



Verbale esterno

11 dicembre 2025

Autore	Damiano Berti
Verificatore	Alessandro Frison
Approvazione	Giulia Romanato

Indice

1 Registro delle versioni	2
2 Informazioni introduttive	2
2.1 Durata e luogo	2
2.2 Partecipanti	2
3 Contenuto della riunione	2
3.1 Ordine del giorno	2
3.2 Riassunto della discussione	3
3.3 Gestione degli accessi e delle repo	3
3.4 Altri punti di discussione	3
3.4.1 Scansioni parziali e storico	3
3.4.2 Pull request	3
3.4.3 Gestione branch	3
3.4.4 Ruoli di un workspace	3
3.4.5 Qualità del codice	4
3.4.6 Visione aggregata	4
3.4.7 Valutazione documentazione	4
3.4.8 Tag delle repo	4
3.4.9 Login	4
4 Decisioni e azioni	5

1 Registro delle versioni

Versione	Data	Autore	Descrizione delle modifiche
1.0.0	15/12/2025	Giulia Romanato	Approvazione del verbale
0.1.0	14/12/2025	Alessandro Frison	Verifica del verbale
0.0.1	12/12/2025	Damiano Berti	Prima stesura del verbale

2 Informazioni introduttive

2.1 Durata e luogo

- **Inizio:** 17:00
- **Fine:** 17:55
- **Luogo:** Meeting Microsoft Teams

2.2 Partecipanti

Nome e Cognome	Presente	Assente
Damiano Berti	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Alessandro Frison	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Lorenzo Grolla	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Nicolò Lattanzio	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Alessandro Morabito	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Giacomo Nalotto	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Giulia Romanato	<input checked="" type="checkbox"/>	<input type="checkbox"/>

3 Contenuto della riunione

3.1 Ordine del giorno

1. Discussione con l'azienda proponente in merito alle funzionalità del programma Code Guardian

3.2 Riassunto della discussione

La riunione si è da subito focalizzata sull'esposizione all'azienda proponente delle funzionalità e del flusso di utilizzo del programma che il gruppo ha definito dopo una prima fase di analisi dei requisiti. I punti di discussione principale sono riportati nella sezione: [Altri punti di discussione](#), e tra questi si è distinto un [punto problematico](#) che ha reso necessario ripensare una parte del programma.

3.3 Gestione degli accessi e delle repo

L'utilizzo delle **organizzazioni** di GitHub per gestire gruppi di repo, ruoli e permessi è troppo limitante, inoltre i ruoli delle organizzazioni interni al programma potrebbero andare in conflitto con quelli delle organizzazioni di GitHub.

La soluzione trovata è quella di utilizzare dei **workspace** creati e gestiti all'interno del programma. Questo comporta diversi vantaggi tra cui:

- non essere più limitati dai ruoli presenti su GitHub;
- semplificazione della gestione dei ruoli e dei permessi;
- la possibilità per chi crea il workspace di diventare l'owner.

3.4 Altri punti di discussione

3.4.1 Scansioni parziali e storico

Discussione in merito alla gestione delle scansioni, con particolare attenzione alle scansioni parziali, ovvero scansioni dedicate solo a una o più sezioni (es: sicurezza, test...) e su come gestirne lo storico.

3.4.2 Pull request

È stata presentata all'azienda l'idea di includere nel programma una sezione dedicata alla visione e/o gestione delle pull request. A seguito di discussione abbiamo concluso che questa funzionalità potrebbe portare alla complicazione dell'alberatura del programma.

3.4.3 Gestione branch

Si è discusso di come il programma debba gestire i vari branch di una repo. La conclusione è che il programma debba permettere di selezionare il branch su cui eseguire la scansione per ogni repo.

3.4.4 Ruoli di un workspace

Si è discusso sulle differenze dei vari ruoli, e su come una personalizzazione complessa per ogni ruolo possa complicare il progetto.

3.4.5 Qualità del codice

Abbiamo esposto come si presenterà la sezione: "qualità del codice" relativa ad un report di una scansione a cui è seguita una discussione per aggiungere dati significativi andando a toglierne altri meno utili.

3.4.6 Visione aggregata

Dopo un'attenta analisi su come gestire la visione aggregata, abbiamo concluso che:

- non deve essere una visuale molto specifica;
- sia piuttosto facile da leggere;
- che comunichi al volo l'andamento della repo mediante grafici.

3.4.7 Valutazione documentazione

Si è discusso su come effettuare la valutazione della documentazione. Sono stati individuati 2 importanti punti di valutazione:

- valutazione tramite LLM della coerenza tra la documentazione e il codice sorgente;
- analisi di quante funzioni pubbliche hanno la documentazione nel codice.

3.4.8 Tag delle repo

Durante una precedente analisi abbiamo pensato di utilizzare un sistema di tag delle repo per raggrupparle secondo una specifica semantica. Dopo averne parlato con Var Group abbiamo individuato 2 tipologie di tag:

1. tag gestito dall'utente per categorizzare le raccolte arbitrariamente;
2. tag preso in automatico da GitHub per taggare le repo in base al linguaggio.

3.4.9 Login

Si è infine discusso in merito al modo ottimale di gestire il login nel programma. In particolare abbiamo vagliato la possibilità di eseguire il login tramite GitHub, senza l'appoggio di strumenti esterni o con un'approccio misto.

4 Decisioni e azioni

Codice	Descrizione	Assegnatario
DEC-RTB-016	Abbandonata l'idea di utilizzare l'importazione automatica e le organizzazioni di GitHub e sostituita con dei workspace	
DEC-RTB-017	Il programma non prevedrà scansioni parziali	
DEC-RTB-018	Rimandata la discussione su come trattare le pull request	
DEC-RTB-019	La visione aggregata non prevedrà informazioni troppo specifiche e sarà tenuta semplice	
DEC-RTB-020	Verranno usati questi 2 punti per la valutazione della documentazione	
DEC-RTB-021	Verranno utilizzati 2 tipi di tag per le repo	
DEC-RTB-022	Il programma dovrà permettere di selezionare il branch su cui eseguire la scansione per ogni repo	

Firma del proponente