



Norme di Progetto

v0.12.0

Registro delle Modifiche

Data	Versione	Descrizione	Redattore	Verificatore
2026/02/23	0.12.0	Verifica per la sezione Analisi dei Requisiti per la RTB		Suar Alberto
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
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

Data	Versione	Descrizione	Redattore	Verificatore
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	12
2.1.4.1 Protocollo di Comunicazione	12
2.1.4.2 Procedura di Consegna	12
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	13
2.1.5.3 Elementi Costituenti del Documento	13
2.1.5.3.1 Analisi dei Rischi	13
2.1.5.3.2 Pianificazione del Ciclo di Vita del Software	15
2.1.5.3.3 Preventivo ed Allocazione delle Risorse	17
2.1.5.3.4 Monitoraggio e Controllo del Rendimento degli Sprint	18
2.1.6 Analisi dei Requisiti	19
2.1.6.1 Standard di Riferimento	19
2.1.6.2 Definizione del Documento	19
2.1.6.3 Elementi Costituenti del Documento	20
2.1.6.3.1 Casi d'Uso	20
2.1.6.3.2 Requisiti di Sistema	21
2.1.7 Glossario	21
2.1.7.1 Definizione del Documento	21
2.1.7.2 Elementi Costituenti del Documento	22
2.1.7.3 Criteri di Verifica del Glossario	22
2.1.8 NdP – Norme di Progetto	22
2.1.8.1 Definizione del Documento	22
2.1.8.2 Elementi Costituenti del Documento	23
2.1.8.2.1 Processo di Fornitura	23
2.1.8.2.2 Processo di Sviluppo	24
2.1.8.2.3 Processi di Supporto	25
2.1.8.2.4 Processi Organizzativi	26
2.1.9 Piano di Qualifica	27
2.1.9.1 Nota sull'adozione dello Standard ISO/IEC 25010 e PDCA	27
2.1.9.2 Definizione del Documento	27
2.1.9.3 Elementi Costituenti del Documento	27
2.1.9.3.1 Qualità di Processo	27
2.1.9.3.2 Qualità del Prodotto	30
2.1.9.3.3 Metodi di Testing	31

2.1.9.3.4 Cruscotto di Valutazione	31
2.1.9.3.5 Automiglioramento	32
2.1.10 Verbali delle Riunioni	32
2.1.10.1 Definizione del Documento	32
2.1.10.2 Elementi Costituenti del Documento	32
2.1.10.3 Tipologie di Verbale	33
2.1.10.4 Criteri di Verifica dei Verbali delle Riunioni	33
2.1.11 Diario di Bordo	33
2.1.11.1 Definizione del Documento	33
2.1.11.2 Elementi Costituenti del Documento	33
2.1.11.3 Criteri di Verifica del Diario di Bordo	33
2.2 Processo di Sviluppo	34
2.2.1 Attività del Processo di Sviluppo	34
2.2.1.1 Implementazione del processo:	34
2.2.1.2 Analisi dei Requisiti:	34
2.2.1.3 Progettazione dell'architettura di sistema:	34
2.2.1.4 Analisi dei Requisiti Software:	34
2.2.1.5 Progettazione dell'architettura Software:	34
2.2.1.6 Progettazione di dettaglio:	35
2.2.1.7 Codifica e testing del Software:	35
2.2.1.8 Integrazione Software:	35
2.2.1.9 Test di qualifica del Software:	35
2.2.1.10 Integrazione e Test di sistema:	35
2.2.1.11 Installazione	35
2.2.1.12 Supporto all'accettazione:	35
2.2.2 Relazione con le Baseline di progetto	35
2.2.2.1 Requirements and Technology Baseline (RTB)	35
2.2.2.2 Product Baseline (PB)	36
2.2.3 Codifica	36
2.2.3.1 Convenzioni	36
3 Processi di Supporto	38
3.1 Documentazione	38
3.1.1 Ciclo di Vita dei Documenti	38
3.1.2 Strumenti di Redazione	38
3.1.3 Struttura dei Documenti	38
3.1.4 Gestione della Configurazione e Versionamento	38
3.1.4.1 Sistema di Versioning	38
3.1.4.2 Branching Strategy	39
3.1.4.3 Denominazione dei File	39
3.1.4.4 Repository GitHub - Struttura delle Cartelle	39
3.1.4.4.1 Librerie Typst (lib/)	39
3.1.5 Tracciabilità delle Attività Documentali	40
3.2 Qualifica	40
3.2.1 Verifica	40
3.2.1.1 Obiettivi della Verifica	40
3.2.1.2 Strumenti a supporto	40
3.2.1.3 Attività proposte	40
3.2.1.4 Implementazione del processo	41

3.2.1.5 Attività di Verifica	41
3.2.1.5.1 Verifica della Documentazione	41
3.2.1.5.2 Verifica del Codice	41
3.2.2 Validazione	44
3.2.2.1 Obiettivi della Validazione	44
3.2.2.2 Attività previste	44
3.2.2.2.1 Implementazione del processo	44
3.2.2.2.2 Attività di Validazione	44
4 Processi Organizzativi	44
4.1 Gestione dei Processi	45
4.1.1 Ruoli di Progetto	45
4.1.2 Rotazione dei Ruoli	46
4.1.3 Gestione dei Task e Pianificazione Operativa	46
4.1.3.1 Strumento di Issue Tracking: Jira	46
4.1.4 Coordinamento: Riunioni e Comunicazioni	46
4.1.4.1 Riunioni Interne	46
4.1.4.2 Riunioni Esterne	47
4.1.4.3 Canali di Comunicazione	47
4.2 Infrastruttura	47
4.2.1 Strumenti Adottati	47
4.2.2 Configurazione dei Principali Strumenti	48
4.2.2.1 GitHub - Struttura del Repository	48
4.2.2.2 Script di Automazione	48
4.2.3 Manutenzione dell'Infrastruttura	48
4.3 Processo di Miglioramento	48
4.3.1 Attività Previste	48
4.3.2 Procedura di Miglioramento	48
4.4 Processo di formazione	49
4.4.1 Competenze Richieste	49
4.4.2 Supporto del PropONENTE	49
4.4.3 Nota sulle Ore di Formazione	50
4.4.4 Spazi per la Formazione	50

Indice tabelle

Table 1 Tabella riepilogativa dei documenti di progetto	12
Table 2 Matrice di Rischio: visualizzazione delle soglie di intervento	14
Table 3 Metriche per il processo di Fornitura	28
Table 4 Metriche per il processo di Sviluppo	29
Table 5 Metriche per il processo di Documentazione e Verifica	29
Table 6 Metriche per i processi Organizzativi	29
Table 7 Metriche di Adeguatezza Funzionale e Affidabilità	30
Table 8 Metriche di Manutenibilità e Sicurezza	31
Table 9 Ruoli di progetto e relative responsabilità	45
Table 10 Canali di comunicazione del team	47
Table 11 Strumenti dell'infrastruttura di progetto	47
Table 12 Tecnologie e fonti di formazione	49

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 gruppo di studenti 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 documento ha valore 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 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 sottolineatura**, qualora **non sia esplicitamente indicato come collegamento a un documento o a una sezione specifica**, 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](#).

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: **04/02/2026**)

- **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: **04/02/2026**)
- **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: **04/02/2026**)

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: **04/02/2026**)
- **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: **04/02/2026**)
- **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: **04/02/2026**)
- **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: **04/02/2026**)
- **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: **04/02/2026**)

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 gruppo di lavoro 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 ^G, il gruppo Skarab Group ^G assume il ruolo di fornitore, mentre l'azienda Var Group ^G 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:

- **Analisi e presa in carico dei requisiti**

Studio sistematico dei requisiti funzionali ^G e dei requisiti non funzionali ^G definiti nel Capitolato C2, finalizzato alla comprensione, classificazione e valutazione di fattibilità tecnica e organizzativa, nonché all'individuazione dei vincoli progettuali rilevanti.

- **Pianificazione delle attività di progetto**

Definizione della struttura delle attività necessarie allo sviluppo del progetto, assegnazione delle responsabilità all'interno del gruppo di lavoro, stima delle tempistiche e delle risorse disponibili e individuazione delle principali milestone di progetto.

- **Produzione, gestione e verifica della documentazione**

Redazione progressiva della documentazione di progetto secondo strutture e convenzioni condivise, gestione delle versioni dei documenti e verifica della loro completezza, coerenza e conformità alle Norme di Progetto e agli standard di riferimento.

- **Monitoraggio dell'avanzamento e controllo della qualità della fornitura**

Controllo periodico dello stato di avanzamento delle attività rispetto alla pianificazione definita, verifica del rispetto delle scadenze e valutazione della qualità degli artefatti prodotti mediante attività di revisione interna.

- **Preparazione e consegna degli artefatti**

Consolidamento degli artefatti prodotti, verifica finale della loro conformità ai requisiti stabiliti e predisposizione della documentazione necessaria alla consegna ufficiale prevista dalle scadenze di progetto.

Nel contesto didattico del progetto, e in particolare nel periodo che conduce alla RTB^G, tali attività si concretizzano prevalentemente nella redazione strutturata e nel controllo della documentazione di progetto, che rappresenta una componente essenziale della fornitura.

2.1.3 Prodotti del Processo

Ai fini delle presenti Norme di Progetto, il processo di fornitura disciplina in particolare la produzione e la gestione dei seguenti prodotti **documentali**:

- **Norme di Progetto**

Documento che definisce le regole metodologiche, organizzative e tecniche adottate dal gruppo di lavoro durante l'intero ciclo di vita del progetto.

- **Piano di Progetto**

Documento che definisce l'organizzazione del lavoro, la pianificazione delle attività, l'allocazione delle risorse, la gestione dei rischi e le modalità di monitoraggio dell'avanzamento.

- **Piano di Qualifica**

Documento che specifica i criteri di qualità del prodotto e della documentazione, nonché le attività di verifica e validazione previste.

- **Analisi dei Requisiti**

Documento che raccoglie e formalizza i requisiti funzionali e non funzionali del sistema, derivati dal Capitolato C2.

- **Verbali di verifica e revisione**

Documenti che registrano esiti e decisioni delle attività di controllo e revisione svolte durante il progetto.

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	Segretario della riunione	Proponente (se esterni), Cardin Riccardo, Vardanega Tullio, Skarab Group	Interno / Esterno

Table 1: Tabella riepilogativa dei documenti di progetto

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

Per garantire professionalità e tracciabilità nei confronti del Proponente (Var Group [©]):

- **Canale Ufficiale:** Le comunicazioni formali (consegne, richieste di chiarimento sui requisiti) avvengono esclusivamente tramite email istituzionale o incontri verbalizzati.
- **Verbalizzazione:** Ogni incontro con il proponente deve produrre un verbale esterno entro **48 ore** lavorative dall'evento.
- **Approvazione Verbali:** Il verbale si considera approvato se non pervengono richieste di rettifica entro 24 ore dalla sua pubblicazione/invio.

2.1.4.2 Procedura di Consegna

In occasione delle scadenze ufficiali, il Responsabile di Progetto deve seguire questa checklist obbligatoria:

1. **Verifica di Conformità:** Accertarsi che tutti i documenti siano nello stato **Approved** e che ogni artefatto in rilascio (documentale o software) soddisfi i criteri di verifica specifici definiti per esso nel Piano di Qualifica.
2. **Consolidamento:** Generare i PDF definitivi dai sorgenti Typst [©] congelati.
3. **Tagging:** Creare una **Release** sul repository GitHub con il tag di versione corrispondente (es. v1.0.0-RTB).
4. **Notifica:** Inviare comunicazione formale di avvenuta consegna al proponente indicando il link alla release note.

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 (Standard for Software Project Management Plans)**. Tale scelta riflette la volontà di

applicare un approccio rigoroso e sistematico alla gestione del progetto, adattando i contenuti alle dimensioni, alla complessità e alle tempistiche del caso di studio considerato.

L'adozione dello standard garantisce una copertura completa degli aspetti critici del ciclo di vita del progetto e si traduce concretamente nei seguenti elementi chiave:

- **Struttura e organizzazione sistematica:**

L'indice e l'articolazione del documento ricalcano la gerarchia proposta dallo standard IEEE 1058, assicurando una pianificazione completa delle aree gestionali, organizzative e tecniche ed evitando lacune nella definizione delle responsabilità e dei processi.

- **Analisi e gestione del rischio:**

In conformità alle linee guida IEEE, è adottato un approccio di risk management basato su valutazioni di probabilità, impatto e strategie di mitigazione/contingenza, consentendo un monitoraggio strutturato delle principali incertezze di progetto.

- **Tracciabilità e controllo:**

Il documento distingue in modo esplicito la fase di pianificazione (**Preventivo**) da quella di monitoraggio e rendicontazione (**Consuntivo**), permettendo un controllo oggettivo dell'avanzamento e l'analisi degli eventuali scostamenti rispetto alle stime iniziali.

- **Efficacia operativa:**

Alcuni moduli previsti dallo standard sono stati ottimizzati e accorpati per migliorare la leggibilità e la comunicazione all'interno del team, mantenendo inalterati rigore metodologico e chiarezza dei flussi informativi richiesti dalla normativa.

2.1.5.2 Definizione del Documento

Il **Piano di Progetto** costituisce il riferimento gestionale vincolante per la conduzione del progetto e deve definire, in modo tracciabile e verificabile:

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

Il Piano di Progetto è utilizzato dal gruppo di lavoro come base per il coordinamento interno e per la rendicontazione dell'avanzamento, garantendo coerenza tra obiettivi, tempi e risorse disponibili.

2.1.5.3 Elementi Costituenti del Documento

Il Piano di Progetto deve includere almeno i seguenti elementi, ciascuno disciplinato da regole operative e criteri minimi di verifica:

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

Gli elementi costituenti del Piano di Progetto sono normati di seguito.

2.1.5.3.1 Analisi dei Rischi

L'Analisi dei Rischi deve individuare e valutare i principali fattori di rischio che possono compromettere il raggiungimento degli obiettivi di progetto. Essa deve includere rischi di natura tecnica, organizzativa e temporale, nonché la valutazione della loro probabilità di occorrenza e del potenziale impatto sul progetto.

2.1.5.3.1.1 Metodologia di Analisi

Il team adotta un approccio **semi-quantitativo** per la valutazione dei rischi. Ogni rischio identificato viene descritto e valutato sulla base di due parametri fondamentali:

- **Probabilità di accadimento (P)**: verosimiglianza con cui l'evento potrebbe verificarsi;
- **Impatto (I)**: gravità delle conseguenze sul progetto qualora l'evento si manifesti.

Il **Livello di Rischio (R)** è calcolato come:

$$R = P \times I$$

I parametri P e I sono valutati su una scala discreta da 1 a 3.

La scelta di una scala ridotta (1–3) rispetto a scale più estese (es. 1–5 o percentuali) è motivata dalla volontà di:

- **ridurre la soggettività**: in assenza di dati storici pregressi, una scala più ampia introdurrebbe un'eccessiva varianza nelle stime individuali; la tripartizione Basso/Medio/Alto facilita una categorizzazione netta e supporta il decision-making;
- **supportare l'approccio iterativo**: il metodo adottato permette una rivalutazione rapida dei rischi a ogni Sprint, evitando overhead non necessario rispetto alla dimensione del progetto.

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

Sulla base del valore di R , i rischi vengono classificati per priorità come segue:

- **Rischio Basso (1–3)**: livello di guardia; il rischio è accettabile e soggetto al solo monitoraggio periodico;
- **Rischio Medio (4)**: livello di attenzione; richiede la definizione preventiva di una strategia di mitigazione e un controllo frequente;
- **Rischio Alto (6–9)**: livello critico; richiede azioni preventive tempestive e la predisposizione di piani di contingenza.

In base al valore di R , si applicano le seguenti strategie vincolanti:

- **Mitigazione (Preventiva)**: Azioni da attuare **subito** per ridurre la Probabilità o l'Impatto (Obbligatoria per $R \geq 4$).
- **Contingenza (Reattiva)**: Piano d'azione da attuare **solo se** il rischio si verifica (Obbligatoria per $R \geq 6$).

2.1.5.3.1.2 Categorie di Rischio

Per favorire l'analisi sistematica e l'assegnazione delle responsabilità, i rischi sono classificati nelle seguenti macro-categorie:

- **RT (Rischi Tecnologici)**: criticità legate alle tecnologie adottate (es. LLM ^G, Agenti ^G), agli strumenti di sviluppo, alle integrazioni e all'infrastruttura;

- **RI (Rischi Interpersonali):** problematiche relative al team di progetto ^G, quali disponibilità dei membri, comunicazione, coordinamento e gestione dei conflitti;
- **RCO (Rischi di Costo e Organizzativi):** rischi derivanti da stime non accurate, pianificazione temporale (Schedulazione ^G), dipendenze esterne e vincoli organizzativi.

2.1.5.3.1.3 Ruoli e Responsabilità

La gestione dei rischi è una responsabilità condivisa, articolata come segue:

- **Responsabile di Progetto:** supervisiona il processo di gestione dei rischi, valuta la criticità degli eventi e decide l'attivazione delle strategie di mitigazione o dei piani di contingenza;
- **Team di Progetto:** contribuisce all'identificazione dei rischi e segnala tempestivamente l'insorgere di eventi critici o condizioni anomale;
- **Stakeholder di riferimento:** vengono informati in caso di rischi ad alto impatto o di variazioni rilevanti rispetto al piano approvato.

2.1.5.3.1.4 Gestione e Monitoraggio dei Rischi

La gestione dei rischi non è limitata alla fase di avvio del progetto, ma costituisce un processo ciclico di controllo che accompagna l'intero ciclo di vita.

Nel caso in cui un rischio si manifesti o emergano nuovi rischi, il team applica la seguente procedura operativa:

1. **Rilevamento (Detection):** Ogni membro del team ha l'obbligo di segnalare al **Responsabile di Progetto** l'insorgere di problematiche tecniche o organizzative **entro il termine della giornata lavorativa** in cui vengono rilevate.
2. **Documentazione:** L'evento viene tracciato nel sistema di issue tracking e, se necessario, discusso nel primo meeting utile.
3. **Valutazione e Risposta:** Il Responsabile di Progetto valuta la criticità (*R*) e attiva la strategia di mitigazione o il piano di contingenza.
4. **Rendicontazione:** Al termine dello Sprint ^G, l'evento viene analizzato nel consuntivo per valutare l'efficacia delle contromisure.
5. **Aggiornamento:** Le strategie vengono aggiornate nel registro dei rischi per gli Sprint successivi.

2.1.5.3.1.5 Criteri di Verifica dell'Analisi dei Rischi

L'elemento **Analisi dei Rischi** del Piano di Progetto è considerato conforme alle Norme di Progetto se risultano soddisfatte tutte le seguenti condizioni:

- ogni rischio è descritto in modo univoco e classificato in una delle categorie definite (RT, RI, RCO)
- per ciascun rischio sono esplicitamente indicati i valori di **Probabilità (P)**, **Impatto (I)** e il corrispondente **Livello di Rischio (R)**
- per tutti i rischi classificati come **Medio** o **Alto** è definita almeno una strategia di mitigazione o un piano di contingenza
- lo stato dei rischi è aggiornato con cadenza coerente con la pianificazione degli Sprint
- l'insorgenza di nuovi rischi o la variazione significativa di rischi esistenti è tracciata nei verbali di progetto o negli strumenti di issue tracking adottati
- gli esiti delle attività di gestione dei rischi sono riportati nei consuntivi di Sprint o di fase (es. RTB)

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.

2.1.5.3.2 Pianificazione del Ciclo di Vita del Software

La pianificazione del ciclo di vita del software stabilisce l'organizzazione temporale del progetto e vincola la suddivisione delle attività nelle diverse fasi previste, definendo milestone, scadenze

intermedie e relazioni di dipendenza tra le attività necessarie alla realizzazione del prodotto software.

La pianificazione costituisce il riferimento formale per il coordinamento delle attività e per il monitoraggio dell'avanzamento del progetto, in relazione alla suddivisione in sprint.

2.1.5.3.2.1 Pianificazione in Corrispondenza della RTB

In corrispondenza della **Requirements and Technology Baseline (RTB)**, la pianificazione del ciclo di vita del software prevede la definizione dettagliata delle attività da svolgere. Per ciascuna attività sono obbligatoriamente specificati la descrizione, il responsabile assegnato e lo stato di avanzamento, al fine di garantire un controllo puntuale delle attività di analisi e di impostazione tecnologica.

Al fine di assicurare la tracciabilità delle attività pianificate, ogni attività definita in questa fase deve corrispondere a una **task** registrata sullo strumento di issue tracking adottato dal team (Jira[©]). Il Responsabile di Progetto, all'avvio di ogni Sprint, definisce e pianifica le task su Jira, assegnando a ciascuna una figura di riferimento del team e una scadenza temporale, così da consentire la verifica della coerenza tra quanto pianificato e quanto effettivamente svolto.

Le attività pianificate devono essere **atomiche**. Una task si definisce atomica se:

- è eseguibile da un singolo membro del team;
- non presenta dipendenze dirette **bloccanti** da altre attività; qualora una dipendenza sia inevitabile, essa deve essere esplicitata su Jira tramite collegamento alla task prerequisita (es. relazione **blocks/blocked-by**);

2.1.5.3.2.2 Pianificazione in Corrispondenza della PB

Per le fasi successive del progetto, in particolare in prossimità della **Product Baseline (PB)**, la pianificazione adotta un livello di dettaglio più ampio e meno vincolante. In questa fase, la pianificazione si limita alla definizione delle macro-attività previste, evitando una scomposizione dettagliata in task operative.

Tale approccio è adottato con l'obiettivo di non formalizzare anticipatamente decisioni progettuali non ancora consolidate, preservando la flessibilità necessaria ad affrontare le scelte architettoniche e implementative che emergeranno nel corso dello sviluppo.

La pianificazione a livello di Product Baseline garantisce comunque la tracciabilità delle principali attività attese, includendo, a titolo esemplificativo, l'analisi post **proof-of-concept**, la progettazione architettonica, la progettazione di dettaglio e le attività che convergono nello sviluppo del **Minimum Viable Product (MVP)**.

2.1.5.3.2.3 Criteri di Verifica della Pianificazione del Ciclo di Vita

L'elemento **Pianificazione del Ciclo di Vita del Software** del Piano di Progetto è considerato conforme alle Norme di Progetto se risultano soddisfatte tutte le seguenti condizioni:

1. Per la **Requirements and Technology Baseline (RTB)** tutte le attività pianificate risultano:
 - [] descritte in modo univoco;
 - [] assegnate a un responsabile identificabile;
 - [] tracciate come task sullo strumento di issue tracking adottato;
 - [] le attività pianificate in RTB risultano definite in modo atomico; eventuali dipendenze sono esplicitate e tracciate sullo strumento di issue tracking;
2. Per la **Product Baseline (PB)** sono definite le macro-attività previste, senza anticipare decisioni progettuali non ancora consolidate;
 - [] le milestone e le scadenze risultano coerenti con la pianificazione degli Sprint;

- [] eventuali variazioni significative della pianificazione sono motivate e tracciate nei verbali di progetto o negli strumenti di gestione delle attività.

Il mancato rispetto di una o più delle condizioni sopra elencate comporta la non conformità dell'elemento e richiede un aggiornamento del Piano di Progetto.

2.1.5.3.3 Preventivo ed Allocazione delle Risorse

Il Preventivo e l'Allocazione delle Risorse definiscono la distribuzione delle risorse disponibili tra le attività di progetto, con particolare riferimento all'impegno temporale richiesto e all'assegnazione delle responsabilità ai membri del gruppo di lavoro.

Il preventivo costituisce una stima iniziale dell'impegno complessivo e rappresenta il riferimento formale per il confronto con i consuntivi rilevati durante l'esecuzione del progetto, al fine di supportare le attività di monitoraggio e controllo.

2.1.5.3.3.1 Definizione della Baseline e Costi

La determinazione del preventivo deve rispettare i seguenti criteri:

- **Esclusione Auto-apprendimento:** il periodo antecedente al primo Sprint è classificato come auto-apprendimento e non concorre al calcolo dei costi rendicontabili;
- **Gestione Variazioni:** eventuali scostamenti (in aumento o diminuzione) rispetto alle stime di candidatura devono essere documentati e motivati nei verbali di progetto o negli strumenti di issue tracking.

Nota di Contesto (Analisi Preliminare): A seguito di un'analisi approfondita del lavoro svolto nel periodo di consolidamento, sono state formalmente definite le baseline di interesse. In conformità al Preventivo di candidatura, la distribuzione pro-capite dell'impegno orario rimane invariata. Tuttavia, il preventivo complessivo attuale riporta un lieve scostamento in diminuzione rispetto alle stime iniziali, determinando una riduzione del costo complessivo.

2.1.5.3.3.2 Strategia di Allocazione per Fasi

L'allocazione delle risorse deve adattarsi alla natura specifica della fase di progetto.

1. **Fase RTB (Requirements and Technology Baseline):** L'allocazione delle risorse deve prevedere un impiego significativo delle figure di **Amministratore** e **Analista**.
Motivazione: scelta dettata dall'elevato impatto delle attività documentali previste in questa fase (definizione procedure, strumenti, requisiti), propedeutiche all'avvio della progettazione post-RTB.
2. **Fase PB (Product Baseline):** Il preventivo per questa fase è definito come **flessibile** e soggetto a revisione **rolling-wave**.
Motivazione: nonostante la pianificazione accurata, la natura delle attività future e i rischi residui richiedono un margine di adattabilità. Il preventivo PB è definitivo rispetto allo stato attuale, ma deve essere riesaminato a cadenza almeno pari alla chiusura di ogni Sprint o al verificarsi di eventi significativi. Ogni revisione deve essere approvata dal Responsabile di Progetto e tracciata nei verbali o negli strumenti di gestione delle attività.

2.1.5.3.3.3 Criteri di Verifica del Preventivo e dell'Allocazione delle Risorse

L'elemento **Preventivo ed Allocazione delle Risorse** del Piano di Progetto è considerato conforme se:

- [] è definito un preventivo iniziale coerente con le baseline di progetto individuate;
- [] la distribuzione dell'impegno orario pro-capite è esplicitata e motivata;

- [] per ciascuna baseline (RTB, PB) è indicata una ripartizione delle risorse coerente con la natura delle attività previste;
- [] eventuali scostamenti rispetto al preventivo iniziale sono motivati e documentati;
- [] il preventivo costituisce il riferimento per il confronto con i consuntivi di Sprint.

In assenza di una o più delle condizioni sopra elencate, l'elemento non è considerato conforme e deve essere revisionato.

2.1.5.3.4 Monitoraggio e Controllo del Rendimento degli Sprint

Il Monitoraggio e Controllo del rendimento degli Sprint definisce le modalità con cui viene verificato in modo sistematico l'andamento del progetto rispetto alla pianificazione approvata. Tale attività è obbligatoriamente basata sulla suddivisione del lavoro in iterazioni temporali (**Sprint**), ciascuna caratterizzata da un insieme di attività definite in modo atomico e tracciabile.

Per ogni Sprint devono essere rilevati e confrontati il **Preventivo di Sprint** e il **Consuntivo di Sprint**, al fine di valutare il grado di aderenza tra quanto pianificato e quanto effettivamente realizzato. I risultati di tale confronto costituiscono l'input principale per l'analisi dell'andamento complessivo del progetto e per l'eventuale aggiornamento della pianificazione.

Il monitoraggio include inoltre una valutazione retrospettiva dello Sprint, finalizzata all'individuazione di criticità operative, al miglioramento continuo dei processi adottati e alla verifica della sostenibilità dell'organizzazione del lavoro del team.

2.1.5.3.4.1 Metodologia di Monitoraggio del Rendimento dello Sprint

Al termine di ogni Sprint, il rendimento deve essere descritto in modo strutturato e uniforme, includendo obbligatoriamente i seguenti elementi:

- **Attività svolte:** elenco dettagliato delle attività completate nel periodo di riferimento, con indicazione delle task concluse (“Definition of Done”);
- **Analisi delle deviazioni:** confronto tra le attività pianificate e quelle effettivamente svolte, con evidenza delle cause principali di eventuali scostamenti (es. stime errate, problemi tecnici imprevisti);
- **Rischi rilevati:** identificazione dei nuovi rischi emersi durante lo Sprint o aggiornamento dello stato dei rischi già noti, in conformità alle procedure definite nella sezione di Analisi dei Rischi;
- **Azioni correttive e preventive:** decisioni adottate per mitigare i rischi rilevati o migliorare l'efficacia delle attività negli Sprint successivi.

2.1.5.3.4.2 Preventivo e Consuntivo di Sprint

Il **Preventivo di Sprint** è definito all'inizio dello Sprint sulla base delle attività assegnate a ciascun membro del team. Per ogni attività devono essere stimate le ore di lavoro previste, in coerenza con le responsabilità assegnate e con le competenze richieste.

Il **Consuntivo di Sprint** è definito al termine dello Sprint sulla base delle attività effettivamente svolte e delle ore rendicontate per ciascuna task. La rendicontazione deve riflettere in modo accurato il lavoro realmente eseguito e costituisce una responsabilità individuale di ciascun membro del team.

Il confronto tra preventivo e consuntivo consente di valutare l'efficacia della pianificazione, l'affidabilità delle stime e l'adeguatezza dell'organizzazione del lavoro adottata.

2.1.5.3.4.3 Strumenti di Rendicontazione

La rendicontazione delle attività e delle ore di lavoro è effettuata esclusivamente tramite lo strumento di issue tracking adottato dal team (Jira[®]), che rappresenta la fonte ufficiale dei dati di consuntivo.

Per ciascuna task devono essere obbligatoriamente indicati:

- lo stato di avanzamento;
- le ore rendicontate;
- il membro del team responsabile dell'attività.

L'aggiornamento dei dati deve avvenire con periodicità giornaliera e comunque **entro il termine della giornata lavorativa** in cui l'attività è stata svolta, per garantire la tempestività del controllo.

2.1.5.3.4.4 Regole di Rendicontazione

Sono considerate rendicontabili le attività che producono valore per il progetto. Al fine di garantire la veridicità dei dati, si applicano le seguenti regole:

- **Attività Pianificate:** devono essere tracciate associandole alla task originale prevista nello Sprint;
- **Attività Non Pianificate:** attività impreviste ma necessarie (es. bugfix urgenti, risoluzione problemi configurazione) devono essere obbligatoriamente tracciate creando una nuova task. L'esclusione di tali attività è vietata in quanto falserebbe il calcolo del costo reale del progetto.

Attività non pertinenti agli obiettivi del progetto o svolte senza autorizzazione non possono essere rendicontate a carico del monte ore.

2.1.5.3.4.5 Criteri di Verifica del Monitoraggio e Controllo degli Sprint

L'elemento **Monitoraggio e Controllo del Rendimento degli Sprint** è considerato conforme se, per ogni Sprint:

- [] è definito un Preventivo di Sprint basato sulle attività assegnate;
- [] è redatto un Consuntivo di Sprint basato esclusivamente sui dati estratti da Jira^G
- [] il confronto tra preventivo e consuntivo è esplicitato e gli scostamenti sono motivati;
- [] tutte le ore lavorate (incluse le attività non pianificate) sono state tracciate;
- [] sono identificati e aggiornati i rischi emersi durante lo Sprint;
- [] sono riportate eventuali azioni correttive o preventive per lo Sprint successivo.

La mancata rendicontazione di uno o più Sprint o l'assenza degli elementi sopra elencati comporta la non conformità dell'elemento.

2.1.6 Analisi dei Requisiti

2.1.6.1 Standard di Riferimento

Per la redazione del documento Analisi dei Requisiti, il gruppo adotta i seguenti standard internazionali, che garantiscono un approccio rigoroso e sistematico alla specifica delle necessità software:

- **IEEE 830-1998 (Recommended Practice for Software Requirements Specifications):** standard di riferimento per la stesura di specifiche dei requisiti software, che definisce una struttura chiara e completa per la documentazione dei requisiti funzionali e non funzionali.
- **ISO/IEC/IEEE 29148:2018 (Systems and Software Engineering – Life Cycle Processes – Requirements Engineering):** standard che fornisce linee guida per l'ingegneria dei requisiti, inclusi processi di analisi, specifica e gestione dei requisiti.
- **UML 2.5:** standard per la modellazione dei casi d'uso, che consente di rappresentare graficamente le interazioni tra attori e sistema, facilitando la comprensione e la tracciabilità dei requisiti funzionali.

2.1.6.2 Definizione del Documento

L'Analisi dei Requisiti^G costituisce il riferimento tecnico vincolante per la formalizzazione rigorosa delle necessità e dei vincoli relativi al sistema **Code Guardian**. Il documento ha lo scopo di tradurre

le esigenze espresse dal proponente in un insieme di specifiche formali, non ambigue e verificabili, che guidano l'intero processo di ingegnerizzazione del software.

Il documento definisce, in modo tracciabile e verificabile:

- **Perimetro del Sistema:** la delimitazione univoca dei confini d'azione del software, specificando le interazioni tra il sistema e gli attori esterni;
- **Classificazione e Prioritizzazione delle Specifiche:** la scomposizione delle necessità del Capitolato in requisiti funzionali, di qualità e di vincolo;
- **Modellazione delle Interazioni (Casi d'Uso):** la descrizione strutturata delle interazioni tra attori e sistema, supportata da diagrammi conformi allo standard UML 2.5;
- **Tracciabilità dei Requisiti:** lo stabilimento di legami esplicativi tra i requisiti e i casi d'uso, garantendo che ogni scelta implementativa sia giustificata.

L'Analisi dei Requisiti rappresenta la colonna portante della **Requirements and Technology Baseline (RTB)**. Essa funge da ponte logico tra gli obiettivi strategici del Capitolato d'Appalto e l'implementazione tecnica.

2.1.6.3 Elementi Costituenti del Documento

Il documento è organizzato in modo chiaro e strutturato, includendo obbligatoriamente i seguenti elementi:

- **Casi d'Uso:** descrivono le interazioni tra gli attori e il sistema attraverso scenari d'uso specifici, costituendo la base per l'identificazione dei requisiti funzionali.
- **Requisiti Funzionali:** definiscono i servizi o le funzioni che il sistema deve eseguire in risposta a determinati input.
- **Requisiti di Qualità:** specificano le caratteristiche non funzionali (prestazioni, usabilità, sicurezza) che il sistema deve garantire.
- **Requisiti di Vincolo:** descrivono limitazioni tecnologiche, normative o implementative che condizionano il progetto.

2.1.6.3.1 Casi d'Uso

I casi d'uso ^G rappresentano scenari specifici di interazione. Per la loro descrizione viene utilizzata la nomenclatura **UCx.y.z** dove:

- **x:** identificatore del caso d'uso principale (macro-scenario);
- **y:** numero relativo alle inclusioni (0 indica il caso d'uso principale);
- **z:** numero relativo alle estensioni.

L'identificatore **x** è univoco a livello globale. Qualora un'inclusione o un'estensione presentino ulteriori ramificazioni, la stringa **UCx.y.z** assumerà il ruolo di radice per la nuova gerarchia.

Ogni caso d'uso include:

- **Attori:** identificazione degli utenti o sistemi esterni coinvolti;
- **Precondizioni e Postcondizioni:** stato del sistema prima e dopo l'esecuzione;
- **Scenario Principale:** sequenza ordinata di azioni;
- **Trigger:** evento che innesca il caso d'uso;
- **Inclusioni:** comportamenti obbligatori comuni a più casi d'uso, necessari per il completamento dello scenario principale.
- **Estensioni:** comportamenti opzionali o condizionali che integrano lo scenario principale in circostanze specifiche.
- **Generalizzazioni:** relazioni gerarchiche dove un caso d'uso (o attore) figlio eredita e specializza i comportamenti di un padre.
- **Diagrammi UML:** rappresentazione grafica conforme allo standard UML 2.5.

2.1.6.3.1.1 Criteri di Verifica dei Casi d’Uso

L’elemento **Casi d’Uso** è considerato conforme se risultano soddisfatte tutte le seguenti condizioni:

- **Univocità**: ogni identificatore UCx.y.z è unico e segue la gerarchia definita.
- **Correttezza degli Attori**: ogni attore citato è stato precedentemente censito e definito.
- **Completezza dei Flussi**: lo scenario copre tutti i percorsi logici necessari al raggiungimento dell’obiettivo.
- **Coerenza UML**: esiste corrispondenza 1:1 tra descrizione testuale e diagramma grafico, ad eccezione dei casi d’uso privi di inclusioni o estensioni.
- **Assenza di Implementazione**: la descrizione si limita al “cosa fa” il sistema, non al “come”.

2.1.6.3.2 Requisiti di Sistema

I requisiti rappresentano le specifiche che il sistema deve soddisfare. Ogni requisito è classificato per garantire la tracciabilità rispetto alle fonti. Viene utilizzata la nomenclatura

TipologiaRPrioritàNumero dove:

- **Tipologia**: F (Funzionale), Q (Qualità), V (Vincolo).
- **R**: acronimo di Requisito.
- **Priorità**: Ob (Obbligatorio), De (Desiderabile), Op (Opzionale).
- **Numero**: numero progressivo univoco per tipologia.

2.1.6.3.2.1 Criteri di Verifica dei Requisiti Funzionali

Un requisito funzionale è considerato conforme se risultano soddisfatte le seguenti condizioni:

- **Atomicità**: esprime una singola necessità non ulteriormente scomponibile.
- **Non Ambiguità**: il linguaggio è tecnico, oggettivo e privo di termini soggettivi.
- **Testabilità**: è possibile definire un test di accettazione con esito binario (Passato/Fallito).
- **Tracciabilità**: indica chiaramente la propria fonte (Capitolato o UC di riferimento).

2.1.6.3.2.2 Criteri di Verifica dei Requisiti di Qualità

Un requisito di qualità è considerato conforme se risultano soddisfatte le seguenti condizioni:

- **Misurabilità**: il requisito è espresso attraverso metriche quantitative o soglie numeriche precise.
- **Pertinenza**: la caratteristica descritta è rilevante per gli standard definiti nel Piano di Qualifica.
- **Verificabilità**: è stata individuata una metodologia o strumento atto a misurare il raggiungimento della soglia.

2.1.6.3.2.3 Criteri di Verifica dei Requisiti di Vincolo

Un requisito di vincolo è considerato conforme se risultano soddisfatte le seguenti condizioni:

- **Giustificazione**: il vincolo deriva da una scelta nella **Technology Baseline**, obblighi normativi o imposizioni del Proponente.
- **Fattibilità**: il vincolo non entra in conflitto con requisiti funzionali obbligatori.
- **Precisione Tecnica**: specifica chiaramente l’ambito di applicazione (versioni, framework, standard).

In assenza di una o più delle condizioni sopra elencate, la sezione non è considerata conforme e deve essere revisionata prima del rilascio ufficiale.

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 lo scopo di definire il vocabolario controllato utilizzato trasversalmente nella

documentazione di progetto e nelle comunicazioni ufficiali, eliminando ambiguità semantiche e garantendo uniformità comunicativa tra il gruppo di lavoro e gli stakeholder esterni.

Il documento assolve alle seguenti funzioni:

1. **Disambiguazione:** fornire definizioni univoche per termini che potrebbero avere significati diversi in contesti differenti.
2. **Accessibilità:** rendere disponibile una base di conoscenza centralizzata, consultabile sia in formato documentale statico (PDF) sia come risorsa ipertestuale dinamica (pagina web).

2.1.7.2 Elementi Costituenti del Documento

Il Glossario deve essere organizzato in ordine alfabetico e deve includere obbligatoriamente le seguenti categorie di voci:

- **Termini Tecnici:** parole chiave relative a tecnologie, framework e strumenti adottati.
- **Acronimi e Abbreviazioni:** espansione completa delle sigle utilizzate e loro significato nel contesto del progetto.
- **Concetti di Dominio:** definizioni specifiche del contesto applicativo **Code Guardian**.

Struttura della voce: Per ogni termine, la definizione deve rispettare i seguenti criteri:

- **Univocità:** una sola definizione per termine (evitare sinonimi duplicati; in caso di sinonimi, scegliere un termine principale e rimandare ad esso).
- **Sintesi:** descrizione concisa, priva di ridondanze, focalizzata sugli aspetti utili al progetto.
- **Contestualizzazione:** il significato deve essere riferito all'ambito del progetto e non generico.

2.1.7.3 Criteri di Verifica del Glossario

Il Glossario è considerato conforme se risultano soddisfatte tutte le seguenti condizioni:

- [] ogni termine marcato come voce di Glossario nella documentazione (tramite sottolineatura o link/definizione) possiede una corrispondente definizione nel Glossario.
- [] in ogni documento di progetto, la prima occorrenza di un termine presente nel Glossario è evidenziata e rimanda alla relativa definizione; occorrenze successive possono non essere linkate per preservare la leggibilità.
- [] la versione web del Glossario è raggiungibile e ogni collegamento (anchor link) inserito nei documenti punta a una sezione esistente della pagina web.
- [] le voci sono presentate in rigoroso ordine alfabetico (A-Z).
- [] le definizioni non sono tautologiche: è vietato definire un termine esclusivamente ripetendo il termine stesso o usando una sua variante non informativa; eventuali rimandi ad altri termini sono ammessi solo se tali termini sono a loro volta definiti nel Glossario.

In assenza di una o più delle condizioni sopra elencate, il Glossario non è considerato conforme e deve essere revisionato prima del rilascio ufficiale.

2.1.8 NdP – Norme di Progetto

2.1.8.1 Definizione del Documento

Le **Norme di Progetto (NdP)** costituiscono il documento normativo di riferimento che disciplina le modalità operative adottate dal gruppo di lavoro durante l'intero ciclo di vita del progetto.

Le NdP definiscono in modo vincolante:

- i **principi organizzativi** e il **Way of Working (WoW[®])** adottato dal team;
- le **norme** e le **convenzioni** per la redazione della documentazione e per lo sviluppo del prodotto software;
- gli **strumenti** e gli **ambienti di lavoro** utilizzati;

- i **processi** e le **procedure operative** che regolano la pianificazione, l'esecuzione, il controllo e la verifica delle attività di progetto.

Le Norme di Progetto si applicano a tutti i membri del gruppo di lavoro e hanno validità per l'intera durata del progetto. Il loro rispetto è obbligatorio e costituisce requisito necessario per garantire uniformità operativa, tracciabilità delle attività, qualità degli artefatti prodotti e conformità agli standard di riferimento adottati.

2.1.8.2 Elementi Costituenti del Documento

Le Norme di Progetto sono strutturate secondo la descrizione normativa dei macro-processi primari rilevanti per il progetto, in conformità allo standard ISO/IEC/IEEE 12207.

In particolare, il documento disciplina:

- il **Processo di Fornitura**;
- il **Processo di Sviluppo**.

Per ciascun macro-processo, le Norme di Progetto definiscono:

- l'ambito di applicazione e gli obiettivi del processo;
- i ruoli e le responsabilità coinvolte;
- i documenti prodotti dal processo;
- le regole di redazione, struttura e verifica dei documenti;
- le procedure operative e gli strumenti adottati.

2.1.8.2.1 Processo di Fornitura

Il Processo di Fornitura disciplina la produzione, la gestione e la consegna dei documenti ufficiali di progetto. Tali documenti costituiscono gli artefatti formali della fornitura e sono vincolanti ai fini della validazione delle milestone previste.

I documenti di fornitura normati dalle presenti Norme di Progetto sono:

- Piano di Progetto (PdP);
- Piano di Qualifica (PdQ);
- Analisi dei Requisiti (AdR);
- Norme di Progetto (NdP);
- Glossario.

2.1.8.2.1.1 Piano di Progetto (PdP)

Documento gestionale che definisce la pianificazione, l'organizzazione, l'allocazione delle risorse, la gestione dei rischi e il monitoraggio dell'avanzamento del progetto.

2.1.8.2.1.2 Piano di Qualifica (PdQ)

Documento che definisce i criteri di qualità del prodotto e del processo, nonché le attività di verifica e validazione applicate agli artefatti di progetto.

2.1.8.2.1.3 Analisi dei Requisiti (AdR)

Documento che raccoglie, classifica e formalizza i requisiti funzionali e non funzionali del sistema, derivati dal Capitolato e dalle successive analisi.

2.1.8.2.1.4 Norme di Progetto (NdP)

Documento normativo che definisce le regole, le convenzioni, i processi e gli strumenti che devono essere adottati dal gruppo di lavoro durante l'intero ciclo di vita del progetto.

2.1.8.2.1.5 Glossario

Documento terminologico che costituisce il riferimento unico e vincolante per la definizione dei termini tecnici, degli acronimi e dei concetti di dominio utilizzati nella documentazione e nel codice sorgente.

2.1.8.2.1.6 Criteri di Verifica dei Documenti di Fornitura

Un documento di fornitura è considerato conforme alle Norme di Progetto se risultano soddisfatte tutte le seguenti condizioni:

- [] il documento è presente tra gli artefatti previsti dal Processo di Fornitura;
- [] il documento rispetta la struttura, i contenuti e i criteri di verifica definiti nella propria sezione normativa dedicata;
- [] il documento è redatto utilizzando le convenzioni tipografiche e stilistiche stabilite nelle NdP;
- [] il documento risulta tracciabile rispetto agli altri artefatti di progetto (riferimenti incrociati coerenti);
- [] il documento è stato sottoposto alle attività di verifica previste e risulta nello stato **Approved** al momento della consegna;
- [] il documento adotta obbligatoriamente la seguente struttura minima:
 - **Definizione del Documento**;
 - **Elementi Constituenti del Documento**;
 - **Descrizione degli Elementi**, con specifica delle sezioni e dei contenuti previsti;
 - **Criteri di Verifica** degli elementi definiti.

Il mancato rispetto di uno o più criteri di verifica specifici, inclusa la struttura minima obbligatoria, comporta automaticamente la non conformità del documento ai fini del Processo di Fornitura.

2.1.8.2.2 Processo di Sviluppo

Il Processo di Sviluppo disciplina la realizzazione tecnica del prodotto software e la produzione degli artefatti di progettazione e implementazione.

Gli artefatti prodotti dal Processo di Sviluppo normati dalle presenti Norme di Progetto sono:

- Codice sorgente;
- Test.

2.1.8.2.2.1 Codice Sorgente

Il codice sorgente costituisce l'implementazione del sistema software e deve rispettare gli standard di codifica definiti nella sezione Processo di Sviluppo.

2.1.8.2.2.2 Test

I Test verificano la correttezza e la conformità del codice rispetto ai requisiti e devono seguire la nomenclatura e le tipologie definite nella sezione Qualifica.

2.1.8.2.2.3 Criteri di Verifica del Processo di Sviluppo

Il Processo di Sviluppo è considerato conforme alle Norme di Progetto se risultano soddisfatte tutte le seguenti condizioni:

- [] il codice sorgente rispetta le convenzioni di nomenclatura definite;
- [] ogni funzione/metodo è documentato con commenti che descrivono scopo, parametri e valore di ritorno;
- [] il codice è organizzato in moduli e cartelle secondo la struttura definita;
- [] non sono presenti funzioni lambda complesse;
- [] ogni funzione ha una singola responsabilità (*Single Responsibility Principle*);
- [] il codice che implementa requisiti include commenti di tracciabilità;

- [] i test seguono la nomenclatura definita;
- [] il codice è stato sottoposto a verifica secondo le procedure definite nella sezione Qualifica.

Il mancato rispetto di uno o più criteri di verifica comporta la non conformità del processo e richiede azioni correttive prima dell'integrazione nel branch principale.

2.1.8.2.3 Processi di Supporto

I Processi di Supporto forniscono le attività trasversali necessarie a garantire la qualità e la tracciabilità del progetto.

I processi di supporto normati dalle presenti Norme di Progetto sono:

- Documentazione;
- Qualifica (Verifica e Validazione).

2.1.8.2.3.1 Processo di Documentazione

Il Processo di Documentazione disciplina la produzione, gestione e versionamento di tutti i documenti di progetto.

2.1.8.2.3.2 Processo di Qualifica

Il Processo di Qualifica garantisce che il prodotto software e la documentazione soddisfino gli standard di qualità definiti attraverso le attività di Verifica e Validazione.

2.1.8.2.3.3 Criteri di Verifica dei Processi di Supporto

I Processi di Supporto sono considerati conformi alle Norme di Progetto se risultano soddisfatte tutte le seguenti condizioni:

Per il Processo di Documentazione

- [] ogni documento è redatto in formato Typst e archiviato nel repository del progetto;
- [] ogni documento segue il ciclo di vita definito: Creazione/Modifica → Verifica → Approvazione;
- [] il versionamento segue lo schema x.y.z con le regole definite;
- [] ogni documento include una tabella di cronologia delle versioni con data, versione, descrizione, redattori e verificatori;
- [] ogni documento è associato a task su Jira per tracciabilità;
- [] i documenti sono organizzati in cartelle secondo la struttura del repository definita;
- [] il sistema di branching è rispettato;
- [] il branch develop è protetto e richiede Pull Request con revisione;
- [] nessun documento viene integrato in main senza aver superato la verifica;
- [] ogni Pull Request è approvata da un Verificatore diverso dal redattore;

Per il Processo di Qualifica

- [] tutte le attività di verifica e validazione sono documentate nel Piano di Qualifica;
- [] ogni artefatto è sottoposto a verifica prima dell'integrazione in main;
- [] i test seguono la nomenclatura e le tipologie definite;
- [] le attività di validazione includono tracciamento requisiti e test di accettazione;
- [] gli esiti di verifica e validazione sono tracciati e aggiornati nel Piano di Qualifica.

Il mancato rispetto di uno o più criteri comporta la non conformità del processo e richiede azioni correttive.

2.1.8.2.4 Processi Organizzativi

I Processi Organizzativi disciplinano l'insieme delle attività trasversali che garantiscono il corretto funzionamento del team e la coerenza operativa del progetto durante l'intero ciclo di vita.

I processi organizzativi normati dalle presenti Norme di Progetto sono:

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

2.1.8.2.4.1 Gestione dei Processi

La Gestione dei Processi definisce i ruoli del team, le responsabilità operative, le modalità di pianificazione e assegnazione delle attività e i meccanismi di coordinamento interno ed esterno.

2.1.8.2.4.2 Infrastruttura

Il Processo di Infrastruttura definisce gli strumenti e gli ambienti di lavoro adottati dal team per lo svolgimento delle attività di progetto.

2.1.8.2.4.3 Processo di Miglioramento

Il Processo di Miglioramento garantisce che i processi adottati vengano periodicamente valutati e migliorati.

2.1.8.2.4.4 Processo di Formazione

Il Processo di Formazione garantisce che tutti i membri del team acquisiscano le competenze necessarie alle tecnologie adottate nel progetto *Code Guardian*.

2.1.8.2.4.5 Criteri di Verifica dei Processi Organizzativi

I Processi Organizzativi sono considerati conformi alle Norme di Progetto se risultano soddisfatte tutte le seguenti condizioni:

Per la Gestione dei Processi

- [] tutti i ruoli attivi sono definiti con responsabilità esplicite e non ambigue;
- [] ogni Sprint è avviato con task definiti su Jira, assegnati e stimati;
- [] ogni Sprint è concluso con una retrospettiva documentata;
- [] le riunioni interne si tengono con cadenza almeno settimanale e producono un verbale;
- [] le riunioni esterne con Var Group producono un verbale entro 48 ore lavorative;
- [] i canali di comunicazione ufficiali sono rispettati secondo gli scopi definiti.

Per l'Infrastruttura

- [] tutti gli strumenti elencati sono operativi e accessibili a tutti i membri del team;
- [] il repository GitHub rispetta la struttura di cartelle definita nelle NdP;
- [] le branch protection rules su develop e main sono attive e rispettate;
- [] Jira è configurato con backlog, Sprint board e workflow definiti;
- [] gli script di automazione sono funzionanti e documentati.

Per il Processo di Miglioramento

- [] al termine di ogni Sprint è condotta una retrospettiva con analisi delle metriche;
- [] le azioni correttive identificate sono tracciate su Jira con responsabile e scadenza;
- [] le modifiche alle NdP derivanti da azioni di miglioramento sono approvate dal Responsabile.

Per il Processo di Formazione

- [] ogni membro ha accesso alle risorse formative identificate per le tecnologie del progetto;
- [] le ore di formazione sono tenute distinte da quelle rendicontabili di progetto.

Il mancato rispetto di una o più condizioni comporta la non conformità del processo e richiede un'azione correttiva.

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 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) 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 almeno i seguenti elementi, ciascuno disciplinato da regole operative e criteri minimi di verifica:

- Qualità di Processo;
- Qualità di Prodotto;
- Metodi di Testing;
- Cruscotto di Valutazione;
- Automiglioramento.

2.1.9.3.1 Qualità di Processo

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

2.1.9.3.1.1 Strategia di Definizione delle Soglie Metriche

La determinazione delle soglie di accettabilità e ottimalità per le metriche adottate dal gruppo non è un processo arbitrario, ma segue una strategia ingegneristica basata sui seguenti criteri:

- **Adesione agli Standard Internazionali:** Per i processi di fornitura, il gruppo adotta il framework (EVM), utilizzando gli intervalli di confidenza standard del settore (0.90 – 1.10) per garantire la stabilità economica e temporale. Per la qualità del prodotto, le soglie sono calibrate sulle caratteristiche del modello .

- Analisi del Dominio e Benchmark:** Le metriche di supporto e sviluppo (es.) sono tarate su benchmark di settore per lo sviluppo software moderno, garantendo che i requisiti di manutenibilità e sicurezza siano oggettivamente verificabili.
- Miglioramento Continuo e Calibrazione (PDCA):** In linea con l'approccio , le soglie sono soggette a revisione periodica. Al termine di ogni , i valori vengono analizzati rispetto alle prestazioni storiche del team; questo permette una ricalibrazione dinamica delle soglie nel , rendendo gli obiettivi sempre sfidanti ma tecnicamente sostenibili (obiettivi S.M.A.R.T.).

2.1.9.3.1.2 Nomenclatura delle Metriche

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

MPC##

dove:

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

2.1.9.3.1.3 Processi Primari

2.1.9.3.1.3.1 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*. Il monitoraggio è cruciale specialmente a seguito di revisioni correttive.

2.1.9.3.1.4 Processi di Supporto

2.1.9.3.1.4.1 Documentazione e Verifica

Codice	Nome	Descrizione
MPC11	Indice di Gulpease	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*.

2.1.9.3.1.5 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

2.1.9.3.1.6 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.

2.1.9.3.2 Qualità del Prodotto

La sezione Qualità di Prodotto definisce l'insieme delle metriche utilizzate per valutare il prodotto software rispetto ai requisiti definiti nell'Analisi dei Requisiti e alle caratteristiche intrinseche definite dallo standard ISO/IEC 25010.

2.1.9.3.2.1 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.

2.1.9.3.2.2 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

2.1.9.3.2.3 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

2.1.9.3.2.4 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 o descrizione operativa del calcolo;
 - soglia di accettabilità;
 - 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 e documentati 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.

2.1.9.3.3 Metodi di Testing

Questa sezione del Piano di Qualifica 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.

2.1.9.3.4 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 che va dall'**aggiudicazione del capitolo** fino alla milestone **RTB**.

Per la milestone **PB** vengono fornite *stime previsionali*. Rispetto a quanto rilevato nella RTB, ci si attende un *incremento dell'efficienza* e una maggiore aderenza alle tempistiche prefissate. La fase precedente, infatti, ha risentito dei rallentamenti dovuti alla concomitanza con la sessione d'esame. Con il superamento di tale periodo, la PB riflette una pianificazione più fluida e una gestione operativa che punta a stabilizzare i ritmi, ottimizzando i risultati.

2.1.9.3.4.1 Frequenza di Aggiornamento

Il cruscotto viene aggiornato con cadenza di Sprint secondo la seguente procedura:

- I dati vengono raccolti al termine di ogni Sprint.
- L'analisi viene condotta durante la retrospettiva di Sprint.
- Il Piano di Qualifica viene aggiornato per riflettere i nuovi valori misurati.

2.1.9.3.4.2 Criteri di Verifica del Cruscotto di Valutazione

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

- [] per ciascuna metrica definita è presente una rappresentazione grafica dell'andamento temporale;
- [] ogni grafico è accompagnato da un'analisi qualitativa che interpreta i dati;
- [] le soglie di accettabilità e ottimalità sono evidenziate graficamente;
- [] il cruscotto è aggiornato con cadenza almeno pari alla chiusura di ogni Sprint;
- [] gli scostamenti significativi sono documentati e motivati;

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

2.1.9.3.5 Automiglioramento

2.1.9.3.5.1 Finalità e Ambito di Applicazione

L'automiglioramento ha lo scopo di:

- identificare criticità operative, organizzative o tecniche emerse durante il progetto;
- analizzare le cause profonde delle inefficienze rilevate;
- definire e implementare azioni correttive o preventive;
- valutare l'efficacia delle soluzioni adottate attraverso metriche oggettive.

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.

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, chiaro e tracciabile lo svolgimento delle riunioni di progetto. Essi hanno lo scopo di documentare le decisioni assunte, le informazioni condivise e le azioni concordate, costituendo evidenza formale delle attività di coordinamento e comunicazione.

2.1.10.2 Elementi Costituenti del Documento

Ogni verbale deve includere obbligatoriamente i seguenti elementi:

- **Data, ora e modalità/luogo** della riunione;
- **Elenco dei partecipanti**;
- **Ordine del giorno**;
- **Sintesi delle discussioni**;

- **Decisioni prese;**
- **Azioni da intraprendere**, con eventuale assegnatario e scadenza.

2.1.10.3 Tipologie di Verbale

I verbali si distinguono in:

- **Verbali Interni:** relativi a riunioni tra i membri del gruppo di lavoro, finalizzate al coordinamento delle attività di progetto;
- **Verbali Esterne:** relativi a riunioni con il Proponente^G o altri stakeholder esterni, finalizzate alla discussione di requisiti, avanzamento dei lavori e feedback.

2.1.10.4 Criteri di Verifica dei Verbali delle Riunioni

Un **Verbale delle Riunioni** è considerato conforme alle Norme di Progetto se:

- [] include tutti gli elementi costituenti obbligatori;
- [] descrive in modo chiaro e non ambiguo decisioni e azioni concordate;
- [] è redatto secondo le convenzioni tipografiche e strutturali definite nelle NdP;
- [] è archiviato nel repository di progetto ed è tracciabile rispetto alla riunione a cui si riferisce.

Il mancato rispetto di uno o più criteri comporta la non conformità del verbale.

2.11 Diario di Bordo

2.11.1 Definizione del Documento

Il **Diario di Bordo** è un documento di monitoraggio a cadenza settimanale finalizzato all'allineamento sullo stato di avanzamento del progetto. Esso consente di fornire una visione sintetica e continuativa delle attività svolte, delle criticità emerse e della pianificazione a breve termine.

2.11.2 Elementi Costituenti del Documento

Ogni Diario di Bordo deve includere obbligatoriamente i seguenti elementi:

- **Periodo di riferimento;**
- **Attività svolte** durante la settimana;
- **Problemi riscontrati** e soluzioni adottate;
- **Pianificazione delle attività** per la settimana successiva.

2.11.3 Criteri di Verifica del Diario di Bordo

Il **Diario di Bordo** è considerato conforme alle Norme di Progetto se:

- [] copre un periodo temporale ben definito;
- [] riporta in modo coerente le attività effettivamente svolte;
- [] evidenzia eventuali problemi o criticità incontrate;
- [] include una pianificazione realistica delle attività future;
- [] è redatto secondo il formato e le convenzioni stabilite nelle NdP.

In assenza di uno o più elementi obbligatori, il Diario di Bordo non è considerato conforme.

2.2 Processo di Sviluppo

Il Processo di Sviluppo adottato da Skarab Group definisce le attività necessarie all’analisi dei requisiti, alla progettazione, alla codifica, al testing e all’accettazione del prodotto Software *Code Guardian*.

Tale processo è conforme allo standard ISO/IEC/IEEE 12207 e garantisce un approccio strutturato, tracciabile e orientato alla qualità durante l’intero ciclo di vita del progetto.

2.2.1 Attività del Processo di Sviluppo

Il processo comprende le seguenti macro-attività, ciascuna delle quali contribuisce alla realizzazione del prodotto finale secondo standard qualitativi definiti.

2.2.1.1 Implementazione del processo:

Questa attività consiste nella selezione e definizione del **ciclo di vita** del Software più appropriato in base alla complessità, agli obiettivi e ai vincoli del progetto. Per *Code Guardian*, Skarab Group adotta un approccio che permette di:

- sviluppare il prodotto in modo graduale;
- rispondere rapidamente ai feedback della Proponente e agli apprendimenti derivanti dall’uso delle tecnologie;
- mantenere alta la qualità del prodotto grazie a continue attività di verifica;
- allineare lo sviluppo alle milestone RTB e PB definite dal corso.

2.2.1.2 Analisi dei Requisiti:

L’Analisi dei Requisiti costituisce il fondamento del progetto e consiste nell’individuazione e formalizzazione delle esigenze dell’utente finale relativamente alle funzionalità che il Software deve offrire. Questa attività si articola in:

- **Studio del capitolato:** analisi approfondita del documento di capitolato per comprendere obiettivi, vincoli e aspettative della Proponente;
- **Identificazione degli attori:** individuazione di tutti gli utenti che interagiranno con *Code Guardian*;
- **Definizione dei casi d’uso:** modellazione delle interazioni tra attori e sistema attraverso diagrammi UML;
- **Classificazione dei requisiti:** suddivisione in requisiti funzionali, di qualità e di vincolo, con assegnazione di priorità;

2.2.1.3 Progettazione dell’architettura di sistema:

Questa attività identifica le componenti *Hardware* e *Software* necessarie a soddisfare i requisiti, garantendo la tracciabilità degli stessi. La progettazione architettonica garantisce separazione delle responsabilità, scalabilità e manutenibilità del sistema.

2.2.1.4 Analisi dei Requisiti Software:

Questa attività approfondisce come il Software soddisfa concretamente i requisiti utente identificati, analizzando:

- **Requisiti funzionali:** definizione dettagliata delle funzionalità che il sistema deve offrire;
- **Requisiti prestazionali:** identificazione degli eventuali vincoli di performance;
- **Requisiti di interfaccia:** specifica delle interfacce di ogni elemento Software;
- **Requisiti di sicurezza.**

2.2.1.5 Progettazione dell’architettura Software:

Essa definisce la struttura portante del sistema e le interazioni tra le sue componenti, focalizzandosi sull’organizzazione d’insieme, senza soffermarsi sui dettagli della futura implementazione del codice.

2.2.1.6 Progettazione di dettaglio:

La progettazione di dettaglio specifica le singole unità Software, definendo:

- **Classi e interfacce:** specifica dettagliata di attributi, metodi e relazioni;
- **Strutture dati:** scelta delle strutture dati appropriate per ciascuna funzionalità;
- **Algoritmi:** definizione degli algoritmi principali con analisi di complessità quando rilevante;
- **Gestione degli errori:** strategia di gestione delle eccezioni a livello di singola unità;

2.2.1.7 Codifica e testing del Software:

Questa attività consiste nell'implementazione effettiva del codice sorgente seguendo le specifiche di dettaglio e nel testing delle singole unità Software, così da verificare il corretto funzionamento di ciascuna.

2.2.1.8 Integrazione Software:

L'integrazione Software combina progressivamente le unità sviluppate in componenti più complessi.

L'integrazione può seguire strategie:

- **Bottom-up:** partendo dalle unità di base verso quelle di livello superiore;
- **Top-down:** partendo dalle componenti di alto livello verso quelle di dettaglio;
- **Incrementale:** integrando progressivamente nuove unità e verificando ad ogni passo.

Per ogni integrazione vengono eseguiti test specifici che verificano la corretta interazione tra le componenti integrate.

2.2.1.9 Test di qualifica del Software:

I test di qualifica verificano che il Software soddisfi gli obiettivi di qualità definiti nel Piano di Qualifica.

2.2.1.10 Integrazione e Test di sistema:

Il sistema completo viene assemblato integrando tutte le componenti sviluppate e viene sottoposto a test che verificano:

- **Scenari d'uso reali:** simulazione di casi d'uso completi dalla richiesta dell'utente alla generazione del risultato;
- **Interazione tra componenti:** verifica che tutte le componenti collaborino correttamente;
- **Robustezza.**

2.2.1.11 Installazione

L'installazione consiste nel rilascio del Software nell'ambiente di destinazione concordato con il cliente.

2.2.1.12 Supporto all'accettazione:

Il gruppo fornisce supporto alla Proponente durante la fase di accettazione del prodotto, assistendolo nella:

- Verifica che tutte le funzionalità richieste siano implementate;
- Esecuzione dei test di accettazione definiti;
- Risoluzione di eventuali problematiche riscontrate.

2.2.2 Relazione con le Baseline di progetto

Le attività del Processo di Sviluppo sono distribuite tra le due baseline principali del progetto in modo da garantire un avanzamento progressivo e controllato.

2.2.2.1 Requirements and Technology Baseline (RTB)

La RTB^G si concentra sulla definizione completa dei requisiti e sulla validazione delle tecnologie scelte. Le attività che il gruppo ritiene fondamentali sono:

- **Analisi dei Requisiti:** attività cardine della RTB;
- **Proof of Concept (PoC);**
- **Definizione degli Standard:** consolidamento delle norme di codifica, delle convenzioni e degli strumenti di sviluppo documentati nelle presenti Norme di Progetto.

2.2.2 Product Baseline (PB)

La PB^G si concentra sulla realizzazione completa del prodotto Software. Le attività che il team ritiene fondamentali sono:

- **Progettazione dell'Architettura;**
- **Progettazione di Dettaglio;**
- **Codifica;**
- **Consolidamento del Prodotto:** finalizzazione del prodotto con tutte le funzionalità implementate, testate e documentate, pronto per la consegna finale e l'accettazione da parte della Proponente.

Le sezioni relative a tali attività verranno aggiornate nel momento opportuno.

2.2.3 Codifica

La codifica ha come obiettivo l'implementazione fedele delle soluzioni progettate, garantendo che ogni componente rispetti gli standard di qualità, sicurezza e manutenibilità definiti dal gruppo.

Il progetto adotta un'architettura full-stack e multi-linguaggio, i cui componenti principali sono:

- **TypeScript**^G: utilizzato per il frontend con React e per il backend con NestJS.
- **Python**^G: impiegato per la logica degli Agenti, l'Orchestratore e gli script di analisi dati.
- **MongoDB**^G: database utilizzato per la gestione flessibile dei dati tramite modelli definiti con Mongoose.
- **Node.js**^G: ambiente di esecuzione per i servizi server-side e gli strumenti di build.

Per la redazione della documentazione tecnica viene utilizzato il linguaggio di markup Typst^G.

2.2.3.1 Convenzioni

Per garantire la massima coerenza stilistica e facilitare il lavoro collaborativo, Skarab Group adotta le seguenti convenzioni:

- **Nomenclatura:**
 - **Variabili e Funzioni:** Si utilizza il camelCase.
 - **Componenti React e Classi:** Si utilizza il PascalCase.
 - **Interfacce e Tipi:** Si utilizza il PascalCase con eventuale prefisso I.
 - **File di logica:** Si utilizza il kebab-case.
 - **Costanti e Variabili d'Ambiente:** Si utilizza lo SCREAMING_SNAKE_CASE.
 - **Collezioni MongoDB:** Si utilizzano nomi in minuscolo e al plurale (es: users, reports).
- **Struttura e Organizzazione del Codice:**
 - **Modularità:** Ogni funzionalità deve essere isolata in moduli (NestJS) o componenti atomici (React).
 - **Separazione dei Concern:** La logica di business deve essere separata dall'interfaccia utente e dall'accesso ai dati.
- **Commenti e Documentazione:**
 - È obbligatorio indicare la descrizione, i parametri e il tipo di ritorno di ogni funzione.
- **Standard di Qualità e Stile:**
 - **Single Responsibility:** Ogni funzione deve svolgere un'unica attività.

- **Automazione:** L'uso di Prettier (indentazione 2 spazi, apici singoli) e ESLint è obbligatorio e integrato nella pipeline di sviluppo.
- **Funzioni Lambda:** Le arrow functions sono incoraggiate per callback brevi, ma la logica complessa deve essere estratta in metodi nominati per favorire il debugging.
- **Gestione Errori e Robustezza:**
 - **Eccezioni:** Ogni chiamata asincrona deve essere protetta da blocchi `try-catch` con logging specifico degli errori per prevenire il crash dei servizi.

Nota di avanzamento: Si prevede che questa sezione venga approfondita ed estesa durante la fase di sviluppo verso la *Product Baseline* (PB), per includere specificità tecniche necessarie al completamento del *Minimum Viable Product*.

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*.

3.1.1 Ciclo di Vita dei Documenti

Ogni documento segue obbligatoriamente 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 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.

3.1.2 Strumenti di Redazione

- **Visual Studio Code** ^G: editor principale per la scrittura dei documenti.
- **Typst** ^G: linguaggio di markup utilizzato per la formattazione di tutti i documenti di progetto. Consente la creazione di documenti strutturati, la gestione di tabelle, figure, riferimenti incrociati e la definizione di funzioni riutilizzabili. I template comuni sono centralizzati nella cartella `lib/` del repository.
- **Tinymist** ^G: estensione per VS Code che fornisce anteprima in tempo reale e supporto per la scrittura in Typst.

3.1.3 Struttura dei Documenti

Tutti i documenti ufficiali devono 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.

3.1.4 Gestione della Configurazione e Versionamento

3.1.4.1 Sistema di Versioning

Ogni documento adotta il seguente schema di versionamento semantico:

x.y.z

dove:

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

Il changelog deve essere aggiornato ad ogni incremento di versione.

3.1.4.2 Branching Strategy

La gestione del codice sorgente segue la seguente struttura di branch:

- main: branch stabile, contenente esclusivamente documenti approvati o verificati.
- develop: branch di integrazione principale, protetto – ogni modifica richiede Pull Request con revisione obbligatoria da parte di almeno un membro diverso dal redattore.
- feature/nome-feature: branch per lo sviluppo di nuove funzionalità software.
- fix/nome-fix: branch per la correzione di bug.
- acronimo-documento (es. NdP, PdP, AdR): branch dedicati alla redazione o aggiornamento di specifici documenti.

Regola fondamentale: il branch `main` contiene esclusivamente artefatti verificati. Il branch `develop` è protetto e richiede Pull Request approvata da un Verificatore diverso dal redattore prima di qualsiasi integrazione.

3.1.4.3 Denominazione dei File

La denominazione dei file sorgente Typst segue lo schema:

SIGLA.typ

dove SIGLA identifica il tipo di documento (es. NdP, PdP, PdQ, AdR).

Le date all'interno dei documenti sono scritte nel formato AAAA/MM/GG.

3.1.4.4 Repository GitHub - Struttura delle Cartelle

La repository DocumentazioneProgetto è organizzata come segue:

- `.github/`: Cartella contenente le configurazioni specifiche di GitHub.
- `assets/`: risorse multimediali utilizzate nei documenti.
- `diarioDiBordo/`: diari di bordo settimanali.
- `Glossario/`: sorgenti e infrastruttura per la versione web del glossario.
- `lib/`: librerie di funzioni Typst riutilizzabili tra i vari documenti.
- `RTB/`: documenti relativi alla **Requirements and Technology Baseline**.
- `verbali/`: verbali di riunioni interne ed esterne.
- `website/`: sorgenti del sito web di Skarab Group.

Nella root della repository si trovano inoltre:

- `README.md`: presentazione del progetto e guida all'uso del repository.
- `glossario.html`: glossario in formato web, generato automaticamente.
- `index.html`: homepage del sito web di Skarab Group.

3.1.4.4.1 Librerie Typst (`lib/`)

La cartella `lib/` contiene i file di funzioni Typst condivise tra i documenti. La gestione di questa cartella è responsabilità dell'Amministratore di Progetto, che garantisce coerenza, aggiornamento e documentazione delle funzioni disponibili. I documenti importano le librerie necessarie tramite `#import ".../lib/..."`.

3.1.5 Tracciabilità delle Attività Documentali

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, avviando lo Sprint successivo con una base documentale aggiornata.

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. Tale processo si articola nelle attività di *Verifica* e *Validazione*, complementari ma distinte negli obiettivi.

La Qualifica risponde a due domande 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 Verifica

Il processo di Verifica ha come scopo accertare che quanto prodotto sia stato realizzato correttamente, in conformità con i requisiti e gli standard definiti.

La Verifica è un’attività continua che accompagna l’intero ciclo di vita del progetto e si applica sia alla documentazione che al codice prodotto.

Tutte le informazioni relative agli esiti di tale attività, incluse misurazioni e risultati dei test, devono essere documentate nel Piano di Qualifica.

3.2.1.1 Obiettivi della Verifica

Skarab Group identifica i seguenti obiettivi per il processo di Verifica:

- garantire che ogni artefatto prodotto sia conforme alle Norme di Progetto;
- identificare e correggere tempestivamente difetti e anomalie;
- fornire evidenze oggettive della qualità del prodotto;
- supportare il miglioramento continuo dei processi attraverso il ciclo PDCA.

3.2.1.2 Strumenti a supporto

Questa sezione verrà ampliata durante lo svolgimento della *Product Baseline* (PB).

Per il calcolo dell’indice di Gulpease è stato utilizzato il servizio farfalla-project.org/readability_static/.

3.2.1.3 Attività proposte

In conformità allo standard ISO/IEC/IECC 12207, il processo di Verifica si articola nelle seguenti attività:

- **Implementazione del processo:** definizione delle criticità da analizzare, delle metodologie di verifica e degli strumenti da utilizzare.
- **Attività di Verifica:** 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.

3.2.1.4 Implementazione del processo

Skarab Group ha stabilito che la principale necessità del progetto *Code Guardian* è assicurare che il prodotto soddisfi le esigenze di **Var Group** attraverso verifiche sistematiche e rigorose.

Per questo motivo, nessun documento o componente software può essere integrato nel branch principale del repository senza aver superato il processo di verifica.

Regola fondamentale: il branch `main` contiene esclusivamente artefatti verificati e di qualità accertata.

Le verifiche sono condotte attraverso il meccanismo delle *Pull Request* su GitHub, che garantisce:

- tracciabilità delle modifiche;
- revisione da parte di un Verificatore;
- discussione strutturata delle criticità;
- approvazione formale prima dell'integrazione.

Eventuali problematiche complesse non risolvibili singolarmente sono discusse in sede di riunione interna e documentate nei verbali.

3.2.1.5 Attività di Verifica

3.2.1.5.1 Verifica della Documentazione

Nella prima fase del progetto, il processo di verifica si è focalizzato sulla qualità della **Documentazione** prodotta, adottando un approccio di revisione formale sistematica. Ogni documento, a seguito della stesura, segue un iter di approvazione obbligatorio prima di essere integrato nel *main branch* del repository.

L'attività si articola nei seguenti controlli:

- **Analisi Linguistica:** verifica della correttezza grammaticale, sintattica e dell'ortografia, con particolare attenzione all'indice di leggibilità per garantire la fruibilità dei contenuti.
- **Ispezione dei Contenuti:** verifica di esaustività, consistenza interna e coerenza rispetto agli obiettivi di fase.
- **Conformità Normativa:** accertamento del rispetto dei template e delle convenzioni grafiche definite internamente.
- **Gestione del Flusso:** le procedure operative di redazione, revisione e approvazione sono vincolate al sistema di *versioning*, seguendo le policy di gestione dei branch e delle issue per garantire la tracciabilità delle modifiche.

3.2.1.5.2 Verifica del Codice

Le metodologie di verifica del software verranno formalizzate e applicate in modo estensivo non appena raggiunta la *Requirements and Technology Baseline* (RTB).

In generale, la strategia di verifica adottata prevede l'integrazione di due approcci metodologici complementari: **Analisi Statica** e **Analisi Dinamica**.

3.2.1.5.2.1 Analisi Statica

L'**Analisi Statica** comprende tutte le attività di verifica che non richiedono l'esecuzione dell'oggetto analizzato.

Si concentra sull'individuazione di errori sintattici, strutturali o concettuali attraverso l'ispezione del codice sorgente o della documentazione.

3.2.1.5.2.1.1 Tecniche di Analisi Statica

L'Analisi Statica può essere realizzata mediante:

Metodi Formali: basati su dimostrazioni matematiche della correttezza. Data la complessità e il costo elevato, Skarab Group riserva questi metodi solo a componenti critiche del sistema *Code Guardian*, dove è richiesta garanzia assoluta di correttezza.

Metodi di Lettura:

I metodi di lettura si suddividono in:

- **Walkthrough:** analisi completa e approfondita dell'oggetto di verifica. Utile quando si è consapevoli dell'esistenza di un problema ma non si conosce natura o posizione. È un metodo costoso e difficilmente automatizzabile, pertanto viene applicato solo a componenti critiche o quando emergono problemi complessi non risolvibili con ispezione.
- **Ispezione:** si basa su checklist di controllo predefinite per identificare categorie note di errori. Meno esaustiva del walkthrough ma più efficiente e facilmente automatizzabile.

Data l'elevata quantità di artefatti da verificare nel progetto *Code Guardian*, Skarab Group privilegia l'**ispezione** in quanto:

- permette la creazione di checklist riutilizzabili e standardizzate;
- garantisce uniformità nei controlli;
- riduce i tempi di verifica mantenendo l'efficacia.

Il team applica l'analisi statica mediante:

- **Review manuale del codice.**
- **Strumenti automatici di linting:**
 - ESLint per codice JavaScript/TypeScript;
 - Pylint/Black per codice Python.

3.2.1.5.2.2 Analisi Dinamica

L'**Analisi Dinamica** prevede l'esecuzione dell'oggetto da verificare al fine di rilevare **failure** (comportamenti non conformi alle aspettative) e identificare i relativi **fault** (difetti nel codice che causano le failure).

L'Analisi Dinamica si realizza mediante **Test**, che devono possedere le seguenti caratteristiche:

- **Ripetibili:** eseguibili multiple volte con risultati consistenti;
- **Indipendenti:** non dipendenti dall'ordine di esecuzione o da altri test;
- **Automatizzabili:** eseguibili automaticamente senza intervento manuale.

L'automazione dei test è supportata da:

- **Driver:** componenti che invocano il codice da testare quando non direttamente eseguibile;
- **Stub:** componenti che simulano dipendenze non ancora implementate;
- **Logger:** strumenti per registrare gli esiti dei test in formato strutturato e analizzabile.

Le principali tipologie di Test sono:

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

La nomenclatura utilizzata per descrivere i Test, reperibili nel Piano di Qualifica, viene presentata di seguito.

3.2.1.5.2.2.1 Nomenclatura dei Test

La nomenclatura adottata per l'identificazione dei Test segue uno schema gerarchico:

[Prefisso] - [Numero]. [Sottonumero]

dove:

- **Prefisso:** identifica il tipo di test
 - **TU:** Test di Unità.
 - **TI:** Test di Integrazione.
 - **TS:** Test di Sistema.
 - **TR:** Test di Regressione.
 - **TA:** Test di Accettazione.
- **Numero:** identificativo progressivo univoco del test all'interno della categoria.
- **Sottonumero:** identifica varianti o sottocasi del test principale.

3.2.1.5.2.2.2 Test di Unità

I **Test di Unità** verificano il corretto funzionamento delle singole unità software, ovvero componenti atomiche con responsabilità ben definita e testabile in isolamento.

I test di unità si suddividono in:

- **Test funzionali (black-box):** verificano input e output senza considerare la logica interna. Per ogni unità vengono testati:
 - Valori non ammissibili inferiori al range valido;
 - Valori limite inferiori del range valido;
 - Valori centrali del range valido;
 - Valori limite superiori del range valido;
 - Valori non ammissibili superiori al range valido.
- **Test strutturali (white-box):** verificano i percorsi logici e la copertura delle istruzioni.

3.2.1.5.2.2.3 Test di Integrazione

I **Test di Integrazione** verificano la corretta interazione tra unità software integrate tra loro, assicurando che le interfacce e i flussi di dati funzionino correttamente.

Strategie di integrazione adottate:

- **Top-Down:** partendo dalle componenti di livello superiore che hanno maggiori dipendenze, ovvero quelle che hanno maggiore responsabilità verso l'esterno del Sistema;
- **Bottom-Up:** partendo dalle componenti di base con minori dipendenze, ovvero quelle con maggiore importanza all'interno del Sistema.

3.2.1.5.2.2.4 Test di Sistema

I Test di Sistema valutano il comportamento del sistema nel suo complesso, verificandone la conformità ai requisiti definiti nell'Analisi dei Requisiti.

3.2.1.5.2.2.5 Test di Regressione

Garantiscono che modifiche o correzioni al codice non introducano nuovi difetti o reintroducano difetti precedentemente risolti.

Procedura dei Test di Regressione:

1. Viene identificato e corretto un difetto nel codice;
2. Viene eseguito il test che aveva rilevato il difetto per verificarne la risoluzione;
3. Viene eseguita l'intera suite di test di regressione;
4. Si verifica che nessun test precedentemente superato sia ora fallito.

3.2.2 Validazione

Il processo di Validazione ha come scopo verificare che il prodotto *Code Guardian* soddisfi le esigenze di Var Group e degli utenti finali, garantendo che il sistema realizzato sia effettivamente quello richiesto.

Mentre la Verifica si concentra sulla conformità tecnica alle specifiche, la Validazione si focalizza sulla conformità del prodotto alle aspettative e ai bisogni reali degli *stakeholder*.

3.2.2.1 Obiettivi della Validazione

Skarab Group identifica i seguenti obiettivi per il processo di Validazione:

- confermare che il sistema *Code Guardian* implementi le funzionalità richieste da Var Group;
- verificare che il sistema sia utilizzabile dagli utenti finali;
- assicurare che il sistema risolva effettivamente i problemi di audit e remediation per cui è stato concepito;
- ottenere l'accettazione formale del prodotto da parte della Proponente;
- identificare eventuali gap tra le aspettative iniziali e la realizzazione finale.

3.2.2.2 Attività previste

In conformità allo standard ISO/IEC/IEEE 12207, il processo di Validazione si articola nelle seguenti attività:

- **Implementazione del processo:** definizione delle strategie di validazione e degli strumenti necessari.
- **Attività di Validazione:** esecuzione delle attività di validazione vera e propria.

3.2.2.2.1 Implementazione del processo

Skarab Group ha analizzato le esigenze di Var Group espresse nel capitolo C2 e ha racchiuso tutti i requisiti da soddisfare nell'**Analisi dei Requisiti**.

Per garantire una validazione efficace, il team adotta le seguenti strategie:

1. **Tracciamento sistematico dei requisiti:** ogni requisito è tracciato dalla sua definizione nell'Analisi dei Requisiti fino alla sua implementazione e validazione;
2. **Test di Accettazione:** rappresentano la verifica finale del prodotto rispetto alle aspettative degli *Stakeholder*. Essi confermano che tutte le funzionalità richieste siano implementate correttamente.

3.2.2.2.2 Attività di Validazione

Skarab Group realizza la Validazione attraverso un approccio strutturato che combina tracciamento e testing. Il **tracciamento sistematico dei requisiti** mantiene collegamenti esplicativi tra requisiti, implementazione e test, garantendo la copertura completa delle funzionalità. I **Test di Accettazione** simulano scenari d'uso reali, verificando criteri di successo oggettivi per ogni requisito.

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*, Skarab Group ha identificato i seguenti Processi Organizzativi rilevanti:

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

- Processo di Formazione.

4.1 Gestione dei Processi

Il processo di Gestione dei Processi ha l'obiettivo di definire e coordinare le attività del progetto, assegnare responsabilità, pianificare le risorse temporali ed economiche e garantire una comunicazione interna ed esterna efficace e continua.

4.1.1 Ruoli di Progetto

La seguente tabella riporta tutti i ruoli presenti all'interno del progetto *Code Guardian* con le relative responsabilità.

Ruolo	Responsabilità
Responsabile di Progetto	Coordina e supervisiona l'intero progetto garantendo il rispetto delle scadenze e del budget. Rappresenta Skarab Group nei rapporti con Var Group, con i docenti Prof. Cardin e Prof. Vardanega e con gli altri stakeholder esterni. Pianifica le attività su Jira, assegna le risorse ai membri del team, monitora l'avanzamento attraverso metriche e indicatori (EVM), gestisce i rischi e le criticità. Approva i documenti ufficiali e autorizza eventuali variazioni al piano. Convoca e presiede le riunioni interne; redige e invia le comunicazioni formali via email.
Amministratore	Gestisce l'infrastruttura tecnica e organizzativa del progetto. Configura e mantiene gli strumenti di versionamento (Git/GitHub), il sistema di issue tracking (Jira) e gli ambienti di sviluppo. Definisce il workflow di branching e le convenzioni di commit. Gestisce i template di documento, redige e mantiene aggiornate le Norme di Progetto. Garantisce backup e sicurezza dei dati e risolve tempestivamente problemi legati all'infrastruttura.
Verificatore	Assicura la qualità degli artefatti prodotti attraverso attività di verifica sistematica. Controlla la conformità dei documenti alle norme redazionali e tipografiche definite nelle NdP, la correttezza tecnica e la coerenza dei contenuti. Esegue review del codice sorgente secondo checklist predefinite, verifica il rispetto degli standard di codifica. Redige gli esiti di verifica nelle Pull Request su GitHub, segnala non conformità e propone azioni correttive. Monitora le metriche di qualità.
Analista	Raccoglie, analizza e formalizza i requisiti del sistema <i>Code Guardian</i> attraverso l'interazione con il committente e gli stakeholder. Identifica attori, casi d'uso, requisiti funzionali, di qualità e di vincolo, classificandoli secondo le norme definite nelle NdP. Redige e mantiene il documento Analisi dei Requisiti. Garantisce la tracciabilità tra requisiti e fonti. Collabora con i progettisti per verificare la fattibilità tecnica delle soluzioni proposte.
Progettista	Definisce l'architettura del sistema e il design dettagliato dei componenti software. Individua pattern architettonici appropriati, decompone il sistema in moduli, definisce interfacce e dipendenze tra componenti. Specifica la struttura delle classi, i diagrammi di sequenza e collaborazione, le strutture dati e gli algoritmi principali. Redige la Specifica Tecnica. Effettua scelte tecnologiche motivate, con attenzione a manutenibilità, estensibilità e

	sicurezza. Fornisce ai programmati le specifiche di dettaglio necessarie all'implementazione.
Programmatore	Implementa il codice seguendo le specifiche fornite dai progettisti e rispettando gli standard di codifica definiti nelle NdP. Scrive codice pulito, manutenibile e ben documentato. Sviluppa unit test per verificare la correttezza delle singole unità software, garantendo un'adeguata code coverage. Gestisce le dipendenze, utilizza strumenti di build automation e integra librerie esterne. Documenta il codice con commenti significativi e mantiene aggiornata la documentazione tecnica. Esegue debugging, risolve difetti segnalati e ottimizza le performance quando necessario. Collabora con verifieri e progettisti per migliorare la qualità del software.

Table 9: Ruoli di progetto e relative responsabilità

4.1.2 Rotazione dei Ruoli

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

4.1.3 Gestione dei Task e Pianificazione Operativa

4.1.3.1 Strumento di Issue Tracking: Jira

La gestione operativa dei task avviene esclusivamente tramite Jira^G, che costituisce la fonte ufficiale di dati per il monitoraggio dell'avanzamento e per la rendicontazione delle ore. Ogni membro del gruppo ha accesso alle proprie attività assegnate.

Ogni task su Jira deve includere obbligatoriamente:

- descrizione chiara e univoca dell'attività;
- assegnatario responsabile;
- stima delle ore previste;
- scadenza temporale;
- collegamento allo Sprint di riferimento.

I task devono essere definiti in modo **atomico** (eseguibili da un singolo membro, senza dipendenze bloccanti non esplicitate).

L'aggiornamento dello stato dei task (ore rendicontate, avanzamento) è responsabilità individuale di ciascun membro, il quale deve segnalare eventuali impedimenti o ritardi.

4.1.4 Coordinamento: Riunioni e Comunicazioni

4.1.4.1 Riunioni Interne

Le riunioni interne si tengono con cadenza settimanale e hanno lo scopo di:

- condividere lo stato di avanzamento delle attività;
- pianificare le attività della settimana successiva;
- identificare e risolvere impedimenti o problemi aperti;
- gestire l'eventuale rotazione dei ruoli.

Le riunioni interne si svolgono in modalità virtuale tramite **Discord**^G e permettono ai membri del team di allinearsi sugli obiettivi comuni. Ogni riunione deve produrre un verbale interno, redatto dal Responsabile di Progetto.

4.1.4.2 Riunioni Esterne

Le riunioni esterne, organizzate dal Responsabile di Progetto, si tengono con il Proponente ^G Var Group, quando necessario per:

- presentare lo stato di avanzamento del progetto;
- discutere requisiti, feedback e modifiche;
- allineare le aspettative e pianificare le fasi successive.

Le comunicazioni per la convocazione avvengono tramite email del gruppo. Le riunioni si svolgono in presenza o in modalità virtuale secondo quanto concordato con il Proponente. Ogni riunione esterna deve produrre un verbale esterno, da condividere con Var Group.

4.1.4.3 Canali di Comunicazione

La comunicazione interna del team avviene attraverso le seguenti piattaforme, ognuna adottata per uno scopo specifico:

Strumento	Utilizzo	Tipo
WhatsApp ^G	Comunicazioni rapide e urgenti, aggiornamenti quotidiani di breve durata.	Asincrono
Telegram ^G	Discussioni più strutturate, condivisione di file e link, coordinamento di gruppo.	Asincrono
Discord ^G	Riunioni virtuali settimanali, sessioni di lavoro condivise, discussione di criticità complesse.	Sincrono
Email	Comunicazioni formali con Var Group, Prof. Cardin e Prof. Vardanega. Gestita dal Responsabile di Progetto.	Formale
Jira ^G	Assegnazione e tracciamento task, monitoraggio avanzamento Sprint, gestione delle scadenze.	Asincrono

Table 10: Canali di comunicazione del team

4.2 Infrastruttura

Il Processo di Infrastruttura è responsabile della creazione, configurazione e manutenzione di tutti gli strumenti e gli ambienti necessari allo svolgimento delle attività di progetto. La gestione dell'infrastruttura è di competenza dell'Amministratore di Progetto.

4.2.1 Strumenti Adottati

Il gruppo utilizza diversi strumenti per supportare l'organizzazione del lavoro. Oltre a quelli già citati, vengono utilizzati i seguenti:

Strumento	Utilizzo nel progetto
Git ^G	Sistema di controllo versione distribuito per la gestione del codice sorgente e della documentazione. Strumento base per il versionamento di tutti gli artefatti del progetto.
GitHub	Hosting remoto dei repository Git. Utilizzato per la sincronizzazione del lavoro tra i membri, la gestione delle Pull Request, la code review e le issue.

Google Calendar	Calendario condiviso per la pianificazione e il promemoria delle riunioni esterne con Var Group e delle scadenze di progetto. Gestito dal Responsabile.
Script di automazione	Script sviluppati internamente per automatizzare attività ripetitive: generazione del Glossario (formati PDF e HTML), compilazione dei documenti Typst, aggiornamento del sito web del gruppo.

Table 11: Strumenti dell’infrastruttura di progetto

4.2.2 Configurazione dei Principali Strumenti

4.2.2.1 GitHub - Struttura del Repository

Il repository DocumentazioneProgetto è strutturato secondo quanto descritto nella sezione **Documentazione** del presente documento. L’Amministratore è responsabile di:

- definire e applicare le regole di protezione dei branch (`develop` e `main`);
- configurare le GitHub Actions per il deploy del sito web;

4.2.2.2 Script di Automazione

Gli script di automazione attualmente in uso includono:

- **Generazione Glossario**: genera automaticamente il file `glossario.html` e il PDF del Glossario a partire dal sorgente Typst.
- **Compilazione documenti**: compila tutti i sorgenti Typst modificati e aggiorna il sito web del gruppo sul branch `main`.

L’Amministratore è responsabile dello sviluppo, del mantenimento e della documentazione di tali script.

4.2.3 Manutenzione dell’Infrastruttura

L’Amministratore monitora il corretto funzionamento di tutti gli strumenti e si impegna ad aggiornarne le configurazioni qualora emergano malfunzionamenti o necessità evolutive. Eventuali problemi bloccanti sull’infrastruttura devono essere segnalati al Responsabile di Progetto e tracciati su Jira.

4.3 Processo di Miglioramento

Il processo di miglioramento è fondamentale per garantire che i prodotti del progetto soddisfino gli standard di qualità definiti e le aspettative degli stakeholder.

4.3.1 Attività Previste

Il processo si articola in tre fasi:

1. **Inizializzazione**: definizione dei processi organizzativi che guidano il lavoro del team, documentati nelle presenti Norme di Progetto.
2. **Valutazione**: monitoraggio continuo dei processi attraverso le metriche definite nel *Piano di Qualifica*. La valutazione avviene con cadenza di Sprint durante la retrospettiva.
3. **Miglioramento**: identificazione delle criticità, definizione di azioni correttive o preventive e aggiornamento delle Norme di Progetto e del Piano di Qualifica per recepire i miglioramenti.

4.3.2 Procedura di Miglioramento

Al termine di ogni Sprint, durante la retrospettiva, il team:

- analizza i valori delle metriche di processo rispetto alle soglie definite nel PdQ;
- identifica processi o attività che hanno mostrato inefficienze o non conformità;

- definisce azioni correttive concrete con responsabile e scadenza, tracciate su Jira;
- aggiorna le NdP o il PdQ per recepire i cambiamenti approvati.

Le modifiche alle Norme di Progetto derivanti da azioni di miglioramento devono essere approvate dal Responsabile di Progetto.

4.4 Processo di formazione

Il processo di formazione ha lo scopo di garantire che tutti i membri del team abbiano le competenze necessarie per svolgere efficacemente i propri ruoli all'interno del progetto.

Questo processo include:

- identificazione delle competenze chiave richieste per ogni ruolo.
- pianificazione di sessioni di formazione e workshop.
- monitoraggio del progresso formativo e valutazione delle competenze acquisite.

4.4.1 Competenze Richieste

A seguito dell'analisi del Capitolato C2 e delle tecnologie scelte, le competenze identificate come necessarie al completamento del progetto sono:

Tecnologia / Ambito	Fonte di formazione
TypeScript ^G e React	Documentazione ufficiale TypeScript e React.
NestJS	Documentazione ufficiale NestJS; guide e tutorial specifici sul framework.
Python ^G (Agenti AI / Orchestratore)	Documentazione ufficiale Python; documentazione dei framework per agenti AI; documentazione AWS Bedrock/OpenAI per LLM.
MongoDB ^G e Mongoose	Documentazione ufficiale MongoDB e Mongoose.
Node.js ^G	Documentazione ufficiale Node.js; tutorial su piattaforme online.
Git ^G e GitHub	Documentazione ufficiale Git.
Typst ^G	Documentazione ufficiale Typst; template e funzioni del repository interno.
Agenti AI e LLM ^G	Documentazione framework specifici; documentazione Var Group fornita nelle sessioni di formazione dedicate.
AWS	Documentazione ufficiale AWS; materiale fornito da Var Group durante le sessioni di formazione.

Table 12: Tecnologie e fonti di formazione

4.4.2 Supporto del PropONENTE

Var Group ha messo a disposizione del team sessioni di formazione dedicate sulle tecnologie necessarie allo sviluppo del progetto e rimane disponibile per la risoluzione di dubbi e domande tramite i canali di comunicazione concordati.

4.4.3 Nota sulle Ore di Formazione

Le ore dedicate alla formazione personale **non sono rendicontabili** nel monte ore di progetto, in quanto rientrano nella parte di apprendimento individuale prevista dal corso universitario.

4.4.4 Spazi per la Formazione

Nel corso degli Sprint, i membri del team potranno dedicare tempo alla formazione personale nelle fasi di attività a peso minore, senza che ciò comprometta il rispetto degli obiettivi di Sprint. Il Responsabile di Progetto può, qualora lo ritenga necessario, pianificare periodi dedicati esplicitamente all’approfondimento tecnologico, specialmente in preparazione alle fasi di progettazione e sviluppo verso la *Product Baseline* (PB).