Università degli Studi di Padova
Laurea: Informatica
Corso: Ingegneria del Software
Anno Accademico: 2025/2026

Gruppo: NullPointers Group
Email: nullpointersg@gmail.com

Norme di Progetto

Stato	In Approvazione
Versione	0.8.0
Data ultima modifica	13/12/2025
Redattori	Lisa Casagrande Matteo Mazzaretto Marco Brunello Laura Pieripolli Luca Marcuzzo
Verificatori	Matteo Mazzaretto Tommaso Ceron Marco Brunello Lisa Casagrande
Destinatari	NullPointers Group

Registro delle modifiche

Vers	Data	Autore	Verificatore	Descrizione
0.8.0	13-12	L. Casagrande	M. Mazzaretto	Continuazione sezione 3.3, iniziata sezione 3.4 e redazione sezioni 2.2.3, 3.1.8, 5.2 e 5.3
0.7.0	12-12	L. Marcuzzo	L. Casagrande	Redazione sezione 6 "Metriche di Qualità" e sezione 5.1. Iniziata sezione 3.3
0.6.0	09-12	L. Casagrande	M. Mazzaretto	Elaborate sezioni 3.2 "Gestione della Configurazione" e sezioni 4.1.3.1, 4.1.4 e 4.1.2.3. Redazione sezione 4.3 "Processo di Miglioramento" e 4.4 "Processo di Formazione"
0.5.0	30-11	M. Brunello	L. Casagrande	Redazione sezione 4.1 "Gestione dei processi" e 4.2 "Gestione dell'Infrastruttura"
0.4.0	30-11	L. Pieripolli	L. Casagrande	Continuazione sezione 2.2 "Sviluppo"
0.3.1	29-11	L. Casagrande	M. Mazzaretto	Creazione macro per virgolette
0.3.0	23-11	L. Casagrande	M. Brunello	Redazione sezione 2.1 "Fornitura" e inizio 2.2 "Sviluppo".
0.2.0	15-11	M. Mazzaretto	T. Ceron	Continuazione sezione 3.1 "Documentazione".
0.1.0	12-11	L. Casagrande	M. Mazzaretto	Creazione e stesura documento. Redazione sezione 1 "introduzione" e inizio sezione 3.1 "Documentazione"

Indice

Elenco delle Tabelle

1 Introduzione

1.1 Scopo del documento

Il presente documento nasce per descrivere il Way of Working^G adottato da **"NullPointers Group"** durante lo svolgimento del progetto SmartOrder.
Lo standard di riferimento è l'ISO/IEC 12207:1995, il quale prevede tre tipologie di processi.

- **Processi primari:** processi fondamentali senza i quali un progetto non può definirsi tale;
- **Processi di Supporto:** processi che coadiuvano i processi primari nello svolgimento delle rispettive azioni;
- **Processi organizzativi:** processi di carattere più generale che aiutano la realizzazione del progetto.

La stesura di questo documento è incrementale, cioè una stesura passo passo con modifiche, aggiunte e cancellazioni a seguito di miglioramenti del metodo di lavoro. I membri del gruppo si impegnano a visionare costantemente questo documento e a rispettare rigorosamente le regole definite in esso, per svolgere il progetto in modo professionale, coerente ed uniforme.

1.2 Scopo del prodotto

La gestione automatizzata^G degli ordini di acquisto in contesti multicanale rappresenta una sfida complessa per le aziende moderne, che devono affrontare la necessità di interpretare richieste provenienti da fonti eterogenee come email, chat, messaggi vocali e immagini.
Il capitolato^G numero C8 di Ergon Informatica propone di sviluppare una piattaforma intelligente in grado di analizzare input multimodali e convertirli automaticamente in ordini strutturati, pronti per l'inserimento nei sistemi gestionali aziendali.
L'obiettivo che si è posto questo gruppo è realizzare un sistema basato su architettura a microservizi che integri tecniche avanzate di intelligenza artificiale^G, machine learning e natural language processing, in grado di riconoscere le intenzioni del cliente, estrarre le informazioni rilevanti e validarle in maniera coerente con il catalogo prodotti aziendale. Questo approccio consentirà di ridurre drasticamente l'intervento umano nelle fasi ripetitive, migliorando al contempo l'efficienza^G complessiva e la soddisfazione del cliente finale.
Il progetto SmartOrder si propone quindi di dimostrare come le tecnologie di intelligenza artificiale^G possano essere applicate con successo a processi reali di business, trasformando un compito complesso e frammentato in un flusso lineare, automatizzato e scalabile. L'obiettivo è realizzare questo progetto entro il 30 Aprile 2026 con un budget a disposizione di: Euro 11.440.

1.3 Glossario

La realizzazione di un sistema software complesso come SmartOrder richiede, ancora prima della scrittura del codice, un'importante operazione di confronto, analisi e progettazione. Per supportare e facilitare il lavoro asincrono tra i membri del gruppo e garantire una comunicazione efficace con il committente, tutte le informazioni derivanti da questa attività saranno appositamente documentate in un glossario condiviso, utile per evitare ambiguità o incomprensioni riguardanti la nomenclatura adottata in tutti i documenti visionabili.
In accordo con quanto stabilito nel verbale interno del 6 novembre 2025, si è deciso di adottare il glossario come strumento ufficiale per la standardizzazione della terminologia di progetto e di assegnare la responsabilità della sua manutenzione^G alla figura dell'Analista.
La nomenclatura utilizzata per segnalare che la definizione di una parola è contenuta nel glossario è la seguente:

termine^G

I termini sono ordinati alfabeticamente per facilitarne la consultazione e vengono inclusi sia termini tecnici che acronimi significativi.
Il gruppo si impegna a visionare il Glossario periodicamente, per permettere la più completa comprensione di ogni tipo di documento pubblicato e per mantenere un allineamento semantico costante tra tutti i partecipanti al progetto.

1.4 Riferimenti

1.4.1 Riferimenti normativi

- **Capitolato^G C8 - Ergon Informatica Srl - SmartOrder**
  `https://www.math.unipd.it/~tullio/IS-1/2025/Progetto/C8.pdf`
  Ultima consultazione: 30 Novembre 2025

1.4.2 Riferimenti informativi

- **Standard ISO/IEC 9126**
  `https://en.wikipedia.org/wiki/ISO/IEC_9126`
  Ultima consultazione: 13 Dicembre 2025

- **Standard ISO/IEC/IEEE^G 12207:1995**
  `https://www.math.unipd.it/~tullio/IS-1/2009/Approfondimenti/ISO_12207-1995.pdf`
  Ultima consultazione: 12 Dicembre 2025

- **Glossario, versione 1.0.0**
  `https://nullpointersgroup.github.io/Documentazione/output/RTB/Documenti%20Interni/Glossario.pdf`
  Ultima consultazione: 13 Dicembre 2025

2 Processi Primari

La creazione di un software di qualità non si basa solo sulla scrittura del codice e su test^G isolati. Per realizzare un prodotto duraturo e ampiamente utilizzato, è essenziale adottare un modello di sviluppo^G che definisca processi chiari da seguire. Lo standard ISO/IEC 12207, a tal proposito, identifica tra i processi primari quelli di **Fornitura** e **Sviluppo**.

2.1 Fornitura

Il processo di Fornitura definisce le modalità con cui il gruppo si impegna a realizzare e consegnare il prodotto **SmartOrder** al proponente Ergon Informatica Srl, nel rispetto di tempi, costi, qualità e requisiti concordati.
Questo processo, che comprende tutte le attività di pianificazione, negoziazione, esecuzione, controllo e consegna, inizia con un'analisi preliminare dei requisiti per definire le funzionalità^G del sistema, sulla base della quale è possibile negoziare i dettagli con la proponente, presentare una pianificazione^G di progetto e stimare una data di consegna plausibile.

2.1.1 Attività

- **Fase iniziale**: definizione degli obiettivi del progetto, studio di fattibilità tecnica ed economica, identificazione dei rischi e delle risorse necessarie, analisi dei capitolati d'appalto.
- **Sviluppo^G della proposta progettuale**: stesura della documentazione progettuale completa, inclusi Lettera di Presentazione, Preventivo dei Costi e Valutazione dei Capitolati.
- **Consolidamento dei requisiti**: negoziazione con la proponente su requisiti, tempi, costi, modalità di collaborazione e criteri di accettazione del prodotto finale.
- **Pianificazione^G operativa**: definizione del Piano di Progetto^G dettagliato, assegnazione delle risorse, definizione delle milestones e delle scadenze critiche.
- **Implementazione e monitoraggio**: monitoraggio dell'avanzamento, gestione delle modifiche e reporting periodico al proponente.
- **Analisi dei risultati**: verifica^G periodica dello stato del progetto rispetto agli obiettivi prefissati, analisi degli scostamenti e valutazione delle performance del team.
- **Rilascio del prodotto**: release del prodotto conforme ai requisiti, documentazione finale completa, chiusura formale del progetto.

2.1.2 Strumenti a supporto

Per la comunicazione interna al gruppo abbiamo deciso di utilizzare:

- **GitHub**: per il versioning del codice e della documentazione, nonché per la gestione del backlog e del sistema di ticketing, strumenti essenziali per monitorare lo stato di avanzamento e le attività da completare;
- **Discord**: per incontri a distanza;
- **WhatsApp**: per la comunicazione asincrona del team.

Mentre per la comunicazione con l'azienda proponente abbiamo concordato l'utilizzo di:

- **Google Mail**: per la comunicazione formale con l'Azienda;
- **Google Meet**: per le riunioni periodiche e il coordinamento;
- **WhatsApp**: per la comunicazione asincrona tra il team e la proponente.

2.1.3 Documentazione fornita

NullPointers Group, in linea con gli obiettivi del progetto, si impegna a fornire a Ergon Informatica Srl e ai committenti tutta la documentazione relativa al progetto:

2.1.3.1 Lettera di Presentazione

Obiettivo del documento:
Il documento si articola in tre tipologie:

- La Lettera di Presentazione per la candidatura al progetto;
- La Lettera di Presentazione per la Requirements and Technology Baseline^G (RTB)^G;

Ciascuna lettera ha lo scopo di presentare formalmente il gruppo NullPointers Group, illustrando l'impegno nel rispettare gli obiettivi e i vincoli delle rispettive Baseline^G, fornendo una panoramica di competenze, risorse e metodologie adottate durante lo sviluppo^G del progetto.

Redattore: Responsabile

Utilizzo: Esterno

Destinatari: - Prof. Tullio Vardanega
- Prof. Riccardo Cardin

2.1.3.2 Preventivo dei Costi

Obiettivo del documento:
Il Preventivo Costi serve a mostrare come NullPointers Group ha pianificato il progetto, stimando ore e costi per ciascun ruolo e definendo le responsabilità dei membri del team.

Redattore: Non Determinato

Utilizzo: Esterno

Destinatari: - NullPointers Group
- Prof. Tullio Vardanega
- Prof. Riccardo Cardin

2.1.3.3 Valutazione dei Capitolati

Obiettivo del documento:
La Valutazione dei Capitolati è un documento redatto da NullPointer Group che analizza ogni capitolato^G proposto, individuando punti di forza e criticità. Ogni analisi comprende la descrizione del capitolato^G (informazioni sull'azienda proponente e sul prodotto da sviluppare), una descrizione dei punti di forza e delle criticità riscontrate, derivanti dalla sua realizzazione. Con riguardo al primo capitolato^G e ai due successivi di interesse, è stato inoltre descritto lo stack tecnologico previsto per la loro realizzazione.

Redattore: Non Determinato

Utilizzo: Esterno

Destinatari: - NullPointers Group
- Prof. Tullio Vardanega
- Prof. Riccardo Cardin

2.1.3.4 Verbali Esterni

Obiettivo del documento:
Il Verbale Esterno è un documento redatto dal gruppo in occasione delle riunioni che coinvolgono soggetti esterni, come i referenti dell'azienda proponente.
Riporta in modo strutturato tutti i contenuti discussi durante l'incontro, incluse richieste, chiarimenti, decisioni, vincoli, feedback e indicazioni fornite dai referenti esterni. Il suo scopo è garantire trasparenza, tracciabilità e un allineamento costante tra il team e la proponente, riducendo il rischio di fraintendimenti o interpretazioni errate.

Redattore: Responsabile

Utilizzo: Esterno

Destinatari: - NullPointers Group
- Ergon Informatica Srl

2.1.3.5 Verbali Interni

Obiettivo del documento:
Il Verbale Interno è un documento redatto dal gruppo in occasione delle riunioni svolte esclusivamente tra i membri del team.
Ha lo scopo di registrare tutte le decisioni prese, le attività pianificate e eventuali problematiche affrontate durante gli incontri interni.
Il documento permette di mantenere una traccia chiara e condivisa dell'avanzamento del lavoro, favorendo il coordinamento tra i membri del gruppo e garantendo continuità anche quando le attività vengono svolte in modo asincrono.

Redattore: Responsabile

Utilizzo: Interno

Destinatari: - NullPointers Group

2.1.3.6 Analisi dei Requisiti

Obiettivo del documento:
L'Analisi dei Requisiti ha lo scopo di definire in modo chiaro e completo le funzionalità, i vincoli e le caratteristiche attese dal sistema SmartOrder. Questo documento costituisce un riferimento univoco per tutto il team, riducendo ambiguità e incomprensioni durante le fasi di progettazione^G e sviluppo.
Nel documento sono individuati gli attori principali e secondari, insieme ai principali casi d'uso.
I requisiti sono organizzati in funzionali, non funzionali e di vincolo, e distinti per priorità in obbligatori, desiderabili e opzionali, fornendo una guida chiara per sviluppo, progettazione^G e verifica.

Redattore: Analista

Utilizzo: Esterno

Destinatari: - NullPointers Group
- Ergon Informatica Srl
- Prof. Tullio Vardanega
- Prof. Riccardo Cardin

2.1.3.7 Glossario

Obiettivo del documento:
Il Glossario definisce in modo univoco termini tecnici, acronimi e concetti rilevanti del progetto SmartOrder, è ritenuto dal team uno strumento fondamentale in quanto permette di standardizzare la terminologia e facilitare la comunicazione tra tutti i soggetti coinvolti.
Le voci sono ordinate alfabeticamente e, quando citate nei documenti, riportano l'apice^G.

Redattore: Analista

Utilizzo: Interno

Destinatari: - NullPointers Group

2.1.3.8 Norme di Progetto

Le Norme di Progetto, ovvero il presente documento, definiscono il Way of Working^G adottato dal team NullPointers Group per lo sviluppo^G del progetto SmartOrder.
Mirano a stabilire regole, metodologie e standard per garantire coerenza, qualità e uniformità nella produzione^G di documenti, codice e artefatti di progetto.
Le norme disciplinano la gestione e il versionamento dei file, la comunicazione

interna ed esterna, il tracciamento dei requisiti e delle decisioni, nonché le procedure di revisione e approvazione.
Il rispetto di queste regole è obbligatorio per tutti i membri del gruppo e rappresenta un riferimento costante per l'esecuzione delle attività di progetto.

Redattore: Amministratore

Utilizzo: Interno

Destinatari: - NullPointers Group

2.1.3.9 Piano di Progetto

Il Piano di Progetto definisce l'organizzazione, le attività, le risorse e i tempi necessari per lo sviluppo^G del progetto da parte del team.
Per ogni sprint^G è inclusa una tabella con i preventivi e i ruoli assunti da ciascun componente.
Vengono inoltre forniti: l'analisi dei rischi, le scadenze e la stima di tempi e costi, con l'aggiornamento del consumo orario e dei costi previsti ed effettivi.
Rappresenta uno strumento operativo fondamentale per coordinare il team, gestire le risorse, tracciare le decisioni e verificare la conformità agli standard di qualità definiti nelle Norme di Progetto, garantendo il rispetto di obiettivi, requisiti e vincoli del progetto.

Redattore: Responsabile

Utilizzo: Esterno

Destinatari: - NullPointers Group
- Ergon Informatica Srl
- Prof. Tullio Vardanega
- Prof. Riccardo Cardin

2.1.3.10 Piano di Qualifica

Il Piano di Qualifica definisce strategie, criteri e procedure per verificare e validare il sistema SmartOrder, garantendo il rispetto dei requisiti funzionali, non funzionali e di vincolo.
Include la pianificazione^G delle attività di test, la descrizione dei casi di verifica, i criteri di accettazione e le modalità di tracciamento delle anomalie.
Rappresenta dunque un riferimento operativo per assicurare la qualità del prodotto, monitorare l'avanzamento dei test^G e supportare decisioni sul rilascio delle componenti del sistema.

Redattore: Responsabile

Utilizzo: Esterno

Destinatari: - NullPointers Group
- Ergon Informatica Srl
- Prof. Tullio Vardanega
- Prof. Riccardo Cardin

2.2 Sviluppo

Il processo di sviluppo^G è un insieme strutturato di attività che guidano la realizzazione del software, dalla definizione dei requisiti fino al rilascio del prodotto finale.
Comprende fasi di analisi, progettazione, implementazione, integrazione, test^G e validazione, seguendo un approccio iterativo e modulare, con ciascuna parte del sistema sviluppata, testata e integrata passo dopo passo.
Lo scopo del processo è garantire che il prodotto soddisfi i requisiti, ridurre i rischi, mantenere la qualità e la tracciabilità delle attività, e assicurare che ogni rilascio sia coerente con gli obiettivi concordati con il committente.

2.2.1 Attività

Il progetto SmartOrder è stato sviluppato seguendo un approccio incrementale e agile, garantendo al contempo la conformità agli standard internazionali ISO/IEC 12207:1995.
Questo approccio permette di mantenere tracciabilità, qualità e gestione efficace delle risorse durante tutte le fasi del progetto.
Le attività di sviluppo^G comprendono:

- **Raccolta e Analisi dei Requisiti**: individuazione delle esigenze della proponente e definizione dei casi d'uso, con distinzione tra requisiti funzionali, non funzionali e di vincolo.
- **Progettazione^G dell'Architettura**: definizione della struttura del sistema, dei componenti principali e delle loro interazioni.
- **Sviluppo^G Incrementale**: implementazione delle funzionalità^G in moduli indipendenti, ciascuno verificato prima dell'integrazione.
- **Monitoraggio e Revisione**: attività continua durante tutto il progetto, dedicata al controllo dell'avanzamento, alla gestione delle modifiche e alla verifica^G della qualità.
- **Integrazione e Test**: unione dei moduli sviluppati e verifica^G del corretto funzionamento complessivo tramite test^G di sistema.
- **Rilascio e Distribuzione**: consegna delle funzionalità^G approvate e della documentazione associata.

Queste attività sono cicliche e iterativamente aggiornate ad ogni sprint, permettendo un progresso costante e la tracciabilità di tutte le decisioni.

2.2.2 Analisi dei Requisiti

L'analisi dei requisiti^G è un processo che ha lo scopo di identificare in maniera univoca tutte le esigenze funzionali ed i requisiti che il sistema software dovrà soddisfare.
Il risultato di questo processo viene formalizzato nell'apposito documento di

Analisi dei Requisiti il quale costituisce la base per le successive fasi di progettazione, codifica e test. Espone nel dettaglio i casi d'uso e i requisiti che compongono gli elementi fondamentali per lo sviluppo^G del progetto.

2.2.2.1 Casi d'uso

Per la definizione e codifica dei casi d'uso è stata adottata la seguente nomenclatura:

UC_Principale.Secondario

dove:

- **UC** è l'acronimo di Use Case (Caso d'Uso);
- **Principale** è un numero intero, incrementale che identifica univocamente il caso d'uso principale;
- **Secondario** è un numero intero, incrementale che identifica un caso d'uso derivato dal caso d'uso principale.

Nel caso in cui un caso d'uso non presenti sotto-casi verrà utilizzato il formato:

UC_Principale

I casi d'uso principali sono univoci all'interno del sistema: non possono esistere due casi d'uso con lo stesso valore Principale.
È invece possibile che casi d'uso secondari condividano il medesimo indice Secondario, purché appartengano a casi d'uso principali differenti.
Questa convenzione ha lo scopo di garantire l'univocità e la tracciabilità di ciascun Caso d'Uso. Ad ogni identificatore inoltre è associato un nome ed una breve descrizione che riassume in maniera concisa lo scopo del caso d'uso.

2.2.2.2 Requisiti

Per la definizione e la codifica dei Requisiti è stata adottata la seguente nomenclatura:

RTipologia-Priorità_#

dove:

- **R** è l'abbreviazione di Requisito;
- **Tipologia** identifica la tipologia del requisito, le possibili tipologie sono:
  - **F**: Requisito Funzionale;
  - **N**: Requisito Non Funzionale;
  - **V**: Requisito di Vincolo.
- **Priorità** identifica la priorità di sviluppo^G del requisito, le possibili tipologie sono:
  - **OB**: che indica Obbligatorio;

- **DE**: che indica Desiderabile;
- **OP**: che indica Opzionale.

- # è un numero intero incrementale che identifica univocamente il requisito all'interno della sua tipologia e priorità.

Per maggiore chiarezza sui casi d'uso ed i requisiti si consiglia la lettura del documento di Analisi dei Requisiti.

2.2.3 Codifica

Il compito del Programmatore è tradurre in codice eseguibile quanto definito durante l'analisi e la progettazione. Per garantire coerenza, manutenibilità e qualità del prodotto software, è imperativo che ogni sviluppatore aderisca alle seguenti regole e convenzioni.

2.2.3.1 Convenzioni sintattiche

La verifica della conformità alle convenzioni di codifica viene effettuata tramite strumenti automatici e revisione manuale durante le Pull Request.
Per garantire coerenza, leggibilità e manutenibilità del codice in tutto il progetto, il team adotta le seguenti convenzioni:

- **Nomenclatura**: i nomi di variabili, funzioni, metodi e classi devono essere scritti in inglese.
- Per il **codice di produzione**: usare `camelCase` per metodi, funzioni, variabili e `PascalCase` per i nomi delle classi.
- Per i **test**: si utilizza il prefisso `test_`.
- **Lingua dei commenti**: i commenti devono essere scritti in italiano. Questo favorisce la comprensione immediata da parte di tutti i membri del team durante le revisioni.
- **Scope delle Variabili**: limitare al massimo l'uso di variabili globali.
- **Funzioni**: favorire funzioni concise e monolitiche (che svolgono un unico compito ben definito).

3 Processi di Supporto

3.1 Documentazione

Il processo di documentazione è un elemento cardine di tutti i processi primari. Il suo output è fondamentale per tracciare le decisioni^G prese e per facilitare il lavoro asincrono, che nel nostro contesto si rivela notevolmente più produttivo di quello sincrono.
Nello specifico, questo processo si occupa di registrare le informazioni generate da ciascuna attività o processo del ciclo di vita del prodotto; comprende quindi tutte le operazioni di pianificazione^G, progettazione^G, sviluppo^G, produzione^G, modifica^G, distribuzione^G e manutenzione^G dei documenti destinati a tutti i soggetti coinvolti.

3.1.1 Linguaggio di Markup

Per la redazione dei documenti il gruppo ha deciso di utilizzare **LaTeX**^{G} ovvero un linguaggio di markup consolidato e ampiamente utilizzato per la stesura di documenti tecnici e scientifici. LaTeX^G consente di mantenere un'elevata qualità tipografica e di gestire in modo efficiente la struttura del documento.
NullPointers Group utilizza LaTeX^G per la produzione^G di tutta la documentazione, facendo uso di pacchetti e template appositamente sviluppati dai membri del gruppo.

3.1.2 Versionamento

Il gruppo utilizza **GitHub**^{G} come piattaforma principale per la gestione del versionamento^G e della collaborazione nella redazione dei documenti. Attraverso il sistema di controllo di versione Git, è possibile mantenere uno storico completo di tutte le modifiche, garantendo tracciabilità, ordine e coordinamento tra i membri del gruppo.

3.1.3 Caricamento in Repository

Ogni volta che si inserisce un nuovo documento o si effettua una modifica^G nel Repository^G si utilizza un branch^G feature/* personalizzato, la cui nomenclatura è definita nella sezione 3.2.1.4.
Una volta creato un commit nel branch^G una GitHub^G Action^G si occupa di creare automaticamente una PR, la quale deve essere approvata dai Verificatori^G.

3.1.4 Struttura base dei documenti

Intestazione

La prima pagina funge da intestazione del documento e contiene i seguenti elementi:

- Logo dell'Università degli Studi di Padova;

- Logo NullPointers Group;
- **Nome del documento**;
- **Stato**: se il documento è stato 'Approvato' o se è ancora 'In Approvazione';
- **Versione**: ultima versione verificata o approvata del documento;
- **Data ultima modifica**: ultima data in cui è stato modificato il documento (se ritenuta necessaria);
- **Redattori**: coloro che hanno partecipato alla redazione del documento;
- **Verificatori**: coloro che hanno partecipato alla verifica^G di parti del documento (presente in documenti diversi da Verbali);
- **Partecipanti**: coloro che partecipano alla riunione, interna o esterna che sia (presente solo nei Verbali);
- **Destinatari** del documento, ovvero a chi è rivolto.

Registro delle modifiche

Segue il Registro delle modifiche in forma tabellare che consente la tracciabilità delle modifiche apportate al documento, nel quale viene specificato:

- **Versione**: numero della versione del documento (identificativo unico);
- **Data**: data di approvazione della versione del documento;
- **Autore**: persona che ha apportato modifiche;
- **Verificatore**: persona che ha approvato le modifiche;
- **Descrizione**: breve descrizione delle modifiche apportate al documento.

Contents

Nella pagina successiva al registro delle modifiche è presente l'indice generale, nel quale vengono elencate tutte le sezioni che compongono il documento.

Indice delle tabelle e delle immagini

Successivamente all'indice, qualora il documento contenga elementi grafici o tabellari, vengono riportati l'indice delle tabelle e l'indice delle immagini. Tali indici descrivono il contenuto di ciascun elemento e ne specificano la collocazione all'interno del documento.

Contenuto principale

Il contenuto del documento è strutturato in modo gerarchico per organizzare al meglio i contenuti:

- **Capitoli**: rappresentano le macro-aree tematiche;
- **Sezioni**: suddividono i capitoli in argomenti specifici;
- **Sottosezioni**: se necessarie, approfondiscono i dettagli di ogni argomento.

3.1.4.1 Verbali Interni

I Verbali Interni seguono la struttura base dei documenti esposta alla sezione 3.1.4 ma diversamente dal resto della documentazione non contengono la sezione del Registro delle modifiche.
Il contenuto principale dei Verbali Interni segue una struttura standardizzata composta dai seguenti elementi:

1. **Informazioni generali**
   - Tipo di riunione: Interna;
   - Luogo della riunione: in presenza o sulla piattaforma Discord;
   - Data della riunione;
   - Orario di inizio;
   - Orario di fine;
   - Scriba, ovvero chi si occupa di redigere il Verbale che, come stabilito nella sezione 2.1.3.5, spetta alla figura del Responsabile.
2. **Ordine del giorno**

Ha lo scopo di delineare in modo strutturato e puntuale gli argomenti che verranno discussi durante la riunione.

3. **Diario della riunione**

Documenta in modo sintetico ma completo lo svolgimento della riunione, registrando le discussioni principali, le decisioni prese e le attività concordate.

4. **Decisioni e Azioni**

Ha lo scopo di registrare in modo formale e strutturato tutte le deliberazioni e i compiti emersi durante l'incontro.
La tabella funge da riferimento ufficiale e chiaro per tutto il gruppo, riassumendo cosa è stato stabilito e quali attività devono essere svolte.
Alcune di queste informazioni non rimangono confinate al documento, ma vengono integrate e tracciate all'interno del nostro sistema di ticketing.
Ciò garantisce che ogni elemento sia documentato per riferimento futuro e attivo per la sua esecuzione, collegando direttamente la decisione^G al task.

3.1.4.2 Verbali Esterni

I verbali esterni sono documenti che registrano ufficialmente gli incontri avvenuti con soggetti esterni al gruppo di lavoro, in particolare con l'ente proponente del progetto.
Tali documenti hanno lo scopo di tracciare le discussioni, le decisioni concordate e gli impegni assunti da entrambe le parti durante la riunione.
Seguono la struttura base dei documenti esposta alla sezione 3.1.4 ma diversamente dal resto della documentazione non contengono la sezione del Registro delle Modifiche.

1. **Informazioni generali**

- Tipo di riunione: Esterna;
- Luogo della riunione: concordato con il proponente;
- Data della riunione;
- Orario di inizio;
- Orario di fine;
- Scriba, ovvero chi si occupa di redigere il Verbale che, come stabilito nella sezione 2.1.3.5, spetta alla figura del Responsabile.;
- Partecipanti: ovvero i referenti dell'azienda proponente con i quali viene svolto l'incontro.

2. **Ordine del giorno**

L'ordine del giorno ha lo scopo di delineare ciò che verrà discusso durante la riunione, con particolare riferimento a chiarimenti dei dubbi emersi e alle domande sollevate dal gruppo in preparazione dell'incontro con l'azienda.

3. **Diario della riunione**

Documenta in modo sintetico ma completo lo svolgimento dell'incontro, registrando le discussioni principali, le decisioni prese e le attività concordate. Fornisce un resoconto strutturato degli argomenti trattati, mantenendo traccia di tutti gli aspetti rilevanti emersi durante il confronto.

4. **Decisioni e Azioni**

Questa sezione ha lo scopo di registrare le decisioni e i relativi compiti emersi dall'incontro, fungendo da riferimento ufficiale per il gruppo e permettendo la tracciabilità delle decisioni.
Eventuali task, se tracciabili, verranno integrati nel sistema di ticketing.

5. **Approvazione esterna**

È l'ultima sezione del documento che attesta che i relativi verbali esterni siano approvati dalla proponente tramite firma ed eventuale timbro del rappresentante.

3.1.4.3 Diari di Bordo

I diari di bordo sono presentazioni utilizzate durante gli incontri settimanali con l'obiettivo di verificare in modo condiviso lo stato di avanzamento di ciascun gruppo ammesso al I lotto.
La struttura tipica di un diario di bordo comprende le seguenti sezioni:

- **Risultati**: descrive le attività completate nel periodo corrente e le confronta con quanto inizialmente pianificato;
- **Problematiche riscontrate**: consente di illustrare le problematiche affrontate e i dubbi ancora irrisolti, sono volti alla richiesta di supporto o chiarimenti;
- **Attività future**: elenca i compiti da svolgere nel prossimo intervallo di lavoro.

3.1.4.4 Altri documenti

Di seguito sono elencati tutti i documenti redatti e mantenuti durante l'intero ciclo di vita del progetto ciascuno dei quali risponde a uno scopo specifico, contribuendo alla tracciabilità, alla gestione e alla comunicazione delle attività di progetto.
La struttura iniziale di tali documenti corrisponde con quanto definito al punto 3.1.4:

- **Valutazione dei Capitolati**;
- **Dichiarazione degli Impegni**;
- **Lettera di Presentazione**;
- **Norme di Progetto**;
- **Analisi dei Requisiti**;
- **Piano di Progetto**;
- **Piano di Qualifica**;
- **Glossario**;
- **Specifica Tecnica**;
- **Manuale Utente**.

3.1.5 Stesura dei documenti

La Redazione di un documento segue i seguenti passaggi:

1. **Creazione della Issue**: viene aperta una issue^G su GitHub, assegnandola a uno o più membri in base al ruolo da loro ricoperto (vedi sezione 4.1.2), per tracciare le attività da svolgere.
2. **Stesura del documento**: coloro a cui è stata assegnata la issue^G avviano o continuano la stesura del documento e procede al caricamento su GitHub^G tramite un branch^G feature/* in base alla funzionalità^G o alla sezione su cui stanno lavorando.
3. **Creazione della Pull Request**: una volta completato il lavoro sul branch, viene aperta una Pull Request^G con richiesta di verifica^G della nuova sezione da un altro membro del gruppo, segnalato nel Registro delle Modifiche.
4. **Revisione e Approvazione**: il membro del team designato procede ad esaminare il codice e il contenuto della PR, fornendo feedback e richiedendo modifiche se necessario.
5. **Merge:** dopo essersi accertato che le modifiche siano conformi, avviene il merge^G con il ramo principale, chiudendo automaticamente la issue^G associata.

Il procedimento che segue si applica solo a verbali o per l'approvazione di Documenti ritenuti definitivi e da approvare per termini di Baseline:

6. **Approvazione e pubblicazione:** A seguito di esito positivo della revisione, il Responsabile^G approva il documento completando la pull request^G e procede al merge^G del branch^G secondario in quello principale.
   Questa operazione attiva una GitHub^G Action^G che, oltre a procedere con la generazione del .pdf dai .tex, aggiorna automaticamente il sito web del progetto con i nuovi documenti approvati.

3.1.6 Nomenclatura

3.1.6.1 File nel repository

Per convenzione i documenti caricati nel repository^G seguiranno la seguente denominazione:
Verbali interni:

AAAA-MM-GG_verbale_interno

Verbali esterni:

AAAA-MM-GG_verbale_esterno

Diari di bordo:

AAAA-MM-GG_Diario_di_Bordo

Le date dovranno essere espresse nel formato ISO^G 8601 (AAAA-MM-GG), che prevede quattro cifre per l'anno, due per il mese e due per il giorno. Questo standard garantisce un ordinamento cronologico automatico e inequivocabile sia per i verbali (interni ed esterni) che per i diario di bordo, semplificandone la consultazione e il raggruppamento.
Per tutti gli altri documenti, la convenzione di denominazione prevede la forma:

Titolo_documento

3.1.6.2 Documenti nel sito web

Diversamente è stato deciso di fare per il sito della Documentazione (https://nullpointersgroup.github.io/Documentazione/) dove, sempre per convenzione, è stato stabilito come segue:

AAAA-MM-GG_TIPO v#

In questo caso sarà presente il TIPO, ovvero:

- VI: Verbale Interno;
- VE: Verbale Esterno;
- DB: Diario di Bordo.

La nomenclatura dei restanti documenti consisterà in:

Titolo_documento v#

Inoltre viene indicata la versione attuale di ogni documento con v# dove # funge da modificatore di versione, che viene incrementato progressivamente a ogni aggiornamento significativo del documento.

3.1.7 Manutenzione

L'attività di manutenzione^G della documentazione viene attivata ogni qualvolta un documento richiede integrazioni o modifiche per rimanere accurato e allineato allo stato del progetto.
Il flusso di aggiornamento ripercorre le fasi principali della stesura iniziale: partendo dalla creazione di una issue^G di tracciamento, si procede con le modifiche in un branch^G dedicato, per concludere con una verifica^G formale attraverso una Pull Request^G prima dell'integrazione nel documento definitivo (il punto 3.1.5 descrive la procedura).

3.1.8 Documentazione del Codice

Questa sezione definisce gli standard da adottare per la scrittura e la documentazione del codice sorgente del progetto SmartOrder.
Il suo scopo è garantire:

- **Leggibilità e Manutenibilità**: un codice ben documentato è più facile da comprendere, correggere ed estendere.
- **Tracciabilità**: collegamento chiaro tra il codice, i requisiti e le decisioni progettuali.
- **Coerenza del Team**: uniformità nello stile, essenziale per il lavoro collaborativo e asincrono.

3.1.8.1 Convenzioni per la Documentazione del Codice

Per garantire tracciabilità e manutenibilità, il codice sorgente deve essere documentato utilizzando uno standard ispirato da **Doxygen**.
I blocchi di documentazione devono essere inseriti immediatamente prima della dichiarazione di:

- Funzioni e metodi in Python.
- Componenti React, funzioni e metodi in JavaScript/TypeScript.

3.1.8.2 Tag previsti

Ogni blocco di documentazione deve includere i seguenti tag, ove applicabili:

Tag	Scopo e Formato
`@brief`	**Brief description**: descrizione concisa dello scopo e della funzionalità.
`@param`	**Type Description**: descrizione di un parametro, specificandone il tipo e una breve descrizione.
`@raise/@throws`	**ExceptionType Condition or description**: eccezioni/errori sollevati, con tipo e condizioni che li provocano.
`@bug`	**Actual problems**: descrizione di problemi noti non ancora risolti nel codice.
`@return`	**Type Description**: descrizione del valore di ritorno e del suo tipo.
`@req`	**Requisito associato**: identificativo del requisito associato (es. `RF-OB__1`). Fondamentale per la tracciabilità.

Table 1: Tag previsti per la documentazione del codice

3.1.8.3 Verifica della Conformità

La conformità a questi standard è parte integrante del **processo di verifica del codice**:

1. Il **Programmatore** è responsabile della documentazione del proprio codice.
2. Il **Verificatore**, durante la revisione di una Pull Request, deve controllare:
   - La presenza dei blocchi di documentazione per le nuove funzionalità.
   - La completezza dei tag corretti (`@brief`, `@param`, `@return`, `@req`).
   - La correttezza del formato, in particolare per il tag `@req` che deve contenere un identificativo valido di requisito.
   - La coerenza tra il codice e quanto descritto nella documentazione.
3. La mancata conformità è motivo di richiesta di modifica prima dell'approvazione della PR.

3.2 Gestione della Configurazione

La gestione delle configurazioni è il processo che garantisce il controllo e la tracciabilità di tutte le componenti di un progetto software. Consente di organizzare le modifiche al codice e alla documentazione, permettendo un controllo su tutte le modifiche e le versioni rilasciate.
Per garantire questo controllo, ogni elemento del progetto viene trattato come elemento di configurazione e sottoposto a versionamento e tracciamento.
Gli elementi sottoposti a verifica sono documenti, esclusi i verbali siccome questi non hanno necessità di registro di modifiche, e il codice.
Questo approccio assicura un avanzamento ordinato dello sviluppo, mantenendo una traccia storica dell'evoluzione di ciascun elemento.

3.2.1 Strumenti di Controllo della Configurazione

Il gruppo utilizza il sistema di Issue Tracking e le Project Board integrate in GitHub come strumenti principali per la gestione e l'organizzazione delle attività.
Questi strumenti consentono di pianificare il lavoro, monitorarne l'avanzamento e mantenere un controllo costante sulla qualità e sulla coerenza del processo di sviluppo.

3.2.1.1 Issue

Ogni attività, modifica o produzione di documentazione viene gestita attraverso una issue, che deve contenere:

- **Titolo**: una descrizione sintetica del compito da svolgere;
- **Descrizione**: informazioni dettagliate sul lavoro richiesto, con eventuali riferimenti ai documenti da aggiornare;
- **Assegnatario**: il membro incaricato;
- **Etichette** (Labels): utilizzate per classificare la tipologia del compito;
- **Milestone**: indica l'obiettivo di riferimento (RTB o PB);
- **Board e Stato**: identifica la project board di appartenenza e lo stato corrente (Todo, In Progress, Done);

Le issue vengono create inizialmente dal Responsabile, ma possono essere aggiornate da qualsiasi membro qualora emergano nuove informazioni, modifiche o necessità di riformulazione del compito. Ogni aggiornamento deve mantenere chiarezza, coerenza e tracciabilità.

Il mantenimento del sistema di Issue Tracking avviene tramite una procedura stabilita:

1. **Creazione della issue** da parte del Responsabile o del membro che individua la necessità del compito.
2. **Aggiornamento continuo della issue** da parte dell'assegnatario quando si presentano cambiamenti o integrazioni.
3. **Registrazione dello stato di avanzamento** tramite lo spostamento della issue nella colonna corrispondente della Project Board.
4. **Revisione obbligatoria**: ogni attività che comporta una modifica ai file del repository richiede una verifica formale.
5. Quando un membro effettua un push nel branch relativo alla issue, il sistema genera automaticamente una **pull request**.
6. **Assegnazione della PR** al Verificatore designato.
7. **Verifica della PR**: il Verificatore controlla che il contenuto sia conforme agli standard, coerente con la documentazione e privo di errori.

8. **Merge nel main**: se la verifica ha esito positivo, il Verificatore effettua il merge e chiude la issue. In caso contrario, richiede ulteriori modifiche.

Questa procedura permette tracciabilità completa del lavoro e sincronizzazione costante tra issue, project board e repository riducendo il rischio di incoerenze dei documenti.

3.2.1.2 Project Board

Il gruppo mantiene due Project Board principali:

- **RTB** (Requirements and Technology Baseline)
- **PB** (Product Baseline)

Queste board permettono di organizzare e visualizzare in modo chiaro tutte le attività necessarie al raggiungimento delle rispettive milestone.
Ogni issue viene collegata alla board pertinente e inserita nella colonna appropriata, così da garantire una visione aggiornata e condivisa dello stato del progetto.
Le board vengono aggiornate:

- quando una issue cambia stato;
- quando viene aggiunto un nuovo compito;
- quando un'attività richiede una modifica o riformulazione.

Il Responsabile verifica periodicamente la coerenza delle board con le milestone attive.

3.2.1.3 Labels

Per migliorare l'organizzazione, il gruppo utilizza un sistema di etichette che identifica chiaramente la tipologia di lavoro, introducendo una famiglia dedicata specificamente alla gestione documentale.
Il principio è semplice: viene sempre aggiunta un'etichetta specifica, come "Piano di Progetto" o "Analisi dei Requisiti", che indica precisamente quale file è coinvolto.
Queste label vengono applicate sia alle issue settimanali, sia alle attività associate alle milestone RTB e PB.
Ciò permette di individuare rapidamente quali attività richiedono attenzione e tracciare il carico di lavoro per tipologia di attività oltre che avere una visione completa dell'avanzamento complessivo del progetto.

3.2.1.4 Nomenclatura dei branch

Per garantire uniformità e tracciabilità nella gestione dei branch relativi alla documentazione, è stata adottata la seguente convenzione di denominazione:

Nome Branch	Descrizione
feature/glossario	Per modifiche al documento Glossario
feature/analisi-requisiti	Per modifiche al documento Analisi dei Requisiti
feature/specifica-tecnica	Per modifiche al documento Specifica Tecnica
feature/manuale-utente	Per modifiche al documento Manuale Utente
feature/norme-progetto	Per modifiche al documento Norme di Progetto
feature/piano-progetto	Per modifiche al documento Piano di Progetto
feature/piano-qualifica	Per modifiche al documento Piano di Qualifica
feature/YYYYMMDDverbint	Per creazione o modifica di un verbale interno (YYYYMMDD = data)
feature/YYYYMMDDverbest	Per creazione o modifica di un verbale esterno (YYYYMMDD = data)
feature/script	Per creazione o modifica di script
feature/aggiornamenti	Per aggiornamenti generali a tutti i documenti
feature/YYYYMMDDdiario	Per creazione o modifica di un diario di bordo (YYYYMMDD = data)

Table 2: Nomenclatura dei branch

3.2.2 Registrazione dello Stato di Configurazione

L'attività di Registrazione dello Stato di Configurazione è fondamentale per garantire la tracciabilità e la conoscenza storica delle evoluzioni di tutti gli Elementi di Configurazione (codice e documentazione) prodotti dal gruppo.

3.2.2.1 Documentazione

In base a quanto definito nel verbale interno del 6 novembre per il tracciamento durante il ciclo di vita dei documenti, NullPointers Group ha deciso di utilizzare una politica di versionamento basata sulla specifica MAJOR.MINOR.PATCH. Questa specifica riflette lo stato di avanzamento e la validazione formale del documento. La sua implementazione avviene come segue:

- **MAJOR**: Incrementato solo in seguito all'Approvazione Formale (Approvazione) del file da parte del Responsabile.
- **MINOR**: Incrementato una volta completato l'intero ciclo di modifica, che include sia lo sviluppo del contenuto sia la verifica formale da parte del Verificatore designato.
- **PATCH**: Incrementato per modifiche di entità minore, come correzioni ortografiche, formattazione, o aggiustamenti di sintassi che non alterano il contenuto sostanziale.

In aggiunta al versionamento, ogni documento conterrà un Registro delle Modifiche dettagliato (come descritto nella sezione 3.1.4), che elenca cronologicamente tutte le variazioni apportate per ciascuna versione rilasciata.

3.3 Gestione della Qualità

La Gestione della Qualità è il processo che definisce, pianifica e coordina tutte le attività necessarie affinché i prodotti sviluppati dal gruppo rispettino gli standard stabiliti nel Piano di Qualifica e risultino coerenti con gli obiettivi progettuali.
Tale processo assicura che ogni artefatto prodotto (documenti, codice, modelli o deliverable intermedi) soddisfi i criteri qualitativi prefissati e fornisce un quadro operativo per il miglioramento continuo.
La qualità è garantita da un sistema integrato di strumenti, procedure e responsabilità, che accompagna e supporta l'intero ciclo di sviluppo. Questo approccio consente di prevenire l'introduzione di errori, rilevare tempestivamente eventuali incongruenze e garantire che ogni output sia verificato prima di essere considerato completo, oltre a mantenere elevata la tracciabilità delle attività.
La Gestione della Qualità opera in stretta sinergia con la Gestione della Configurazione (sezione 3.2) e con il processo di Verifica (sezione 3.4), costituendo un sistema integrato di controllo e miglioramento del prodotto.

3.3.1 Piano di Qualifica

Il gruppo adotta una pianificazione esplicita delle attività di qualità, formalizzata nel Piano di Qualifica, che definisce:

- Gli obiettivi qualitativi specifici da raggiungere, distinti per il prodotto finale e per i processi di sviluppo;
- Metriche quantitative per misurare oggettivamente la qualità del prodotto ma anche del processo.
- Strumenti utilizzati sia per l'analisi statica che dinamica;
- Strategie di testing specifiche (unitario, di integrazione, di sistema e di accettazione) per verificare il prodotto e garantire che soddisfi i requisiti;
- Un cruscotto di valutazione che consente il monitoraggio, per la durata del ciclo di vita del progetto, delle metriche di qualità definite.

Il Piano di Qualifica viene aggiornato nel corso del progetto per l'aggiornamento dei dati e la continuazione del cruscotto di valutazione, oltre che per eventuali miglioramenti o variazioni negli strumenti adottati.

3.3.2 Metriche di Qualità

Per garantire un controllo oggettivo e ripetibile della qualità, il gruppo adotta un insieme di metriche che vengono applicate regolarmente ai prodotti sviluppati.
Le metriche (dettaglio nella sezione 6) sono suddivise in due categorie:

- **Metriche di Qualità del prodotto**: indicatori per misurare le caratteristiche del software finale, come affidabilità e usabilità.
- **Metriche di Qualità del processo**: indicatori per valutare l'efficacia e l'efficienza delle attività di sviluppo.

3.3.3 Struttura e identificazione delle metriche

Per garantire chiarezza, coerenza interna e una tracciabilità efficace, tutte le metriche adottate nel progetto e riportate nella sezione 6 sono strutturate secondo un formato standardizzato.
Ogni metrica è definita dai seguenti elementi essenziali:

- **Codice Identificativo Univoco**, che utilizza il formato

  `M[Abbreviazione]_[Numero]`

  dove:

  - `MQD_xx`: identifica le Metriche di Qualità del Prodotto.
  - `MQC_xx`: identifica le Metriche di Qualità del Processo.

  Il numero progressivo (xx) garantisce l'unicità all'interno di ciascuna categoria.
- **Nome**: il nome completo e descrittivo della metrica.
- **Descrizione**: una spiegazione chiara della sua finalità e ambito di applicazione.
- **Formula**: la definizione matematica o procedurale per il suo calcolo, specificata per le metriche quantitative.

Questa struttura facilita il riconoscimento immediato della tipologia di metrica (di prodotto o di processo), standardizza la documentazione e ne supporta l'utilizzo sistematico in tutte le fasi di monitoraggio e valutazione.

3.3.4 Miglioramento Continuo

Il gruppo segue un approccio in cui la qualità migliora progressivamente, analizzando i risultati dopo ogni ciclo di lavoro.
Se emergono problemi, come errori frequenti o metriche insufficienti, si valutano azioni correttive. Questo può significare introdurre nuovi strumenti, aggiornare quelli esistenti o rivedere procedure e checklist.
Le modifiche importanti vengono discusse in gruppo, approvate e registrate formalmente nei verbali interni, per assicurare la tracciabilità e una corretta applicazione futura.

3.4 Verifica

3.4.1 Scopo e descrizione

Il processo di Verifica è un'attività sistematica che deve essere applicata a tutti i prodotti di progetto (documenti e codice) prima che possano essere considerati completi e rilasciati.
Il suo scopo principale è rispondere alla domanda "*Did I build the system right?*", ovvero assicurare che quanto prodotto sia stato realizzato correttamente, in

piena conformità con i vincoli, le regole e gli standard qualitativi definiti nel Piano di Qualifica e nelle Norme di Progetto.
Le attività di Verifica sono svolte dai Verificatori, il cui compito è controllare, correggere e segnalare ogni discrepanza rispetto ai requisiti prestabiliti.

3.4.2 Strumenti a supporto

Per supportare le attività di Verifica descritte, il gruppo utilizza GitHub, imponendo che ogni modifica alla documentazione o al codice venga prima sottoposta a revisione attraverso una Pull Request.
Le regole di protezione del ramo principale obbligano che ogni PR sia approvata da almeno un Verificatore diverso dall'autore, garantendo un controllo qualità sistematico e tracciabile prima che le modifiche vengano integrate.

3.4.3 Analisi statica

L'analisi statica è una tecnica di verifica che si applica senza eseguire il codice o il prodotto.
Il suo scopo è individuare problemi di sintassi, logica o conformità agli standard prima che si manifestino durante l'esecuzione.
Si può eseguire tramite metodi formali (dimostrazioni matematiche) oppure, più frequentemente, tramite metodi di lettura manuali o semi-automatici.
I metodi di lettura più diffusi sono due:

- **Walkthrough**: è una tecnica di verifica basata sull'ipotesi che esista un difetto, pur in assenza di informazioni sulla sua natura o localizzazione. Ne consegue l'obbligo di un'analisi approfondita ed esaustiva dell'oggetto sotto esame. Questa caratteristica lo rende una metodologia ad alto impegno di risorse, di limitata applicabilità pratica e scarsamente automatizzabile.
- **Ispezione**: utilizza una lista di controllo predeterminata, concentrandosi su possibili errori o criteri di qualità specifici. Pur essendo meno esaustiva di un walkthrough (specie nelle prime fasi di sviluppo) offre un vantaggio decisivo: è facilmente automatizzabile. Per questo risulta la scelta preferibile quando occorre verificare grandi quantità di codice o documenti in modo sistematico e ripetibile.

3.4.4 Analisi dinamica

L'analisi dinamica viene condotta eseguendo il prodotto software attraverso una serie di test. Questa procedura misura la qualità funzionale del codice, assicurandosi che raggiunga il suo scopo e che il suo comportamento sia conforme alle specifiche.
Permette di individuare e risolvere gli errori nel codice che causano comportamenti indesiderati.
Ogni test è definito da uno stato iniziale, input specifici e output attesi.
Le attività di test devono essere ripetibili, per poter confermare le correzioni,

e automatizzabili, per essere eseguiti in modo continuo durante tutto il ciclo di vita del prodotto.

3.4.5 Classificazione dei Test

Tutti i test pianificati, eseguiti e i loro esiti sono tracciati nel Piano di Qualifica. I test sono classificati in categorie gerarchiche, ognuna con un codice identificativo univoco nel formato:

4 Processi Organizzativi

I processi organizzativi definiscono un insieme di operazioni di supporto per lo sviluppo^G software che operano trasversalmente rispetto al ciclo di vita del software garantendo che il gruppo possieda l'organizzazione, le infrastrutture e le competenze necessarie per sostenere i processi primari.
Assicurano la buona esecuzione di tutti i processi adottati e eventuali miglioramenti.
Si individuano i seguenti processi:

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

4.1 Gestione dei Processi

Secondo lo standard ISO^G 12207:1995, "La gestione dei processi comprende le attività e i compiti che possono essere svolti da qualsiasi soggetto che debba gestire i propri processi".
Sulla base di questo principio, il suo scopo principale è stabilire come un processo deve essere pianificato e monitorato secondo le relative responsabilità dei membri del gruppo.
Un altro obiettivo fondamentale è garantire un flusso comunicativo efficace, sia interno che esterno assicurando: coerenza, controllo e miglioramento continuo.

4.1.1 Attività previste

Per assicurare il raggiungimento degli obiettivi nel rispetto di tempi e qualità, il processo è strutturato nelle seguenti attività, che definiscono un flusso di lavoro chiaro e responsabilità precise:

4.1.1.1 Inizializzazione

L'avvio del processo avviene tramite la selezione dei Requisiti, presenti nel documento "Analisi dei Requisiti", da portare a termine tramite le attività di tale processo.
Il Responsabile^G valuta preliminarmente la fattibilità del processo: se alcuni requisiti risultano irrealizzabili per vincoli di tempo, risorse o competenze, e previo accordo di tutto il gruppo, i requisiti del processo possono essere modificati in questa fase per garantire il raggiungimento dei criteri di completamento.

4.1.1.2 Pianificazione

L'attività di pianificazione, portata a termine dal Responsabile, ha lo scopo di preparare il piano di esecuzione delle attività del processo. In particolare deve verificare la disponibilità delle risorse necessarie (budget residuo, della disponibilità dei componenti del gruppo, competenza, etc.) per completare il

processo entro i tempi stabiliti.
Infine assegna le attività del processo ai membri del team in base ai loro ruoli.

4.1.1.3 Esecuzione e controllo

Durante l'esecuzione: i membri del team portano a termine le attività assegnategli, mentre il Responsabile^G ha il compito di monitorare l'andamento delle attività.
Qualora si presentassero problemi, il Responsabile^G deve essere immediatamente notificato e in caso di stallo, contattare la proponente o il committente per un chiarimento.

4.1.1.4 Verifica

La Verifica^G del prodotto realizzato dai membri del team è di competenza del Verificatore, che assicura la corretta realizzazione del singolo requisito tramite la procedura di Verifica^G.

4.1.1.5 Chiusura

La chiusura del processo consegue la terminazione di tutte le attività che ne hanno preso parte.
È compito del Responsabile^G o del Verificatore approvare il merge^G della Pull Request^G nel branch^G main; una volta fatto il merge, il processo è da definirsi chiuso.

4.1.2 Tracciamento delle ore

Per monitorare il tempo dedicato ai diversi ruoli durante il progetto, viene utilizzato uno spreadsheet dedicato, accessibile al gruppo su Google Drive.
Al termine di ogni sprint, viene generato il consuntivo dello sprint appena concluso e il preventivo di quello successivo.
Il membro che ricopre il ruolo di Responsabile ha il compito di inserire tali informazioni nel documento "Piano di Progetto".

4.1.2.1 Preventivo

Il preventivo delle ore è presentato in forma tabellare e riporta le ore stimate per ciascun membro del gruppo, suddivise per ruolo e riferite al singolo sprint.
Viene inoltre creato un diagramma circolare che illustra la distribuzione delle ore previste per ogni ruolo, fornendo una rappresentazione immediata e intuitiva delle risorse che si prevede di allocare per quello sprint.

4.1.2.2 Consuntivo

Il consuntivo delle ore è anch'esso presentato tramite una tabella che riporta le ore effettivamente registrate da ciascun membro, suddivise per ruolo e relative allo sprint.
Anche in questo caso viene incluso un diagramma circolare che visualizza la ripartizione delle ore effettive per ruolo, fornendo una visione immediata delle risorse realmente impiegate durante lo sprint.

4.1.3 Ruoli

Ruolo	Compiti
Responsabile^G	Il Responsabile^G coordina le attività del gruppo garantendo una pianificazione^G efficace.
Amministratore	L'Amministratore^G si occupa della configurazione e gestione dell'infrastruttura IT di supporto al progetto.
Analista^G	L'Analista^G si occupa di identificare e chiarire i requisiti, interpretando le esigenze degli utilizzatori finali per garantire una corretta definizione delle funzionalità.
Verificatore^G	Il Verificatore^G si occupa di assicurare la qualità dei prodotti e dei processi adottati, effettuando revisioni e test.
Programmatore	Il Programmatore è responsabile^G dello sviluppo^G del codice sorgente del progetto, traducendo il design in codice funzionante e testabile dal proponente.
Progettista^G	Il Progettista^G traduce i requisiti del sistema in un'architettura software dettagliata, definendo moduli, interfacce, flussi dati e vincoli tecnici.

Table 3: Ruoli

4.1.3.1 Rotazione dei ruoli

La rotazione dei ruoli avviene ogni due settimane, in modo da garantire una distribuzione equilibrata delle competenze e favorire l'apprendimento trasversale all'interno del gruppo.
Ci riserviamo tuttavia la possibilità di effettuare modifiche ai ruoli anche a metà sprint, qualora necessario.
Tale decisione può derivare dall'esito dell'incontro intermedio, durante il quale viene valutato l'avanzamento delle attività e individuati eventuali ritardi, blocchi o membri sottoutilizzati.
In questi casi, l'assegnazione di nuovi incarichi o la riorganizzazione dei ruoli permette una migliore suddivisione dei task in unità più piccole e gestibili, favorendo il completamento efficace delle attività pianificate.

4.1.4 Coordinamento

Il coordinamento è un fattore fondamentale per il buon andamento del progetto, perché permette di gestire in modo efficace le attività del team e le relazioni con la proponente e i committenti.

Per assicurare un flusso informativo continuo e aggiornato, sono pianificate riunioni regolari e l'impiego di canali di comunicazione adeguati.

4.1.4.1 Riunioni

Al fine di garantire un flusso di comunicazione efficace, un costante allineamento del gruppo e un confronto attivo con la proponente, sono previsti due tipi distinti di incontri:

4.1.4.1.1 Riunioni interne

Le riunioni interne vengono organizzate regolarmente, in genere ogni lunedì della settimana.
Durante questi incontri il gruppo effettua un punto della situazione: vengono analizzate le attività svolte, quelle ancora in corso e le eventuali criticità emerse. Il Responsabile, grazie al confronto tra i membri, può così ottenere una visione aggiornata dell'avanzamento del progetto e pianificare al meglio le attività successive.

4.1.4.1.2 Riunioni esterne

Le riunioni esterne coinvolgono i membri del gruppo e la proponente.
Tali incontri non seguono una periodicità fissa: vengono programmati secondo necessità tramite richiesta via canale concordato con la proponente (come specificato al punto 2.1.2).
Durante le riunioni esterne il gruppo presenta lo stato di avanzamento del lavoro, chiarisce eventuali dubbi e riceve indicazioni utili per proseguire nelle attività di sviluppo.
Al termine di ogni riunione, interna o esterna, viene redatto un verbale (specifiche a sezioni 3.1.4.1 e 3.1.4.2) che documenta gli argomenti trattati e le decisioni prese, garantendo tracciabilità e condivisione delle informazioni.

4.1.4.2 Comunicazioni

Le comunicazioni costituiscono un elemento fondamentale per garantire coordinamento, continuità informativa e tempestività nelle attività del gruppo. Gli strumenti utilizzati per la comunicazione sono descritti nella sezione 2.1.2; in questa sezione viene invece specificato come tali strumenti vengono impiegati all'interno dei processi organizzativi.

4.1.4.2.1 Comunicazioni interne

Le comunicazioni interne hanno lo scopo di supportare il lavoro quotidiano del team, favorire la condivisione di informazioni e consentire un rapido confronto operativo.

- Le comunicazioni rapide o di servizio avvengono mediante il canale di messaggistica adottato dal gruppo (WhatsApp);
- Le discussioni che richiedono un confronto più approfondito o decisioni condivise vengono svolte tramite incontri a distanza (Discord);
- La documentazione dei task, degli avanzamenti e delle attività pianificate è gestita attraverso lo strumento di versionamento e ticketing adottato dal

team (GitHub).

Questa organizzazione consente di distinguere in modo chiaro tra comunicazioni informali, operative e momenti di coordinamento strutturato.

4.1.4.2.2 Comunicazioni esterne

Le comunicazioni esterne garantiscono il mantenimento di un flusso regolare e trasparente con la proponente e con i committenti.

- Le comunicazioni formali, come approvazioni di verbali esterni, avvengono tramite l'indirizzo e-mail istituzionale del gruppo (Google Mail);
- Le riunioni esterne sono organizzate attraverso la piattaforma concordata per gli incontri a distanza (Google Meet);
- Per eventuali comunicazioni asincrone o rapide è previsto anche un canale condiviso con la proponente (WhatsApp), come stabilito in fase iniziale.

Tutte le comunicazioni esterne rilevanti ai fini del progetto vengono tracciate mediante la produzione dei verbali esterni.

4.2 Gestione dell'Infrastruttura

Il processo di infrastruttura ha lo scopo di fornire, configurare e mantenere l'ambiente di lavoro necessario all'esecuzione di tutti i processi di sviluppo^G e documentazione.
Esso comprende la gestione delle risorse, siano esse hardware o software, garantendone la disponibilità e l'efficienza^G per l'intera durata del progetto.

4.2.1 Attività previste

4.2.1.1 Implementazione

Per supportare il lavoro asincrono, la tracciabilità e la qualità dei prodotti NullPointers Group adotta i seguenti strumenti che costituiscono l'infrastruttura del progetto:

- Gestione del versionamento: Git
- Piattaforma: GitHub
- Automazione: GitHub^G Actions, script Python^G e Lua
- Comunicazione: Discord e Whatsapp

4.2.1.2 Predisposizione

L'attività di predisposizione stabilisce le regole di interazione tra i membri del gruppo e l'ambiente di lavoro, inoltre definisce la natura dell'infrastruttura utilizzata.
L'infrastruttura adottata è finalizzata a minimizzare gli errori e a garantire la coerenza del prodotto.
Vengono riportati gli strumenti principali:

Strumento	Predisposizione
Git^G	Definizione di un file .gitignore condiviso per escludere i file temporanei e di build garantendo che la repository^G contenga solamente i file sorgente.
GitHub^G	È stata creata una repository^G dedicata alla documentazione del progetto. È stata applicata una branch^G protection rule sul ramo main: ogni modifica^G deve provenire da una pull request^G e richiede l'approvazione di un Verificatore^G per il suo merge.
Discord e Whatsapp^G	Per consentire al gruppo di riunirsi settimanalmente, e venire incontro al fatto che ci sono significative distanze tra i membri, è stato creato un server sulla piattaforma Discord, un'applicazione che consente videochiamate e scambio di messaggi; ideale per il nostro scopo. È stato inoltre creato un gruppo Whatsapp^G per questioni minori che non richiedono una videochiamata.
Labels GitHub^G	Sono state implementate delle Labels per categorizzare le attività e Milestones per tracciare l'avanzamento del progetto.
GitHub^G Actions, script Python^G e script Lua	Sono state configurate le GitHub^G Actions per l'esecuzione automatica degli script che compilano i file sorgente LaTeX^G ad ogni push garantendo che la versione PDF visibile sul sito sia sempre sincronizzata con l'ultima versione dei documenti.

Table 4: Strumenti di predisposizione

4.2.1.3 Manutenzione

Data la complessità del progetto è probabile che l'infrastruttura subisca dei cambiamenti nel corso del tempo per l'aggiornamento o il miglioramento delle sue funzionalità.
È compito dell'Amministratore^G la manutenzione^G dell'infrastruttura ovvero le attività di controllo delle funzionalità^G ed aggiornamento/creazione degli script di automazione.
Successivamente verranno illustrate le norme da seguire per mantenere e aggiornare l'infrastruttura affinché il flusso di lavoro non venga spezzato.

4.2.1.3.1 Git

Git^G non ha bisogno di particolari configurazioni, è sufficiente accedere localmente con le credenziali che il membro usa per accedere a Github.

4.2.1.3.2 GitHub

Su Github, l'account di NullPointers Group è gestito come organizzazione, ovvero un account che serve da contenitore per il lavoro condiviso tra mem-

bri di un team.
Sono state create 3 repository^G dentro l'organizzazione:

- **Documentazione**: repository^G dove viene salvata e versionata tutta la documentazione in merito al capitolato^G SmartOrder e non solo.
- **SmartOrder**: repository^G dove viene salvato e versionato il codice sorgente dell'applicativo SmartOrder.
- **.github**: repository che serve a creare il README.md del gruppo, gestire i workflow condivisi fra le varie repository. In generale, serve a centralizzare configurazioni e contenuti condivisi.

Affinché le nuove impostazioni vengano effettivamente applicate nelle repository^G di documentazione o di codice sorgente, il verificatore^G dovrà assicurarsi che la nuova infrastruttura proposta superi le metriche di qualità di processo definite in seguito.

4.2.1.3.3 Actions e script ausiliari

Nelle repository^G di Documentazione e SmartOrder, si impiegano strumenti di CI messi a disposizione da Github: le Github^G Actions, le quali vengono definite in un file ".yml" dentro la cartella ".github/workflows".
Affinché le Github Actions potessero assolvere allo scopo per cui sono state configurate, sono stati sviluppati script ausiliari, in Python e Lua, da far eseguire a quest'ultime.
La configurazione di nuove Github^G Actions o la modifica^G di Github^G Action^G esistenti spetta agli Amministratori.
La creazione di nuove Github^G Actions viene richiesta agli Amministratori dal Responsabile, sotto comune accordo dai membri del gruppo.

4.2.1.3.4 Infrastruttura di Tracking

La configurazione e la manutenzione ordinaria degli strumenti di GitHub (la cui organizzazione è descritta nella sezione 3.2.1) seguono le procedure definite successivamente.

Sistema di Label
Viene definita una palette standard di label e le convenzioni di nomenclatura. Periodicamente si esegue una pulizia per rimuovere quelle obsolete o unificarne di duplicate.
Qualsiasi modifica allo standard viene discussa e approvata dal gruppo.

Project Board
Le board per le milestone (RTB, PB) vengono create da un template predefinito. Al termine di una milestone, le board vengono archiviate dopo aver chiuso le issue residue.
Mensilmente si verifica che la posizione delle issue nelle colonne corrisponda al loro stato effettivo.

Issue e Automazioni
Si mantengono aggiornati i template predefiniti per la creazione di nuove issue. Vengono identificate e chiuse periodicamente le issue inattive o completate.

4.2.1.3.5 Discord

Discord è la piattaforma principale per la comunicazione interna del gruppo.
Il server è organizzato in canali testuali dedicati a specifiche aree, come la documentazione e il codice sorgente, per mantenere le discussioni focalizzate.
La creazione di nuovi canali avviene su richiesta di qualsiasi membro, ma è riservata all'Amministratore del server, che provvede a configurarli, garantire l'accesso a tutti e mantenere la moderazione.

4.3 Processo di Miglioramento

Il miglioramento è un processo finalizzato a valutare, misurare, controllare e ottimizzare il ciclo di vita del software.
Il suo obiettivo è garantire che il prodotto non solo risponda alle aspettative, ma raggiunga e mantenga standard elevati di qualità ed efficienza, attraverso revisioni periodiche e perfezionamenti incrementali.
L'approccio è ciclico e, secondo il modello Plan-Do-Check-Act (PDCA), si compone di quattro fasi principali, applicate periodicamente per garantire un progresso continuo.

4.3.1 Ciclo PDCA

Il miglioramento continuo si articola nelle seguenti quattro fasi, iterabili su base periodica o al verificarsi di eventi significativi:

- **Plan** (Pianificazione): identificazione e documentazione dei processi organizzativi da applicare al ciclo di vita del software.
- **Do** (Implementazione): integrazione dei miglioramenti nei processi esistenti, con aggiornamento della documentazione e raccolta di dati storici e tecnici.
- **Check** (Valutazione): analisi dei processi rispetto agli obiettivi prefissati e alle metriche adottate, mediante i dati raccolti e revisioni periodiche.
- **Act** (Standardizzazione): consolidamento dei miglioramenti efficaci nei processi, per evitarne la regressione e assicurarne l'evoluzione costante.

4.4 Processo di Formazione

Il processo di formazione è un'attività di supporto volta a garantire che tutti i membri del gruppo possiedano le competenze necessarie per svolgere i compiti assegnati e gestire le tecnologie richieste dal progetto.

4.4.1 Attività

La formazione del team si articola in un percorso strutturato che comprende:

- **Analisi dei bisogni**: identificare le competenze richieste dal progetto e le eventuali lacune del gruppo, partendo dai requisiti della proponente e dalle tecnologie scelte.
- **Formazione con la Proponente**: partecipazione a incontri tecnici dedicati, organizzati dalla proponente, sulle tecnologie cardine del progetto (Docker, FastAPI, LangChain, React). Queste sessioni accelerano la curva di apprendimento e allineano il gruppo con gli standard tecnici e le metodologie di sviluppo adottate dall'azienda.
- **Apprendimento individuale**: studio autonomo, da parte di ciascun membro, degli strumenti, linguaggi e framework necessari per il proprio ruolo e per le attività comuni.
- **Condivisione delle conoscenze** (Knowledge Sharing): diffusione interna delle competenze, facilitata dai membri più esperti in un dato ambito attraverso discussioni tecniche, sessioni di pair programming o documentazione informale.
- **Consolidamento e uniformità**: l'obiettivo è garantire un livello di preparazione omogeneo e adeguato agli obiettivi di progetto, promuovendo la crescita professionale di tutti i componenti.

4.4.2 Implementazione del processo

In base a quanto previsto dallo standard ISO/IEC 12207:1995, è necessario realizzare una revisione dei requisiti del progetto.
Questo passaggio serve a comprendere le competenze che i membri del NullPointers Group dovranno sviluppare per completare il progetto didattico.
Come gruppo abbiamo quindi definito quali sono le tecnologie necessarie da approfondire ed utilizzare per la corretta realizzazione del progetto.
Queste tecnologie sono:

1. **React**: framework JavaScript/TypeScript per la realizzazione dell'interfaccia web.
2. **Python**: linguaggio di programmazione versatile, utilizzato per implementare API REST, gestire le comunicazioni con modelli di machine learning e con il database, e realizzare il back-end.
3. **PostgreSQL**: sistema di gestione di database relazionale open source, utilizzato per memorizzare e gestire i dati dell'applicazione.
4. **Docker**: piattaforma per la containerizzazione, che consente di distribuire e eseguire l'applicazione in ambienti isolati e replicabili.
5. **LaTeX**: linguaggio di markup utilizzato per produrre la documentazione

5 Standard di Progetto

5.1 Standard ISO/IEC 12207 - 1995

5.1.1 Scopo dello standard

Lo standard ISO/IEC 12207 definisce un insieme strutturato di processi, attività e compiti che coprono l'intero ciclo di vita del software.
Fornisce un modello di riferimento per organizzare in modo sistematico le attività di sviluppo, verifica, gestione e manutenzione, garantendo coerenza e tracciabilità in tutte le fasi del progetto.
L'adozione di tale standard consente al gruppo di:

- definire responsabilità e flussi operativi chiari;
- assicurare un controllo costante sulla qualità dei processi;
- mantenere allineamento tra pianificazione, sviluppo e verifica;
- favorire la riproducibilità e la trasparenza del lavoro svolto.

5.1.2 Applicazione nel progetto SmartOrder

Per il progetto SmartOrder, lo standard ISO/IEC 12207 definisce un modello di riferimento per l'organizzazione del ciclo di vita del software.
La sua adozione consente di strutturare le attività del progetto in modo controllato e tracciabile, garantendo che ogni fase sia condotta secondo criteri uniformi di qualità.
Lo standard suddivide il ciclo di vita in tre categorie di processi:

- **Processi primari**: rappresentano le attività direttamente legate alla realizzazione del prodotto. Comprendono l'analisi e la definizione dei requisiti, la progettazione dell'architettura, lo sviluppo del software, le attività di verifica e validazione necessarie per garantire che il sistema soddisfi quanto concordato con la proponente. Attraverso questi processi, il progetto avanza in modo controllato verso la costruzione del prodotto finale.
- **Processi di supporto**: sono attività trasversali che affiancano e sostengono i processi primari. Includono la gestione della configurazione, il controllo della qualità, la produzione della documentazione e le attività di verifica. Il loro ruolo è assicurare ordine, coerenza e affidabilità durante tutte le fasi del progetto.
- **Processi organizzativi**: regolano la gestione complessiva del progetto, la pianificazione, il coordinamento interno e le attività di miglioramento. Attraverso questi processi vengono definiti i ruoli, le responsabilità, gli obiettivi e le modalità operative del gruppo, garantendo un ambiente di lavoro strutturato e orientato al raggiungimento dei risultati.

L'utilizzo di ISO/IEC 12207 permette di mantenere un controllo metodico sul ciclo di vita, particolarmente importante in un progetto che integra componenti

di intelligenza artificiale, modelli multimodali e pipeline di elaborazione complesse.

5.2 Standard ISO/IEC 9126

5.2.1 Scopo dello standard

Lo standard ISO/IEC 9126 definisce un modello strutturato per valutare la qualità di un prodotto software. Il suo scopo è trasformare il concetto generico di "qualità" in un insieme di caratteristiche misurabili, permettendo valutazioni oggettive e confrontabili.
Il modello identifica sei dimensioni fondamentali della qualità, che sono:

- **Funzionalità:** capacità del sistema di soddisfare requisiti e bisogni specificati.
- **Affidabilità:** misura la capacità del sistema di funzionare senza interruzioni e di resistere a condizioni anomale o a guasti.
- **Usabilità:** considera la facilità con cui un utente può imparare a usare il software e interagire con esso in modo efficace.
- **Efficienza:** si riferisce alle prestazioni del sistema e al suo utilizzo ottimale delle risorse hardware (come tempo di risposta e consumo di memoria).
- **Manutenibilità:** valuta la facilità con cui il software può essere modificato per correggere errori, migliorarlo o adattarlo a nuove esigenze.
- **Portabilità:** analizza la capacità del sistema di essere installato e di funzionare in ambienti diversi (ad esempio, su sistemi operativi diversi).

L'adozione di questo approccio garantisce che la qualità non sia un controllo solo finale, ma una caratteristica intrinseca, costruita e verificata progressivamente in ogni passaggio dello sviluppo.

5.2.2 Applicazione nel progetto SmartOrder

Nel progetto SmartOrder, questo standard costituisce il quadro di riferimento principale per definire gli obiettivi di qualità del prodotto e le relative metriche (dettagliate nella Sezione 6). Viene applicato per guidare lo sviluppo verso traguardi concreti:

- validare la correttezza dell'elaborazione multimodale (testi, immagini, audio);
- garantire robustezza e stabilità all'interno dei moduli di AI e NLP;
- assicurare tempi di risposta adeguati per un'interazione fluida con l'utente;
- preservare la separazione e la modularità tra i componenti, semplificando manutenzione e futuri aggiornamenti;
- offrire un'interfaccia chiara e intuitiva nonostante la complessità architetturale del sistema.

L'adozione dello standard permette quindi di stabilire criteri di qualità precisi e misurabili, supportando sia la verifica finale che il monitoraggio continuo durante tutto lo sviluppo.

5.3 Motivazione dell'adozione degli standard

La scelta di adottare gli standard ISO/IEC 12207 (per il processo) e ISO/IEC 9126 (per il prodotto) porta dei vantaggi strategici al progetto SmartOrder:

- consente di organizzare tutte le attività, dall'analisi iniziale fino al rilascio finale;
- permette di esercitare un controllo accurato e continuo sulla qualità sia del processo di sviluppo sia del prodotto realizzato;
- introduce metriche oggettive, misurabili e verificabili a supporto della valutazione delle prestazioni;
- supporta una gestione metodica di componenti tecnologicamente avanzati, tra cui modelli di intelligenza artificiale e pipeline multimodali;
- garantisce affidabilità, manutenibilità e coerenza nel progetto.

L'adozione di tali standard costituisce quindi un fondamento essenziale per mantenere coerenza, trasparenza e qualità nello sviluppo di SmartOrder.

6 Metriche di Qualità

Le metriche rappresentano uno strumento essenziale per garantire un monitoraggio continuo e oggettivo della qualità del prodotto software e dei processi impiegati per realizzarlo.
Permettono di individuare tempestivamente anomalie, inefficienze o deviazioni dagli standard prefissati, favorendo una gestione consapevole dell'evoluzione del progetto.
Per garantire chiarezza e coerenza interna, ogni metrica è classificata secondo la seguente notazione:

- MQC_xx - Metriche di qualità del processo
- MQD_xx - Metriche di qualità del prodotto

6.1 Metriche di qualità di Processo

Le metriche qui definite consentono di monitorare l'andamento dei processi primari e di supporto, permettendo di verificare l'aderenza alla pianificazione, l'efficacia delle attività svolte e la qualità complessiva del ciclo di sviluppo.

6.1.1 Processi Primari

- **MQC_01 - Earned Value**:
  - **Formula**:

$$\text{Earned Value} = \text{Planned Value} \times \text{Completeness Issue}$$

  - **Descrizione**: misura il valore del lavoro effettivamente completato rispetto alla pianificazione. Riflette l'avanzamento reale del progetto in termini economici ed è utile per identificare scostamenti precoci rispetto al piano.
- **MQC_02 - Planned Value**:
  - **Formula**:

$$\text{Planned Value} = \text{Costo preventivato}$$

  - **Descrizione**: indica il valore del lavoro che avrebbe dovuto essere completato in un dato momento secondo la pianificazione. È utilizzato come riferimento per confrontare l'avanzamento reale con quello previsto.
- **MQC_03 - Actual Cost**:
  - **Formula**:

$$\text{Actual Cost} = \text{Consuntivo}$$

  - **Descrizione**: rappresenta il costo reale sostenuto fino a una certa data. Confrontato con l'Earned Value consente di valutare l'efficienza economica del progetto.

- **MQC_04 - Cost Performance Index**:
  - **Formula**:

$$\text{Cost Performance Index} = \frac{\text{Earned Value}}{\text{Actual Cost}}$$

  - **Descrizione**: Misura l'efficienza economica del progetto confrontando il valore realizzato con il costo effettivamente sostenuto.
    Un valore maggiore di 1 indica che il lavoro è stato svolto a un costo inferiore al previsto (sotto budget), un valore uguale a 1 indica che i costi sono in linea con le previsioni, mentre un valore inferiore a 1 segnala che il costo supera il previsto (sopra budget).
    Valori persistentemente inferiori a 1 richiedono interventi correttivi per riallineare il progetto al budget.

- **MQC_05 - Schedule Performance Index**:
  - **Formula**:

$$\text{Schedule Performance Index} = \frac{\text{Earned Value}}{\text{Planned Value}}$$

  - **Descrizione**: valuta l'efficacia dell'avanzamento temporale. Un valore pari o superiore a 1 indica che il progetto è in linea con la pianificazione o in anticipo rispetto ad essa.

- **MQC_06 - Estimate at Completion**:
  - **Formula**:

$$\text{Estimate at Completion} = \frac{\text{Budget at Completion}}{\text{Cost Performance Index}}$$

  - **Descrizione**: stima il costo totale previsto a completamento. Viene utilizzato per prevedere scostamenti rispetto al budget pianificato e per aggiornare le proiezioni economiche.

- **MQC_07 - Estimate to Complete**:
  - **Formula**:

$$\text{Estimate to Complete} = \text{Estimate at Completion} - \text{Actual Cost}$$

  - **Descrizione**: stima il costo necessario per completare le attività rimanenti. Consente di valutare l'impatto delle performance correnti sul budget finale.

- **MQC_08 - Time Estimate at Completion**:
  - **Formula**:
$$\text{Time Estimate at Completion} = \frac{\text{Tempo previsto}}{\text{Schedule Performance Index}}$$
  - **Descrizione**: indica la data stimata di completamento del progetto sulla base dell'andamento attuale. È utile per valutare possibili ritardi e adottare misure correttive.
- **MQC_09 - Completeness Issue**:
  - **Formula**:
$$\text{Completeness Issue} = \frac{\text{Issue chiuse}}{\text{Issue totali}}$$
  - **Descrizione**: misura il grado di completamento delle attività pianificate, considerando il rapporto tra le issue chiuse e quelle registrate. Un valore elevato indica un buon avanzamento delle attività di sviluppo.

6.1.2 Processi di Supporto

- **MQC_10 - Indice di Gulpease**:
  - **Formula**:
$$\text{Indice di Gulpease} = 89 + \frac{300 \times \text{frasi} - 10 \times \text{lettere}}{\text{parole}}$$
  - **Descrizione**: valuta la leggibilità dei documenti prodotti. Un indice elevato indica testi più chiari e facilmente comprensibili, migliorando la qualità complessiva della documentazione.
- **MQC_11 - Test Success Rate**:
  - **Formula**:
$$\text{Test Success Rate} = \frac{\text{Test superati}}{\text{Test totali}} \times 100$$
  - **Descrizione**: indica la percentuale di test automatizzati o manuali che si concludono con esito positivo. Riflette la stabilità delle funzionalità verificate.
- **MQC_12 - Test Density and Automation**:
  - **Formula**:
$$\text{Test Density and Automation} = \frac{\text{Test automatizzati}}{\text{Test totali}} \times 100$$
  - **Descrizione**: misura il grado di automazione del processo di verifica. Un'elevata densità di test automatizzati migliora l'affidabilità della pipeline di testing e riduce i tempi di validazione.

6.1.3 Processi Organizzativi

- **MQC_13 - Quality Metrics Satisfied**:
  - **Formula**:

$$\text{Quality Metrics Satisfied} = \frac{\text{Metriche Rispettate}}{\text{Metriche Totali}} \times 100$$

  - **Descrizione**: valuta la percentuale di metriche qualitative che hanno raggiunto il valore accettabile o ottimale. È un indicatore complessivo dell'efficacia del processo di gestione della qualità.

6.2 Metriche di qualità di Prodotto

6.2.1 Funzionalità

- **MQD_01 - Requisiti obbligatori soddisfatti**:
  - **Formula**:

$$\text{Requisiti obbligatori soddisfatti} = \left(\frac{\text{Requisiti obbligatori soddisfatti}}{\text{Requisiti obbligatori totali}}\right) \times 100$$

  - **Descrizione**: misura la percentuale dei requisiti classificati come obbligatori che risultano implementati e verificati con successo. Una percentuale pari al 100% è condizione necessaria per garantire la correttezza funzionale minima del prodotto.
- **MQD_02 - Requisiti desiderabili soddisfatti**:
  - **Formula**:

$$\text{Requisiti desiderabili soddisfatti} = \left(\frac{\text{Requisiti desiderabili soddisfatti}}{\text{Requisiti desiderabili totali}}\right) \times 100$$

  - **Descrizione**: indica il grado di completamento dei requisiti desiderabili, utili a migliorare l'esperienza d'uso o arricchire il sistema con funzionalità non essenziali.
- **MQD_03 - Requisiti opzionali soddisfatti**:
  - **Formula**:

$$\text{Requisiti opzionali soddisfatti} = \left(\frac{\text{Requisiti opzionali soddisfatti}}{\text{Requisiti opzionali totali}}\right) \times 100$$

  - **Descrizione**: valuta l'implementazione di funzionalità opzionali che aggiungono valore al sistema ma non influiscono sulla sua correttezza di base.

6.2.2 Affidabilità

- **MQD_04 - Code Coverage**:
  - **Formula**:
    $$\text{Code Coverage} = \frac{\text{LOC testate}}{\text{LOC totali}} \times 100$$
    Dove LOC = Lines of code
  - **Descrizione**: misura la percentuale di linee di codice che risultano coperte da test automatici. Una copertura elevata contribuisce a ridurre la probabilità di malfunzionamenti e garantisce maggiore stabilità nelle fasi di manutenzione.
- **MQD_05 - Branch Coverage**:
  - **Formula**:
    $$\text{Branch Coverage} = \frac{\text{Rami coperti}}{\text{Rami totali}} \times 100$$
  - **Descrizione**: misura la percentuale di rami decisionali testati. È una metrica fondamentale per verificare la robustezza del sistema in condizioni operative variabili.
- **MQD_06 - Statement Coverage**:
  - **Formula**:
    $$\text{Statement Coverage} = \frac{\text{Istruzioni eseguite}}{\text{Istruzioni totali}} \times 100$$
  - **Descrizione**: misura la percentuale di istruzioni eseguibili del codice che vengono effettivamente eseguite durante l'esecuzione della suite di test. Essa consente di valutare quanto i test riescano a coprire il flusso operativo del software, identificando eventuali sezioni di codice non esercitate e quindi potenzialmente soggette a difetti non rilevati.

6.2.3 Usabilità

- **MQD_07 - Time on Task**:
  - **Formula**:
    $$\text{Time on Task} = \frac{\sum_{i=1}^{n} T_i}{n}$$
    dove:
    T_i = tempo impiegato dall'utente i per completare il task specifico
    n = numero di utenti che hanno completato il task
  - **Descrizione**: misura quanto tempo, in media, un utente impiega a completare un'attività specifica nel sistema. Serve a capire se l'interfaccia è facile da usare e se ci sono passaggi che rallentano l'utente.

6.2.4 Efficienza

- **MQD_08 - Response time**:
  - **Formula**:

$$\text{Response time} = \text{Tempo risposta medio}$$

  - **Descrizione**: misura il tempo di risposta del sistema alle interazioni dell'utente. Un valore ridotto è indicatore di un'applicazione performante e reattiva.

6.2.5 Manutenibilità

- **MQD_09 - Code Smells per KLOC**:
  - **Formula**:

$$\text{Code Smells per KLOC} = \frac{\text{Numero di code smells}}{\text{KLOC}}$$

    Dove KLOC = Kilo Lines of Code
  - **Descrizione**: valuta la qualità del codice attraverso l'analisi dei "code smell", ovvero pattern che segnalano complessità o debolezza nella struttura del software. Un numero contenuto favorisce la manutenibilità e la pulizia architetturale.
- **MQD_10 - Coefficient of Coupling**:
  - **Formula**:

$$\text{Coefficient of Coupling} = \frac{\text{Dipendenze esterne}}{\text{Moduli totali}}$$

  - **Descrizione**: misura il livello di accoppiamento tra i moduli del sistema. Un basso accoppiamento è desiderabile perché rende il prodotto più stabile, testabile e semplice da estendere nel tempo.
- **MQD_11 - Cyclomatic Complexity**:
  - **Formula**:

$$\text{Cyclomatic Complexity} = E - N + 2P$$

    Dove:
    - E: numero di archi
    - N: numero di nodi
    - P: componenti connessi
  - **Descrizione**: misura la complessità del flusso di controllo del codice contando i percorsi logicamente indipendenti. Valori elevati indicano funzioni difficili da comprendere e testare.