo .....</b>	<b>10</b>
<b>Table 3 Soglie metriche Documentazione .....</b>	<b>11</b>
<b>Table 4 Soglie metriche Verifica .....</b>	<b>11</b>
<b>Table 5 Soglie metriche Gestione Qualità .....</b>	<b>11</b>
<b>Table 6 Soglie metriche Organizzative .....</b>	<b>11</b>
<b>Table 7 Metriche Adeguatezza Funzionale .....</b>	<b>12</b>
<b>Table 8 Metriche Affidabilità .....</b>	<b>12</b>
<b>Table 9 Metriche Usabilità .....</b>	<b>13</b>
<b>Table 10 Metriche Manutenibilità .....</b>	<b>13</b>
<b>Table 11 Metriche Sicurezza .....</b>	<b>13</b>
<b>Table 12 Tabella dei Test di Sistema .....</b>	<b>14</b>
<b>Table 13 Tabella dei Test di Unità .....</b>	<b>17</b>
<b>Table 14 Tabella dei Test di Accettazione .....</b>	<b>19</b>
<b>Table 15 Storico delle azioni di miglioramento (Periodo RTB) .....</b>	<b>23</b>

## Introduzione

### 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**.

### 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 capitolo.
- **Miglioramento Continuo**: applicazione di meccanismi retroattivi (basati sul ciclo PDCA) che utilizzano i risultati delle misurazioni per ottimizzare i processi e il **Way of Working** in corso d'opera.

In linea con la metodologia Agile e il principio del miglioramento continuo, il Piano di Qualifica adotta il ciclo di Deming (PDCA: Plan-Do-Check-Act). Questo approccio permette di monitorare costantemente l'efficacia del **Way of Working**, attuando azioni correttive e preventive per ottimizzare le performance del gruppo parallelamente all'avanzamento dello sviluppo.

### Traguardi Qualitativi

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

#### Revisione dei Requisiti e della Tecnologia (RTB)

Per la milestone RTB (**05/02/2026**), 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.

## 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.

## 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](#)

## Riferimenti

### 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

### 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.
- **Dispense del corso di Ingegneria del Software – Qualità del software**  
<https://www.math.unipd.it/~tullio/IS-1/2025/Dispense/T06.pdf>

### Nota sull’adozione dello Standard ISO/IEC 25010 e PDCA

La redazione del presente documento e la definizione delle metriche si ispirano alla famiglia di standard ISO/IEC 25010 (SQuaRE). Tale scelta garantisce una classificazione rigorosa delle caratteristiche di qualità del prodotto (es. Sicurezza, Affidabilità, Usabilità). Inoltre, l’adozione strutturale del ciclo PDCA (Plan-Do-Check-Act) assicura che il processo di controllo qualità non sia statico, ma evolva:

- **Plan:** Definizione delle metriche e delle soglie nel presente documento.
- **Do:** Esecuzione delle misurazioni durante gli Sprint.
- **Check:** Analisi degli scostamenti nei periodi di retrospettiva.
- **Act:** Aggiornamento delle Norme di Progetto e ricalibrazione delle metriche per lo Sprint successivo.

## 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.

La strategia adottata si fonda su tre pilastri fondamentali:

- **Modelli di Riferimento:** ISO/IEC 12207 e CMMI.
- **Metriche di Processo:** Misurazione oggettiva dell'efficienza.
- **Revisioni Periodiche:** Ciclo PDCA applicato agli Sprint.

## Processi Primari

I Processi Primari costituiscono il nucleo operativo del ciclo di vita del software.

### Fornitura

*Riferimento: Norme di Progetto, Sezione [Monitoraggio della Fornitura]*

Il processo di fornitura comprende le attività necessarie a consegnare il prodotto al committente rispettando gli accordi contrattuali presi. Vengono monitorati sia i valori assoluti (Misure) che gli indici di efficienza (Metriche) derivati dallo standard Earned Value Management (EVM).

ID	Nome	Formula	V. Accettabile	V. Ottimo
MPC01	Budget At Completion (BAC)	Preventivo	Preventivo	Preventivo
MPC02	Planned Value (PV)	$PV$	$\geq 0$	Da Piano
MPC03	Actual Cost (AC)	$AC$	$\leq EAC$	$\leq EV$
MPC04	Earned Value (EV)	$EV$	$\geq 95\% PV$	$\geq PV$
MPC05	Cost Variance (CV)	$BV = EV - AC$	$\geq 0$	$> 0$
MPC06	Schedule Variance (SV)	$SV = EV - PV$	$> -5\% BAC$	$\geq 0$
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)

**Sviluppo***Riferimento: Norme di Progetto, Sezione [Stabilità dei Requisiti]*Si monitora la stabilità dell'ambito di progetto per evitare modifiche incontrollate (**Scope Creep**).

ID	Nome	Formula	V. Accettabile	V. Ottimo
MPC10	Requirements Stability Index	$RSI = \frac{R_{tot} - \Delta R}{R_{tot}} \times 100$	$\geq 80\%$	100%

Table 2: Soglie metriche per il processo di Sviluppo

**Processi di Supporto****Documentazione***Riferimento: Norme di Progetto, Sezione [Documentazione]*

Si monitora la leggibilità per garantire l'accessibilità delle informazioni.

ID	Nome	Formula	V. Accettabile	V. Ottimo
MPC11	Gulpease Index	$89 + \frac{300(L_f) - 10(L_p)}{F_p}$	$\geq 40$	$\geq 60$
MPC12	Correttezza Ortografica	Errori segnalati	0	0

Table 3: Soglie metriche Documentazione

### Verifica

Riferimento: Norme di Progetto, Sezione [Verifica]

Si monitora la copertura dei test per garantire la robustezza del codice.

ID	Nome	Formula	V. Accettabile	V. Ottimo
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 4: Soglie metriche Verifica

### Gestione della Qualità

Riferimento: Norme di Progetto, Sezione [Gestione Qualità]

ID	Nome	Formula	V. Accettabile	V. Ottimo
MPC15	Metrics Satisfaction	$\frac{\text{Metriche OK}}{\text{Metriche Tot}} \times 100$	$\geq 90\%$	100%

Table 5: Soglie metriche Gestione Qualità

## Processi Organizzativi

### Gestione Organizzativa

Riferimento: Norme di Progetto, Sezione [Gestione Agile]

Si misura l'affidabilità del team nel raggiungere gli obiettivi di Sprint.

ID	Nome	Formula	V. Accettabile	V. Ottimo
MPC16	Sprint Goal Achievement	$\frac{\text{Completati}}{\text{Pianificati}} \times 100$	$\geq 80\%$	100%

Table 6: Soglie metriche Organizzative

## Qualità di Prodotto

La qualità di prodotto è il grado con cui il software soddisfa i requisiti dichiarati e le aspettative implicite degli utenti. A differenza della qualità di processo (che misura *come* lavoriamo), questa sezione misura il risultato finale.

Il gruppo fa riferimento al modello di qualità definito dallo standard internazionale [ISO/IEC 25010](#).

### Adeguatezza Funzionale

Riferimento: Norme di Progetto, Sezione [Requisiti]

Misura il grado di copertura dei requisiti tracciati nell'Analisi dei Requisiti.

ID	Nome	Formula	V. Accettabile	V. Ottimo
MPD01	Copertura Req. Obbligatori	$\frac{\text{Soddisfatti}}{\text{Totale Obbl.}} \times 100$	100%	100%
MPD02	Copertura Req. Desiderabili	$\frac{\text{Soddisfatti}}{\text{Totale Des.}} \times 100$	0%	100%
MPD03	Copertura Req. Opzionali	$\frac{\text{Soddisfatti}}{\text{Totale Opz.}} \times 100$	0%	100%

Table 7: Metriche Adeguatezza Funzionale

### Affidabilità (Reliability)

Riferimento: Norme di Progetto, Sezione [Affidabilità]

Capacità del sistema di mantenere un determinato livello di prestazioni in un dato periodo di tempo.

ID	Nome	Formula	V. Accettabile	V. Ottimo
MPD04	Failure Density	$\frac{\text{N. guasti}}{\text{KLOC}}$	$\leq 0.5$	0
MPD05	Availability	$\frac{\text{Tempo Up}}{\text{Tempo Tot}} \times 100$	$\geq 98\%$	$\geq 99.9\%$

Table 8: Metriche Affidabilità

### Usabilità (Usability)

Riferimento: Norme di Progetto, Sezione [Usabilità]

Grado con cui un prodotto può essere usato da specifici utenti per raggiungere specifici obiettivi con efficacia, efficienza e soddisfazione.

ID	Nome	Formula	V. Accettabile	V. Ottimo
MPD06	Comprehensibility	$\frac{\text{Compresa}}{\text{Totale funzioni}} \times 100$	$\geq 80\%$	100%
MPD07	Error Prevention	$\frac{\text{Undo}}{\text{Azioni Totali}} \times 100$	$\leq 5\%$	$\leq 2\%$

Table 9: Metriche Usabilità

**Manutenibilità (Maintainability)***Riferimento: Norme di Progetto, Sezione [Manutenibilità]*

Grado di efficacia ed efficienza con cui il prodotto può essere modificato.

ID	Nome	Formula	V. Accettabile	V. Ottimo
MPD08	Comment Density	$\frac{\text{Linee commento}}{\text{Linee codice}} \times 100$	$\geq 10\%$	20% – 30%
MPD09	Cyclomatic Complexity	$V(G) = E - N + 2P$	$\leq 15$	$\leq 10$
MPD10	Coupling (Fan-out)	Dipendenze esterne	$\leq 6$	$\leq 3$

Table 10: Metriche Manutenibilità

**Sicurezza (Security)***Riferimento: Norme di Progetto, Sezione [Sicurezza]*

Grado con cui il prodotto protegge le informazioni e i dati.

ID	Nome	Formula	V. Accettabile	V. Ottimo
MPD11	Vulnerability Detection	N. vulnerabilità critiche	0	0

Table 11: Metriche Sicurezza

## Metodi di Testing

Questa sezione definisce la strategia di testing per il progetto *CodeGuardian*. Skarab Group ha adottato un approccio di testing multilivello che copre:

- **Test di Sistema (TS).**
- **Test di Unità (TU).**
- **Test di Accettazione (TA).**

## Convenzioni di Nomenclatura

I test sono identificati secondo il seguente schema:

**Formato:** [Tipologia] - [Gruppo]. [Sottogruppo]

Ogni test è esplicitamente collegato ai **Casi d'Uso (UC)** relativi per garantire copertura funzionale completa.

## Test di Sistema

Di seguito, viene riportata la tabella che definisce i Test di Sistema (TS) necessari per validare il comportamento descritto nei Casi d'Uso.

ID Test	Descrizione	Tipo	UC Riferimento
TS-1.0	Verifica procedura di registrazione completa con dati validi.	Funzionale	UC1
TS-1.1	Verifica validazione formato Nome: rifiuto di numeri e caratteri speciali non ammessi.	Validazione	UC1.1.1
TS-1.2	Verifica validazione formato Cognome: rifiuto di numeri e caratteri speciali non ammessi.	Validazione	UC1.2.1
TS-1.3	Verifica unicità Username: tentativo di registrazione con username già esistente nel DB.	Sicurezza	UC1.3.2
TS-1.4	Verifica validazione formato Username: lunghezza e caratteri non ammessi.	Validazione	UC1.3.1
TS-1.5	Verifica unicità Email: tentativo di registrazione con email già presente nel DB.	Sicurezza	UC1.4.2
TS-1.6	Verifica validazione formato Email: sintassi non valida.	Validazione	UC1.4.1
TS-1.7	Verifica complessità Password: mancato rispetto dei criteri di sicurezza.	Sicurezza	UC1.5.1

ID Test	Descrizione	Tipo	UC Riferimento
TS-2.0	Verifica login con credenziali corrette.	Funzionale	UC2
TS-2.1	Verifica gestione errore per Username non esistente.	Sicurezza	UC2.1.2
TS-2.2	Verifica gestione errore per Password errata.	Sicurezza	UC2.2.2
TS-2.3	Verifica validazione formato input in fase di login (Username/ Password malformati).	Validazione	UC2.1.1, UC2.2.1
TS-3.0	Verifica flusso completo OAuth con GitHub.	Integrazione	UC3, UC3.2
TS-3.1	Verifica rifiuto collegamento da parte dell'utente.	Funzionale	UC3.1.1
TS-3.2	Verifica gestione errore ricezione codice mancante da GitHub.	Integrazione	UC3.2.1
TS-3.3	Verifica gestione errore codice GitHub già associato ad altro utente.	Sicurezza	UC3.2.2
TS-4.0	Verifica invio richiesta analisi con URL valido e opzioni selezionate.	Funzionale	UC4
TS-4.1	Verifica validazione URL Repository: formato non valido.	Validazione	UC4.1.1
TS-4.2	Verifica accessibilità Repository: URL privato o inesistente.	Integrazione	UC4.1.2
TS-4.3	Verifica obbligatorietà selezione aree di interesse (Codice/ Documentazione).	Validazione	UC4.2.1
TS-4.4	Verifica blocco analisi per report già "Up-to-date" (Hash commit invariato).	Ottimizzazione	UC4.3.1
TS-4.5	Verifica blocco analisi concorrente (Analisi già in corso sulla stessa repo).	Stato	UC4.3.2

ID Test	Descrizione	Tipo	UC Riferimento
TS-5.0	Verifica visualizzazione lista report e apertura dashboard di dettaglio.	Funzionale	UC5
TS-5.1	Verifica gestione caso “Nessun report disponibile” per una repo.	Funzionale	UC5.2.1
TS-6.0	Verifica filtro temporale sui report passati (Intervallo valido).	Funzionale	UC6
TS-6.1	Verifica errore filtro temporale: intervallo incoerente (Start > End) o troppo ampio.	Validazione	UC6.2.2, UC6.2.3
TS-6.2	Verifica messaggio “Nessun report nel periodo selezionato”.	Funzionale	UC6.2.1
TS-7.0	Verifica rendering grafico comparativo.	UI/UX	UC7
TS-8.0	Verifica rendering tabella comparativa (Corrispondenza dati con il grafico).	UI/UX	UC8
TS-9.0	Verifica presenza e contenuto sezioni Analisi Codice (Statica, Dipendenze, OWASP).	Contenuto	UC9
TS-9.1	Verifica conteggio vulnerabilità totali (Deve corrispondere alla somma delle vulnerabilità rilevate).	Logica	UC9.4
TS-10.0	Verifica presenza e contenuto sezioni Analisi Documentazione.	Contenuto	UC10
TS-12.0	Verifica coerenza Metadati Report.	Integrazione	UC12

Table 12: Tabella dei Test di Sistema

## Test di Unità

In questa sezione vengono definiti i Test di Unità volti a verificare il corretto funzionamento delle singole componenti software.

ID Test	UC Riferimento	Descrizione	Risultato Atteso
TU-1.1	UC1.1	Verifica funzione di normalizzazione nome: input con spazi iniziali e finali.	La stringa restituita non presenta spazi esterni.
TU-1.2	UC1.1.1	Verifica validatore Nome: input contenente caratteri numerici.	La funzione restituisce <b>false</b> e l'errore specifico.
TU-1.3	UC1.1.1	Verifica validatore Nome: input contenente simboli speciali non ammessi.	La funzione restituisce <b>false</b> .
TU-1.4	UC1.3.1	Verifica validatore Username: lunghezza inferiore al minimo consentito.	La funzione restituisce <b>false</b> .
TU-1.5	UC1.4.1	Verifica validatore Email: formato sintattico errato.	La funzione restituisce <b>false</b> .
TU-1.6	UC1.5.1	Verifica validatore Password: assenza di carattere maiuscolo o numero.	La funzione restituisce <b>false</b> indicando il criterio mancante.
TU-2.1	UC2.1.1	Verifica sanitizzazione input Username in fase di login.	Eventuali caratteri di escape pericolosi vengono rimossi.
TU-2.2	UC2.2.1	Verifica gestione input vuoti su campi obbligatori.	La validazione blocca la richiesta prima dell'invio al server.
TU-3.1	UC3.2.3	Verifica parser codice OAuth: gestione stringa vuota o null.	Il metodo solleva un'eccezione gestita senza effettuare chiamate esterne.
TU-4.1	UC4.1.1	Verifica validatore URL: input non appartenente al dominio github.com.	La funzione restituisce errore "Dominio non supportato".
TU-4.2	UC4.1.1	Verifica validatore URL: formato malformato o incompleto.	La funzione segnala l'URL come non valido.

ID Test	UC Riferimento	Descrizione	Risultato Atteso
TU-4.3	UC4.3.1	Verifica logica “Up-to-date”: confronto hash commit identici.	La funzione restituisce true.
TU-4.4	UC4.3.1	Verifica logica “Up-to-date”: confronto hash commit differenti.	La funzione restituisce false (necessaria nuova analisi).
TU-6.1	UC6.2.2	Verifica validatore intervallo date: Data Inizio successiva a Data Fine.	La funzione restituisce errore di incoerenza temporale.
TU-9.1	UC9.4	Verifica calcolo totale vulnerabilità: somma corretta dei parziali per categoria.	Il valore restituito corrisponde alla somma aritmetica delle issue.
TU-10.1	UC10.2	Verifica calcolo percentuale coverage documentazione.	Il calcolo restituisce il valore corretto arrotondato a due decimali.
TU-12.1	UC12.1	Verifica formattazione data report: conversione timestamp in formato locale.	La data viene renderizzata nel formato “YYYY/MM/DD HH:mm”.

Table 13: Tabella dei Test di Unità

## Test di Accettazione

In questa sezione vengono definiti i Test di Accettazione, volti a validare il sistema rispetto ai requisiti utente e agli scenari d'uso reali, inclusa la verifica della qualità dell'analisi AI.

ID Test	UC Riferimento	Descrizione	Risultato Atteso
TA-1.0	UC1	Utente completa la registrazione con dati validi.	L'account viene creato, l'email di conferma inviata e l'utente reindirizzato al login.
TA-1.1	UC1.1.1	Utente tenta la registrazione con caratteri non ammessi nel nome.	Il sistema mostra un errore visuale e disabilita il tasto di conferma.
TA-2.0	UC2	Utente effettua il login con credenziali corrette.	L'utente accede alla dashboard e il token di sessione viene generato.
TA-2.1	UC2.2.2	Utente tenta il login con password errata.	Il sistema nega l'accesso mostrando un messaggio di errore generico.
TA-3.0	UC3	Utente collega con successo l'account GitHub.	La dashboard mostra lo stato "Collegato" e l'avatar dell'utente GitHub.
TA-3.1	UC3.1.1	Utente rifiuta il collegamento durante il redirect.	L'operazione viene annullata e il sistema mostra un messaggio informativo.
TA-4.0	UC4	Utente richiede analisi di una repository pubblica valida.	L'analisi si avvia, mostrando lo stato di avanzamento/ spinner.
TA-4.1	UC4.1.2	Utente inserisce URL di una repository privata non accessibile.	Il sistema termina immediatamente segnalando l'errore di accesso.
TA-4.2	UC4.3.1	Utente richiede analisi su repo già analizzata senza modifiche.	Il sistema notifica che il report è già aggiornato senza avviare l'analisi.
TA-5.0	UC5	Utente visualizza il report finale.	Tutte le sezioni sono popolate e leggibili.
TA-9.0	UC9	Verifica qualità AI: Rilevamento vulnerabilità nota (es. SQL Injection).	Il report OWASP segnala correttamente la vulnerabilità presente nel codice di test.

ID Test	UC Riferimento	Descrizione	Risultato Atteso
TA-10.0	UC10	Verifica qualità AI: Rilevamento assenza documentazione.	L'agente segnala correttamente la bassa coverage su file privi di commenti.
TA-10.1	UC10.1	Verifica correzione spelling documentazione.	Gli errori di battitura presenti nei commenti vengono identificati e listati.
TA-6.0	UC6	Utente filtra i report per un intervallo temporale specifico.	La lista e i grafici mostrano solo i dati compresi nell'intervallo selezionato.
TA-7.0	UC7	Visualizzazione grafico comparativo tra analisi diverse.	Il grafico mostra l'evoluzione temporale delle metriche di qualità.
TA-12.0	UC12	Verifica correttezza metadati.	I dati visualizzati corrispondono esattamente all'ultimo commit su GitHub.

Table 14: Tabella dei Test di Accettazione

## Cruscotto di Valutazione

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.

### 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.

#### 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.

#### 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.

#### 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 progettata la stima del costo finale a finire (**Estimate At Completion**) confrontandola con il budget iniziale.

### Processi Primari: Sviluppo

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

#### 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**).

### Processi di Supporto: Documentazione

Monitoraggio della qualità formale e della fruibilità della documentazione prodotta.

#### 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.

### Processi di Supporto: Verifica

Monitoraggio dell'efficacia delle attività di testing dinamico.

## **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.

## **Processi di Supporto: Gestione della Qualità**

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

## **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.

## **Processi Organizzativi: Gestione dei Processi**

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

## **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.

## **Qualità di Prodotto**

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

## **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.

## **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.

## **Usabilità e Sicurezza**

*Metriche: MPD06, MPD07, MPD11*

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

## 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.

## 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
<b>Area: Comunicazione</b>		
AM01	<b>Inefficienza nella Comunicazione Interna</b> 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.	<b>Ristrutturazione dei Flussi Informativi</b> 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.
<b>Area: Ruoli e Pianificazione</b>		
AM02	<b>Pressione sulle Scadenze (Time-to-Result)</b> La necessità di produrre risultati tangibili (PoC) in tempi brevi per la revisione RTB rischiava di non essere soddisfatta con la pianificazione lineare iniziale.	<b>Ridistribuzione del Budget Orario</b> 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.
<b>Area: Strumenti e Tecnologie</b>		
AM03	<b>Disomogeneità nella Documentazione</b> La stesura parallela dei documenti da parte di più persone ha	<b>Adozione di Template e Funzioni Comuni</b> 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.
<b>AM04</b>	<b>Overhead Nuovi Strumenti</b> L'adozione contemporanea di nuovi strumenti (Jira, GitHub, Typst) ha comportato un rallentamento iniziale dovuto alla curva di apprendimento.	<b>Consolidamento della Toolchain</b> 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 15: Storico delle azioni di miglioramento (Periodo RTB)

## 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.