

# Norme di Progetto

v0.2



7Last



## Versioni

Ver.	Data	Autore	Verificatore	Descrizione
0.2	05/04/2024	Matteo Tiozzo		Modificato tabella versioni
0.1	29/03/2024	Raul Seganfreddo		Creazione struttura documento

# Indice

<b>1</b>	<b>Introduzione</b>	<b>8</b>
1.1	Scopo del documento . . . . .	8
1.2	Scopo del progetto . . . . .	8
1.3	Glossario . . . . .	9
1.4	Riferimenti . . . . .	9
1.4.1	Riferimenti normativi . . . . .	9
1.4.2	Riferimenti informativi . . . . .	9
<b>2</b>	<b>Processi primari</b>	<b>10</b>
2.1	Fornitura . . . . .	10
2.1.1	Introduzione . . . . .	11
2.1.2	Attività . . . . .	11
2.1.3	Comunicazioni con l'azienda proponente . . . . .	11
2.1.4	Documentazione fornita . . . . .	11
2.1.4.1	Valutazione dei capitolati . . . . .	11
2.1.4.2	Analisi dei requisiti . . . . .	11
2.1.4.3	Piano di progetto . . . . .	11
2.1.4.4	Piano di qualifica . . . . .	11
2.1.4.5	Glossario . . . . .	11
2.1.4.6	Lettera di presentazione . . . . .	11
2.1.5	Strumenti . . . . .	11
2.2	Sviluppo . . . . .	11
2.2.1	Introduzione . . . . .	11
2.2.2	Analisi dei requisiti . . . . .	11
2.2.2.1	Descrizione . . . . .	11
2.2.2.2	Obiettivi . . . . .	11
2.2.2.3	Documentazione . . . . .	11
2.2.2.4	Casi d'uso . . . . .	11
2.2.2.5	Diagrammi dei casi d'uso . . . . .	11
2.2.2.6	Requisiti . . . . .	11
2.2.2.7	Metriche . . . . .	11
2.2.2.8	Strumenti . . . . .	11
2.2.3	Progettazione . . . . .	11
2.2.3.1	Descrizione . . . . .	11



2.2.3.2	Obiettivi . . . . .	11
2.2.3.3	Documentazione . . . . .	11
2.2.3.4	Qualità dell'architettura . . . . .	11
2.2.3.5	Diagrammi UML . . . . .	11
2.2.3.6	Design pattern . . . . .	11
2.2.3.7	Test . . . . .	11
2.2.3.8	Metriche . . . . .	11
2.2.3.9	Strumenti . . . . .	11
2.2.4	Codifica . . . . .	11
2.2.4.1	Descrizione . . . . .	11
2.2.4.2	Obiettivi . . . . .	11
2.2.4.3	Norme di codifica . . . . .	11
2.2.4.4	Strumenti . . . . .	11
2.2.4.5	Metriche . . . . .	11
2.2.5	Configurazione dell'ambiente di esecuzione . . . . .	11
2.2.5.1	Docker . . . . .	11
2.2.5.2	Strumenti . . . . .	11

### **3 Processi di supporto 13**

3.1	Documentazione . . . . .	13
3.1.1	Introduzione . . . . .	13
3.1.2	Documentation as Code . . . . .	13
3.1.3	Sorgente documenti . . . . .	13
3.1.4	Ciclo di vita dei documenti . . . . .	13
3.1.5	Procedure correlate alla redazione di documenti . . . . .	13
3.1.5.1	I redattori . . . . .	13
3.1.5.2	I verificatori . . . . .	13
3.1.5.3	Il responsabile . . . . .	13
3.1.5.4	L'amministratore . . . . .	13
3.1.6	Struttura del documento . . . . .	13
3.1.6.1	Registro delle modifiche . . . . .	13
3.1.6.2	Prima pagina . . . . .	13
3.1.6.3	Indice . . . . .	13
3.1.6.4	Piè di pagina . . . . .	13
3.1.6.5	Verbali: struttura generale . . . . .	13
3.1.7	Norme tipografiche . . . . .	13



3.1.8	Abbreviazioni . . . . .	13
3.1.9	Strumenti . . . . .	13
3.2	Verifica . . . . .	13
3.2.1	Introduzione . . . . .	13
3.2.2	Verifica dei documenti . . . . .	13
3.2.3	Analisi . . . . .	13
3.2.3.1	Analisi statica . . . . .	13
3.2.3.2	Walkthrough . . . . .	13
3.2.3.3	Inspection . . . . .	13
3.2.3.4	Analisi dinamica . . . . .	13
3.2.4	Testing . . . . .	13
3.2.4.1	Test di unità . . . . .	13
3.2.4.2	Test di integrazione . . . . .	13
3.2.4.3	Test di sistema . . . . .	13
3.2.4.4	Test di regressione . . . . .	13
3.2.4.5	Test di accettazione . . . . .	13
3.2.4.6	Sequenza delle fasi di test . . . . .	13
3.2.4.7	Codici dei test . . . . .	13
3.2.4.8	Stato dei test . . . . .	13
3.2.4.9	Continuous Integration . . . . .	13
3.2.5	Strumenti . . . . .	13
3.3	Validazione . . . . .	13
3.3.1	Introduzione . . . . .	13
3.3.2	Procedura di validazione . . . . .	13
3.3.3	Strumenti . . . . .	13
3.4	Gestione della configurazione . . . . .	13
3.4.1	Introduzione . . . . .	13
3.4.2	Versionamento . . . . .	13
3.4.3	Repository . . . . .	13
3.4.3.1	Struttura repository . . . . .	13
3.4.4	Sincronizzazione e branching . . . . .	13
3.4.4.1	Documentazione . . . . .	13
3.4.4.2	Sviluppo . . . . .	13
3.4.4.3	Pull Request . . . . .	13
3.4.5	Controllo di configurazione . . . . .	13
3.4.5.1	Change Request . . . . .	13



3.4.6	Configuration Status Accounting . . . . .	13
3.4.7	Release management and delivery . . . . .	13
3.4.7.1	Procedura per la creazione di una release . . . . .	13
3.4.8	Strumenti . . . . .	13
3.5	Joint review . . . . .	13
3.5.1	Introduzione . . . . .	13
3.5.2	Implementazione del processo . . . . .	13
3.5.2.1	Revisioni periodiche . . . . .	13
3.5.2.2	SAL . . . . .	13
3.5.2.3	Revisioni ad hoc . . . . .	13
3.5.2.4	Risorse per le revisioni . . . . .	13
3.5.2.5	Elementi da concordare . . . . .	13
3.5.2.6	Documenti e distribuzione dei risultati . . . . .	13
3.5.3	Project management reviews . . . . .	13
3.5.3.1	Introduzione . . . . .	13
3.5.3.2	Stato del progetto . . . . .	13
3.5.4	Revisioni tecniche . . . . .	13
3.5.5	Strumenti . . . . .	13
3.6	Risoluzione dei problemi . . . . .	13
3.6.1	Introduzione . . . . .	13
3.6.2	Gestione dei rischi . . . . .	13
3.6.2.1	Codifica dei rischi . . . . .	13
3.6.2.2	Metriche . . . . .	13
3.6.3	Identificazione dei problemi . . . . .	13
3.6.4	Strumenti . . . . .	13
3.7	Gestione della qualità . . . . .	13
3.7.1	Introduzione . . . . .	13
3.7.2	Attività . . . . .	13
3.7.3	Piano di qualifica . . . . .	13
3.7.4	PDCA . . . . .	13
3.7.5	Strumenti . . . . .	13
3.7.6	Struttura e identificazioni metriche . . . . .	13
3.7.7	Criteri di accettazione . . . . .	13
3.7.8	Metriche . . . . .	13

## 4 Processi organizzativi

**14**



4.1	Gestione dei processi . . . . .	14
4.1.1	Introduzione . . . . .	14
4.1.2	Pianificazione . . . . .	15
4.1.2.1	Descrizione . . . . .	15
4.1.2.2	Obiettivi . . . . .	15
4.1.2.3	Assegnazione dei ruoli . . . . .	15
4.1.2.4	Responsabile . . . . .	16
4.1.2.5	Amministratore . . . . .	16
4.1.2.6	Analista . . . . .	17
4.1.2.7	Progettista . . . . .	17
4.1.2.8	Programmatore . . . . .	17
4.1.2.9	Verificatore . . . . .	18
4.1.2.10	Ticketing . . . . .	18
4.1.2.11	Strumenti . . . . .	20
4.1.3	Coordinamento . . . . .	20
4.1.3.1	Descrizione . . . . .	20
4.1.3.2	Obiettivi . . . . .	21
4.1.3.3	Comunicazioni sincrone . . . . .	21
4.1.3.4	Comunicazioni asincrone . . . . .	21
4.1.3.5	Riunioni interne . . . . .	22
4.1.3.6	Riunioni esterne . . . . .	23
4.1.3.7	Strumenti . . . . .	23
4.1.3.8	Metriche . . . . .	24
4.2	Miglioramento . . . . .	24
4.2.1	Introduzione . . . . .	24
4.2.2	Analisi . . . . .	24
4.2.3	Miglioramento . . . . .	24
4.3	Formazione . . . . .	25
4.3.1	Introduzione . . . . .	25
4.3.2	Metodo di formazione . . . . .	25
4.3.2.1	Individuale . . . . .	25
4.3.2.2	Gruppo . . . . .	25
<b>5</b>	<b>Standard per la qualità</b>	<b>26</b>
5.1	Caratteristiche del sistema, STANDARD CHE USEREMO NOI . . . . .	26
5.1.1	Funzionalità . . . . .	26

5.1.2	Affidabilità . . . . .	26
5.1.3	Usabilità . . . . .	26
5.1.4	Efficienza . . . . .	26
5.1.5	Manutenibilità . . . . .	26
5.1.6	Portabilità . . . . .	26
5.2	Suddivisione secondo standard, STANDARD CHE USEREMO NOI . . . . .	26
5.2.1	Processi primari . . . . .	26
5.2.2	Processi di supporto . . . . .	26
5.2.3	Processi organizzativi . . . . .	26
<b>6</b>	<b>Metriche di qualità</b>	<b>27</b>
6.1	Metriche per la qualità di processo . . . . .	27
6.2	Metriche per la qualità di prodotto . . . . .	27

## Elenco delle tabelle

## Elenco delle figure





# 1 Introduzione

## 1.1 Scopo del documento

Questo documento ha lo scopo di descrivere le regole e le procedure che il gruppo adotterà per lo svolgimento del progetto. Lo scopo quindi è quello di definire il *Way of Working* del gruppo, in modo da garantire un lavoro efficiente e di qualità.

Il processo seguirà le linee guida descritte dallo standard ISO/IEC 12207:1995.

## 1.2 Scopo del progetto

Lo scopo del progetto è quello di realizzare una piattaforma di monitoraggio per una smart city, in grado di raccogliere e analizzare in tempo reale dati provenienti da diverse fonti, come sensori, dispositivi indossabili e macchine. La piattaforma avrà lo scopo di:

- Migliorare la qualità della vita dei cittadini: la piattaforma consentirà alle autorità locali di prendere decisioni informate e tempestive sulla gestione delle risorse e sull'implementazione di servizi, basandosi su dati reali e aggiornati.
- Coinvolgere i cittadini: i dati monitorati saranno resi accessibili al pubblico attraverso portali online e applicazioni mobili, permettendo ai cittadini di essere informati sullo stato della loro città e di partecipare attivamente alla sua gestione.
- Gestire il big data: la piattaforma sarà in grado di gestire grandi volumi di dati provenienti da diverse fonti, aggregandoli, normalizzandoli e analizzandoli per estrarre informazioni significative.

La piattaforma si baserà su tecnologie di data streaming processing per l'analisi in tempo reale dei dati e su una piattaforma OLAP per la loro archiviazione e visualizzazione. La parte "IoT" del progetto sarà simulata attraverso tool di generazione di dati realistici.

In sintesi, il progetto mira a creare una piattaforma che sia:

- Efficiente: in grado di raccogliere e analizzare grandi volumi di dati in tempo reale.
- Efficace: in grado di fornire informazioni utili per la gestione della città e il miglioramento della qualità della vita dei cittadini.
- Accessibile: in grado di rendere i dati disponibili al pubblico in modo chiaro e comprensibile.

Il progetto si pone come obiettivo di contribuire allo sviluppo di città più intelligenti e sostenibili, in cui la tecnologia viene utilizzata per migliorare il benessere dei cittadini.



## 1.3 Glossario

Al fine di evitare ambiguità e di facilitare la comprensione del documento, si allega il *Glossario* contenente la definizione dei termini tecnici e degli acronimi utilizzati.

## 1.4 Riferimenti

### 1.4.1 Riferimenti normativi

- Glossario: **TODO inserire link**
- *ISO/IEC 12207:1995*: **TODO inserire link**

### 1.4.2 Riferimenti informativi



## 2 Processi primari

### 2.1 Fornitura

Il processo di fornitura ha lo scopo di stabilire un accordo contrattuale tra il fornitore e il cliente, in cui vengono definiti i servizi che il fornitore si impegna a fornire e le condizioni di fornitura. Il processo di fornitura comprende le seguenti attività:

- **Preparazione della proposta**
- **Contrattazione**
- **Pianificazione**
- **Esecuzione**
- **Revisione**
- **Consegna**

TODO Mettere una descrizione di ogni attività?



### **2.1.1 Introduzione**

### **2.1.2 Attività**

### **2.1.3 Comunicazioni con l'azienda proponente**

### **2.1.4 Documentazione fornita**

#### **2.1.4.1 Valutazione dei capitolati**

#### **2.1.4.2 Analisi dei requisiti**

#### **2.1.4.3 Piano di progetto**

#### **2.1.4.4 Piano di qualifica**

#### **2.1.4.5 Glossario**

#### **2.1.4.6 Lettera di presentazione**

### **2.1.5 Strumenti**

## **2.2 Sviluppo**

### **2.2.1 Introduzione**

### **2.2.2 Analisi dei requisiti**

#### **2.2.2.1 Descrizione**

#### **2.2.2.2 Obiettivi**

#### **2.2.2.3 Documentazione**

#### **2.2.2.4 Casi d'uso**

#### **2.2.2.5 Diagrammi dei casi d'uso**

#### **2.2.2.6 Requisiti**

#### **2.2.2.7 Metriche**

#### **2.2.2.8 Strumenti**

### **2.2.3 Progettazione**

#### **2.2.3.1 Descrizione**

#### **2.2.3.2 Obiettivi**

#### **2.2.3.3 Documentazione**

#### **2.2.3.4 Qualità dell'architettura**





## **3 Processi di supporto**

### **3.1 Documentazione**

#### **3.1.1 Introduzione**

#### **3.1.2 Documentation as Code**

#### **3.1.3 Sorgente documenti**

#### **3.1.4 Ciclo di vita dei documenti**

#### **3.1.5 Procedure correlate alla redazione di documenti**

##### **3.1.5.1 I redattori**

##### **3.1.5.2 I verificatori**

##### **3.1.5.3 Il responsabile**

##### **3.1.5.4 L'amministratore**

#### **3.1.6 Struttura del documento**

##### **3.1.6.1 Registro delle modifiche**

##### **3.1.6.2 Prima pagina**

##### **3.1.6.3 Indice**

##### **3.1.6.4 Piè di pagina**

##### **3.1.6.5 Verballi: struttura generale**

#### **3.1.7 Norme tipografiche**

#### **3.1.8 Abbreviazioni**

#### **3.1.9 Strumenti**

### **3.2 Verifica**

#### **3.2.1 Introduzione**

#### **3.2.2 Verifica dei documenti**

#### **3.2.3 Analisi**

##### **3.2.3.1 Analisi statica**

##### **3.2.3.2 Walkthrough**

##### **3.2.3.3 Inspection**



## 4 Processi organizzativi

Lo sviluppo software è un processo complesso e multidisciplinare che richiede una pianificazione, una gestione del tempo e delle risorse accurata, efficiente ed efficace. L'adozione di processi organizzativi ben strutturati è punto cruciale per garantire il successo dello sviluppo software.

### 4.1 Gestione dei processi

#### 4.1.1 Introduzione

La gestione dei processi si occupa di determinare, migliorare, ottimizzare i processi che fanno da guida alla realizzazione del software. Le attività di gestione dei processi sono:

- **Definizione dei processi:**
  - Identificare e documentare i processi chiave coinvolti nello sviluppo software.
  - Stabilire le linee guida e procedure per l'esecuzione di ciascun processo.
- **Pianificazione e monitoraggio:**
  - Elaborare piani dettagliati per l'esecuzione dei processi.
  - Monitorare costantemente l'avanzamento, l'efficacia e la conformità ai requisiti pianificati.
  - Stimare i tempi, le risorse ed i costi.
- **Valutazione e miglioramento continuo**
  - Condurre valutazioni periodiche dei processi per identificare aree di miglioramento.
  - Implementare azioni correttive e preventive per ottimizzare i processi.
- **Formazione e Competenze**
  - Assicurare che il personale coinvolto nei processi sia adeguatamente formato.
  - Mantenere e sviluppare le competenze necessarie per l'efficace gestione dei processi.
- **Gestione dei rischi**



- Identificare e valutare i rischi associati ai processi.
- Definire le strategie per mitigare o gestire i rischi identificati.

## **4.1.2 Pianificazione**

### **4.1.2.1 Descrizione**

La pianificazione riveste un ruolo centrale nella gestione dei processi, poiché mira a creare un piano organizzato e coerente per assicurare un'efficace esecuzione delle attività durante l'intero ciclo di vita del software. Il responsabile del progetto assume il compito di coordinare ogni aspetto della pianificazione delle attività, che include l'allocatione delle risorse, la definizione dei tempi e la redazione di piani dettagliati. Inoltre, il responsabile si assicura che il piano elaborato sia fattibile e possa essere eseguito correttamente ed efficientemente dai membri del team. I piani associati all'esecuzione del processo devono comprendere descrizioni dettagliate delle attività e delle risorse necessarie, specificando le tempistiche, le tecnologie impiegate, le infrastrutture coinvolte e il personale assegnato.

### **4.1.2.2 Obiettivi**

L'obiettivo primario della pianificazione è assicurare che ciascun membro del team assuma ogni ruolo almeno una volta durante lo svolgimento del progetto, promuovendo così una distribuzione equa delle responsabilità e un arricchimento delle competenze all'interno del team. La pianificazione, stilata dal responsabile, è integrata nel documento del Piano di Progetto. Questo documento fornisce una descrizione completa delle attività e dei compiti necessari per raggiungere gli obiettivi prefissati in ogni periodo del progetto.

### **4.1.2.3 Assegnazione dei ruoli**

Durante l'intero periodo del progetto, i membri del gruppo assumeranno sei ruoli distinti, ovvero assumeranno le responsabilità e svolgeranno le mansioni tipiche dei professionisti nel campo dello sviluppo software. Nei successivi paragrafi sono descritti in dettaglio i seguenti ruoli:

- Responsabile
- Amministratore





- Analista
- Progettista
- Programmatore
- Verificatore

#### **4.1.2.4 Responsabile**

Figura fondamentale che coordina il gruppo, fungendo da punto di riferimento per il committente e il team, svolgendo il ruolo di mediatore tra le due parti. In particolare si occupa di:

- Gestire le relazioni con l'esterno.
- Pianificare le attività: quali svolgere, data di inizio e fine, assegnazione delle priorità.
- Valutare i rischi delle scelte da effettuare.
- Controllare i progressi del progetto.
- Gestire le risorse umane.
- Approvazione della documentazione.

#### **4.1.2.5 Amministratore**

Questa figura professionale è incaricata del controllo e dell'amministrazione dell'ambiente di lavoro utilizzato dal gruppo ed è anche il punto di riferimento per quanto concerne le norme di progetto. Le sue mansioni principali sono:

- Affrontare e risolvere le problematiche associate alla gestione dei processi.
- Gestire versionamento della documentazione.
- Gestire la configurazione del prodotto.
- Redigere ed attuare le norme e le procedure per la gestione della qualità.
- Amministrare le infrastrutture e i servizi per i processi di supporto.



#### 4.1.2.6 Analista

Figura professionale con competenze avanzate riguardo l'attività di analisi dei requisiti e il dominio applicativo del problema. Il suo ruolo è quello di identificare, documentare e comprendere a fondo le esigenze e le specifiche del progetto, traducendole in requisiti chiari e dettagliati. Si occupa di:

- Analizzare il contesto di riferimento, definire il problema in esame e stabilire gli obiettivi da raggiungere.
- Comprendere il problema e definire la complessità e i requisiti.
- Redigere il documento *Analisi dei requisiti*.
- Studiare i bisogni espliciti ed impliciti.

#### 4.1.2.7 Progettista

Il progettista è la figura di riferimento per quanto riguarda le scelte progettuali partendo dal lavoro dell'analista. Spetta al progettista assumere decisioni di natura tecnica e tecnologica, oltre a supervisionare il processo di sviluppo. Tuttavia, non è responsabile della manutenzione del prodotto. In particolare si occupa di:

- Progettare l'architettura del prodotto secondo specifiche tecniche dettagliate.
- Prendere decisioni per sviluppare soluzioni che soddisfino i criteri di affidabilità, efficienza, sostenibilità e conformità ai requisiti.
- Redige la Specifica Architetture e la parte pragmatica del Piano di Qualifica.

#### 4.1.2.8 Programmatore

Il programmatore è la figura professionale incaricata della scrittura del codice software. Il suo compito primario è implementare il codice conformemente alle specifiche fornite dall'analista e all'architettura definita dal progettista. In particolare, il programmatore:

- Scrive codice manutenibile in conformità con le Specifiche Tecniche.
- Codifica le varie componenti dell'architettura seguendo quanto ideato dai progettisti.



- Realizza gli strumenti per verificare e validare il codice.
- Redige il *Manuale Utente*.

#### 4.1.2.9 Verificatore

La principale responsabilità del verificatore consiste nell'ispezionare il lavoro svolto da altri membri del team per assicurare la qualità e la conformità alle attese prefissate. Stabilisce se il lavoro è stato svolto correttamente sulla base delle proprie competenze tecniche, esperienza e conoscenza delle norme. In particolare il verificatore si occupa di:

- Verificare che il lavoro svolto sia conforme alle Norme di progetto.
- Verificare che il lavoro svolto sia conforme alle Specifiche Tecniche.
- Ricercare ed in caso segnalare eventuali errori.
- Redigere la sezione retrospettiva del *Piano di Qualifica*, descrivendo le verifiche e le prove effettuate durante il processo di sviluppo del prodotto.

#### 4.1.2.10 Ticketing

GitHub è adottato come sistema di tracciamento delle issue (ITS), garantendo così una gestione agevole e trasparente delle attività da svolgere. L'amministratore ha la facoltà di creare e assegnare specifiche issue sulla base delle attività identificate dal responsabile, assicurando chiarezza sulle responsabilità di ciascun membro del team e stabilendo tempi definiti entro cui ciascuna attività deve essere completata. Inoltre, ogni membro del gruppo può monitorare i progressi compiuti nel periodo corrente, consultando lo stato di avanzamento delle varie issue attraverso le Dashboard:

- DashBoard: per una panoramica dettagliata sullo stato delle issue.
- RoadMap: per una panoramica temporale dettagliata delle issue.

Procedura per la creazione delle issue:

Le issue vengono create dall'amministratore e devono essere specificati i seguenti attributi:

- Titolo: breve descrizione dell'attività da svolgere.



- Descrizione:
  - Descrizione testuale oppure “to-do” tramite bullet points.
  - Nell’ultima riga viene specificato il verificatore della issue nel formato: “Verificatore: Nome Cognome”.
- Assegnatario: membro del team responsabile dell’issue.
- Milestone: periodo di riferimento in cui l’attività deve essere completata.
- Labels: etichette per categorizzare le issue.  
Per associare ad ogni issue un Configuration Item vengono utilizzati i seguenti label:
  - **NdP**: Norme di Progetto
  - **PdP**: Piano di Progetto
  - **PdQ**: Piano di Qualifica
  - **AdR**: Analisi dei Requisiti
  - **PoC**: Proof of Concept
  - **Gls**: Glossario
- Milestone: milestone associata alla issue.
- Projects: progetti a cui la issue è associata. Se sono presenti dashboard associate ad un progetto, le issue correlate a tale progetto verranno visualizzate nella relativa/e dashboard di progetto.
- Development: branch e Pull Request associate alla issue. Quando una Pull Request viene accettata, la relativa issue viene automaticamente chiusa ed eventualmente spostata nella sezione “Done” della dashboard di progetto.

Ciclo di vita di una issue:

Il ciclo di vita è il seguente:

- Creazione: l’amministratore crea la issue e la assegna al membro del team responsabile.
- L’amministratore accede alla dashboard di progetto e sposta la issue dalla colonna “No Status” alla colonna “To Do”.



- L'assegnatario apre un branch su GitHub seguendo la denominazione suggerita in "Sincronizzazione e Branching".
- Quando la issue viene presa in carico dall'assegnatario, questo accede alla Dashboard e sposta la issue dalla colonna "To Do" alla colonna "In Progress".
- Una volta che la issue è considerata terminata, l'assegnatario apre una Pull Request su GitHub seguendo la convenzione descritta in dettaglio nella sezione "Procedura per la creazione di Pull Request".
- All'interno della Dashboard GitHub la issue deve essere spostata dalla colonna "In Progress" alla colonna "Da revisionare".
- il verificatore o i verificatori designati seguono le procedure esposte nella sezione 3.2 per verificare le modifiche apportate al progetto.
- Se la verifica ha esito positivo, la issueG viene trasferita dalla colonna "Da revisionare" alla colonna "Done" della Dashboard di GitHub. Nel caso in cui la issue sia associata ad una Pull Request, una volta che quest'ultima viene accettata dal verificatore, la issue viene automaticamente chiusa e spostata nella colonna "Done" della Dashboard di progetto.

#### 4.1.2.11 Strumenti

- GitHub: utilizzato per la condivisione del codice tra i membri del gruppo.
- ClickUp: piattaforma utilizzata per il tracciamento e la gestione delle issue e dei compiti.

### 4.1.3 Coordinamento

#### 4.1.3.1 Descrizione

Il coordinamento rappresenta l'attività che sovrintende la gestione della comunicazione e la pianificazione degli incontri tra le diverse parti coinvolte in un progetto di ingegneria del software. Questo comprende sia la gestione della comunicazione interna tra i membri del team del progetto, sia la comunicazione esterna con il proponente e i committenti. Il coordinamento risulta essere cruciale per assicurare che il progetto proceda in modo efficiente e che tutte le parti coinvolte siano informate e partecipino attivamente in ogni fase del progetto.



#### 4.1.3.2 Obiettivi

Il coordinamento in un progetto è fondamentale per gestire la comunicazione e pianificare gli incontri tra gli stakeholder. L'obiettivo principale è garantire efficienza, evitando ritardi e confusioni, assicurando che tutte le parti in causa siano informate e coinvolte in ogni fase del progetto. Inoltre, promuove la collaborazione e la coesione nel team, facilitando lo scambio di idee e la risoluzione dei problemi in modo collaborativo, creando un ambiente lavorativo positivo e produttivo.

#### Comunicazione

Il gruppo 7Last mantiene comunicazioni attive, sia interne che esterne al team, le quali possono essere sincrone o asincrone, a seconda delle necessità.

#### 4.1.3.3 Comunicazioni sincrone

- **Comunicazione sincrone interne**

Per le comunicazioni sincrone interne, il gruppo 7Last, ha scelto di adottare Discord in quanto permette di comunicare tramite chiamate vocali, videochiamate, messaggi di testo, media e file in chat private o come membri di un "server Discord".

- **Comunicazione sincrone esterne**

Per le comunicazioni sincrone esterne, in accordo con l'azienda proponente si è deciso di utilizzare un canale Discord.

#### 4.1.3.4 Comunicazioni asincrone

- **Comunicazione asincrone interne**

Per le comunicazioni asincrone interne, il gruppo 7Last, ha scelto di adottare Telegram in quanto permette di comunicare tramite messaggi di testo, media e file in chat private o come membri di un "gruppo Telegram".

- **Comunicazione asincrone esterne**

Per le comunicazioni asincrone esterne sono stati adottati due canali differenti:

- **Email:** per comunicazioni formali e ufficiali.
- **Discord**



#### 4.1.3.5 Riunioni interne

Si è scelto di svolgere i meeting interni a cadenza settimanale, al fine di facilitare una comunicazione costante e coordinare il progresso delle attività. Generalmente le riunioni sono programmate per ogni venerdì alle ore:

- Mercoledì dalle 15:00 - 16:00 per riunioni interne.

Se qualche membro del gruppo non può partecipare alla riunione nella data e nell'orario stabiliti, si procede programmando un nuovo incontro, concordando data e ora tramite un sondaggio sul canale Telegram dedicato. Ogni membro del gruppo ha la facoltà di richiedere una riunione supplementare se necessario. In questo caso, la data e l'orario saranno concordati sempre attraverso il canale Telegram dedicato, mediante la creazione di un sondaggio. Le riunioni interne rivestono un ruolo cruciale nel monitorare il progresso delle mansioni assegnate, valutare i risultati conseguiti e affrontare i dubbi e le difficoltà che possono sorgere. Durante i meeting interni, i membri del team condividono gli aggiornamenti sulle proprie attività, identificano le problematiche riscontrate e discutono di opportunità di miglioramento nei processi di lavoro. Questo ambiente aperto e collaborativo favorisce l'interazione, l'innovazione e la condivisione di nuove prospettive. Per agevolare la comunicazione sincrona, il canale utilizzato per i meeting interni è Discord, ritenuto particolarmente efficace per tali scopi. Relativamente ai meeting interni, sarà compito del responsabile:

- Stabilire preventivamente i principali temi da trattare durante la riunione, considerando la possibilità di aggiungerne di nuovi nel corso della riunione stessa.
- Guidare la discussione e raccogliere i pareri dei membri in maniera ordinata.
- Nominare un segretario per la riunione.
- Pianificare e proporre le nuove attività da svolgere.

#### Verballi interni

Lo svolgimento di una riunione interna ha come obiettivo la retrospettiva del periodo precedente, la discussione dei punti stilati nell'ordine del giorno e la pianificazione delle nuove attività. Alla conclusione di ciascuna riunione, l'amministratore apre un'issue nell'ITS di GitHub e assegna l'incarico di redigere il verbale interno al segretario della riunione. È compito quindi del segretario redigere il verbale, includendo tutte le informazioni rilevanti emerse durante la riunione. Le indicazioni dettagliate per la compilazione dei verbali interni sono disponibili nella sezione 3.1.6.5.



#### 4.1.3.6 Riunioni esterne

Durante lo svolgimento del progetto, è essenziale organizzare vari incontri con i Committenti e/o il Proponente al fine di valutare lo stato di avanzamento del prodotto e chiarire eventuali dubbi o questioni. La responsabilità di convocare tali incontri ricade sul responsabile, il quale è incaricato di pianificarli e agevolarne lo svolgimento in modo efficiente ed efficace. Sarà compito del responsabile anche l'esposizione dei punti di discussione al proponente/committente, lasciando la parola ai membri del gruppo interessati quando necessario. Questo approccio assicura una comunicazione efficace tra le varie parti in causa, garantendo una gestione ottimale del tempo e una registrazione accurata delle informazioni rilevanti emerse durante gli incontri. I membri del gruppo si impegnano a garantire la propria presenza in modo costante alle riunioni, facendo il possibile per riorganizzare eventuali altri impegni al fine di partecipare. Nel caso in cui gli obblighi inderogabili di un membro del gruppo rendessero impossibile la partecipazione, il responsabile assicurerà di informare tempestivamente il proponente o i committenti, richiedendo la possibilità di rinviare la riunione ad una data successiva.

##### **Riunioni con l'azienda proponente**

In accordo con l'azienda proponente, si è stabilito di tenere incontri di stato avanzamento lavori (SAL) con cadenza bisettimanale tramite Google Meet. Durante tali incontri, si affrontano diversi aspetti, tra cui:

- Discussione delle attività svolte nel periodo precedente, valutando l'aderenza a quanto concordato e identificando eventuali problematiche riscontrate.
- Pianificazione delle attività per il prossimo periodo, definendo gli obiettivi e le azioni necessarie per il loro raggiungimento.
- Chiarezza e risoluzione di eventuali dubbi emersi nel corso delle attività svolte.

##### **Verbalisti esterni**

Come nel caso delle riunioni interne, anche per le riunioni esterne verrà redatto un verbale con le stesse modalità descritte nella sezione relativa ai Verbalisti Interni. Le linee guida per la redazione dei verbalisti esterni sono reperibili alla sezione 3.1.6.5.

#### 4.1.3.7 Strumenti

- **Discord:** impiegato per la comunicazione sincrona e i meeting interni del team e per le riunioni esterne con il proponente.





- **Telegram:** utilizzato per la comunicazione asincrona interna.
- **Gmail:** come servizio di posta elettronica.

#### 4.1.3.8 Metriche

AGGIUNGERE TABELLA

## 4.2 Miglioramento

### 4.2.1 Introduzione

Secondo lo standard ISO/IEC 12207:1995, il processo di miglioramento nel ciclo di vita del software è finalizzato a stabilire, misurare, controllare e migliorare i processi che lo compongono. L'attività di miglioramento è composta da:

- **Analisi:** valutazione dei processi per identificare le aree di miglioramento.
- **Miglioramento:** attuazione di azioni correttive e preventive per ottimizzare i processi.

### 4.2.2 Analisi

Questa operazione richiede di essere eseguita regolarmente e ad intervalli di tempo appropriati e costanti. L'analisi fornisce un ritorno sulla reale efficacia e correttezza dei processi implementati, permettendo di identificare prontamente quelli che necessitano di miglioramenti. Durante ogni riunione, il team dedica inizialmente del tempo per condurre una retrospettiva sulle attività svolte nell'ultimo periodo. Questa pratica implica una riflessione approfondita su ciò che è stato realizzato, coinvolgendo tutti i membri nella identificazione delle aree di successo e di possibili miglioramenti. L'obiettivo principale è formulare azioni correttive da implementare nel prossimo sprint, promuovendo così un costante feedback e un adattamento continuo per migliorare le prestazioni complessive del team nel corso del tempo.

### 4.2.3 Miglioramento

Il team implementa le azioni correttive stabilite durante la retrospettiva, successivamente valuta la loro efficacia e le sottopone nuovamente a esame durante la retrospettiva successiva. L'esito di ogni azione correttiva sarà documentato nella sezione "Revisione del periodo precedente" di ogni verbale.



## **4.3 Formazione**

### **4.3.1 Introduzione**

L'obiettivo di questa iniziativa è stabilire standardG per il processo di apprendimento all'interno del team, assicurando la comprensione adeguata delle conoscenze necessarie per la realizzazione del progetto. Si prevede che il processo di formazione del Team assicuri che ciascun membro acquisisca una competenza adeguata per utilizzare consapevolmente le tecnologie selezionate dal gruppo per la realizzazione del progetto.

### **4.3.2 Metodo di formazione**

#### **4.3.2.1 Individuale**

Ciascun membro del team si impegnerà in un processo di autoformazione per adempiere alle attività assegnate al proprio ruolo. Durante la rotazione dei ruoli, ogni membro del gruppo condurrà una riunione con il successivo occupante del suo attuale ruolo, trasmettendo le conoscenze necessarie. Al contempo, terrà una riunione con chi ha precedentemente svolto il ruolo che esso assumerà, con l'obiettivo di apprendere le competenze richieste.

#### **4.3.2.2 Gruppo**

Sono programmate sessioni formative, condotte dalla proponente, al fine di trasferire competenze relative alle tecnologie impiegate nel contesto del progetto. La partecipazione del team a tali riunioni è obbligatoria.



## **5 Standard per la qualità**

Nel corso dell'analisi e della valutazione della qualità dei processi e del software, adotteremo standard internazionali ben definiti per garantire una valutazione rigorosa e conforme agli standard globali. In particolare, l'utilizzo dello standard ISO/IEC 9126 fornirà una solida struttura per valutare la qualità del software, concentrandosi su attributi quali la funzionalità, l'affidabilità, l'usabilità, l'efficienza, la manutenibilità e la portabilità. Questo framework ci consentirà di misurare in modo accurato e completo la qualità del prodotto software. Parallelamente, la suddivisione dei processi in primari, di supporto e organizzativi sarà guidata dall'adozione dello standard ISO/IEC 12207:1995. Infine, l'adozione dello standard ISO/IEC 25010 ci fornirà un quadro completo per la definizione e la suddivisione delle metriche di qualità del software. L'utilizzo congiunto di questi standard consentirà un approccio completo e strutturato alla valutazione della qualità dei processi e del software, assicurando un'elevata coerenza, affidabilità e conformità agli standard riconosciuti a livello internazionale.

### **5.1 Caratteristiche del sistema, STANDARD CHE USEREMO NOI**

#### **5.1.1 Funzionalità**

#### **5.1.2 Affidabilità**

#### **5.1.3 Usabilità**

#### **5.1.4 Efficienza**

#### **5.1.5 Manutenibilità**

#### **5.1.6 Portabilità**

### **5.2 Suddivisione secondo standard, STANDARD CHE USEREMO NOI**

#### **5.2.1 Processi primari**

#### **5.2.2 Processi di supporto**

#### **5.2.3 Processi organizzativi**



## **6 Metriche di qualità**

### **6.1 Metriche per la qualità di processo**

### **6.2 Metriche per la qualità di prodotto**