# Code7Crusaders

Software Development Team

**Norme Di Progetto**

**Membri del Team:**
Enrico Cotti Cottini, Gabriele Di Pietro, Tommaso Diviesti
Francesco Lapenna, Matthew Pan, Eddy Pinarello, Filippo Rizzolo

**Versioni**

| Ver | Data | Redattore | Verificatore | descrizione |
|---|---|---|---|---|
| 0.3 | 25/11/2024 | Matthew Pan | | Stesura sezione 3 |
| 0.3 | 21/11/2024 | Matthew Pan | | Completamento sezione 2 |
| 0.2 | 18/11/2024 | Matthew Pan | | Stesura sezione 2 |
| 0.1 | 12/11/2024 | Matthew Pan | Enrico Cotti Cottini | Prima stesura delle sezioni 1 e 2 |

# Indice


**1 Introduzione** **4**
- 1.1 Scopo del documento . . . 4
- 1.2 Scopo del progetto . . . 4
- 1.3 Glossario . . . 4
- 1.4 Riferimenti . . . 4
  - 1.4.1 Normativi . . . 4
  - 1.4.2 Informativi . . . 4

**2 Processi primari** **6**
- 2.1 Fornitura . . . 6
  - 2.1.1 Introduzione . . . 6
  - 2.1.2 Contatti con l'azienda proponente . . . 7
  - 2.1.3 Analisi dei Requisiti . . . 7
  - 2.1.4 Piano di Progetto . . . 7
  - 2.1.5 Piano di Qualifica . . . 8
  - 2.1.6 Glossario . . . 9
  - 2.1.7 Strumenti . . . 9
- 2.2 Sviluppo . . . 9
  - 2.2.1 Introduzione . . . 9
  - 2.2.2 Analisi dei Requisiti . . . 9
    - 2.2.2.1 Descrizione . . . 9
    - 2.2.2.2 Scopo . . . 10
    - 2.2.2.3 Codifica dei casi d'uso . . . 10
    - 2.2.2.4 Diagrammi Casi D'uso . . . 10
    - 2.2.2.5 Requisiti . . . 14
    - 2.2.2.6 Fonti dei requisiti . . . 15
    - 2.2.2.7 Codifica dei requisiti . . . 15
  - 2.2.3 Progettazione . . . 15
  - 2.2.4 Codifica . . . 16

**3 Processi di Supporto** **17**
- 3.1 Documentazione . . . 17
  - 3.1.1 Introduzione . . . 17
  - 3.1.2 Ciclo di Vita del Documento . . . 17
  - 3.1.3 Template . . . 17
  - 3.1.4 Documenti Prodotti . . . 17
  - 3.1.5 Struttura del Documento . . . 18
  - 3.1.6 Verbali . . . 19
  - 3.1.7 Nome del File . . . 19
  - 3.1.8 Stile del Testo . . . 19
  - 3.1.9 Glossario . . . 19
  - 3.1.10 Tabelle . . . 19
  - 3.1.11 Immagini . . . 19
- 3.2 Gestione della Configurazione . . . 19
  - 3.2.1 Versionamento . . . 20
  - 3.2.2 Repository . . . 20
- 3.3 Gestione della Qualità . . . 20

3.3.1 Descrizione . . . 20
3.3.2 Piano di Qualifica . . . 20
3.4 Verifica . . . 21
3.4.1 Introduzione . . . 21
3.4.2 Tipi di Verifica . . . 21
3.5 Validazione . . . 22
3.5.1 Introduzione . . . 22

**4 Processi Organizzativi** **23**
4.1 Gestione dei Processi . . . 23
4.1.1 Introduzione . . . 23
4.1.2 Pianificazione . . . 23
4.1.2.1 Descrizione . . . 23
4.1.2.2 Obiettivi . . . 23
4.1.3 Assegnazione dei Ruoli . . . 24
4.1.4 Ticketing . . . 24
4.1.5 Gestione dei rischi . . . 25
4.1.5.1 Struttura dei rischi . . . 25
4.2 Procedure Comunicative . . . 25
4.2.1 Comunicazioni Asincrone . . . 25
4.2.2 Comunicazioni Sincrone . . . 25
4.2.3 Riunioni Interne . . . 25
4.2.4 Riunioni Esterne . . . 26
4.3 Formazione . . . 26
4.3.1 Introduzione . . . 26
4.3.2 Metodo di formazione . . . 26

# Elenco delle tabelle

# Elenco delle figure

1 Esempio di attore . . . 11
2 Esempio di sistema . . . 11
3 Esempio di caso d'uso . . . 11
4 Esempio di sottocaso d'uso . . . 12
5 Esempio di associazione . . . 12
6 Esempio di Generalizzaione casi d'uso . . . 13
7 Esempio di inclusione . . . 13
8 Esempio di estensione . . . 14
9 Esempio di Generalizzaione attore . . . 14

# 1 Introduzione

## 1.1 Scopo del documento

Questo documento ha lo scopo di definire le regole e le procedure che ogni membro del team deve seguire durante lo sviluppo del progetto. In particolare, mira a stabilire il *Way of Working* del gruppo.

La sua redazione inizia nelle prime fasi del progetto e continua anche durante le fasi successive, per essere costantemente aggiornato e adattato alle esigenze del team.

Il processo seguirà le linee guida dello standard ISO/IEC 12207:1995, suddivise in:

- Processi primari
- Processi di supporto
- Processi organizzativi

## 1.2 Scopo del progetto

Il progetto si propone di sviluppare un Assistente Virtuale intelligente per aziende che operano nel settore della vendita multiprodotto. Questo assistente avrà il compito di semplificare l'accesso alle informazioni sui prodotti disponibili, rispondendo alle domande più frequenti poste dai clienti in modo rapido ed efficace.

Grazie all'uso di tecnologie avanzate come il Machine Learning[G] e il Natural Language Processing[G], il sistema sarà in grado di analizzare i dati contenuti nei cataloghi aziendali e negli archivi digitali, fornendo risposte precise e personalizzate.

L'obiettivo principale è ridurre la dipendenza dagli specialisti aziendali, che attualmente rappresentano l'unico canale di accesso per ottenere dettagli approfonditi sui prodotti. Questo migliorerà l'efficienza operativa, ottimizzerà le risorse e offrirà una migliore esperienza ai clienti, che potranno interagire con il sistema in modo intuitivo e diretto attraverso piattaforme digitali come siti web o chatbot.

In sintesi, il progetto intende rendere l'accesso alle informazioni aziendali più semplice, veloce e scalabile, migliorando al contempo la qualità del servizio offerto ai clienti.

## 1.3 Glossario

Per evitare ambiguità e facilitare la comprensione del documento, si farà uso di un glossario, contenente la definizione dei termini tecnici e degli acronimi utilizzati, che sarà incluso all'interno del file *glossario*.

## 1.4 Riferimenti

### 1.4.1 Normativi

- **Capitolato C7:**
  `https://www.math.unipd.it/~tullio/IS-1/2024/Progetto/C7.pdf`
- **ISO/IEC 12207:1995**
  `https://www.math.unipd.it/~tullio/IS-1/2009/Approfondimenti/ISO_12207-1995.pdf`

### 1.4.2 Informativi

- **Glossario RTB**
  `https://code7crusaders.github.io/docs/RTB/documentazione_interna/glossario.html`
- **Documentazione Git**
  `https://git-scm.com/docs`

- **Documentazione Latex**
  `https://www.latex-project.org/help/documentation/`

# 2 Processi primari

## 2.1 Fornitura

### 2.1.1 Introduzione

Il processo di fornitura rappresenta un percorso ben definito che stabilisce un contratto tra fornitore e cliente, accompagnando la creazione e la consegna del software. Fondamentale per garantire che il software risponda ai requisiti del cliente, rispetti i tempi e i costi, e soddisfi gli standard di qualità, il processo include anche un continuo dialogo tra le parti per chiarire le necessità, risolvere eventuali difficoltà tecniche e stabilire le basi per il corretto sviluppo del prodotto, attraverso un'accurata definizione dei requisiti e dei vincoli tecnologici.

Il processo di fornitura si articola nelle seguenti fasi principali:

1. **Preparazione della proposta** Questa fase iniziale si concentra sulla raccolta delle informazioni necessarie e sulla stesura di una proposta formale per il cliente. Include:
   - Analisi delle esigenze del cliente.
   - Studio di fattibilità.
   - Elaborazione della proposta di candidatura.
2. **Pianificazione** Qui si stabilisce l'organizzazione e la programmazione delle attività del progetto, con particolare attenzione a:
   - Definizione delle milestone.
   - Creazione del piano di progetto.
   - Assegnazione di compiti e risorse.
3. **Esecuzione** Durante questa fase si procede con la realizzazione pratica del progetto, che comprende:
   - Sviluppo del software.
   - Test e verifiche.
   - Preparazione della documentazione.
4. **Revisione** Questa fase consiste nel valutare approfonditamente il lavoro svolto per verificarne la conformità agli s tandard di qualità e ai requisiti contrattuali. Le attività principali sono:
   - Revisione del codice.
   - Esecuzione dei test di accettazione.
   - Risoluzione di eventuali discrepanze.
5. **Consegna** Infine, il prodotto finale viene consegnato al cliente. Questa fase comprende:
   - Consegna del software.
   - Formazione del personale.

### 2.1.2 Contatti con l'azienda proponente

7Crusaders dispone di un indirizzo email(code7crusaders@gmail.com) e un canale Discord per le riunioni telematiche. Gli incontri online si svolgeranno settimanalmente, con la possibilità di pianificare riunioni aggiuntive su richiesta del team. Ad ogni incontro settimanale verrà redatto un verbale che riporterà gli argomenti discussi e le scelte intraprese. Per ogni meeting con l'azienda proponente sarà preparato un verbale che riepilogherà i punti principali discussi. Tutti i verbali interni e esterni per discussioni durante lo svolgimento dell'RTB[G] saranno accessibili al seguente link: `https://code7crusaders.github.io/docs/RTB/index.html`. Inoltre per una comunicazione più rapida e informale, il team e l'azienda utilizzeranno Telegram.

### 2.1.3 Analisi dei Requisiti

L'**Analisi dei Requisiti[G]** v1.0, redatto dagli Analisti, rappresenta un documento fondamentale per lo sviluppo del sistema software. Il suo obiettivo principale è definire in dettaglio le funzionalità necessarie affinché il prodotto soddisfi pienamente le richieste della Proponente.

Il documento comprende i seguenti elementi essenziali:

- **Definizione degli attori**: entità o persone che interagiscono con il sistema.
- **Definizione dei casi d'uso**: rappresentazione narrativa di scenari specifici che descrivono come gli attori interagiscono con il sistema. I casi d'uso offrono una visione chiara delle azioni eseguibili all'interno del sistema e delle interazioni degli utenti con esso. All'interno di ciascun caso d'uso, viene fornito:
  - un elenco preciso delle azioni intraprese dall'attore per attivare il caso d'uso;
  - una base per facilitare l'estrazione dei requisiti corrispondenti.
- **Definizione di requisiti**: individuazione e categorizzazione dei requisiti in:
  - **Requisiti funzionali**: specificano le operazioni che il sistema deve essere in grado di eseguire;
  - **Requisiti di qualità**: si concentrano sulla definizione degli standard e degli attributi che il software deve possedere per garantire prestazioni, affidabilità, usabilità e sicurezza ottimali;
  - **Requisiti di vincolo**: delineano vincoli e limitazioni che il sistema deve rispettare, includendo restrizioni tecnologiche, normative o di risorse.

### 2.1.4 Piano di Progetto

Il **Piano di Progetto[G]** v1.0, redatto dal Responsabile, descrive in dettaglio il processo di sviluppo del progetto. Esso funge da guida fondamentale per il team al fine di:

- mantenere l'allineamento con gli obiettivi;
- gestire le risorse in modo efficace;
- affrontare e mitigare eventuali criticità durante le varie fasi.

Si articola nelle seguenti sezioni:

- **Analisi dei rischi**: identifica, valuta e gestisce i rischi potenziali che possono influenzare il successo del progetto. I rischi sono classificati in:
  - tecnologici;
  - di comunicazione;

  - di pianificazione.

  Per ciascun rischio vengono definiti segnali di manifestazione, probabilità, impatto e strategie di mitigazione.

- **Modello di sviluppo**: descrive l'approccio metodologico scelto, nel nostro caso il framework *agile Scrum*. Include:
  - gli eventi principali del framework;
  - le pratiche adottate dal team.
- **Pianificazione**: include una roadmap dettagliata che descrive:
  - le attività necessarie per raggiungere gli obiettivi di ogni sprint;
  - la distribuzione temporale delle risorse.
- **Preventivo**: fornisce una stima delle ore produttive disponibili, distribuite tra:
  - i ruoli assegnati ai membri del team;
  - ogni sprint pianificato.

  Inoltre, include il costo stimato di ogni sprint.
- **Consuntivo**: analizza a posteriori la ripartizione effettiva delle ore e dei costi. Contiene:
  - una retrospettiva sulle discrepanze rispetto al preventivo;
  - eventuali miglioramenti nella pianificazione futura.

#### 2.1.5 Piano di Qualifica

Il **Piano di Qualifica**$^{G}$ v1.0, redatto dall'Amministratore, descrive le strategie e gli approcci adottati per garantire la qualità del prodotto o servizio sviluppato. Si compone delle seguenti sezioni:

- **Qualità di processo**: specifica gli standard e le procedure seguite per garantire la qualità dei processi di sviluppo. Include:
  - metodologie utilizzate;
  - criteri per la misurazione e il miglioramento dei processi.
- **Qualità di prodotto**: descrive gli standard e le specifiche che il prodotto deve soddisfare per essere considerato di qualità. Include:
  - metriche e criteri di valutazione;
  - specifiche tecniche richieste.
- **Specifiche dei test**: fornisce una descrizione dettagliata dei test pianificati durante lo sviluppo per verificare che i requisiti siano soddisfatti.
- **Cruscotto delle metriche**: presenta un resoconto delle attività di valutazione svolte, utile per:
  - monitorare l'andamento del progetto rispetto agli obiettivi prefissati;
  - identificare azioni correttive necessarie.

#### 2.1.6 Glossario

Il Glossario[G] rappresenta un riferimento completo che raccoglie e definisce i termini tecnici utilizzati nel progetto. Questo documento garantisce una comprensione uniforme della terminologia specifica del settore, riducendo il rischio di equivoci e favorendo una comunicazione chiara. Inoltre, contribuisce a migliorare la coerenza e la qualità della documentazione prodotta dal team.

#### 2.1.7 Strumenti

Di seguito sono elencati gli strumenti software utilizzati nel processo di fornitura:

- **Discord**: piattaforma utilizzata per le riunioni interne.
- **Google Meet**: utilizzato per le riunioni formali online con l'azienda proponente.
- **Telegram**: piattaforma utilizzata come metodo informale per comunicare con l'azienda proponente.
- **LaTeX**: sistema per la creazione di documenti e slide di presentazione.
- **GitHubProject**: sistema di ticketing e roadmap integrato nella piattaforma di GitHub.

### 2.2 Sviluppo

#### 2.2.1 Introduzione

Il processo di sviluppo ha l'obiettivo di stabilire i compiti e le attività che il team deve eseguire per realizzare il prodotto software richiesto, assegnando ruoli specifici.

Per garantire che il software soddisfi le aspettative del committente, il gruppo SWEG definisce gli obiettivi di sviluppo e design necessari, assicurando una corretta implementazione del prodotto finale. Quest'ultimo deve:

rispettare le richieste del committente, come descritto nell'Analisi dei Requisiti[G] e sviluppato dal designER; aderire agli obiettivi di sviluppo e superare con successo i test di verifica e validazione G. In particolare, vengono dettagliate le seguenti attività:

#### 2.2.2 Analisi dei Requisiti

##### 2.2.2.1 Descrizione

L'**Analisi dei Requisiti[G]** v1.0 è un documento redatto dagli Analisti che comprende i seguenti aspetti:

- **Introduzione**: descrive l'obiettivo del documento, il fine del prodotto e i riferimenti utilizzati per la sua stesura;
- **Descrizione**: illustra le funzionalità attese del prodotto e le caratteristiche principali degli utenti;
- **Attori**: definisce i soggetti che utilizzeranno il sistema finale;
- **Casi d'uso**: identifica gli attori e descrive tutte le possibili interazioni con il sistema;
- **Requisiti**: raccoglie le caratteristiche essenziali da soddisfare e le fonti da cui queste sono state derivate.

#### 2.2.2.2 Scopo

Lo scopo principale dell'**Analisi dei Requisiti**[G] v1.0 è quello di specificare in modo completo e preciso le funzionalità e le caratteristiche che il prodotto software deve offrire. Tale analisi consente di comprendere appieno:

- le necessità degli utenti;
- gli obiettivi principali del sistema;
- il contesto operativo in cui il sistema sarà utilizzato.

Gli obiettivi fondamentali di questa attività includono:

- Individuare e chiarire le finalità e le aspettative legate al prodotto da sviluppare;
- Fornire ai Progettisti una base dettagliata per definire l'architettura e il design del sistema;
- Offrire un supporto per la pianificazione del progetto utilizzando i requisiti identificati;
- Facilitare lo scambio di informazioni tra il team di sviluppo e la Proponente;
- Servire come riferimento per la fase di verifica del sistema.

#### 2.2.2.3 Codifica dei casi d'uso

I casi d'uso sono codificati utilizzando la seguente notazione:

- **UC[ID-Principale][ID-Sottocaso]**: Identificativo univoco del caso d'uso, composto da un ID principale che identifica il caso principale e, se necessario, da un ID del sottocaso.
- **Titolo**: Breve descrizione del caso d'uso.
- **Attori**: Elenco degli attori coinvolti nel caso d'uso.
- **Precondizioni**: Condizioni che devono essere vere prima che il caso d'uso possa iniziare.
- **Postcondizioni**: Condizioni che devono essere vere dopo che il caso d'uso è stato completato con successo.
- **Scenario principale**: Descrizione dettagliata del flusso di eventi principale del caso d'uso.
- **Generalizzaioni**: Eventuali casi d'uso generalizzati.
- **Estensioni**: Eventuali casi d'uso estesi.

#### 2.2.2.4 Diagrammi Casi D'uso

I diagrammi dei casi d'uso rappresentano visivamente le interazioni tra attori e sistema, illustrando i vari scenari di utilizzo. Ogni caso d'uso descrive una sequenza di azioni necessarie per raggiungere un obiettivo specifico, aiutando a identificare i requisiti funzionali e a chiarire le aspettative degli utenti. Questi diagrammi facilitano la comunicazione tra sviluppatori e stakeholder, garantendo che tutte le funzionalità richieste siano considerate e implementate correttamente. Di seguito sono elencati i principali componenti di un diagramma dei casi d'uso.

- **Attori**: I soggetti che interagiscono con il sistema, rappresentati come uomini stilizzati possono essere persone, altri applicativi o dispositivi che utilizzano le funzionalità del sistema. 1

Figura 1: Esempio di attore

- **Sistema**: Indica il contesto del sistema software, indicando funzionalità interne al contesto definito. 2

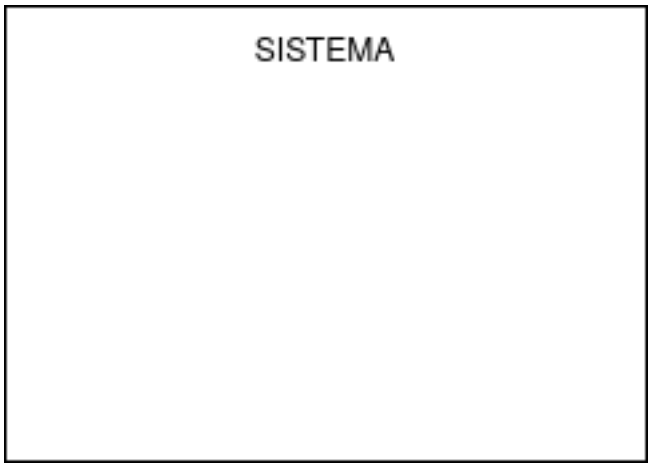


Figura 2: Esempio di sistema

- **Casi d'uso**: funzionalità offerte dal sistema che soddisfano le necessità di un Attore. Ogni caso d'uso descrive una sequenza specifica di interazioni tra gli attori e il sistema. 3

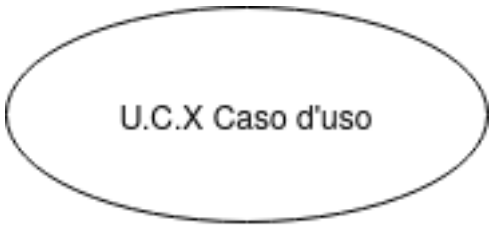


Figura 3: Esempio di caso d'uso

- **Sottocasi d'uso**: Scenari specifici che si verificano all'interno del caso d'uso principale. 4

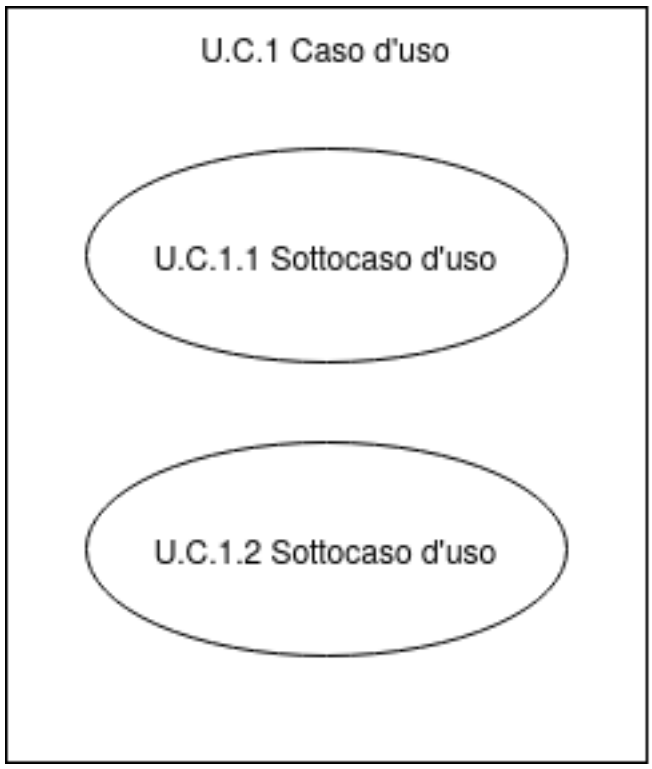


Figura 4: Esempio di sottocaso d'uso

- **Relazioni tra Attori e Casi d'Uso**:
  - **Associazione**: Collegamento tra un attore e un caso d'uso, indicando che l'attore è coinvolto nel caso d'uso. 5

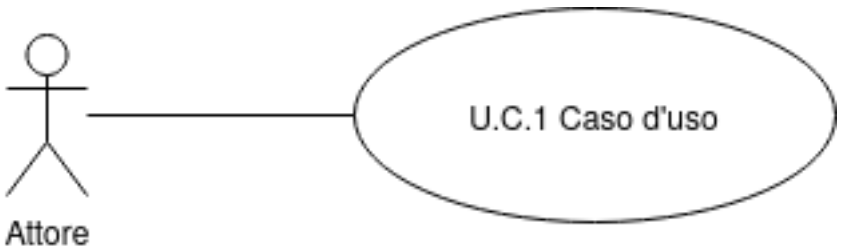


Figura 5: Esempio di associazione

- **Relazioni tra Attori**:
  - **Generalizzazione**: Un attore eredita le funzionalità di un altro attore. Una relazione padre figlio dove il figlio eredita almeno una funzionalità del padre. Utilizzata nel caso in cui due attori condividano funzionalità. 6

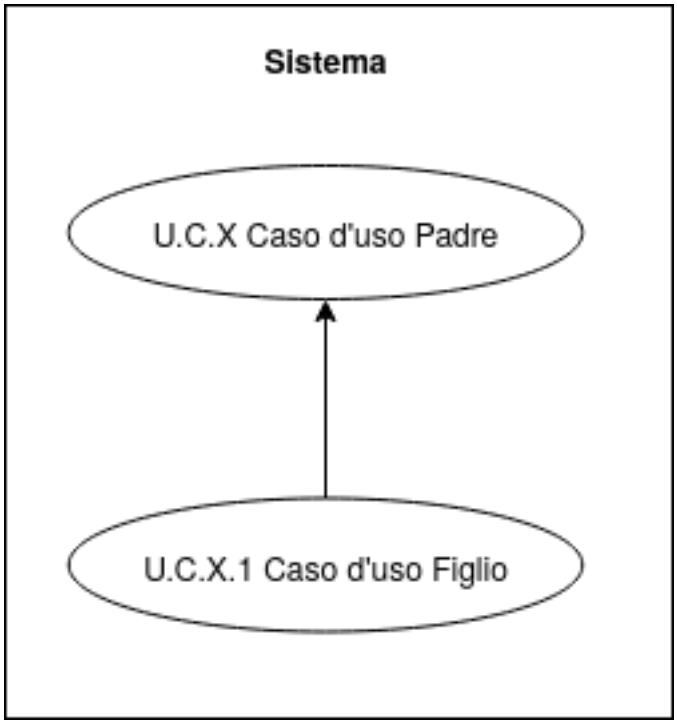


Figura 6: Esempio di Generalizzaione casi d'uso

- **Relazioni tra Casi d'Uso**:
  - **Inclusione**: Indica che un caso d'uso include un altro caso d'uso. Questo significa che durante l'esecuzione di un caso d'uso si eseguono anche i casi d'uso inclusi. Questo per evitare la ripetizione di funzionalità uguali in più casi d'uso. 7

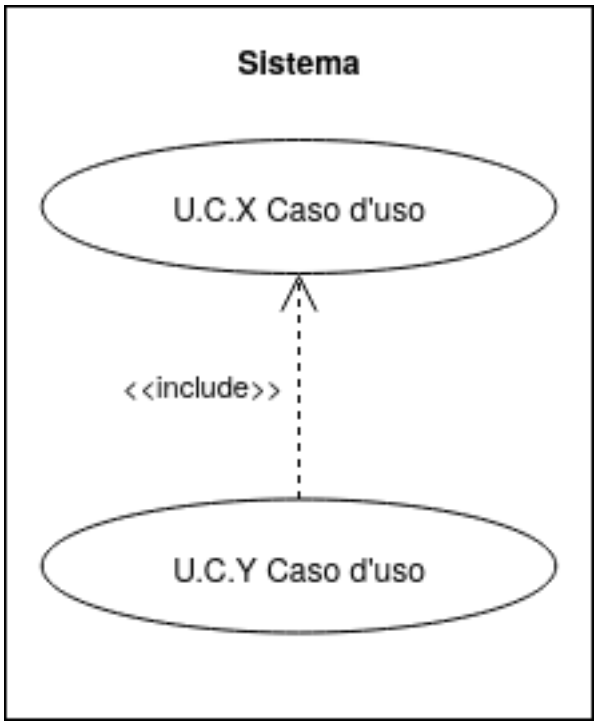


Figura 7: Esempio di inclusione

– **Estensione**: Un caso d'uso esteso aggiunge funzionalità al caso d'uso principale, ma viene attivato solo in specifiche circostanze. Quando ciò accade, il flusso del caso d'uso principale si interrompe temporaneamente per consentire l'esecuzione del caso d'uso esteso. 8

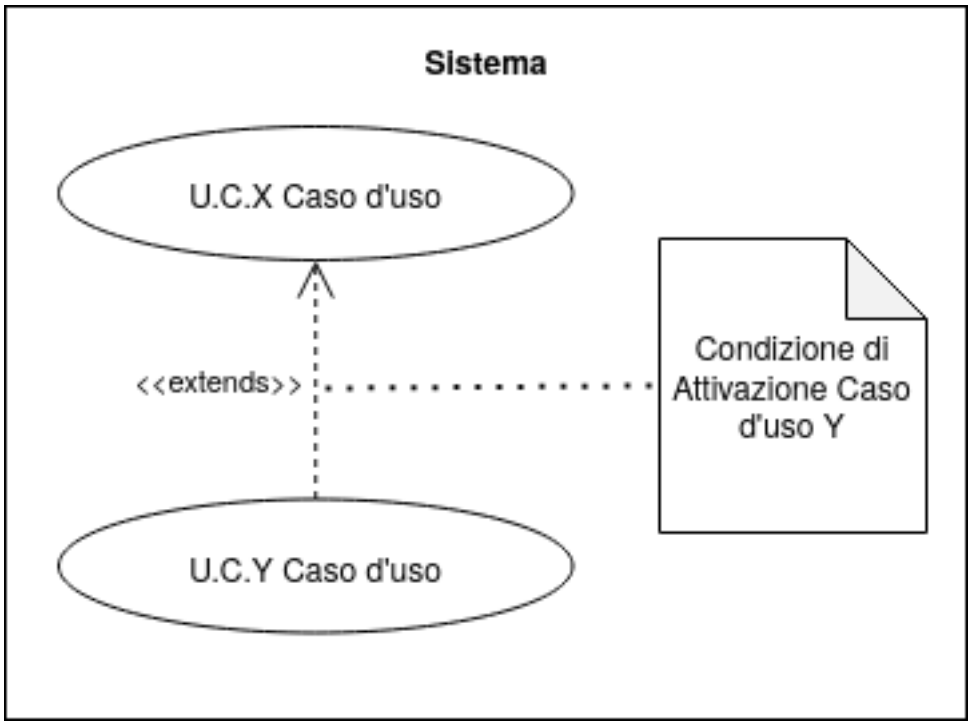


Figura 8: Esempio di estensione

– **Generalizzazione**: Un caso d'uso eredita le funzionalità di un altro caso d'uso. Una relazione padre figlio dove il figlio eredita almeno una funzionalità del padre. Utilizzata nel caso in cui due casi d'uso condividano funzionalità. 9

Figura 9: Esempio di Generalizzaione attore

#### 2.2.2.5 Requisiti

I requisiti sono classificati in tre categorie principali:

- **Funzionali**: riguardano l'usabilità del prodotto finale;
- **Di qualità**: includono gli strumenti e la documentazione da fornire;
- **Di vincolo**: fanno riferimento alle tecnologie da utilizzare.

Ciascun requisito è indicato da:

- **Codice Identificativo**: codice univoco che identifica il requisito;

- **Descrizione**: breve spiegazione del requisito;
- **Fonte**: origine del requisito (es. capitolato, interno, ecc..);
- **Priorità**: importanza del requisito rispetto agli altri;

#### 2.2.2.6 Fonti dei requisiti

I requisiti sono stati identificati a partire dalle seguenti fonti:

- **Capitolato**: Requisiti individuati tramite analisi del capitolato;
- **interno**: requisiti individuati durante riunioni interne al gruppo di lavoro;
- **Esterno**: requisiti individuati durante incontri con il proponente;
- **Piano di Qualifica**[G]: Requisiti necessari per rispettare standard di qualità definiti nel documento Piano di Qualifica[G];
- **Norme di Progetto**[G]: Requisiti necessari per rispettare le norme di progetto definite nel documento Norme di Progetto[G];

#### 2.2.2.7 Codifica dei requisiti

I requisiti sono codificati come segue: **R[Tipo][Importanza][Numero]**
Dove **Tipo** può essere:

- **F (funzionale)**
- **Q (di qualità)**
- **V (di vincolo)**

**Importanza** può essere:

- **O (obbligatorio)**
- **D (desiderabile)**
- **F (facoltativo )**

**Numero** è un numero identificativo univoco del requisito.

### 2.2.3 Progettazione

La fase di progettazione riveste un ruolo cruciale nel definire la struttura principale del progetto, basandosi sui requisiti individuati durante l'analisi e descritti nell'Analisi dei Requisiti[G]. Questa attività è affidata ai progettisti, i quali elaborano un piano dettagliato per implementare tutti i requisiti specificati.

Per questa fase del ciclo di vita del software, il nostro gruppo si pone i seguenti obiettivi: Trasformare i requisiti in specifiche tecniche dettagliate che coprano tutti gli aspetti del sistema; Garantire una struttura facilmente comprensibile per agevolare la manutenzione futura; Ottenere l'approvazione per il passaggio alla fase di sviluppo. Il processo di progettazione si articola in tre livelli principali: Design dell'interfaccia: questa fase si concentra su un livello di astrazione elevato rispetto al funzionamento interno del sistema. Durante la progettazione dell'interfaccia, l'attenzione è rivolta alle tecnologie da utilizzare nella fase di sviluppo del software, portando alla creazione di un Proof of Concept[G]. Progettazione architettonica: si definisce la struttura generale del sistema a un alto livello, senza entrare nei dettagli interni dei componenti principali. In questa fase vengono anche definiti i test di integrazione. Progettazione dettagliata: si specificano gli elementi interni di ciascun componente principale, incluse le specifiche architetturali del prodotto. Si producono inoltre i diagrammi delle classi e si definiscono i test di unità per ogni componente. Questa fase culmina nella creazione della Product Baseline[G].

### 2.2.4 Codifica

Dopo la fase di progettazione, i membri del gruppo con il ruolo di Programmatori avviano la fase di codifica, implementando le specifiche dei requisiti e seguendo i documenti di progettazione. L'obiettivo di questa fase è trasformare le idee in realtà, sviluppando il prodotto software desiderato tramite attività di programmazione. Le aspettative per la fase di codifica includono:

- Completare lo sviluppo del prodotto finale garantendo la qualità e il rispetto delle richieste del committente;
- Assicurare che il codice prodotto sia chiaro e leggibile per facilitare eventuali modifiche future.

# 3 Processi di Supporto

## 3.1 Documentazione

### 3.1.1 Introduzione

La documentazione software si riferisce al testo che accompagna un programma, descrivendo il prodotto sia per gli sviluppatori che per gli utilizzatori. Essa ha l'obiettivo di supportare i membri del team durante lo sviluppo, monitorando i processi e documentando tutte le attività, per facilitare anche la manutenzione del software e migliorare la qualità del prodotto finale.

In base a quanto sopra, la documentazione svolge un ruolo cruciale nel ciclo di vita del software. Le aspettative nei suoi confronti includono:

- Definizione di regole chiare e concise per la redazione dei documenti.
- Adozione di una struttura uniforme e standard per tutti i documenti nel ciclo di vita del software, per garantire omogeneità.

### 3.1.2 Ciclo di Vita del Documento

Il ciclo di vita di un documento software si articola in tre fasi principali:

- **Redazione**: la fase di creazione del documento, che viene suddivisa tra i membri del gruppo e supportata dall'uso di un sistema di versionamento.
- **Verifica**: una volta completata la stesura, il documento passa alla fase di verifica, che può essere effettuata su parti del documento o su tutto il contenuto. Ogni sezione deve essere verificata da una persona distinta dal redattore della sezione stessa.
- **Approvazione**: il documento, una volta completato e verificato, viene approvato dal Responsabile di Progetto.

### 3.1.3 Template

Il gruppo ha deciso di utilizzare un template semplice, creato con Latex. Questo modello è stato standardizzato e viene utilizzato per la redazione di tutti i documenti ufficiali.

### 3.1.4 Documenti Prodotti

I documenti generati durante il ciclo di vita del software sono suddivisi in due categorie principali:

#### Formali

I documenti formali sono quelli con un nome univoco e utilizzati per regolare le attività interne al gruppo durante tutto il ciclo di vita del software. Sono versionati e approvati dal Responsabile di Progetto. Questi documenti si suddividono in:

- **Ad uso interno**: destinati esclusivamente ai membri del gruppo, come ad esempio:
  - Norme di progetto
  - Verbali interni
- **Ad uso esterno**: destinati a enti esterni come il committente o il proponente, e consegnati nell'ultima versione approvata. Tra questi:
  - Analisi dei Requisiti[G]

- Piano di Progetto[G]
- Piano di Qualifica[G]
- Glossario[G]
- Verbali esterni

#### Informali

I documenti informali comprendono:

- Documenti non ancora approvati dal Responsabile di Progetto.
- Bozze e appunti brevi.
- Documenti che non necessitano di essere versionati.

Questi documenti sono gestiti in una sezione separata, dove il gruppo ha creato un Google Drive condiviso per facilitarne la gestione.

### 3.1.5 Struttura del Documento

Tutti i documenti ufficiali seguono una struttura rigida che deve essere rispettata. La struttura include:

#### Prima Pagina

La prima pagina include:

- Il titolo del gruppo.
- Il nome del documento.
- Il logo del gruppo.
- Il membri del team
- Le informazioni sul documento, che elencano i redattori, i verificatori, l'amministratore, i destinatari e la versione.

#### Registro dei Cambiamenti - Changelog

Il registro dei cambiamenti tiene traccia della storia del documento. In questa sezione sono inclusi:

- La versione del documento.
- La data di ogni modifica.
- L'autore che ha effettuato la modifica.
- Il verificatore delle modifiche.
- Una breve descrizione delle modifiche.

#### Indice

Ogni documento include un indice subito dopo il registro dei cambiamenti. Questo indice aiuta a navigare nel documento, rendendo più facile la ricerca di sezioni specifiche.

#### 3.1.6 Verbali

I verbali sono documenti speciali con una struttura diversa rispetto agli altri. Non includono il registro dei cambiamenti né l'indice. La struttura dopo la prima pagina prevede:

- **Partecipanti**: orario di inizio e fine dell'incontro, seguito da una tabella con i nomi e le durate di presenza dei partecipanti.
- **Sintesi ed elaborazione incontro**: un riassunto degli argomenti trattati e una sezione per eventuali dubbi o indicazioni per i prossimi incontri.

I verbali sono suddivisi in interni (tra i membri del gruppo) ed esterni (con l'azienda o il committente).

#### 3.1.7 Nome del File

I file devono avere nomi coerenti, con la lettera iniziale maiuscola e le restanti lettere minuscole, eccetto i verbali, che sono nominati con la data dell'incontro in formato GG-MM-AAAA.

#### 3.1.8 Stile del Testo

Lo stile del testo nei documenti ufficiali include:

- **Grassetto**: per titoli e parole di rilevanza.
- Sottolineato: solo per i link e parole del glossario.

#### 3.1.9 Glossario

Il glossario è un documento contenente termini e definizioni utili per comprendere meglio il linguaggio tecnico, evitando ambiguità. I termini sono registrati in ordine alfabetico.

#### 3.1.10 Tabelle

Le tabelle nei documenti ufficiali devono avere un titolo che descriva il contenuto e devono essere centrate orizzontalmente. Le righe della tabella sono alternate in colori per migliorare la leggibilità.

#### 3.1.11 Immagini

Le immagini devono essere centrate orizzontalmente. Anche i diagrammi UML sono trattati come immagini.

### 3.2 Gestione della Configurazione

La gestione della configurazione è un processo fondamentale per mantenere il software in uno stato coerente, garantendo che il sistema continui a funzionare correttamente nonostante le modifiche apportate nel tempo. Problemi di configurazione possono causare incoerenze o non conformità, con un impatto negativo sulle operazioni del sistema. Il gruppo, attraverso la gestione della configurazione, mira a:

- Individuare e risolvere i problemi prima che diventino critici;
- Facilitare il tracciamento delle modifiche e l'identificazione degli errori.

### 3.2.1 Versionamento

Il versionamento è un processo che consente di tracciare le modifiche apportate a un documento. Inoltre, permette di ripristinare il documento a uno stadio precedente e di visualizzare i cambiamenti effettuati nel tempo, associandoli al relativo autore. Il nostro gruppo ha adottato il seguente formato per identificare la versione di un documento:

$$[x].[y].[z]$$

Dove:

- **x**: numero intero che parte da 0 e viene incrementato dal Responsabile di Progetto (**RdP**) dopo l'approvazione del documento (versione di produzione);
- **y**: numero intero incrementato dal Verificatore (**Ve**) a ogni verifica del documento;
- **z**: numero intero incrementato dal Redattore (**Re**) per ogni piccola modifica effettuata.

### 3.2.2 Repository

Il nostro gruppo ha deciso di utilizzare per la gestione della configurazione il servizio GitHub, basato sul sistema di controllo di versione distribuito Git. La repository è pubblica e si può facilmente trovare al seguente link:

## 3.3 Gestione della Qualità

### 3.3.1 Descrizione

La gestione della qualità di progetto comprende i processi e le attività svolte all'interno di un progetto per garantire che la qualità dei deliverable e delle performance siano in linea con gli obiettivi e i requisiti definiti. I membri del nostro gruppo si pongono i seguenti obiettivi:

- Comprendere, valutare e gestire le aspettative del committente, assicurandosi che i requisiti siano rispettati;
- Definire chiaramente i requisiti di qualità e documentare tutte le procedure necessarie per completare il progetto in conformità con le aspettative richieste;
- Consegnare il progetto in linea con il piano di qualità, garantendo che il prodotto finale sia consegnato nei tempi previsti, rispettando il budget e soddisfacendo i requisiti e le aspettative del committente.

### 3.3.2 Piano di Qualifica

Per garantire il rispetto di tutti gli aspetti del processo di gestione della qualità, utilizziamo il **Piano di Qualifica**$^{G}$, un documento che include un elenco strutturato dei dati necessari per assicurare un piano di alta qualità. In particolare, il Piano di Qualifica$^{G}$ prevede:

- La definizione dei requisiti richiesti dal committente;
- L'identificazione di metriche e parametri per l'analisi dei dati;
- L'implementazione di un sistema per il controllo della qualità durante l'intero ciclo di vita del progetto;
- La pianificazione di un sistema di miglioramento che descriva le azioni necessarie per analizzare le prestazioni di qualità e individuare le attività utili a incrementare il valore del progetto.

## 3.4 Verifica

### 3.4.1 Introduzione

La verifica del software è il processo di valutazione del prodotto per garantire che la fase di sviluppo sia eseguita correttamente, al fine di costruire il prodotto desiderato. Questo processo si svolge durante lo sviluppo del software, consentendo di rilevare difetti e guasti nelle fasi iniziali del ciclo di vita e di verificare che il prodotto soddisfi i requisiti del cliente.

Le aspettative per questo processo includono:

- Incrementare la fiducia del gruppo nel proseguire lo sviluppo del progetto in modo corretto;
- Garantire il raggiungimento del prodotto finale atteso;
- Identificare precocemente gli errori, riducendo così i costi e il tempo necessari per le correzioni.

### 3.4.2 Tipi di Verifica

Il processo di verifica si compone di due tipi principali, ciascuno focalizzato su diversi aspetti del software. Insieme, questi due tipi garantiscono che il software sia conforme ai requisiti specificati. Inoltre, viene considerato un terzo tipo per la verifica della documentazione.

#### Analisi Statica

L'analisi statica consiste nell'ispezione del codice prima della sua esecuzione, assicurando che il software soddisfi i requisiti e le specifiche definiti. Poiché non richiede l'esecuzione dell'oggetto in verifica, questo tipo di analisi può essere applicato non solo al codice, ma anche alla documentazione. Questo approccio analizza gli aspetti statici del sistema software, come le convenzioni del codice o il calcolo di metriche. Include sia tecniche di test manuali che automatizzate, come quelle orientate alla coerenza. L'analisi statica si divide in due metodi principali:

- **Walkthroughs**: una lettura di ampio spettro che consente di esaminare e discutere eventuali errori o difetti trovati. Questo metodo è utile quando non si ha certezza sulla posizione dei problemi.
- **Inspection**: un metodo mirato per identificare e rimuovere errori e difetti. Si utilizza un approccio più focalizzato, avendo già un'idea delle possibili problematiche.

#### Analisi Dinamica

L'analisi dinamica viene eseguita durante l'esecuzione del software e consiste nella fase di test. A differenza della verifica statica, comporta l'esecuzione del sistema e dei suoi componenti.

Il gruppo adotterà un insieme di test ripetibili e automatizzati. L'automatizzazione sarà possibile attraverso strumenti dedicati, che verranno definiti successivamente.

#### Verifica della Documentazione

La verifica della documentazione si compone delle seguenti attività:

- Controllo di ortografia e sintassi;
- Controllo dell'utilizzo corretto delle norme tipografiche o di altre norme di stile e formattazione concordate;
- Verifica della pertinenza dei contenuti scritti.

## 3.5 Validazione

### 3.5.1 Introduzione

La convalida del software è un processo di valutazione del prodotto, finalizzato a garantire che il software soddisfi i requisiti predefiniti e specificati dal richiedente, nonché le richieste e le aspettative degli utenti finali/proponente. Un processo di validazione ha successo quando è stata effettuata una buona verifica durante tutta la fase di sviluppo. L'esito finale positivo della validazione assicura che il prodotto finale sia allineato con le aspettative.

- Rilevare possibili errori ignorati o trascurati durante la fase di verifica;
- Soddisfare i requisiti specificati nell'Analisi dei Requisiti$^{G}$ per il prodotto finale;
- Contribuire a migliorare la qualità e il valore del prodotto software finale.

# 4 Processi Organizzativi

## 4.1 Gestione dei Processi

### 4.1.1 Introduzione

La gestione dei processi rappresenta una fase cruciale per il successo di un progetto, garantendo che venga completato in conformità agli obiettivi e ai requisiti predefiniti. Questa fase si concentra sulla pianificazione, organizzazione, monitoraggio e controllo delle attività coinvolte nel ciclo di vita del software, assicurando che il lavoro svolto rispetti gli standard di qualità e soddisfi le aspettative del cliente.

Le principali attività di gestione dei processi sono le seguenti:

- **Definizione dei processi:** Documentazione dei processi chiave adottati nel progetto, inclusi quelli relativi allo sviluppo del software, controllo di versione, gestione dei cambiamenti e assicurazione della qualità.
- **Pianificazione dei processi:** Definizione degli obiettivi del progetto, delle fasi, delle risorse necessarie e delle scadenze. In questa fase vengono stabiliti i criteri di successo e redatto un piano di lavoro dettagliato.
- **Assegnazione delle risorse:** Allocazione dei membri del team alle attività specifiche, tenendo conto delle loro competenze e disponibilità.
- **Monitoraggio e controllo:** Controllo continuo dei progressi rispetto al piano stabilito, comprendente tempi, costi e qualità, oltre a gestione dei rischi.
- **Gestione dei cambiamenti:** Valutazione e gestione delle modifiche ai requisiti, alla pianificazione o alla distribuzione delle risorse.
- **Assicurazione della qualità:** Implementazione di procedure per garantire che il software soddisfi i requisiti e le aspettative del cliente.
- **Comunicazione e coordinamento:** Facilitazione della comunicazione tra membri del team e stakeholder per mantenere tutte le parti informate sullo stato del progetto.
- **Miglioramento continuo:** Analisi dei processi per identificare aree di miglioramento e ottimizzare l'efficienza e la qualità.

### 4.1.2 Pianificazione

#### 4.1.2.1 Descrizione

La pianificazione dei processi consiste nell'identificare, organizzare e controllare le attività necessarie per il successo del progetto. È un'attività strategica che garantisce una direzione chiara e una solida struttura gestionale lungo il ciclo di vita del progetto.

#### 4.1.2.2 Obiettivi

L'obiettivo principale della pianificazione è assicurare l'esecuzione efficiente ed efficace del progetto, rispettando gli obiettivi e i requisiti stabiliti. Inoltre:

- Ogni membro del team deve assumere almeno una volta ciascun ruolo, favorendo crescita e collaborazione.
- Ridurre i rischi e affrontare le sfide in modo anticipato, permettendo al team di superare eventuali difficoltà.

### 4.1.3 Assegnazione dei Ruoli

Durante l'implementazione del progetto, i membri del team di Code7Crusaders ricopriranno ruoli distinti. Ogni ruolo comporta specifiche responsabilità:

- **Responsabile:**
  - Coordina il gruppo di lavoro.
  - Pianifica e controlla le attività.
  - Gestisce le risorse e le comunicazioni esterne.
  - Redige il Piano di Progetto$^{G}$.
- **Amministratore:**
  - Gestisce l'ambiente di lavoro e le procedure.
  - Gestisce la configurazione del prodotto.
  - Redige le Norme di Progetto$^{G}$.
- **Analista:**
  - Analizza i requisiti del progetto e il dominio applicativo.
  - Redige l'Analisi dei Requisiti$^{G}$.
- **Progettista:**
  - Progetta l'architettura del prodotto.
  - Prende decisioni tecniche e tecnologiche.
  - Redige la Specifica Tecnica.
- **Programmatore:**
  - Scrive il codice e implementa le funzionalità richieste.
  - Redige il Manuale Utente.
- **Verificatore:**
  - Verifica che il lavoro svolto sia conforme alle norme e alle specifiche.
  - Redige il Piano di Qualifica$^{G}$.

### 4.1.4 Ticketing

Il gruppo code7crusaders utilizza GitHub Projects per il ticketing. La roadmap è organizzata in tre colonne: *To Do*, *In Progress*, e *Completed*. Ogni attività è classificata in base a:

- **Priorità:** Bassa, media o alta.
- **Dimensione:** XS, S, M, L, XL.
- **Stima ore:** Numero di ore necessarie per completare l'attività.

#### 4.1.5 Gestione dei rischi

##### 4.1.5.1 Struttura dei rischi

I rischi sono classificati in tre categorie principali:

- Rischi di natura tecnologica;
- Rischi legati alla comunicazione;
- Rischi relativi alla pianificazione.

Ogni rischio è identificato tramite un codice univoco con la seguente struttura:

**R[Categoria][Indice] - [Nome]**

Dove:

- **Categoria**: indica il tipo di rischio e può assumere i seguenti valori:
  - **T**: per i rischi tecnologici;
  - **C**: per i rischi comunicativi;
  - **P**: per i rischi di pianificazione.
- **Indice**: un identificatore progressivo univoco all'interno della categoria di appartenenza;
- **Nome**: il nome descrittivo del rischio.

### 4.2 Procedure Comunicative

#### 4.2.1 Comunicazioni Asincrone

- **Interne:** Abbiamo deciso di usare un gruppo Whatsapp come canale di comunicazione asincrono, consentendo comunicazione semplice e veloce tramite un gruppo.
- **Esterne:** Gestite tramite la piattaforma .

#### 4.2.2 Comunicazioni Sincrone

- **Interne:** Viene adottata la piattaforma Discord per la velocità nell'effettuare riunioni in chiamata vocale.
- **Esterne:** L'azienda Ergon adotta Zoom come piattaforma di riunioni esterne.

#### 4.2.3 Riunioni Interne

Le riunioni interne del gruppo Code7Crusaders si tengono ogni venerdì, utilizzando Discord come piattaforma di comunicazione. Questi incontri servono principalmente per monitorare i progressi delle attività in corso, discutere eventuali difficoltà riscontrate e pianificare i passi successivi. L'orario delle riunioni è fissato dalle 15:00 alle 16:00, salvo necessità particolari che richiedano un adattamento. Nel caso in cui un membro non possa partecipare, è tenuto a informare tempestivamente il resto del team. Se necessario, i membri assenti potranno recuperare le informazioni rilevanti consultando il verbale della riunione. Durante queste riunioni, il team lavora in modo collaborativo e dinamico: ogni membro condivide lo stato delle proprie attività, proponendo soluzioni a eventuali problemi emersi. L'approccio informale permette di discutere liberamente idee, priorità e obiettivi futuri, favorendo una comunicazione aperta ed efficace.

#### 4.2.4 Riunioni Esterne

Durante lo sviluppo del progetto, è fondamentale organizzare incontri periodici con i committenti o con la proponente per valutare lo stato di avanzamento del lavoro, risolvere eventuali dubbi e discutere questioni rilevanti. La pianificazione e la gestione di questi incontri sono affidate al responsabile, che si occupa di convocarli e garantirne un'efficace organizzazione. Il responsabile ha anche il compito di presentare i punti principali della discussione alla proponente o ai committenti, coinvolgendo i membri del gruppo direttamente interessati in base agli argomenti trattati. Questo metodo garantisce una comunicazione chiara e mirata, evitando dispersioni di tempo e favorendo la comprensione reciproca. La partecipazione alle riunioni è considerata una priorità per tutti i membri del gruppo. Ogni membro si impegna a riorganizzare i propri impegni, quando possibile, per assicurare una presenza costante. Nel caso in cui un membro sia impossibilitato a partecipare per cause inderogabili, il responsabile si farà carico di informare prontamente i committenti o la proponente e, se necessario, proporrà il rinvio dell'incontro a una data più adeguata. Durante ogni incontro, sarà inoltre garantita una registrazione accurata delle informazioni discusse, per mantenerne traccia e facilitarne il recupero in futuro.

### 4.3 Formazione

#### 4.3.1 Introduzione

La formazione è una componente essenziale per garantire che tutti i membri del team siano adeguatamente preparati per affrontare le sfide tecniche e gestionali del progetto. Questo processo si concentra sullo sviluppo delle competenze e delle conoscenze necessarie per utilizzare strumenti, tecnologie e metodologie specifiche del progetto, promuovendo così l'efficienza e la qualità del lavoro svolto.

#### 4.3.2 Metodo di formazione

**Individuale:** Ogni individuo del team dovrà compiere un processo di autoformazione per riuscire a svolgere al meglio il ruolo assegnato. La rotazione dei ruoli permetterà al nuovo occupante di un ruolo di apprendere le competenze necessarie da chi lo ha precedentemente svolto, nel caso avesse delle lacune. Questo metodo permette di avere una formazione continua e di garantire che ogni membro del team sia in grado di svolgere ogni ruolo. **Di gruppo:** Oltre alla formazione individuale, il team parteciperà a sessioni di formazione collettiva per condividere conoscenze, affrontare problematiche comuni e rafforzare la collaborazione. Questi incontri includeranno workshop, presentazioni tecniche e sessioni di revisione del lavoro, mirate a uniformare le competenze tra i membri del gruppo. La formazione di gruppo consente inoltre di sviluppare una visione condivisa del progetto e di affrontare sfide complesse in modo coordinato, promuovendo il lavoro di squadra e la comunicazione efficace.