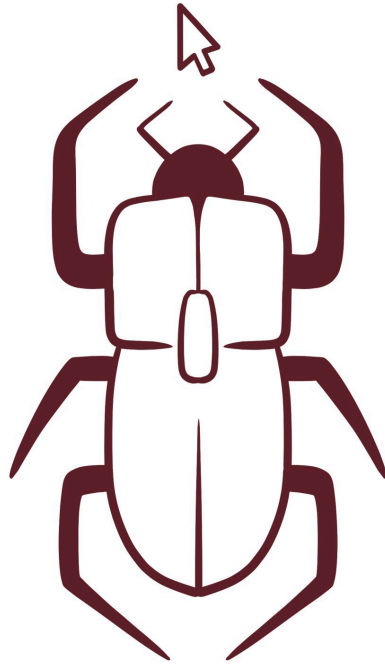




Norme di Progetto

v1.0.0

Registro delle Modifiche

Data	Versione	Descrizione	Redattore	Verificatore
2026/03/03	1.0.0	Revisione per RTB		Suar Alberto
2026/02/28	0.14.0	Verifica di tutto lo stato attuale del documento	Sgreva Andrea	Suar Alberto
2026/02/27	0.13.1	Verifica per la sezione Processo di Sviluppo e sezione Processi Organizzativi	Sgreva Andrea	Suar Alberto
2026/02/27	0.13.0	Verifica per la sezione Piano di Qualifica	Berengan Riccardo	Sgreva Andrea
2026/02/23	0.12.0	Verifica per la sezione Analisi dei Requisiti per la RTB		Suar Alberto, Basso Kevin
2026/02/21	0.11.0	Rielaborazione Processi di Supporto e Organizzativi	Zago Alice	Suar Alberto
2026/02/20	0.10.1	Fix sezione AdR in processo di fornitura	Sgreva Andrea	Suar Alberto
2026/02/14	0.10.0	Riorganizzazione sezione Processo di Sviluppo	Zago Alice	Suar Alberto
2026/02/13	0.9.0	Ampliamento sezioni Verifica e Validazione. Criteri di Verifica Processi di Supporto e Sviluppo	Zago Alice	Suar Alberto
2026/02/11	0.8.1	Ultimo fix/ miglioramento sezione AdR	Sgreva Andrea	Suar Alberto

Data	Versione	Descrizione	Redattore	Verificatore
2026/02/11	0.8.0	Sistemazione sezione AdR con nuove sezioni e miglioramenti	Sgreva Andrea	Basso Kevin
2026/02/07	0.7.0	Rielaborazione sezioni relative al Piano di Qualifica	Zago Alice	Basso Kevin
2026/02/04	0.6.0	Rielaborazione delle Norme di Progetto relative al Piano di Progetto	Suar Alberto	Basso Kevin
2026/02/03	0.5.0	Riscrittura delle sezioni relative all'AdR	Sgreva Andrea	Suar Alberto
2026/02/03	0.4.0	Integrazioni a PdQ e Metriche	Zago Alice	Suar Alberto
2026/01/15	0.3.0	Aggiunta prima versione sezione documenti	Basso Kevin	Suar Alberto
2026/01/09	0.2.1	Correzioni minori e miglioramenti alla documentazione	Basso Kevin	Martinello Riccardo
2026/01/02	0.2.0	Arricchimento con standard industriali, norme di codifica dettagliate e integrazioni progetto-specifiche	Martinello Riccardo	Suar Alberto, Basso Kevin
2025/12/28	0.1.0	Rilascio iniziale con norme, processi e strumenti aggiornati	Martinello Riccardo	Suar Alberto, Basso Kevin
2025/12/20	0.0.1	Prima bozza iniziale	Martinello Riccardo	Suar Alberto, Basso Kevin

Indice

1 Introduzione	8
1.1 Contesto del Progetto	8
1.2 Finalità del Documento	8
1.3 Glossario	8
1.4 Riferimenti	8
1.4.1 Riferimenti Normativi	8
1.4.2 Riferimenti Informativi	9
2 Processi Primari	10
2.1 Processo di Fornitura	10
2.1.1 Descrizione del Processo	10
2.1.2 Attività del Processo	10
2.1.3 Prodotti del Processo	11
2.1.4 Procedure Operative del Processo	11
2.1.4.1 Protocollo di Comunicazione (PR-FOR-01)	11
2.1.4.2 Procedura di Consegna (PR-FOR-02)	11
2.1.5 Piano di Progetto	12
2.1.5.1 Nota sull'Adozione dello Standard IEEE 1058	12
2.1.5.2 Definizione del Documento	12
2.1.5.3 Elementi Costituenti del Documento	12
2.1.5.3.1 PR-FOR-03: Procedura di Analisi e Gestione dei Rischi	12
2.1.5.3.2 PR-FOR-04: Procedura di Pianificazione del Ciclo di Vita	14
2.1.5.3.3 PR-FOR-05: Procedura di Preventivazione e Allocazione delle Risorse	14
2.1.5.3.4 PR-FOR-06: Procedura di Monitoraggio e Controllo del Rendimento	15
2.1.6 Analisi dei Requisiti	16
2.1.6.1 Standard di Riferimento	16
2.1.6.2 Definizione del Documento	16
2.1.6.3 Elementi Costituenti del Documento	16
2.1.6.3.1 PR-FOR-07: Procedura di Ingegneria dei Requisiti	16
2.1.7 Glossario	17
2.1.7.1 Definizione del Documento	17
2.1.7.2 Elementi Costituenti del Documento	17
2.1.7.2.1 PR-FOR-08: Procedura di Gestione Terminologica	17
2.1.8 NdP – Norme di Progetto	17
2.1.8.1 Definizione del Documento	17
2.1.8.2 Elementi Costituenti del Documento	17
2.1.8.2.1 Processo di Fornitura	18
2.1.8.2.2 Processo di Sviluppo	18
2.1.8.2.3 Processi di Supporto	18
2.1.8.2.4 Processi Organizzativi	18
2.1.9 Piano di Qualifica	19
2.1.9.1 Nota sull'adozione dello Standard ISO/IEC 25010 e PDCA	19
2.1.9.2 Definizione del Documento	19
2.1.9.3 Elementi Costituenti del Documento	19
2.1.9.3.1 Qualità di Processo	19
2.1.9.3.2 PR-FOR-09: Procedura di Definizione e Gestione delle Metriche	19

2.1.9.3.3	Qualità del Prodotto	22
2.1.9.3.4	PR-FOR-10: Procedura di Controllo Qualità del Prodotto	22
2.1.9.3.5	Strategie di Testing	24
2.1.9.3.6	Cruscotto di Valutazione	24
2.1.9.3.7	PR-FOR-11: Procedura di Gestione del Cruscotto	24
2.1.9.3.8	Automiglioramento	24
2.1.10	Verbali delle Riunioni	24
2.1.10.1	Definizione del Documento	24
2.1.10.2	Elementi Costituenti del Documento	25
2.1.10.2.1	PR-FOR-12: Procedura di Verbalizzazione	25
2.1.11	Diario di Bordo	25
2.1.11.1	Definizione del Documento	25
2.1.11.2	Elementi Costituenti del Documento	25
2.1.11.2.1	PR-FOR-13: Procedura di Aggiornamento del Diario	25
2.2	Processo di Sviluppo	26
2.2.1	Attività del Processo di Sviluppo	26
2.2.1.1	Implementazione del processo	26
2.2.1.2	Analisi e Progettazione	26
2.2.1.2.1	PR-SVIL-01: Procedura di Specifica Tecnica	26
2.2.1.3	Implementazione e Integrazione	26
2.2.1.3.1	PR-SVIL-02: Procedura di Sviluppo e Build	26
2.2.1.4	Testing, Installazione e Qualifica	27
2.2.1.4.1	PR-SVIL-03: Procedura di Validazione	27
2.2.2	Relazione con le Baseline di progetto	27
2.2.3	Codifica e Standard Tecnici	27
2.2.3.1	PR-SVIL-04: Standard di Codifica (Coding Standards)	27
2.2.3.2	PR-SVIL-05: Procedura di Gestione della Configurazione	28
2.2.3.3	PR-SVIL-06: Procedura di Verifica (Testing)	28
2.2.4	Evoluzione delle Norme verso l'MVP	28
3	Processi di Supporto	29
3.1	Documentazione	29
3.1.1	PR-SUPP-01: Procedura di Gestione Documentale (Ciclo di Vita)	29
3.1.1.1	Strumenti di Redazione	29
3.1.1.2	Struttura dei Documenti	29
3.1.2	PR-SUPP-02: Gestione della Configurazione e Versionamento	30
3.1.2.1	Sistema di Versioning	30
3.1.2.2	Denominazione dei File e Tracciabilità	30
3.1.2.3	Repository GitHub - Struttura e Branching	30
3.2	Qualifica	30
3.2.1	PR-SUPP-03: Procedura di Verifica	31
3.2.1.1	Strumenti a supporto	31
3.2.1.2	Attività previste (ISO/IEC/IEEE 12207)	31
3.2.1.3	Implementazione: Regola del Main e Pull Request	31
3.2.1.4	Verifica della Documentazione	31
3.2.1.5	Verifica del Codice: Analisi Statica e Dinamica	31
3.2.1.5.1	Analisi Statica	31
3.2.1.5.2	Analisi Dinamica e Testing	32
3.2.2	PR-SUPP-04: Procedura di Validazione	32

3.2.2.1	Obiettivi e Strategie di Validazione	33
4	Processi Organizzativi	34
4.1	Gestione dei Processi	34
4.1.1	Ruoli di Progetto e Rotazione	34
4.1.2	PR-ORG-01: Procedura di Gestione Task e Pianificazione	34
4.1.3	PR-ORG-02: Procedura di Coordinamento e Comunicazione	35
4.2	Infrastruttura	35
4.2.1	PR-ORG-03: Procedura di Gestione della Configurazione (GitHub)	35
4.2.1.1	Organizzazione del Repository DocumentazioneProgetto	35
4.3	Processo di Miglioramento	36
4.3.1	PR-ORG-04: Procedura di Miglioramento (Retrospettiva)	36
4.4	Processo di Formazione	36
4.4.1	PR-ORG-05: Procedura di Formazione Continua	36

Indice tabelle

Table 1	Tabella riepilogativa dei prodotti del processo di fornitura	11
Table 2	Matrice di Rischio: visualizzazione delle soglie di intervento obbligatorio	13
Table 3	Metriche per il processo di Fornitura	20
Table 4	Metriche per il processo di Sviluppo	20
Table 5	Metriche per il processo di Documentazione e Verifica	21
Table 6	Metriche per i processi Organizzativi	22
Table 7	Metriche di Adeguatezza Funzionale e Affidabilità	23
Table 8	Metriche di Manutenibilità e Sicurezza	23
Table 9	Standard di nomenclatura tecnica	27
Table 10	Livelli di test e strumenti (Vitest, Pytest, SonarQube)	28
Table 11	Strumenti di redazione documentale	29
Table 12	Struttura gerarchica del repository di documentazione	30
Table 13	Nomenclatura e Classificazione delle Analisi Dinamiche (Test)	32
Table 14	Matrice di responsabilità per i ruoli di progetto	34
Table 15	Piano di comunicazione del team	35
Table 16	Strumenti dell'infrastruttura di progetto	35

1 Introduzione

1.1 Contesto del Progetto

Il presente documento descrive le Norme di Progetto ^G applicate al progetto Code Guardian ^G, proposto dall'azienda Var Group ^G e realizzato dal team di sviluppo Skarab Group ^G nell'ambito del corso di Ingegneria del Software presso l'Università degli Studi di Padova.

Il progetto ha come obiettivo lo sviluppo di una piattaforma software ad agenti ^G finalizzata all'audit ^G e alla remediation ^G automatizzata delle vulnerabilità presenti nei repository di codice sorgente. Il sistema è progettato in conformità ai requisiti e ai vincoli definiti nel Capitolato C2.

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

1.2 Finalità del Documento

Le Norme di Progetto stabiliscono il quadro di riferimento metodologico, organizzativo e tecnologico adottato per lo sviluppo del sistema software.

Il presente documento costituisce lo standard normativo interno del team; esso è vincolante per l'intero gruppo di lavoro e persegue i seguenti obiettivi:

- definire un insieme di regole comuni e convenzioni condivise al fine di garantire uniformità e coerenza nello sviluppo del prodotto software;
- individuare e descrivere gli strumenti, le tecnologie e gli ambienti di lavoro adottati;
- formalizzare i processi operativi, organizzativi e di controllo applicati durante l'intero ciclo di vita ^G del progetto;
- assicurare l'allineamento delle attività progettuali agli standard di riferimento e alle buone pratiche dell'ingegneria del software.

Il rispetto delle norme definite nel presente documento è obbligatorio per tutti i membri del gruppo di lavoro e costituisce un requisito fondamentale per garantire qualità, tracciabilità e controllo dell'intero ciclo di sviluppo.

1.3 Glossario

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

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

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

1.4 Riferimenti

1.4.1 Riferimenti Normativi

I seguenti documenti hanno valore vincolante per la redazione delle Norme di Progetto e per la definizione delle regole da rispettare durante il ciclo di vita del prodotto software:

- **Regolamento del Progetto Didattico:** Regole e vincoli del corso di Ingegneria del Software.
<https://www.math.unipd.it/~tullio/IS-1/2025/Dispense/PD1.pdf>
(ultimo accesso: 25/02/2026)

- **Capitolato C2:** Piattaforma ad agenti per l'audit e la remediation dei repository software.
<https://www.math.unipd.it/~tullio/IS-1/2025/Progetto/C2.pdf>
(ultimo accesso: 25/02/2026)
- **ISO/IEC/IEEE 12207 – Software Life Cycle Processes:** framework internazionale per la gestione dei processi di sviluppo, manutenzione e dismissione del software.
<https://ieeexplore.ieee.org/document/8100771>
(ultimo accesso: 25/02/2026)

1.4.2 Riferimenti Informativi

- **Dispense del corso di Ingegneria del Software relative ai Processi di Ciclo di Vita del Software**
<https://www.math.unipd.it/~tullio/IS-1/2025/Dispense/T02.pdf>
(ultimo accesso: 25/02/2026)
- **Dispense del corso di Ingegneria del Software relative alla Gestione di Progetto**
<https://www.math.unipd.it/~tullio/IS-1/2025/Dispense/T04.pdf>
(ultimo accesso: 25/02/2026)
- **IEEE 829 – Standard for Software and System Test Documentation:** definizione della struttura dei piani di test, casi di test, procedure e report.
<https://ieeexplore.ieee.org/document/4578383>
(ultimo accesso: 25/02/2026)
- **IEEE 830 – Standard for Software Requirements Specifications:** guida alla redazione e classificazione dei requisiti funzionali e non funzionali.
<https://ieeexplore.ieee.org/document/720574>
(ultimo accesso: 25/02/2026)
- **IEEE 1016 – Recommended Practice for Software Design Descriptions:** linee guida per la descrizione dell'architettura software e dei principali design pattern.
<https://ieeexplore.ieee.org/document/741934>
(ultimo accesso: 25/02/2026)

2 Processi Primari

In conformità allo standard ISO/IEC/IEEE 12207, il ciclo di vita di un sistema software è descritto attraverso un insieme strutturato di processi, ciascuno dei quali copre specifici ambiti di responsabilità e attività. Tra questi, i processi primari rappresentano il nucleo operativo del ciclo di vita e descrivono le interazioni fondamentali tra chi richiede il sistema, chi lo realizza e chi ne cura lo sviluppo tecnico.

I processi primari definiscono cosa viene fatto, da chi e con quale responsabilità contrattuale e operativa, indipendentemente dalle metodologie o dagli strumenti adottati. Essi costituiscono quindi il riferimento concettuale per l'organizzazione delle attività progettuali e per la distribuzione dei ruoli all'interno del progetto.

I processi primari, come indicato nello standard sopracitato, sono:

- Acquisizione (Acquisition);
- Fornitura (Supply);
- Sviluppo (Development);
- Operazione (Operation);
- Manutenzione (Maintenance).

Nel contesto del presente progetto, sono considerati rilevanti principalmente i processi di Fornitura e Sviluppo, in quanto direttamente pertinenti alla natura didattica e contrattuale dell'iniziativa.

2.1 Processo di Fornitura

2.1.1 Descrizione del Processo

Il Processo di Fornitura definisce l'insieme delle attività svolte dal team di sviluppo in qualità di fornitore al fine di pianificare, coordinare e consegnare il prodotto software e la relativa documentazione in conformità ai requisiti stabiliti dal proponente.

Nel contesto del progetto Code Guardian, il gruppo Skarab Group assume il ruolo di fornitore, mentre l'azienda Var Group ricopre il ruolo di proponente. Il Capitolato C2 costituisce il riferimento contrattuale che disciplina requisiti, vincoli e obiettivi della fornitura.

Il processo di fornitura è responsabile della gestione complessiva del progetto dal punto di vista organizzativo e qualitativo e coordina le attività necessarie a garantire la conformità degli artefatti prodotti rispetto a quanto concordato.

2.1.2 Attività del Processo

Nel presente progetto, il processo di fornitura si articola nelle seguenti attività principali, regolate da specifiche procedure operative:

- **Pianificazione della Fornitura:** Il Responsabile ^G **ha l'obbligo di** definire la struttura delle attività e assegnare le responsabilità operative all'inizio di ogni Sprint ^G, garantendo la saturazione delle risorse;
- **Gestione della Documentazione:** Ogni membro del team **è tenuto a** redigere e verificare gli artefatti seguendo le convenzioni stilistiche e i criteri di qualità normati, assicurando la tracciabilità delle modifiche;
- **Monitoraggio e Controllo:** Il Responsabile **deve** verificare periodicamente l'avanzamento rispetto alla baseline ^G e attivare tempestivamente azioni correttive qualora si rilevino scostamenti dalle soglie di qualità definite nel PdQ ^G
- **Rilascio e Consegna:** Il team **deve** assicurare il consolidamento degli artefatti e la loro consegna formale nel rispetto delle scadenze contrattuali, seguendo il protocollo di rilascio ufficiale.

2.1.3 Prodotti del Processo

I prodotti documentali della fornitura non costituiscono il fine ultimo del processo, ma rappresentano gli output formali risultanti dall'applicazione sistematica delle **procedure operative** di gestione e controllo. Essi sono:

- **Norme di Progetto:** definiscono le regole operative e tecniche adottate dal team;
- **Piano di Progetto:** definisce pianificazione, risorse e gestione dei rischi;
- **Piano di Qualifica:** specifica criteri di qualità e attività di verifica;
- **Analisi dei Requisiti:** formalizza le necessità del sistema derivate dal Capitolato;
- **Verbali:** registrano decisioni e attività di revisione.

Documento	Redattori	Destinatari	Uso
Lettera di Presentazione	Responsabile	Proponente, Cardin Riccardo, Vardanega Tullio, Skarab Group	Esterno
AdR – Analisi dei Requisiti	Analisti	Proponente, Cardin Riccardo, Vardanega Tullio, Skarab Group	Esterno
Glossario	Tutti i membri del gruppo	Stakeholder interni ed esterni	Esterno
NdP – Norme di Progetto	Responsabili di processo	Skarab Group, Cardin Riccardo, Vardanega Tullio	Interno
PdP – Piano di Progetto	Project Manager	Proponente, Cardin Riccardo, Vardanega Tullio, Skarab Group	Esterno
PdQ – Piano di Qualifica	Responsabili Qualità	Skarab Group, Cardin Riccardo, Vardanega Tullio, Proponente	Esterno
Verbali delle Riunioni	Responsabile	Proponente (se esterni), Cardin Riccardo, Vardanega Tullio, Skarab Group	Interno / Esterno

Table 1: Tabella riepilogativa dei prodotti del processo di fornitura

2.1.4 Procedure Operative del Processo

Le seguenti procedure sono vincolanti per l'esecuzione del processo di fornitura e regolano l'interazione con l'esterno.

2.1.4.1 Protocollo di Comunicazione (PR-FOR-01)

Per garantire professionalità e tracciabilità nei confronti del Proponente Var Group:

- **Canale Ufficiale:** Le comunicazioni formali **devono** avvenire esclusivamente tramite email istituzionale o incontri verbalizzati;
- **Verbalizzazione:** È **obbligatorio** produrre un verbale esterno entro 48 ore lavorative da ogni incontro con il proponente;
- **Approvazione Verbali:** Il silenzio-assenso del proponente dopo 24 ore dalla pubblicazione **costituisce l'approvazione formale** del contenuto.

2.1.4.2 Procedura di Consegna (PR-FOR-02)

In occasione delle scadenze ufficiali, il Responsabile di Progetto **deve obbligatoriamente** eseguire la seguente checklist operativa tramite gli strumenti di supporto:

1. **Verifica di Conformità:** Accertarsi che tutti i documenti siano nello stato **Approved** e che ogni artefatto soddisfi i criteri definiti nel PdQ;
2. **Consolidamento:** Generare i PDF definitivi dai sorgenti Typst ^G congelati;
3. **Tagging:** Creare una **Release** sul repository GitHub ^G con tag semantico (es. v1.0.0-RTB);

4. **Notifica:** Inviare comunicazione formale di avvenuta consegna al proponente indicando il link alla release.

2.1.5 Piano di Progetto

2.1.5.1 Nota sull'Adozione dello Standard IEEE 1058

Il documento **Piano di Progetto** adotta come riferimento metodologico lo standard **IEEE 1058-1998**. Tale scelta **vincola il team** ad applicare un approccio rigoroso e sistematico alla gestione, adattando i contenuti alle dimensioni e alla complessità del progetto.

L'adozione dello standard garantisce una copertura completa degli aspetti critici del ciclo di vita e si traduce nei seguenti obblighi operativi:

- **Sistematicità:** l'articolazione del documento **deve** ricalcare la gerarchia IEEE 1058 per evitare lacune nella definizione delle responsabilità;
- **Gestione del Rischio:** è **obbligatorio** adottare un approccio basato su probabilità e impatto per il monitoraggio delle incertezze;
- **Controllo Dinamico:** il team **deve** aggiornare il Piano a ogni Sprint, distinguendo esplicitamente tra **Preventivo** e **Consuntivo** per identificare tempestivamente gli scostamenti.

2.1.5.2 Definizione del Documento

Il **Piano di Progetto** ^G costituisce lo **standard operativo vincolante** per la conduzione del progetto. Esso **deve** definire, in modo tracciabile e verificabile:

- l'organizzazione del lavoro e le responsabilità operative;
- la pianificazione temporale (milestone e scadenze);
- l'allocazione delle risorse e le stime di impegno;
- la gestione dei rischi e le modalità di monitoraggio.

2.1.5.3 Elementi Costituenti del Documento

Il Piano di Progetto **deve obbligatoriamente integrare** i seguenti elementi, ciascuno governato dalle procedure operative e dai criteri di verifica descritti nelle sezioni successive:

- Analisi dei Rischi;
- Pianificazione del Ciclo di Vita del Software;
- Preventivo ed Allocazione delle Risorse;
- Monitoraggio e Controllo del Rendimento degli Sprint.

2.1.5.3.1 PR-FOR-03: Procedura di Analisi e Gestione dei Rischi

L'attività di Analisi dei Rischi **deve** individuare e valutare sistematicamente i fattori che possono compromettere gli obiettivi di progetto. Il team **ha l'obbligo** di applicare tale procedura con approccio ciclico durante l'intero ciclo di vita.

2.1.5.3.1.1 Metodologia di Analisi

Il team **adotta obbligatoriamente** un approccio **semi-quantitativo**. Ogni rischio identificato **deve** essere valutato tramite i seguenti parametri:

- **Probabilità di accadimento (P):** verosimiglianza dell'evento;
- **Impatto (I):** gravità delle conseguenze.

Il **Livello di Rischio (R)** **deve** essere calcolato come:

$$R = P \times I$$

I parametri **P** e **I** **devono** essere valutati su una scala discreta da 1 a 3 per ridurre la soggettività e garantire la reattività dell'approccio iterativo.

Probabilità	Impatto		
	Basso (1)	Medio (2)	Alto (3)
Bassa (1)	1 (Basso)	2 (Basso)	3 (Basso)
Media (2)	2 (Basso)	4 (Medio)	6 (Alto)
Alta (3)	3 (Basso)	6 (Alto)	9 (Critico)

Table 2: Matrice di Rischio: visualizzazione delle soglie di intervento obbligatorio

In base al valore di R , il team è **vincolato** all'applicazione delle seguenti strategie:

- **Rischio Basso (1– 3):** monitoraggio periodico;
- **Rischio Medio (4):** obbligo di definizione preventiva di una strategia di mitigazione;
- **Rischio Alto (6– 9):** obbligo di azioni preventive e predisposizione di piani di contingenza.

2.1.5.3.1.2 Categorie di Rischio

I rischi **devono** essere classificati nelle seguenti macro-categorie per l'assegnazione delle responsabilità:

- **RT (Rischi Tecnologici):** LLM, Agenti, integrazioni;
- **RI (Rischi Interpersonali):** disponibilità del team, comunicazione;
- **RCO (Rischi di Costo e Organizzativi):** stime, schedulazione, vincoli esterni.

2.1.5.3.1.3 Ruoli e Responsabilità

- **Responsabile di Progetto:** deve supervisionare il processo, valutare le criticità e autorizzare l'attivazione dei piani di contingenza;
- **Team di Sviluppo:** ha il compito di identificare e segnalare tempestivamente ogni anomalia;
- **Stakeholder:** devono essere informati in caso di rischi con impatto critico sulla baseline.

2.1.5.3.1.4 Procedura Operativa di Monitoraggio

Qualora un rischio si manifesti, il team **deve** seguire rigorosamente questa sequenza:

1. **Rilevamento (Detection):** Ogni membro **ha l'obbligo** di segnalare l'insorgere di problematiche al Responsabile **entro il termine della giornata lavorativa**;
2. **Documentazione:** L'evento **deve** essere tracciato immediatamente nel sistema di issue tracking;
3. **Valutazione e Risposta:** Il Responsabile **deve** attivare la strategia di mitigazione o il piano di contingenza;
4. **Rendicontazione:** Al termine dello Sprint, l'evento **deve** essere analizzato nel consuntivo per validare l'efficacia delle contromisure.

2.1.5.3.1.5 Criteri di Verifica dell'Analisi dei Rischi

L'elemento **Analisi dei Rischi** è conforme se e solo se:

- ogni rischio è descritto e classificato univocamente (RT, RI, RCO);
- sono indicati esplicitamente i valori di P , I e R ;
- per ogni rischio è definita una strategia di mitigazione e contingenza;
- lo stato dei rischi è aggiornato con cadenza pari alle iterazioni di Sprint;
- l'insorgenza di nuovi rischi è tracciata nei verbali o negli strumenti di gestione.

2.1.5.3.2 PR-FOR-04: Procedura di Pianificazione del Ciclo di Vita

Questa procedura norma l'organizzazione temporale del progetto e vincola la scomposizione delle attività necessarie alla realizzazione del prodotto. Il team **deve** utilizzare la pianificazione come riferimento formale per il coordinamento e il monitoraggio costante dell'avanzamento.

2.1.5.3.2.1 Gestione Operativa della Pianificazione

La pianificazione **deve** adattarsi alla granularità richiesta dalla fase corrente del ciclo di vita:

- **Pianificazione in RTB:** Il Responsabile **ha l'obbligo** di definire nel dettaglio ogni attività. Per ciascuna di esse **devono** essere specificati: descrizione univoca, responsabile assegnato e scadenza.
- **Pianificazione in PB:** È **obbligatorio** adottare un approccio **rolling-wave**. La pianificazione si limita alla definizione delle macro-attività (es. Progettazione Architettuale, MVP) per preservare la flessibilità necessaria alle scelte implementative non ancora consolidate.

2.1.5.3.2.2 Requisiti di Atomicità e Tracciabilità

Per garantire l'efficacia del monitoraggio tramite lo strumento ufficiale Jira ^G, il team **deve** attenersi alle seguenti regole:

1. **Corrispondenza 1:1:** Ogni attività definita **deve** corrispondere a una **task** registrata su Jira.
2. **Atomicità delle Task:** Le attività pianificate in RTB **devono** essere atomiche. Una task è definita tale se:
 - è eseguibile da un singolo membro del team;
 - non presenta dipendenze bloccanti non tracciate; qualora esistano, **devono** essere esplicitate tramite relazioni "blocks/blocked-by".
3. **Assegnazione:** Il Responsabile, all'avvio di ogni Sprint, **deve** assegnare ogni task a un componente del team, definendo lo stato di avanzamento atteso.

2.1.5.3.2.3 Criteri di Verifica della Pianificazione

L'attività di pianificazione è considerata conforme se e solo se risultano soddisfatte le seguenti condizioni:

1. **Per la fase RTB:**
 - Tutte le attività sono tracciate su Jira come task atomiche;
 - Ogni task possiede un responsabile identificabile;
 - Le dipendenze sono esplicitate graficamente o tramite collegamenti tecnici.
2. **Per la fase PB:**
 - Sono identificate le macro-attività senza forzare decisioni progettuali premature;
 - Le milestone sono coerenti con la cadenza degli Sprint.
3. **Generali:**
 - Ogni variazione significativa alla pianificazione è motivata e tracciata nei verbali di progetto.

Il mancato rispetto di una sola condizione comporta lo stato di "Non Conformità", richiedendo l'immediata correzione del Piano di Progetto.

2.1.5.3.3 PR-FOR-05: Procedura di Preventivazione e Allocazione delle Risorse

Questa procedura norma la distribuzione dell'impegno temporale e la gestione economica. Il preventivo approvato costituisce la baseline formale per il confronto con i consuntivi e supporta le attività di controllo.

2.1.5.3.3.1 Regole di Definizione del Preventivo e Costi

La determinazione del preventivo **deve** rispettare i seguenti vincoli normativi:

- **Esclusione Auto-apprendimento:** Il periodo antecedente al primo Sprint è classificato come auto-apprendimento; è **fatto divieto** di rendicontare tali ore nel calcolo dei costi di progetto.

- **Gestione Variazioni:** Ogni scostamento rispetto alle stime di candidatura **deve** essere documentato e motivato nei verbali di progetto o negli strumenti di issue tracking.

Nota di Contesto (Analisi Preliminare): A seguito del consolidamento delle baseline, la distribuzione pro-capite rimane invariata, pur registrando una riduzione del costo complessivo rispetto alle stime iniziali.

2.1.5.3.3.2 Strategia di Allocazione per Milestone

L'allocazione delle risorse **deve** adattarsi alla natura specifica della milestone di progetto:

1. **Milestone RTB (Requirements and Technology Baseline):** L'allocazione **deve** prevedere un impiego prevalente delle figure di **Amministratore** ^G e **Analista** ^G. Tale scelta è **obbligatoria** per gestire l'alto carico documentale e normativo di questa fase.
2. **Milestone PB (Product Baseline):** Il preventivo per questa fase **deve** essere gestito con metodologia **rolling-wave**. Esso **ha l'obbligo** di essere riesaminato alla chiusura di ogni Sprint; ogni revisione **deve** essere approvata dal Responsabile di Progetto.

2.1.5.3.3.3 Criteri di Verifica del Preventivo e dell'Allocazione

L'attività è considerata conforme se e solo se risultano soddisfatte le seguenti condizioni:

- è definito un preventivo iniziale coerente con le baseline individuate;
- la distribuzione dell'impegno orario pro-capite è esplicitata e motivata;
- la ripartizione dei ruoli per fase (RTB/PB) è coerente con le attività pianificate;
- eventuali scostamenti dal preventivo iniziale sono motivati e tracciati;
- il preventivo è utilizzato come riferimento esplicito nel confronto con i consuntivi.

2.1.5.3.4 PR-FOR-06: Procedura di Monitoraggio e Controllo del Rendimento

Questa procedura norma la verifica sistematica dell'andamento del progetto rispetto alla pianificazione. L'attività **deve** basarsi su iterazioni temporali (**Sprint**) composte da task atomiche e tracciabili.

2.1.5.3.4.1 Metodologia di Monitoraggio dello Sprint

Al termine di ogni Sprint, il rendimento **deve** essere descritto in modo uniforme includendo:

- **Attività svolte:** Elenco delle task concluse secondo la **Definition of Done**;
- **Analisi delle deviazioni:** Confronto tra pianificato ed effettivo con analisi delle cause;
- **Rischi rilevati:** Aggiornamento dello stato dei rischi emersi (rif. **PR-FOR-03**);
- **Azioni correttive:** Decisioni adottate per migliorare l'efficacia delle iterazioni successive.

2.1.5.3.4.2 Regole di Rendicontazione e Strumenti

La rendicontazione **deve** avvenire esclusivamente tramite lo strumento Jira ^G:

1. **Frequenza:** L'aggiornamento dei dati **deve** avvenire con periodicità giornaliera, **entro il termine della giornata lavorativa**.
2. **Precisione:** Per ogni task **devono** essere indicati obbligatoriamente lo stato di avanzamento, le ore spese e il responsabile.
3. **Attività Non Pianificate:** È **obbligatorio** tracciare anche le attività impreviste (bugfix, criticità) creando nuove task; l'esclusione di tali ore è **vietata**.

2.1.5.3.4.3 Criteri di Verifica del Monitoraggio e Controllo

L'attività è considerata conforme se e solo se, per ogni Sprint:

- è definito un Preventivo di Sprint basato sulle attività assegnate;
- è redatto un Consuntivo di Sprint basato esclusivamente sui dati di Jira;
- il confronto tra preventivo e consuntivo è esplicitato e giustificato;

- tutte le ore lavorate (incluse le extra-piano) sono state tracciate correttamente;
- sono riportate le azioni correttive o preventive validate dal Responsabile.

2.1.6 Analisi dei Requisiti

2.1.6.1 Standard di Riferimento

Per la redazione del documento Analisi dei Requisiti, il gruppo **adotta obbligatoriamente** i seguenti standard internazionali:

- **IEEE 830-1998**: standard di riferimento per la stesura di specifiche dei requisiti software;
- **ISO/IEC/IEEE 29148:2018**: standard per i processi di Requirements Engineering;
- **UML 2.5**: standard per la modellazione grafica dei casi d'uso e delle interazioni.

2.1.6.2 Definizione del Documento

L'Analisi dei Requisiti ^G costituisce il **riferimento tecnico vincolante** per la formalizzazione delle necessità e dei vincoli del sistema **Code Guardian**. Il documento **ha l'obbligo** di tradurre le esigenze del proponente in specifiche formali, non ambigue e verificabili.

2.1.6.3 Elementi Costituenti del Documento

Il documento **deve essere organizzato** includendo i seguenti elementi, normati dalle procedure operative interne:

- **Casi d'Uso**: base per l'identificazione dei requisiti funzionali;
- **Requisiti Funzionali, di Qualità e di Vincolo**: specifiche atomiche e misurabili.

2.1.6.3.1 PR-FOR-07: Procedura di Ingegneria dei Requisiti

Questa procedura norma la formalizzazione delle necessità del sistema. Il team **deve** attenersi ai seguenti requisiti operativi:

- **Modellazione dei Casi d'Uso**: Ogni scenario **deve** essere identificato dalla nomenclatura UCx.y.z dove **x** è l'identificatore principale, **y** il numero di inclusioni e **z** il numero di estensioni. Qualora esistano ulteriori ramificazioni, la stringa UCx.y.z fungerà da radice gerarchica.
- **Contenuto del Caso d'Uso**: Ogni scenario **deve includere** Attori, Pre/Post-condizioni, Scenario Principale, Trigger e, se presenti, Inclusioni, Estensioni o Generalizzazioni.
- **Diagrammi UML**: È **obbligatorio** corredare la descrizione testuale con diagrammi conformi allo standard UML 2.5.
- **Classificazione dei Requisiti**: Ogni requisito **deve** essere catalogato secondo la sintassi TipologiaRPrioritàNumero dove:
 - **Tipologia**: F (Funzionale), Q (Qualità), V (Vincolo);
 - **Priorità**: Ob (Obbligatorio), De (Desiderabile), Op (Opzionale).

2.1.6.3.1.1 Criteri di Verifica dei Casi d'Uso

L'elemento è considerato conforme se e solo se risultano soddisfatte le seguenti condizioni:

- **Univocità**: ogni identificatore UCx.y.z è unico;
- **Correttezza degli Attori**: ogni attore citato è stato precedentemente censito;
- **Coerenza UML**: esiste corrispondenza 1:1 tra descrizione testuale e diagramma;
- **Assenza di Implementazione**: la descrizione **deve** limitarsi al "cosa fa", non al "come".

2.1.6.3.1.2 Criteri di Verifica dei Requisiti di Sistema

I requisiti sono conformi se e solo se soddisfano i seguenti vincoli:

- **Atomicità**: ogni requisito esprime una singola necessità non scomponibile;
- **Non Ambiguità**: il linguaggio utilizzato è tecnico, oggettivo e privo di termini soggettivi;

- **Testabilità:** permette la definizione di un test di accettazione binario;
- **Misurabilità (Qualità):** il requisito **deve** essere espresso attraverso metriche quantitative o soglie numeriche;
- **Tracciabilità:** ogni requisito **deve** indicare chiaramente la propria fonte (Capitolato o UC) e la sua giustificazione (per i vincoli).

2.1.7 Glossario

2.1.7.1 Definizione del Documento

Il **Glossario** costituisce il riferimento terminologico unico e **vincolante** per l'intero ciclo di vita del progetto. Esso **ha l'obbligo** di eliminare ambiguità semantiche e garantire uniformità comunicativa.

2.1.7.2 Elementi Costituenti del Documento

Il Glossario **deve essere organizzato** in ordine alfabetico e includere: Termini Tecnici, Acronimi e Concetti di Dominio.

2.1.7.2.1 PR-FOR-08: Procedura di Gestione Terminologica

Questa procedura norma la manutenzione del vocabolario controllato del team:

- **Redazione della Voce:** La definizione **deve** essere sintetica, non tautologica e contestualizzata. È **vietato** l'uso di sinonimi duplicati senza rimandi.
- **Marcatura nei Documenti:** In ogni documento di progetto, la prima occorrenza di un termine di glossario **deve obbligatoriamente** essere marcata con l'apice ^G e linkata alla versione ufficiale.
- **Accessibilità:** Il Glossario **deve** essere disponibile sia in formato PDF statico che come risorsa ipertestuale dinamica.

2.1.7.2.1.1 Criteri di Verifica del Glossario

Il Glossario è considerato conforme se e solo se soddisfa le seguenti condizioni:

- **Completezza:** ogni termine marcato nella documentazione possiede una definizione corrispondente;
- **Integrità Web:** ogni collegamento (anchor link) punta a una sezione esistente della pagina web;
- **Ordine:** le voci sono presentate in rigoroso ordine alfabetico (A-Z);
- **Qualità:** le definizioni non sono tautologiche e i rimandi sono circolari solo verso termini definiti.

2.1.8 NdP – Norme di Progetto

2.1.8.1 Definizione del Documento

Le **Norme di Progetto** ^G (NdP) costituiscono lo standard normativo di riferimento che disciplina le modalità operative dell'intero ciclo di vita del progetto. Esse definiscono in modo **vincolante**:

- i **principi organizzativi** e il **Way of Working**;
- le **norme** per la redazione della documentazione e lo sviluppo del software;
- gli **strumenti** e gli **ambienti di lavoro**;
- i **processi** e le **procedure operative** di pianificazione, esecuzione e controllo.

Il rispetto delle NdP è **obbligatorio** per tutti i membri del gruppo e costituisce requisito necessario per la qualità e la tracciabilità degli artefatti.

2.1.8.2 Elementi Costituenti del Documento

Le Norme di Progetto **devono** essere strutturate secondo la descrizione normativa dei macro-processi primari (**ISO/IEC/IEEE 12207**), disciplinando obbligatoriamente il **Processo di Fornitura** e il **Processo di Sviluppo**. Per ciascun processo, le NdP **devono** esplicitare obiettivi, ruoli, documenti prodotti, procedure operative e criteri di verifica.

2.1.8.2.1 Processo di Fornitura

Il Processo di Fornitura disciplina la produzione e la consegna degli artefatti formali (PdP, PdQ, AdR, NdP, Glossario). Ogni documento di fornitura **deve** rispettare la seguente struttura minima obbligatoria:

- **Definizione del Documento;**
- **Elementi Costituenti del Documento;**
- **Descrizione degli Elementi** (sezioni e contenuti);
- **Criteri di Verifica** specifici.

2.1.8.2.1.1 Criteri di Verifica dei Documenti di Fornitura

Un documento è considerato conforme se e solo se soddisfa le seguenti condizioni:

- È presente tra gli artefatti previsti e rispetta la struttura minima obbligatoria;
- Adotta le convenzioni tipografiche e stilistiche stabilite nelle NdP;
- Risulta tracciabile tramite riferimenti incrociati coerenti;
- È nello stato **Approved** al momento della consegna, a seguito di verifica formale.

2.1.8.2.2 Processo di Sviluppo

Il Processo di Sviluppo disciplina la realizzazione tecnica (Codice e Test).

2.1.8.2.2.1 Criteri di Verifica del Processo di Sviluppo

Il processo è conforme se e solo se soddisfa le seguenti condizioni:

- Il codice rispetta le convenzioni di nomenclatura e il **Single Responsibility Principle**;
- Ogni funzione è documentata con commenti standard (scopo, parametri, ritorno);
- Il codice include commenti di tracciabilità verso i requisiti dell'AdR;
- I test seguono la nomenclatura definita e sono stati validati secondo le procedure di **Qualifica**.

2.1.8.2.3 Processi di Supporto

I Processi di Supporto garantiscono la qualità trasversale tramite le seguenti procedure operative:

- **PR-SUPP-01: Procedura di Documentazione:** Ogni documento **deve** essere redatto in **Typst**, versionato con schema x.y.z e associato a task su **Jira**. È **vietata** l'integrazione in main senza Pull Request approvata da un verificatore diverso dal redattore.
- **PR-SUPP-02: Procedura di Qualifica:** Ogni artefatto **deve** essere sottoposto a verifica prima dell'integrazione. Gli esiti **devono** essere tracciati nel **Piano di Qualifica**.

2.1.8.2.4 Processi Organizzativi

I Processi Organizzativi garantiscono la coerenza del team tramite le seguenti procedure operative:

- **PR-ORG-01: Gestione dei Processi:** Ogni Sprint **deve** essere avviato con task stimate su Jira e concluso con una retrospettiva documentata. Le riunioni **devono** produrre un verbale entro 48 ore.
- **PR-ORG-02: Infrastruttura:** Il repository **deve** rispettare la struttura definita. Le **branch protection rules** su develop e main **devono** essere attive.
- **PR-ORG-03: Miglioramento e Formazione:** Le azioni correttive **devono** essere tracciate su Jira. Le ore di formazione **devono** essere tenute distinte da quelle rendicontabili.

2.1.8.2.4.1 Criteri di Verifica dei Processi Organizzativi

I processi sono conformi se e solo se soddisfano le seguenti condizioni:

- Tutti i ruoli sono definiti; Jira è configurato con workflow attivi;
- I verbali interni ed esterni sono pubblicati entro le scadenze;
- Le metriche di Sprint sono analizzate e le modifiche alle NdP sono approvate dal Responsabile.

2.1.9 Piano di Qualifica

2.1.9.1 Nota sull'adozione dello Standard ISO/IEC 25010 e PDCA

La redazione del Piano di Qualifica ^G 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.

Inoltre, l'adozione strutturale del ciclo **Plan-Do-Check-Act (PDCA)** 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.

2.1.9.2 Definizione del Documento

Il **Piano di Qualifica** costituisce il riferimento **vincolante** per la gestione della qualità nel progetto e **deve** definire, in modo tracciabile e verificabile:

- le strategie di assicurazione qualità (*Quality Assurance*);
- le attività di controllo qualità (*Quality Control*);
- le metriche di processo e prodotto con relative soglie di accettabilità e ottimalità;
- le procedure di verifica e validazione degli artefatti;
- i meccanismi di miglioramento continuo.

2.1.9.3 Elementi Costituenti del Documento

Il Piano di Qualifica **deve** includere obbligatoriamente i seguenti elementi: Qualità di Processo, Qualità di Prodotto, Metodi di Testing, Cruscotto di Valutazione e Automiglioramento.

2.1.9.3.1 Qualità di Processo

La sezione Qualità ^G di Processo definisce l'insieme delle metriche utilizzate per monitorare e controllare i processi produttivi adottati dal gruppo. Il monitoraggio **è essenziale** per garantire efficienza, efficacia e miglioramento continuo del **Way of Working**.

2.1.9.3.2 PR-FOR-09: Procedura di Definizione e Gestione delle Metriche

La determinazione delle soglie di accettabilità e ottimalità per le metriche adottate dal gruppo **Skarab Group** segue una strategia ingegneristica basata sui seguenti criteri vincolanti:

- **Adesione agli Standard Internazionali:** Per i processi di fornitura, il gruppo **deve** adottare il framework **Earned Value Management (EVM)** con intervalli di confidenza standard (0.90 – 1.10). Per la qualità del prodotto, le soglie **devono** essere calibrate sul modello **ISO/IEC 25010**;
- **Analisi del Dominio e Benchmark:** Le metriche di supporto e sviluppo (es. **Gulpease**, **Complessità Ciclomantica**) **devono** essere tarate su benchmark di settore per garantire la verificabilità oggettiva;
- **Miglioramento Continuo (PDCA):** Al termine di ogni **Sprint**, i valori **devono** essere analizzati rispetto alle prestazioni storiche del team per una ricalibrazione dinamica delle soglie (obiettivi **S.M.A.R.T.**).

2.1.9.3.2.1 Nomenclatura delle Metriche

La nomenclatura utilizzata per le metriche di processo **deve** essere la seguente:

MPC##

dove:

- M sta per Metrica;
- PC indica Processo;
- ## è un numero progressivo crescente da 01.

2.1.9.3.2.2 Processi Primari

2.1.9.3.2.2.1 Fornitura e Sviluppo

Codice	Nome	Descrizione
MPC01	Budget At Completion	Budget totale preventivato per il progetto
MPC02	Planned Value	Valore del lavoro che si era pianificato di completare entro un determinato momento
MPC03	Actual Cost	Costo effettivamente sostenuto per il lavoro svolto fino a un determinato momento
MPC04	Earned Value	Valore del lavoro effettivamente completato
MPC05	Budget Variance	Differenza tra il budget totale e la stima del costo finale. Un valore positivo indica che si prevede di risparmiare budget
MPC06	Schedule Variance	Misura lo scostamento temporale: valori negativi indicano ritardi, positivi anticipi
MPC07	Cost Performance Index	Indice di efficienza economica del progetto. CPI = 1 indica perfetta aderenza al budget, < 1 sforamento, > 1 risparmio
MPC08	Schedule Performance Index	Indice di efficienza della schedulazione. SPI = 1 indica perfetta aderenza ai tempi, < 1 ritardo, > 1 anticipo
MPC09	Estimate At Completion	Stima del costo totale finale del progetto basata sulla performance attuale. Utilizza il CPI per proiettare il costo a completamento considerando l'efficienza dimostrata

Table 3: Metriche per il processo di Fornitura

Codice	Nome	Descrizione
MPC10	Requirements Stability Index	Misura la volatilità dei requisiti. Indica la percentuale di requisiti che non hanno subito modifiche, aggiunte o cancellazioni rispetto alla baseline iniziale

Table 4: Metriche per il processo di Sviluppo

Nota: Valori bassi di RSI possono segnalare un'analisi iniziale insufficiente o cambiamenti significativi nelle esigenze degli *stakeholder*^G. Il monitoraggio è cruciale specialmente a seguito di revisioni correttive.

2.1.9.3.2.3 Processi di Supporto**2.1.9.3.2.3.1 Documentazione e Verifica**

Codice	Nome	Descrizione
MPC11	Gulpease Index	Indice di leggibilità. Valuta la complessità linguistica basandosi sulla lunghezza delle parole e delle frasi. Valori: ≥ 80 molto facile; 60-80 media difficoltà; 40-60 abbastanza difficile; < 40 molto difficile
MPC12	Correttezza Ortografica	Numero di errori ortografici rilevati nel documento
MPC13	Code Coverage	Percentuale di codice sorgente che viene eseguita durante il lancio della suite di test automatizzati. Indica il grado di copertura della verifica dinamica
MPC14	Test Success Rate	Percentuale di test automatizzati che superano con successo l'esecuzione. Un valore del 100% è necessario per garantire la stabilità del sistema prima di ogni rilascio

Table 5: Metriche per il processo di Documentazione e Verifica

Rilevanza: Il Test Success Rate deve rimanere costantemente al 100% per evitare regressioni e garantire l'affidabilità del *Software*.

2.1.9.3.2.4 Processi Organizzativi

Codice	Nome	Descrizione
MPC15	Metrics Satisfaction	Percentuale di metriche che rispettano le soglie di accettabilità. Fornisce una visione d'insieme sulla qualità complessiva dei processi adottati
MPC16	Sprint Goal Achievement	Percentuale degli obiettivi prefissati durante lo Sprint Planning che sono stati effettivamente raggiunti e validati al termine dello Sprint. Misura l'efficacia della pianificazione e dell'esecuzione

Table 6: Metriche per i processi Organizzativi

2.1.9.3.2.5 Criteri di Verifica della Qualità di Processo

L'elemento **Qualità di Processo** del Piano di Qualifica è considerato conforme se risultano soddisfatte tutte le seguenti condizioni:

- ogni metrica è identificata in modo univoco tramite codice (MPC##);
- per ciascuna metrica sono definiti:
 - formulazione matematica o descrizione operativa del calcolo;
 - soglia di accettabilità;
 - soglia di ottimalità.
- le metriche sono classificate secondo i processi di riferimento (Primari, Supporto, Organizzativi);
- le soglie adottate sono motivate in base a standard di settore, benchmark o analisi storiche;
- i valori misurati sono confrontati con le soglie durante le retrospettive di Sprint;
- gli scostamenti rilevati sono documentati e danno luogo ad azioni correttive tracciate.

Il mancato rispetto di una o più delle condizioni sopra elencate comporta la non conformità dell'elemento e richiede un'azione correttiva prima della validazione del documento.

2.1.9.3.3 Qualità del Prodotto

La sezione Qualità di Prodotto definisce l'insieme delle metriche utilizzate per valutare il prodotto software rispetto ai requisiti e alle caratteristiche intrinseche dello standard **ISO/IEC 25010**.

2.1.9.3.4 PR-FOR-10: Procedura di Controllo Qualità del Prodotto

Questa procedura norma le attività di misurazione e valutazione del software **Code Guardian**. Il team **ha l'obbligo** di monitorare le metriche di prodotto per garantire il rispetto dei requisiti di qualità:

- **Estrazione Dati:** Le metriche relative al codice (MPD04-07) **devono** essere estratte tramite strumenti di analisi statica automatizzati ad ogni chiusura di Sprint;
- **Validazione:** Le metriche funzionali (MPD01-03) **devono** essere validate attraverso l'esecuzione dei test di sistema e di accettazione;
- **Gestione Scostamenti:** Il superamento delle soglie di accettabilità **deve** comportare l'immediata apertura di una segnalazione su **Jira** per la risoluzione del debito tecnico.

2.1.9.3.4.1 Nomenclatura delle Metriche

La nomenclatura utilizzata per le metriche di prodotto è la seguente:

MPD##

dove:

- M sta per Metrica;
- PD indica Prodotto;
- ## è un numero progressivo crescente da 01.

2.1.9.3.4.2 Adeguatezza Funzionale e Affidabilità

Codice	Nome	Descrizione
MPD01	Copertura Requisiti Obbligatori	Percentuale di requisiti obbligatori soddisfatti rispetto al totale dei requisiti obbligatori identificati. Questi requisiti sono vincolanti per l'accettazione del prodotto
MPD02	Failure Density	Misura la densità di errori nel software. Valori elevati indicano problemi di qualità del codice
MPD03	Availability	Percentuale di tempo in cui il sistema è operativo e disponibile all'uso

Table 7: Metriche di Adeguatezza Funzionale e Affidabilità

2.1.9.3.4.3 Manutenibilità e Sicurezza

Codice	Nome	Descrizione
MPD04	Comment Density	Percentuale di linee di commento rispetto alle linee di codice. Una documentazione adeguata migliora la comprensibilità e facilita la manutenzione futura del codice
MPD05	Cyclomatic Complexity	Misura la complessità del codice contando i percorsi linearmente indipendenti. Valori elevati indicano codice difficile da testare e mantenere
MPD06	Coupling (Fan-out)	Numero di dipendenze esterne di un modulo. Misura il grado di accoppiamento tra componenti. Valori elevati riducono la modularità
MPD07	Vulnerability Detection	Numero di vulnerabilità di sicurezza critiche rilevate attraverso analisi statica e dinamica del codice

Table 8: Metriche di Manutenibilità e Sicurezza

2.1.9.3.4.4 Criteri di Verifica della Qualità di Prodotto

L'elemento **Qualità di Prodotto** del Piano di Qualifica è considerato conforme se risultano soddisfatte tutte le seguenti condizioni:

- ogni metrica è identificata in modo univoco tramite codice (MPD##);
- per ciascuna metrica sono definiti: formulazione matematica, soglia di accettabilità e soglia di ottimalità;
- le metriche sono riconducibili a caratteristiche del modello **ISO/IEC 25010**;
- le soglie adottate sono motivate in base a standard di settore o analisi di benchmark;
- i valori misurati sono tracciati nel Cruscotto di Valutazione;

- gli scostamenti rispetto alle soglie sono analizzati e danno luogo ad azioni correttive.

Il mancato rispetto di una o più delle condizioni sopra elencate comporta la non conformità dell'elemento.

2.1.9.3.5 Strategie di Testing

Questa sezione elenca le metodologie di testing adottate per garantire la qualità del prodotto. I dettagli operativi riguardanti le procedure, le tipologie di test e le relative nomenclature sono approfonditi normativamente nella sezione **3.2.1 Verifica** delle presenti Norme di Progetto.

2.1.9.3.6 Cruscotto di Valutazione

Nella sezione Cruscotto di Valutazione, per ciascuna metrica viene fornita una rappresentazione grafica dell'andamento temporale accompagnata da un'analisi qualitativa.

Le misurazioni coprono il periodo dall'**aggiudicazione del capitolato** fino alla milestone **RTB**. Per la milestone **PB** vengono fornite *stime previsionali* che riflettono un incremento dell'efficienza previsto dopo il superamento della sessione d'esame.

2.1.9.3.7 PR-FOR-11: Procedura di Gestione del Cruscotto

Il monitoraggio dei risultati **deve** avvenire tramite il Cruscotto di Valutazione. Il team **ha l'obbligo** di aggiornare tale strumento secondo la seguente procedura:

- **Raccolta:** I dati vengono estratti al termine di ogni **Sprint**;
- **Analisi:** L'analisi qualitativa **deve** essere condotta collegialmente durante la retrospettiva di Sprint;
- **Formalizzazione:** Il Piano di Qualifica **deve** essere aggiornato per riflettere i nuovi valori entro l'inizio dello Sprint successivo.

2.1.9.3.7.1 Criteri di Verifica del Cruscotto di Valutazione

L'elemento **Cruscotto di Valutazione** è considerato conforme se e solo se:

- è presente una rappresentazione grafica dell'andamento temporale per ogni metrica;
- ogni grafico è accompagnato da un'analisi qualitativa interpretativa;
- le soglie di accettabilità e ottimalità sono evidenziate graficamente;
- l'aggiornamento è coerente con la chiusura di ogni Sprint;
- gli scostamenti significativi sono documentati e motivati.

2.1.9.3.8 Automiglioramento

2.1.9.3.8.1 Finalità e Ambito di Applicazione

L'automiglioramento si applica a tutti i processi adottati dal gruppo e costituisce un'attività trasversale che accompagna l'intero ciclo di vita del progetto. Esso ha lo scopo di:

- identificare criticità operative, organizzative o tecniche;
- analizzare le cause profonde delle inefficienze rilevate (**Root Cause Analysis**);
- definire e implementare azioni correttive o preventive;
- valutare l'efficacia delle soluzioni adottate attraverso metriche oggettive.

2.1.10 Verbali delle Riunioni

2.1.10.1 Definizione del Documento

I **Verbali delle Riunioni** sono documenti ufficiali che registrano in modo sintetico e tracciabile lo svolgimento degli incontri. Essi **hanno l'obbligo** di documentare decisioni, informazioni e azioni concordate, costituendo evidenza formale delle attività di coordinamento.

2.1.10.2 Elementi Costituenti del Documento

Ogni verbale **deve** includere obbligatoriamente:

- Data, ora e modalità/luogo;
- Elenco dei partecipanti;
- Ordine del giorno;
- Sintesi delle discussioni e decisioni prese;
- Azioni da intraprendere (assegnatario e impegno orario previsto).

2.1.10.2.1 PR-FOR-12: Procedura di Verbalizzazione

Questa procedura norma la gestione delle comunicazioni formali:

- **Tempistiche:** I verbali interni **devono** essere redatti entro 24 ore. I verbali esterni **devono** essere inviati al proponente entro 48 ore lavorative (rif. **PR-FOR-01**).
- **Identificazione:** Ogni verbale **deve** essere identificato in modo univoco (es. VI_2026-03-02 per interni, VE_2026-03-02 per esterni).
- **Approvazione:** Per i verbali esterni, si applica la regola del silenzio-assenso dopo 24 ore dalla notifica.

2.1.10.2.1.1 Criteri di Verifica dei Verbali

Un verbale è considerato conforme se e solo se soddisfa le seguenti condizioni:

- include tutti gli elementi obbligatori sopra citati;
- descrive decisioni e azioni in modo non ambiguo;
- è archiviato nel repository ufficiale in formato PDF generato da **Typst**;
- ogni azione tracciata ha un responsabile identificabile.

2.1.11 Diario di Bordo

2.1.11.1 Definizione del Documento

Il **Diario di Bordo** è un documento di monitoraggio a cadenza settimanale. Esso **deve** fornire una visione sintetica e continuativa delle attività svolte, delle criticità emerse e della pianificazione a breve termine.

2.1.11.2 Elementi Costituenti del Documento

Ogni voce del diario **deve** includere obbligatoriamente:

- Periodo di riferimento (Settimana n);
- Sintesi delle attività svolte;
- Problemi riscontrati e soluzioni adottate;
- Pianificazione per la settimana successiva.

2.1.11.2.1 PR-FOR-13: Procedura di Aggiornamento del Diario

Questa procedura norma l'allineamento continuativo:

- **Redazione:** Il **Responsabile** di turno **ha l'obbligo** di aggiornare il diario entro la fine di ogni settimana lavorativa.
- **Tracciabilità:** Le attività riportate **devono** trovare riscontro nelle task gestite su **Jira**.

2.1.11.2.1.1 Criteri di Verifica del Diario di Bordo

Il Diario è considerato conforme se e solo se soddisfa le seguenti condizioni:

- copre il periodo temporale in modo continuo senza lacune settimanali;
- riporta in modo coerente le attività effettivamente svolte;
- evidenzia esplicitamente le soluzioni adottate per le criticità riscontrate;
- include una pianificazione realistica approvata dal Responsabile.

2.2 Processo di Sviluppo

Il Processo di Sviluppo adottato da Skarab Group ^G definisce le attività necessarie all'analisi dei requisiti, alla progettazione, alla codifica, al testing e all'accettazione del prodotto Software *Code Guardian* ^G. In conformità allo standard ISO/IEC/IEEE 12207, il team **garantisce** un approccio strutturato, orientato alla qualità e alla tracciabilità in ogni fase del ciclo di vita.

2.2.1 Attività del Processo di Sviluppo

Il processo si articola nelle seguenti macro-attività, ciascuna delle quali contribuisce alla realizzazione del prodotto finale.

2.2.1.1 Implementazione del processo

Questa attività consiste nella selezione e definizione del **ciclo di vita** del Software. Per *Code Guardian*, il team adotta un approccio che permette di sviluppare il prodotto in modo graduale, rispondere rapidamente ai feedback della Proponente, mantenere alta la qualità tramite continue attività di verifica e allineare lo sviluppo alle milestone RTB e PB.

2.2.1.2 Analisi e Progettazione

Questa attività norma la definizione logica e strutturale del prodotto.

- **Analisi dei Requisiti:** Studio del capitolato, identificazione di attori e casi d'uso (UML 2.5) e classificazione (funzionali, qualità, vincolo).
- **Progettazione Architettuale:** Identificazione delle componenti hardware e software per garantire scalabilità e separazione delle responsabilità.
- **Progettazione di Dettaglio:** Specifica dettagliata di classi, interfacce, strutture dati e algoritmi.

2.2.1.2.1 PR-SVIL-01: Procedura di Specifica Tecnica

- **Obbligo:** Definire interfacce, classi e algoritmi prima della fase di codifica.
- **Strumenti:** StarUML, Draw.io, Typst.

2.2.1.2.1.1 Criteri di Verifica di Analisi e Progettazione

L'attività è conforme se e solo se:

- ogni requisito dell'AdR è tracciato verso almeno una componente progettuale;
- la progettazione specifica la strategia di gestione delle eccezioni;
- i diagrammi UML sono aggiornati e coerenti con la documentazione tecnica.

2.2.1.3 Implementazione e Integrazione

Questa attività norma la trasformazione dei requisiti in codice sorgente eseguibile e la combinazione delle singole unità. L'integrazione può seguire strategie **bottom-up**, **top-down** o **incrementale**.

2.2.1.3.1 PR-SVIL-02: Procedura di Sviluppo e Build

- **Codifica:** Implementazione fedele delle specifiche seguendo gli standard stilistici.
- **Integrazione:** Combinazione progressiva delle unità con test ad ogni iterazione.
- **Strumenti:** VS Code, Docker.

2.2.1.3.1.1 Criteri di Verifica di Codifica e Integrazione

L'attività è conforme se e solo se:

- il codice supera l'analisi statica (ESLint/Prettier);
- non sono presenti regressioni a seguito dell'integrazione di nuove unità;
- ogni unità è corredata dai relativi test di unità (Unit Test).

2.2.1.4 Testing, Installazione e Qualifica

Il team **ha l'obbligo** di verificare che il Software soddisfi gli obiettivi del **Piano di Qualifica**. Il sistema completo viene sottoposto a test su scenari d'uso reali, seguito dall'installazione nell'ambiente di destinazione e dal supporto all'accettazione.

2.2.1.4.1 PR-SVIL-03: Procedura di Validazione

- **Test di Sistema:** Simulazione di scenari reali per verificare interazione e robustezza.
- **Supporto all'Accettazione:** Fornire evidenza del soddisfacimento dei requisiti alla Proponente e risolvere eventuali problematiche.
- **Strumenti:** Postman, Playwright, Selenium.

2.2.2 Relazione con le Baseline di progetto

Le attività del Processo di Sviluppo sono distribuite tra le due baseline principali per garantire un avanzamento controllato:

- **Requirements and Technology Baseline (RTB):** Si concentra su Analisi dei Requisiti, Proof of Concept (PoC) e definizione degli standard normativi.
- **Product Baseline (PB):** Si concentra su Progettazione dell'Architettura, Codifica e consolidamento del prodotto.

2.2.3 Codifica e Standard Tecnici

Lo stack tecnologico è multi-linguaggio e basato su:

- **TypeScript:** Frontend (React) e Backend (NestJS).
- **Python:** Agenti AI, Orchestratore e script di analisi dati.
- **MongoDB e Node.js:** Database con Mongoose e ambiente di esecuzione per servizi server-side.

2.2.3.1 PR-SVIL-04: Standard di Codifica (Coding Standards)

Ogni membro del team è **vincolato** al rispetto delle seguenti norme tecniche:

Tipo	Standard Obbligatorio
camelCase	Variabili e funzioni
PascalCase	Classi, componenti React, interfacce (prefisso I)
kebab-case	Nomi dei file di logica e Collezioni MongoDB (al plurale)
SCREAMING_SNAKE_CASE	Costanti e variabili d'ambiente (.env)

Table 9: Standard di nomenclatura tecnica

Regole di Qualità, Stile e Robustezza:

- **Modularità:** La logica di business **deve** essere separata dall'interfaccia utente; ogni funzionalità **deve** essere isolata in componenti atomici.
- **Single Responsibility:** Ogni funzione **ha l'obbligo** di svolgere un'unica attività.
- **Documentazione:** Uso **obbligatorio** di JSDoc/Docstring (descrizione, parametri e ritorno) per ogni metodo.
- **Automazione:** L'uso di Prettier (2 spazi, apici singoli) ed ESLint è **mandatorio** in pipeline CI.
- **Gestione Errori:** Ogni chiamata asincrona **deve** essere protetta da blocchi try-catch con logging specifico per prevenire crash.
- **Funzioni Lambda:** Incoraggiate per callback brevi; la logica complessa **deve** essere estratta in metodi nominati.

2.2.3.1.0.1 Criteri di Verifica della Codifica

Il codice è conforme se e solo se:

- supera i controlli di linting senza errori;
- la densità di commenti rispetta la metrica **MPD04**;
- la complessità ciclomatica (**MPD05**) non supera la soglia definita nel PdQ.

2.2.3.2 PR-SVIL-05: Procedura di Gestione della Configurazione

Norma l'archiviazione e il versionamento tramite **Git** e **GitHub**.

- **Strategia Branching:** main (stabile), develop (integrazione), feature/ (task atomiche).
- **Commits:** Standard **Conventional Commits** (es. feat(parser): add python support).
- **Pull Request:** Integrazione **vietata** senza Peer Review di un componente terzo.

2.2.3.3 PR-SVIL-06: Procedura di Verifica (Testing)

Il team **adotta** una strategia di test a livelli per assicurare la correttezza del software.

Livello	Metodologia Obbligatoria
Unit Test	Verifica delle singole unità tramite Mocking.
Integration Test	Verifica interazione tra moduli (es. Python e NestJS).
Analisi Statica	Controllo automatico della qualità prima di ogni merge.

Table 10: Livelli di test e strumenti (Vitest, Pytest, SonarQube)

2.2.3.3.0.1 Criteri di Verifica della Qualità Tecnica

L'attività è conforme se e solo se:

- non sono presenti commit diretti sui rami protetti;
- ogni PR supera i controlli automatizzati di CI;
- la Code Coverage ^G non scende sotto la soglia definita nel PdQ.

2.2.4 Evoluzione delle Norme verso l'MVP

Le norme qui definite costituiscono la baseline tecnica per la fase **RTB**.

Nota Operativa: Con l'avvio della fase **PB**, Skarab Group **ha l'obbligo** di includere configurazioni di deployment, standard di sicurezza API/LLM e soglie di accoppiamento per librerie terze.

3 Processi di Supporto

In conformità allo standard ISO/IEC/IEEE 12207, i Processi di Supporto comprendono le attività trasversali che accompagnano l'intero ciclo di vita del progetto, garantendo la qualità, la tracciabilità e la verificabilità degli artefatti prodotti.

Nel contesto del progetto *Code Guardian*, Skarab Group ha identificato i seguenti Processi di Supporto rilevanti:

- Documentazione;
- Qualifica (Verifica e Validazione).

3.1 Documentazione

Il processo di documentazione è trasversale a tutti i processi primari e di supporto: i suoi artefatti costituiscono la memoria del progetto e il mezzo principale per garantire tracciabilità, verificabilità e comunicazione uniforme tra i membri del gruppo e gli stakeholder esterni.

Nel contesto di Skarab Group ^G, il processo di documentazione disciplina la pianificazione, lo sviluppo, la produzione, la revisione e la manutenzione di tutti i documenti ufficiali del progetto *Code Guardian* ^G.

3.1.1 PR-SUPP-01: Procedura di Gestione Documentale (Ciclo di Vita)

Ogni documento **ha l'obbligo** di seguire il seguente ciclo di vita:

1. **Creazione / Modifica**: redazione o aggiornamento del contenuto in formato sorgente (Typst ^G).
L'attività **deve** essere tracciata da una task ^G su Jira ^G
2. **Verifica**: controllo di conformità (ortografia, stile, contenuti, rispetto delle NdP) da parte di un membro del team **diverso** dal redattore;
3. **Approvazione**: validazione finale da parte del Responsabile di Progetto ^G per il rilascio ufficiale.

3.1.1.1 Strumenti di Redazione

Il team è **vincolato** all'utilizzo del seguente toolkit tecnologico:

Strumento	Descrizione e Utilizzo Obbligatorio
VS Code ^G	Editor principale per la scrittura dei documenti.
Typst ^G	Linguaggio di markup per la formattazione. Consente la creazione di documenti strutturati, tabelle, figure e riferimenti. I template comuni sono centralizzati nella cartella <code>lib/</code> .
Tinymist ^G	Estensione per VS Code che fornisce anteprima in tempo reale e supporto per la scrittura.

Table 11: Strumenti di redazione documentale

3.1.1.2 Struttura dei Documenti

Tutti i documenti ufficiali **devono obbligatoriamente** rispettare la struttura minima seguente:

- **Pagina di copertina**, contenente: nome del documento e versione;
- **Registro delle modifiche**, con data, versione, descrizione della modifica e nomi di redattori e verificatori;
- **Indice dei contenuti**, generato automaticamente;
- **Corpo del documento**, organizzato in sezioni e sottosezioni numerate;
- Ogni pagina (eccetto la copertina) **deve** riportare un header con nome del gruppo e titolo del documento, e un footer con numero di pagina.

3.1.1.2.0.1 Criteri di Verifica della Documentazione

L'attività è conforme se e solo se il documento rispetta la struttura minima, il Registro delle modifiche riporta il corretto incremento di versione e la verifica è stata effettuata da un membro indipendente tramite task tracciata su Jira.

3.1.2 PR-SUPP-02: Gestione della Configurazione e Versionamento

3.1.2.1 Sistema di Versioning

Ogni documento **deve** adottare il seguente schema di versionamento semantico:

x.y.z

- x (**Major**): incrementato solo all'approvazione formale del documento da parte del Responsabile ^G
- y (**Minor**): incrementato al completamento di una modifica sostanziale;
- z (**Patch**): incrementato per modifiche minori (correzioni ortografiche, aggiustamenti di forma).

Il changelog **deve** essere aggiornato ad ogni incremento di versione.

3.1.2.2 Denominazione dei File e Tracciabilità

La denominazione dei file sorgente Typst segue lo schema SIGLA.typ (es. NdP, PdP, PdQ, AdR). Le date all'interno dei documenti sono scritte nel formato AAAA/MM/GG.

Ogni attività documentale (redazione, verifica, aggiornamento) **deve** essere associata a una task su Jira ^G con descrizione, assegnatario e scadenza. Al termine di ogni Sprint, i documenti completati e verificati vengono integrati nel branch develop tramite **Pull Request**.

3.1.2.3 Repository GitHub - Struttura e Branching

La gestione avviene nel repository DocumentazioneProgetto seguendo la struttura di branching ufficiale (fatta eccezione per nome-componente, riservata al repository PoC).

La repository **deve** rispettare la seguente organizzazione:

Cartella / File	Contenuto
.github/	Configurazioni specifiche di GitHub.
Glossario/	Sorgenti e infrastruttura per la versione web del glossario.
RTB/ e candidatura/	Documenti ufficiali delle relative milestone.
assets/	Risorse multimediali utilizzate nei documenti.
diarioDiBordo/	Diari di bordo settimanali.
lib/	Librerie di funzioni Typst riutilizzabili (gestite dall'Amministratore). I documenti le importano tramite <code>#import "../lib/..."</code> .
verbali/	Verbali di riunioni interne ed esterne.
website/	Sorgenti del sito web di Skarab Group.
README.md & index.html	Presentazione del progetto e homepage del sito web nella root.

Table 12: Struttura gerarchica del repository di documentazione

3.2 Qualifica

Il processo di **Qualifica** garantisce che il prodotto software e la documentazione soddisfino gli standard di qualità definiti e i requisiti specificati. Si articola in due attività fondamentali:

- **Verifica:** “*Did I build the system right?*” (Ho costruito correttamente il sistema?);
- **Validazione:** “*Did I build the right system?*” (Ho costruito il sistema giusto?).

3.2.1 PR-SUPP-03: Procedura di Verifica

La Verifica ha come scopo accertare che quanto prodotto sia stato realizzato correttamente. È un’attività continua applicata sia alla documentazione che al codice. Gli esiti **devono** essere documentati nel Piano di Qualifica per supportare il miglioramento continuo (ciclo PDCA) e garantire la conformità alle NdP.

3.2.1.1 Strumenti a supporto

Per il calcolo dell’indice di Gulpease ^G **deve** essere utilizzato il servizio farfalla-project.org/readability_static/. Ulteriori strumenti verranno ampliati durante la PB.

3.2.1.2 Attività previste (ISO/IEC/IEEE 12207)

Il processo si articola nell’implementazione (definizione criticità, metodologie e strumenti) e nelle attività di controllo vero e proprio su:

- **Processi:** verifica dell’efficacia dei processi adottati;
- **Requisiti:** verifica di esaustività, consistenza e tracciabilità;
- **Progettazione:** verifica di efficacia rispetto ai requisiti;
- **Codice:** verifica del soddisfacimento dei requisiti implementativi;
- **Integrazione:** verifica della corretta interazione tra componenti;
- **Documentazione:** verifica di completezza, correttezza e conformità.

3.2.1.3 Implementazione: Regola del Main e Pull Request

Nessun documento o componente software può essere integrato senza aver superato il processo di verifica.

- **Regola fondamentale:** Il branch main contiene esclusivamente artefatti verificati e di qualità accertata.
- Le verifiche **devono** essere condotte tramite *Pull Request* su GitHub ^G, garantendo: tracciabilità, revisione da parte di un Verificatore ^G diverso dal redattore e approvazione formale. Problematiche complesse sono discusse in riunione e verbalizzate.

3.2.1.4 Verifica della Documentazione

Ogni documento **deve** seguire un approccio di revisione formale sistematica articolato in:

- **Analisi Linguistica:** Verifica di grammatica, sintassi, ortografia e indice di leggibilità.
- **Ispezione dei Contenuti:** Verifica di esaustività, consistenza interna e coerenza.
- **Conformità Normativa:** Rispetto dei template e convenzioni grafiche interne.
- **Gestione del Flusso:** Rispetto rigoroso del systema di versioning e policy dei branch.

3.2.1.5 Verifica del Codice: Analisi Statica e Dinamica

La strategia di verifica del software adotta due approcci complementari, formalizzati dalla RTB ed estesi in PB.

3.2.1.5.1 Analisi Statica

Non richiede l’esecuzione dell’oggetto e individua errori sintattici, strutturali o concettuali. Si realizza tramite:

- **Metodi Formali:** Dimostrazioni matematiche, riservate solo a componenti critiche.
- **Walkthrough:** Analisi completa e approfondita. È costoso e applicato solo a componenti critiche o quando la natura del problema è ignota.

- **Ispezione:** Utilizzo di checklist predefinite. **Skarab Group privilegia l'ispezione** in quanto garantisce uniformità e riduce i tempi. (Nota: Il walkthrough preliminare è necessario per stabilire i criteri della checklist).

Il team **ha l'obbligo** di applicare l'analisi statica mediante Review manuale del codice e strumenti di linting: ESLint ^G per JavaScript/TypeScript; Pylint/Black per Python.

3.2.1.5.2 Analisi Dinamica e Testing

Prevede l'esecuzione dell'oggetto per rilevare **failure** (comportamenti non conformi) e **fault** (difetti nel codice). I test **devono** essere: Ripetibili, Indipendenti e Automatizzabili. L'automazione è supportata da:

- **Driver:** invocano il codice da testare se non direttamente eseguibile;
- **Stub:** simulano dipendenze non ancora implementate;
- **Logger:** registrano gli esiti in formato strutturato.

I Test seguono lo schema gerarchico:

[Prefisso] - [Numero] . [Sottonumero]

Prefisso	Tipologia	Obiettivi e Dettagli Metodologici
TU	Test di Unità	Verificano il funzionamento atomico in isolamento. Si suddividono in Funzionali (black-box) (verificano input/output tramite valori: non ammissibili inf., limite inf., centrali, limite sup., non ammissibili sup.) e Strutturali (white-box) (verificano percorsi logici e copertura).
TI	Integrazione	Verificano la corretta interazione tra unità. Strategie adottate: Top-Down (dalle componenti con maggiori dipendenze esterne) o Bottom-Up (dalle componenti di base con minori dipendenze).
TS	Sistema	Valutano il comportamento del sistema nel suo complesso conformemente all'AdR.
TR	Regressione	Garantiscono l'assenza di nuovi difetti. Procedura obbligatoria: 1. Correzione difetto; 2. Esecuzione del test che aveva rilevato il difetto; 3. Esecuzione intera suite di regressione; 4. Verifica che nessun test superato in precedenza sia fallito.
TA	Accettazione	Verifica finale per la validazione del prodotto rispetto alle aspettative.

Table 13: Nomenclatura e Classificazione delle Analisi Dinamiche (Test)

3.2.1.5.2.1 Criteri di Verifica della Qualifica

L'attività è conforme se e solo se: ogni Pull Request è approvata da un verificatore indipendente, il Test Success Rate è pari al 100% prima del merge in main, gli indici di leggibilità rispettano il PdQ e ogni anomalia è tracciata su Jira.

3.2.2 PR-SUPP-04: Procedura di Validazione

Il processo di Validazione ha come scopo verificare che **Code Guardian** soddisfi le esigenze di Var Group ^G e degli utenti finali, focalizzandosi sulla conformità alle aspettative reali.

3.2.2.1 Obiettivi e Strategie di Validazione

Il team **ha l'obbligo** di: confermare l'implementazione delle funzionalità richieste, verificare l'utilizzabilità del sistema, assicurare la risoluzione dei problemi di audit/remediation e ottenere l'accettazione formale.

L'attività si realizza tramite un approccio strutturato basato su:

1. **Tracciamento sistematico dei requisiti:** Ogni requisito raccolto nell'**Analisi dei Requisiti** **deve** essere tracciato fino alla sua implementazione e validazione, garantendo la copertura completa.
2. **Test di Accettazione (TA):** Simulano scenari d'uso reali per verificare criteri di successo oggettivi per ogni requisito, confermando che il sistema realizzato sia effettivamente quello richiesto.

4 Processi Organizzativi

In conformità allo standard ISO/IEC/IEEE 12207, i Processi Organizzativi costituiscono l'insieme delle attività trasversali che garantiscono il corretto andamento del progetto dal punto di vista gestionale, infrastrutturale e della crescita professionale del team.

Nel contesto del progetto *Code Guardian*^G, Skarab Group^G ha identificato i seguenti Processi Organizzativi rilevanti:

- Gestione dei Processi;
- Infrastruttura;
- Processo di Miglioramento;
- Processo di Formazione.

4.1 Gestione dei Processi

Il processo ha l'obiettivo di definire e coordinare le attività, assegnare responsabilità, pianificare le risorse e garantire una comunicazione interna ed esterna efficace.

4.1.1 Ruoli di Progetto e Rotazione

Ogni membro di Skarab Group ricopre ruoli differenti nel corso del progetto, garantendo la rotazione ciclica prevista. La distribuzione dei ruoli per ciascuno Sprint^G è pianificata dal Responsabile di Progetto^G all'inizio dello Sprint e tracciata nel Piano di Progetto^G.

Ruolo	Responsabilità Vincolante
Responsabile	Coordina il progetto, garantisce scadenze e budget. Rappresenta il team con Var Group e i docenti. Monitora l'avanzamento (EVM), gestisce rischi e approva documenti. Convoca riunioni e gestisce email formali.
Amministratore	Gestisce l'infrastruttura (Git/GitHub, Jira) e gli ambienti di sviluppo. Definisce workflow di branching e convenzioni di commit. Mantiene le NdP e i template di documento. Garantisce backup e sicurezza.
Verificatore	Assicura la qualità degli artefatti e la conformità alle NdP. Esegue review del codice sorgente e verifica il rispetto degli standard di codifica. Redige esiti di verifica nelle Pull Request e monitora le metriche.
Analista	Formalizza i requisiti (funzionali, qualità, vincolo) tramite interazione con gli stakeholder. Redige l'AdR e garantisce la tracciabilità tra requisiti e fonti.
Progettista	Definisce l'architettura del sistema e i design pattern. Specifica la struttura delle classi, algoritmi e interfacce. Redige la Specifica Tecnica ed effettua scelte tecnologiche motivate.
Programmatore	Implementa il codice rispettando gli standard NdP. Sviluppa unit test e garantisce code coverage. Gestisce dipendenze e build automation. Documenta il codice e collabora alla risoluzione dei difetti.

Table 14: Matrice di responsabilità per i ruoli di progetto

4.1.2 PR-ORG-01: Procedura di Gestione Task e Pianificazione

La gestione operativa delle task^G avviene esclusivamente tramite Jira^G, fonte ufficiale per monitoraggio e rendicontazione.

- **Requisiti Task:** Ogni task **deve** includere: descrizione univoca, assegnatario, stima ore, scadenza e Sprint di riferimento.

- **Atomicità:** Le task **devono** essere eseguibili da un singolo membro senza dipendenze bloccanti non esplicitate.
- **Rendicontazione:** È responsabilità individuale aggiornare quotidianamente lo stato e le ore, segnalando tempestivamente impedimenti.

4.1.3 PR-ORG-02: Procedura di Coordinamento e Comunicazione

Il coordinamento avviene tramite canali specifici per garantire efficienza e formalità.

Canale	Utilizzo e Obblighi	Tipo
Discord	Riunioni settimanali per condividere avanzamento e risolvere impedimenti. Ogni riunione deve produrre un verbale interno.	Sincrono
WhatsApp / Telegram	Comunicazioni rapide, urgenze e coordinamento di gruppo breve.	Asincrono
Email	Convocazioni e comunicazioni formali con Var Group e Docenti. Ogni riunione esterna deve produrre un verbale esterno.	Formale
Jira	Assegnazione task e monitoraggio avanzamento Sprint.	Gestionale

Table 15: Piano di comunicazione del team

4.2 Infrastruttura

L'Amministratore ^G è responsabile della creazione, configurazione e manutenzione del toolkit tecnologico.

Strumento	Utilizzo nel progetto
Git ^G / GitHub ^G	Controllo versione, hosting repository, gestione Pull Request e code review. L'Amministratore definisce le regole di protezione branch.
Google Calendar ^G	Pianificazione riunioni esterne e scadenze gestite dal Responsabile.
Automazione	Script interni per generazione Glossario (HTML/PDF), compilazione Typst e aggiornamento sito web (GitHub Actions).

Table 16: Strumenti dell'infrastruttura di progetto

4.2.1 PR-ORG-03: Procedura di Gestione della Configurazione (GitHub)

La struttura dei branch **deve** seguire rigorosamente lo schema seguente:

- **main:** Branch stabile con artefatti verificati e approvati. Commit diretti **vietati**.
- **develop:** Branch di integrazione protetto. Richiede Pull Request con revisione obbligatoria.
- **feature/nome - feature:** Sviluppo funzionalità (fase PB).
- **fix/nome - fix:** Correzione bug.
- **[ACRONIMO - DOC]:** Branch per redazione documenti (es. NdP, AdR). Verbali raggruppati in verbali.
- **nome - componente:** Branch per componenti PoC (deprecato in PB).

4.2.1.1 Organizzazione del Repository DocumentazioneProgetto

Le cartelle **devono** essere così strutturate: .github/ (configurazioni), Glossario/ (sito web glossario), RTB/ (documenti milestone), assets/ (multimedia), candidatura/, diarioDiBordo/, lib/ (librerie Typst condivise), verbali/, website/.

4.3 Processo di Miglioramento

Il processo garantisce il soddisfacimento degli standard di qualità tramite il monitoraggio continuo.

4.3.1 PR-ORG-04: Procedura di Miglioramento (Retrospettiva)

Il processo si articola in: Inizializzazione (NdP), Valutazione (PdQ) e Miglioramento. Al termine di ogni Sprint, il team **ha l'obbligo** di:

1. Analizzare i valori delle metriche rispetto alle soglie del PdQ.
2. Identificare inefficienze operative.
3. Definire azioni correttive concrete su Jira con responsabile e scadenza.
4. Aggiornare le NdP o il PdQ previo approvazione del Responsabile.

4.4 Processo di Formazione

Garantisce che il team acquisisca le competenze necessarie per gli obiettivi del Capitolato C2.

4.4.1 PR-ORG-05: Procedura di Formazione Continua

- **Competenze Chiave:** TypeScript/React, NestJS, Python (Agenti AI), MongoDB, Node.js, Git/GitHub, Typst, AWS.
- **Auto-formazione:** Le ore di studio individuale **non sono rendicontabili** nel monte ore di progetto.
- **Spazi per la Formazione:** Il Responsabile può pianificare periodi di approfondimento tecnologico nelle fasi a peso minore, senza compromettere gli obiettivi di Sprint.
- **Supporto Proponente:** Il team **ha l'obbligo** di partecipare alle sessioni dedicate fornite da **Var Group**.

4.4.1.0.0.1 Criteri di Verifica della Formazione

Il processo è conforme se lo studio autonomo è escluso dal consuntivo e se il team dimostra operatività sulle tecnologie richieste senza blocchi critici.