



# Norme di Progetto

---

## Bot4Me

---

[skynet.swe@gmail.com](mailto:skynet.swe@gmail.com)

12 Settembre 2022

*Redattori:* Anna Cisotto Bertocco, Davide Dinato, Davide Sut, Nicholas Pilotto

*Verificatori:* Davide Dinato, Alberto Matterazzo, Anna Cisotto Bertocco, Kaltrina Collaku

*Responsabile:* Davide Sut

*Destinatari:* Prof. Tullio Vardanega, Prof. Riccardo Cardin, SkyNet

*Uso:* **Interno**

*Stato:* Approvato

*Versione:* **2.0.0**

## Registro delle Modifiche

Versione	Autore (verificatore)	Data	Ruolo	Descrizione
<b>2.0.0</b>	Anna Cisotto Bertocco (-)	12-09-2022	Responsabile	Approvazione documento
<b>1.2.0</b>	Davide Sut (-)	12-09-2022	Verificatore	Verifica documento
<b>1.1.4</b>	Davide Sut (Anna Cisotto Bertocco)	05-09-2022	Responsabile	Stesura §3.2.3 , §4.1.3 e modifica §3.5.5.2, §3.5.6
<b>1.1.3</b>	Davide Dinato (Anna Cisotto Bertocco)	05-09-2022	Programmatore	Stesura §2.1.3 , §2.2.4.2.7 e §2.2.4.2.8
<b>1.1.2</b>	Nicholas Pilotto (Anna Cisotto Bertocco)	05-09-2022	Progettista	Stesura §2.2.4.2.2 e §3.4.3
<b>1.1.1</b>	Nicholas Pilotto (Alberto Matterazzo)	03-09-2022	Responsabile	Aggiunta §3.5.5.3
<b>1.1.0</b>	Davide Dinato (-)	27-08-2022	Verificatore	Verifica documento
<b>1.0.4</b>	Anna Cisotto Bertocco (Davide Dinato)	24-08-2022	Amministratore	Stesura §3.1.4.4.7
<b>1.0.3</b>	Anna Cisotto Bertocco (Davide Dinato)	23-08-2022	Amministratore	Modificata struttura Registro delle modifiche
<b>1.0.2</b>	Anna Cisotto Bertocco (Davide Dinato)	22-08-2022	Amministratore	Modifica struttura §3 e §4
<b>1.0.1</b>	Anna Cisotto Bertocco (Kaltrina Collaku)	20-08-2022	Amministratore	Stesura §1.4 e modifica struttura §2
<b>1.0.0</b>	Davide Sut (-)	10-08-2022	Responsabile	Approvazione documento
<b>0.2.1</b>	Anna Cisotto Bertocco (-)	10-08-2022	Verificatore	Modifica sezione §2.1.4
<b>0.2.0</b>	Anna Cisotto Bertocco (-)	28-07-2022	Verificatore	Verifica documento
<b>0.1.7</b>	Alberto Matterazzo (Davide Sut)	27-07-2022	Responsabile	Modifica §2.2.5, §3.1.11, §3.3.6, §3.5.5.2
<b>0.1.6</b>	Davide Sut (-)	22-07-2022	Amministratore	Stesura sezione §4.2
<b>0.1.5</b>	Davide Dinato (-)	20-07-2022	Verificatore	Aggiornamento §3.2.4 e §3.2.3.2
<b>0.1.4</b>	Davide Sut (Nicholas Pilotto)	17-07-2022	Progettista	Stesura sezione §4.1



Versione	Autore (verificatore)	Data	Ruolo	Descrizione
0.1.3	Alberto Matterazzo (Anna Cisotto Bertocco)	15-07-2022	Amministratore	Aggiornamento §3.6
0.1.2	Anna Cisotto Bertocco (-)	11-07-2022	Verificatore	Modifica §3.5
0.1.1	Anna Cisotto Bertocco (Kaltrina Collaku)	06-06-2022	Amministratore	Modifica §2.2.4, stesura §2.2.4.2 , §2.2.4.3, §3.4
0.1.0	Davide Sut (-)	22-04-2022	Verificatore	Verifica documento
0.0.3	Davide Dinato (Davide Sut)	21-04-2022	Amministratore	Stesura di: §3, §3.1.7, §3.1.9, §3.1.10, §3.1.11, §3.1.12, §3.2, §3.4
0.0.2	Kaltrina Collaku (-)	11-04-2022	Amministratore	Stesura di: §1; §2; §3: §3.1, §3.3, §3.5
0.0.1	Davide Sut (-)	22-03-2022	-	Creazione struttura documento

## Indice

<b>1</b>	<b>Introduzione</b>	<b>1</b>
1.1	Scopo del documento . . . . .	1
1.2	Scopo del capitolato . . . . .	1
1.3	Glossario . . . . .	1
1.4	Organizzazione delle sezioni . . . . .	1
1.5	Riferimenti . . . . .	2
1.5.1	Normativi . . . . .	2
1.5.2	Informativi . . . . .	2
<b>2</b>	<b>Processi Primari</b>	<b>3</b>
2.1	Fornitura . . . . .	3
2.1.1	Scopo . . . . .	3
2.1.2	Descrizione . . . . .	3
2.1.3	Aspettative . . . . .	3
2.1.4	Attività . . . . .	3
2.1.4.1	Redazione Piano di Progetto . . . . .	3
2.1.4.1.1	Descrizione . . . . .	3
2.1.4.1.2	Procedure . . . . .	4
2.1.4.2	Redazione Piano di Qualifica . . . . .	4
2.1.4.2.1	Descrizione . . . . .	4
2.1.4.2.2	Procedure . . . . .	4
2.1.5	Metriche di qualità . . . . .	4
2.1.5.1	MQPC01 - Budgeted Cost of Work Scheduled (BCWS) . . . . .	4
2.1.5.2	MQPC02 - Actual Cost of Work Performed (ACWP) . . . . .	4
2.1.5.3	MQPC03 - Budgeted Cost of Work Performed (BCWP) . . . . .	4
2.1.5.4	MQPC04 - Cost Variance (CV) . . . . .	4
2.1.5.5	MQPC05 - Schedule Variance (SV) . . . . .	5
2.1.5.6	MQPC06 - Estimated At Completion (EAC) . . . . .	5
2.1.5.7	MQPC07 - Schedule Performance Index (SPI) . . . . .	5
2.1.5.8	MQPC08 - Cost Performance Index (CPI) . . . . .	5
2.1.6	Strumenti . . . . .	5
2.2	Sviluppo . . . . .	6
2.2.1	Scopo . . . . .	6
2.2.2	Descrizione . . . . .	6
2.2.3	Aspettative . . . . .	6
2.2.4	Attività . . . . .	6
2.2.4.1	Analisi dei requisiti . . . . .	6
2.2.4.1.1	Scopo . . . . .	6
2.2.4.1.2	Descrizione . . . . .	6
2.2.4.1.3	Procedure . . . . .	6
2.2.4.1.4	Classificazione dei requisiti . . . . .	6
2.2.4.1.5	Classificazione dei casi d'uso . . . . .	7
2.2.4.1.6	Strumenti . . . . .	7
2.2.4.2	Progettazione . . . . .	7
2.2.4.2.1	Scopo . . . . .	7
2.2.4.2.2	Descrizione . . . . .	8
2.2.4.2.3	Procedure . . . . .	8
2.2.4.2.4	Qualità dell'architettura . . . . .	8
2.2.4.2.5	Technology Baseline <sub>G</sub> . . . . .	8
2.2.4.2.6	Product Baseline <sub>G</sub> . . . . .	8
2.2.4.2.7	Diagrammi delle classi . . . . .	9
2.2.4.2.8	Diagrammi di sequenza . . . . .	9
2.2.4.3	Codifica . . . . .	9
2.2.4.3.1	Scopo . . . . .	9



2.2.4.3.2	Descrizione . . . . .	9
2.2.4.3.3	Procedure . . . . .	9
2.2.4.3.4	Strumenti . . . . .	9
2.2.5	Metriche di qualità . . . . .	10
2.2.5.1	MQPC09 - Requirements Stability Index (RSI) . . . . .	10
2.2.5.2	MQPD03 - Copertura requisiti obbligatori . . . . .	10
2.2.5.3	MQPD04 - Copertura requisiti desiderabili . . . . .	10
2.2.5.4	MQPD05 - Copertura requisiti opzionali . . . . .	10
2.2.5.5	MQPD06 - Facilità di utilizzo . . . . .	10
2.2.5.6	MQPD07 - Numero di browser <sub>G</sub> supportati . . . . .	10
2.2.6	Strumenti . . . . .	10
<b>3</b>	<b>Processi di supporto</b>	<b>11</b>
3.1	Documentazione . . . . .	11
3.1.1	Scopo . . . . .	11
3.1.2	Descrizione . . . . .	11
3.1.3	Aspettative . . . . .	11
3.1.4	Attività . . . . .	11
3.1.4.1	Stesura dei documenti . . . . .	11
3.1.4.1.1	Descrizione . . . . .	11
3.1.4.1.2	Ciclo di vita del documento . . . . .	11
3.1.4.2	Strutturazione dei documenti . . . . .	11
3.1.4.2.1	Descrizione . . . . .	11
3.1.4.2.2	Template . . . . .	11
3.1.4.3	Classificazione dei documenti . . . . .	11
3.1.4.3.1	Descrizione . . . . .	11
3.1.4.3.2	Documenti informali . . . . .	12
3.1.4.3.3	Documenti formali . . . . .	12
3.1.4.4	Redazione Verbali . . . . .	12
3.1.4.4.1	Descrizione . . . . .	12
3.1.4.4.2	Prima pagina . . . . .	12
3.1.4.4.3	Indice . . . . .	12
3.1.4.4.4	Informazioni sull'incontro . . . . .	13
3.1.4.4.5	Punti discussi . . . . .	13
3.1.4.4.6	Esito incontro . . . . .	13
3.1.4.4.7	Tracciamento delle decisioni . . . . .	13
3.1.4.4.8	Struttura delle pagine . . . . .	13
3.1.4.5	Redazione altri documenti . . . . .	13
3.1.4.5.1	Descrizione . . . . .	13
3.1.4.5.2	Documenti . . . . .	14
3.1.4.5.3	Prima pagina . . . . .	14
3.1.4.5.4	Registro delle modifiche . . . . .	14
3.1.4.5.5	Indice . . . . .	14
3.1.4.5.6	Elenco delle tabelle e delle figure . . . . .	14
3.1.4.5.7	Struttura delle pagine . . . . .	15
3.1.4.6	Glossario . . . . .	15
3.1.4.6.1	Descrizione . . . . .	15
3.1.4.7	Definizione norme tipografiche . . . . .	15
3.1.4.7.1	Descrizione . . . . .	15
3.1.4.7.2	Nomi dei file e delle <i>directory</i> . . . . .	15
3.1.4.7.3	Stile di testo . . . . .	16
3.1.4.7.4	Sigle . . . . .	16
3.1.4.7.5	Formato della data . . . . .	16
3.1.4.7.6	Verifica ortografica . . . . .	16
3.1.4.7.7	Indice di Gulpease <sub>G</sub> . . . . .	16
3.1.4.8	Elementi grafici . . . . .	16



	3.1.4.8.1	Descrizione . . . . .	16
	3.1.4.8.2	Tabelle . . . . .	16
	3.1.4.8.3	Immagini . . . . .	17
	3.1.4.8.4	Grafici . . . . .	17
	3.1.4.9	Metriche di qualità . . . . .	17
	3.1.4.9.1	MQPD01 - Indice di Gulpease <sub>G</sub> . . . . .	17
	3.1.4.9.2	MQPD02 - Errori ortografici . . . . .	17
	3.1.4.10	Strumenti . . . . .	17
3.2		Gestione della configurazione . . . . .	17
3.2.1		Scopo . . . . .	17
3.2.2		Descrizione . . . . .	17
3.2.3		Aspettative . . . . .	18
3.2.4		Attività . . . . .	18
	3.2.4.1	Versionamento <sub>G</sub> . . . . .	18
	3.2.4.1.1	Descrizione . . . . .	18
	3.2.4.1.2	Codice di versione . . . . .	18
	3.2.4.1.3	Sistemi software utilizzati . . . . .	18
	3.2.4.1.4	Formato dei file . . . . .	19
	3.2.4.1.5	Norme sui commit . . . . .	19
	3.2.4.2	Struttura del repository <sub>G</sub> . . . . .	19
	3.2.4.2.1	Descrizione . . . . .	19
	3.2.4.2.2	SkyNet-UniPd/Docs . . . . .	19
	3.2.4.2.3	SkyNet-UniPd/Documentation . . . . .	19
	3.2.4.2.4	SkyNet-UniPd/PoC . . . . .	19
	3.2.4.2.5	SkyNet-UniPd/ChatBot . . . . .	19
	3.2.4.2.6	SkyNet-UniPd/Dev . . . . .	19
	3.2.5	Metriche di qualità . . . . .	19
	3.2.6	Strumenti . . . . .	19
3.3		Gestione della qualità . . . . .	20
3.3.1		Scopo . . . . .	20
3.3.2		Descrizione . . . . .	20
3.3.3		Aspettative . . . . .	20
3.3.4		Attività . . . . .	20
	3.3.4.1	Definizione obiettivi di qualità . . . . .	20
	3.3.4.1.1	Descrizione . . . . .	20
	3.3.4.1.2	Procedure . . . . .	20
	3.3.4.1.3	Denominazione metriche di qualità . . . . .	20
	3.3.5	Metriche di qualità . . . . .	21
	3.3.5.1	MQPC013 - Metrics Satisfied (MS) . . . . .	21
	3.3.6	Strumenti . . . . .	21
3.4		Gestione dei cambiamenti . . . . .	21
3.4.1		Scopo . . . . .	21
3.4.2		Descrizione . . . . .	21
3.4.3		Aspettative . . . . .	21
3.4.4		Attività . . . . .	21
	3.4.4.1	Ciclo risolutivo . . . . .	21
	3.4.4.2	Denominazione cambiamenti . . . . .	22
	3.4.5	Metriche di qualità . . . . .	22
	3.4.6	Strumenti . . . . .	22
3.5		Verifica . . . . .	22
3.5.1		Scopo . . . . .	22
3.5.2		Descrizione . . . . .	22
3.5.3		Aspettative . . . . .	22
3.5.4		Attività . . . . .	23
	3.5.4.1	Analisi . . . . .	23
	3.5.4.1.1	Descrizione . . . . .	23



3.5.4.1.2	Analisi statica . . . . .	23
3.5.4.1.3	Analisi dinamica . . . . .	23
3.5.4.2	Test . . . . .	23
3.5.4.2.1	Descrizione . . . . .	23
3.5.4.2.2	Test di unità . . . . .	24
3.5.4.2.3	Test di integrazione . . . . .	24
3.5.4.2.4	Test di sistema . . . . .	24
3.5.4.2.5	Test di regressione . . . . .	24
3.5.5	Metriche di qualità . . . . .	24
3.5.5.1	MQPC10 - Tests Passed (TP) . . . . .	24
3.5.5.2	MQPC11 - Code Coverage (CC) . . . . .	25
3.5.5.3	MQPC12 - Static Code Analysis (SCA) . . . . .	25
3.5.6	Strumenti . . . . .	25
3.6	Validazione . . . . .	25
3.6.1	Scopo . . . . .	25
3.6.2	Descrizione . . . . .	25
3.6.3	Aspettative . . . . .	25
3.6.4	Attività . . . . .	26
3.6.4.1	Testing con il committente . . . . .	26
3.6.4.1.1	Descrizione . . . . .	26
3.6.4.1.2	Test di accettazione . . . . .	26
3.6.5	Metriche di qualità . . . . .	26
3.6.6	Strumenti . . . . .	26
<b>4</b>	<b>Processi Organizzativi</b>	<b>27</b>
4.1	Gestione dei processi . . . . .	27
4.1.1	Scopo . . . . .	27
4.1.2	Descrizione . . . . .	27
4.1.3	Aspettative . . . . .	27
4.1.4	Attività . . . . .	27
4.1.4.1	Assegnazione ruoli di progetto . . . . .	27
4.1.4.1.1	Descrizione . . . . .	27
4.1.4.1.2	Responsabile di progetto . . . . .	28
4.1.4.1.3	Amministratore di progetto . . . . .	28
4.1.4.1.4	Analista . . . . .	28
4.1.4.1.5	Progettista . . . . .	28
4.1.4.1.6	Programmatore . . . . .	28
4.1.4.1.7	Verificatore . . . . .	29
4.1.4.2	Assegnazione delle attività . . . . .	29
4.1.4.2.1	Descrizione . . . . .	29
4.1.4.3	Gestione delle comunicazioni . . . . .	29
4.1.4.3.1	Descrizione . . . . .	29
4.1.4.3.2	Comunicazioni interne . . . . .	29
4.1.4.3.3	Comunicazioni esterne . . . . .	29
4.1.4.4	Gestione degli incontri . . . . .	29
4.1.4.4.1	Descrizione . . . . .	29
4.1.4.4.2	Incontri interni . . . . .	30
4.1.4.4.3	Incontri esterni . . . . .	30
4.1.4.5	Gestione degli strumenti di coordinamento . . . . .	30
4.1.4.5.1	Descrizione . . . . .	30
4.1.4.5.2	Software per la gestione dei task . . . . .	30
4.1.4.6	Gestione degli strumenti di <i>versionamento</i> <sub>G</sub> . . . . .	30
4.1.4.6.1	Descrizione . . . . .	30
4.1.4.6.2	Repository <sub>G</sub> . . . . .	30
4.1.4.6.3	Tipi di file e <i>.gitignore</i> . . . . .	30
4.1.4.7	Gestione dei rischi . . . . .	31



---

	4.1.4.7.1	Descrizione . . . . .	31
	4.1.4.7.2	Codifica dei rischi . . . . .	31
4.1.5		Metriche di qualità . . . . .	31
	4.1.5.1	MQPC14 - Non-Calculated Risks (NCR) . . . . .	31
4.1.6		Strumenti . . . . .	31
4.2		Formazione . . . . .	31
	4.2.1	Scopo . . . . .	31
	4.2.2	Descrizione . . . . .	31
	4.2.3	Aspettative . . . . .	31
	4.2.4	Attività . . . . .	32
	4.2.4.1	Modalità di formazione . . . . .	32
	4.2.4.1.1	Descrizione . . . . .	32
	4.2.4.1.2	Fonti . . . . .	32
	4.2.5	Metriche di qualità . . . . .	32
	4.2.6	Strumenti . . . . .	32



# 1 Introduzione

## 1.1 Scopo del documento

Il seguente documento definisce le regole e le procedure fondamentali che ciascun membro del gruppo *SkyNet* si impegna a visionare e rispettare.

Il team ha deciso di intraprendere un approccio di tipo incrementale, in modo da aggiornare il documento di volta in volta al seguito di ogni decisione presa dal gruppo.

## 1.2 Scopo del capitolato

Oggigiorno, nelle realtà aziendali si fa uso di sempre più strumenti informatici che permettono ai dipendenti di svolgere varie operazioni che in passato venivano svolte su carta. Tuttavia, con l'aumentare delle diverse piattaforme e strumenti utilizzati, aumenta la complessità di gestione e il tempo speso dai dipendenti per l'accesso e l'inserimento dei dati. Il capitolato in questione si pone quindi l'obiettivo di fornire un'unica piattaforma (un *ChatBot<sub>G</sub>* aziendale) che, integrandosi con i sistemi già in uso dall'azienda, permetta ai dipendenti di svolgere operazioni come il tracciamento della presenza in sede, la consuntivazione oraria per un'attività o l'apertura del cancello di una sede con un unico servizio accessibile tramite dispositivo *mobile<sub>G</sub>*.

## 1.3 Glossario

Per evitare incomprensioni e ambiguità durante la lettura del documento, vengono utilizzati due simboli a pedice di alcuni termini, con le seguenti funzioni:

- *G* per indicare i termini la cui definizione si trova nel Glossario v3.0.0<sub>D</sub>;
- *D* per indicare il nome di un documento esterno.

## 1.4 Organizzazione delle sezioni

Il documento presenta le seguenti sezioni:

- **Processi Primari;**
- **Processi di supporto;**
- **Processi organizzativi.**

Ognuna di queste sezioni contiene i processi che vengono istanziati per lo svolgimento del progetto. Ogni processo presenta la seguente struttura organizzativa:

- **Scopo:** obiettivi del processo;
- **Descrizione:** descrizione del processo;
- **Aspettative:** aspettative del gruppo per il processo;
- **Attività:** elenco delle attività che compongono il processo.  
Ogni attività può contenere le seguenti sotto sezioni:
  - **Scopo:** scopo dell'attività;
  - **Descrizione:** descrizione dell'attività;
  - **Procedure:** procedure per lo svolgimento dell'attività;
  - **Documenti:** documenti prodotti durante l'attività;
  - **Metriche di qualità:** misurazioni per garantire la qualità dell'attività;
  - **Strumenti:** strumenti utilizzati per lo svolgimento dell'attività.
- **Metriche di qualità:** misurazioni per garantire la qualità del processo;
- **Strumenti:** strumenti utilizzati per la gestione del processo.

## 1.5 Riferimenti

### 1.5.1 Normativi

- Regolamento del progetto didattico  
<https://www.math.unipd.it/tullio/IS-1/2021/Dispense/PD2.pdf>
- Capitolato d'appalto - Bot4Me  
<https://www.math.unipd.it/tullio/IS-1/2021/Progetto/C1p.pdf>

### 1.5.2 Informativi

- Standard ISO/IEC 12207-1995  
[https://www.math.unipd.it/tullio/IS-1/2009/Approfondimenti/ISO\\_12207-1995.pdf](https://www.math.unipd.it/tullio/IS-1/2009/Approfondimenti/ISO_12207-1995.pdf)
- Slide del corso - Processi di ciclo di vita del software  
<https://www.math.unipd.it/tullio/IS-1/2021/Dispense/T03.pdf>

## 2 Processi Primari

### 2.1 Fornitura

#### 2.1.1 Scopo

In questa sezione del documento verranno descritte le norme che ogni membro del gruppo si impegna a rispettare durante l'intera realizzazione del progetto, al fine di soddisfare i requisiti richiesti dal proponente *Imola Informatica*.

#### 2.1.2 Descrizione

Il processo di fornitura stabilisce le attività e le risorse necessarie alla realizzazione del progetto e alla stesura dei documenti *Piano di Qualifica v2.0.0<sub>D</sub>* e *Piano di Progetto v1.0.0<sub>D</sub>*, i quali si occupano rispettivamente di fornire metriche per misurare la qualità del prodotto e di gestire le scadenze temporali e il preventivo dei costi che il gruppo si impegna a rispettare. Il processo di fornitura è suddiviso nelle seguenti fasi:

1. Avvio;
2. Approntamento di risposte alle richieste;
3. Contrattazione;
4. Pianificazione;
5. Esecuzione e controllo;
6. Revisione e valutazione;
7. Consegna e completamento.

#### 2.1.3 Aspettative

Le aspettative riguardo il processo di fornitura sono:

1. Dialogare in maniera costante con il proponente per determinare i bisogni da soddisfare con il prodotto finale;
2. Definire il piano di lavoro, stimando tempistiche e costi;
3. Stabilire vincoli sui requisiti richiesti dal proponente;
4. Risolvere dubbi riguardo ai vincoli stabiliti;
5. Delineare la struttura documentale;
6. Effettuare una verifica continua sul prodotto.

#### 2.1.4 Attività

##### 2.1.4.1 Redazione Piano di Progetto

###### 2.1.4.1.1 Descrizione

Il documento *Piano di Progetto<sub>D</sub>* contiene il piano di lavoro che il gruppo deve adottare per lo sviluppo del progetto.

#### 2.1.4.1.2 Procedure

Redazione del documento *Piano di Progetto<sub>D</sub>*, che deve contenere le seguenti sezioni:

- **Analisi dei rischi:** definizione dei rischi potenziali e riscontrati nelle varie fasi di sviluppo;
- **Modello di sviluppo:** definizione del modello di sviluppo scelto;
- **Pianificazione:** pianificazione preventiva delle attività del progetto;
- **Preventivo:** preventivo dei costi e delle ore per ogni fase;
- **Consuntivo:** consuntivo dei costi e delle ore effettivamente svolte e analisi retrospettiva per effettuare eventuali miglioramenti.

#### 2.1.4.2 Redazione Piano di Qualifica

##### 2.1.4.2.1 Descrizione

Il documento *Piano di Qualifica<sub>D</sub>* contiene tutte le misure da adottare per garantire la qualità del prodotto.

##### 2.1.4.2.2 Procedure

Redazione del documento *Piano di Qualifica<sub>D</sub>*, che deve contenere le seguenti sezioni:

- **Qualità di processo:** metriche per garantire la qualità dei processi;
- **Qualità di prodotto:** metriche per garantire la qualità delle componenti prodotte;
- **Test:** specifica sui test da effettuare sulle componenti software;
- **Resoconto attività di verifica:** resoconto dell'andamento delle metriche nei vari periodi e considerazioni finali.

#### 2.1.5 Metriche di qualità

##### 2.1.5.1 MQPC01 - Budgeted Cost of Work Scheduled (BCWS)

Metrica che indica il costo in euro pianificato per realizzare le attività di progetto alla data corrente.

##### 2.1.5.2 MQPC02 - Actual Cost of Work Performed (ACWP)

Metrica che indica il costo in euro effettivamente sostenuto per realizzare le attività di progetto alla data corrente.

##### 2.1.5.3 MQPC03 - Budgeted Cost of Work Performed (BCWP)

Metrica che indica il valore in euro del prodotto alla data corrente.

##### 2.1.5.4 MQPC04 - Cost Variance (CV)

Metrica che indica il rapporto tra i costi sostenuti e i costi preventivati mediante la formula:

$$CV = BCWP - ACWP$$

Il risultato deve essere interpretato nel seguente modo:

- **0:** i costi sostenuti non si discostano da quelli previsti;
- **> 0:** i costi sostenuti sono minori di quelli previsti;
- **< 0:** i costi sostenuti sono maggiori di quelli previsti.

#### 2.1.5.5 MQPC05 - Schedule Variance (SV)

Metrica che indica il rapporto tra lo stato di avanzamento attuale e quello programmato mediante la formula:

$$SV = BCWP - BCWS$$

Il risultato deve essere interpretato nel seguente modo:

- **0**: il progetto avanza con il ritmo di lavoro previsto;
- **> 0**: il progetto avanza ad un ritmo superiore rispetto a quello previsto;
- **< 0**: il progetto avanza ad un ritmo inferiore rispetto a quello previsto.

#### 2.1.5.6 MQPC06 - Estimated At Completion (EAC)

Indice della stima del costo complessivo a fine progetto basandosi sui costi alla data corrente:

$$EAC = ACWP + \frac{(BAC - BCWP)}{CPI}$$

dove *BAC* indica il budget previsto per lo sviluppo del progetto.

#### 2.1.5.7 MQPC07 - Schedule Performance Index (SPI)

Indice della stima del rapporto tra la parte di budget già spesa alla data corrente e la parte di budget prevista, dato dalla formula:

$$SPI = \frac{BCWP}{BCWS}$$

Il risultato deve essere interpretato nel seguente modo:

- **1**: l'avanzamento del progetto è uguale a quanto previsto;
- **> 1**: l'avanzamento del progetto è minore rispetto a quanto previsto;
- **< 1**: l'avanzamento del progetto è maggiore rispetto a quanto previsto.

#### 2.1.5.8 MQPC08 - Cost Performance Index (CPI)

Indice della stima del rapporto tra il costo previsto per le attività svolte e quanto è stato effettivamente speso alla data corrente, dato dalla formula:

$$CPI = \frac{BCWP}{ACWP}$$

Il risultato deve essere interpretato nel seguente modo:

- **1**: il costo del progetto è uguale a quanto previsto;
- **> 1**: il costo del progetto è minore rispetto a quanto previsto;
- **< 1**: il costo del progetto è maggiore rispetto a quanto previsto.

### 2.1.6 Strumenti

Non sono stati utilizzati strumenti in particolare per la gestione di questo processo.

## 2.2 Sviluppo

### 2.2.1 Scopo

Lo scopo del processo di sviluppo è quello di descrivere le attività di analisi, progettazione, codifica, integrazione, test, installazione ed accettazione, relative allo sviluppo del prodotto software. In questa sezione vengono descritte le norme adottate per la realizzazione di tale processo.

### 2.2.2 Descrizione

Il processo di sviluppo è caratterizzato dalle seguenti attività:

- Analisi dei requisiti;
- Progettazione;
- Codifica.

### 2.2.3 Aspettative

Le aspettative riguardanti il processo di sviluppo sono le seguenti:

- determinare vincoli tecnologici;
- determinare gli obiettivi di sviluppo;
- determinare vincoli di design;
- realizzare un prodotto finale che superi i test e soddisfi i requisiti del proponente.

### 2.2.4 Attività

#### 2.2.4.1 Analisi dei requisiti

##### 2.2.4.1.1 Scopo

Lo scopo dell'analisi dei requisiti è di individuare i requisiti (sia impliciti che espliciti) richiesti dal proponente per il prodotto ed effettuare lo studio dei diversi casi d'uso del prodotto stesso.

##### 2.2.4.1.2 Descrizione

Analisi del capitolato e studio dei diversi casi d'uso del prodotto al fine di definire i requisiti che il prodotto deve soddisfare.

##### 2.2.4.1.3 Procedure

Redazione del documento di *Analisi dei Requisiti v3.0.0<sub>D</sub>*, che deve essere composto dalle seguenti sezioni:

- **Descrizione del prodotto:** analisi del prodotto e definizione degli attori<sub>G</sub> principali e secondari;
- **Casi d'uso:** elenco degli scenari designabili nel prodotto;
- **Requisiti:** classificazione dei requisiti individuati;
- **Tracciamento requisiti:** mappatura dei requisiti individuati.

##### 2.2.4.1.4 Classificazione dei requisiti

Ogni requisito viene identificato tramite un codice identificativo univoco che rispetta il seguente pattern:

**R[Importanza][Tipologia][Codice]**

1. **Importanza:** indica l'importanza di un dato requisito, può assumere un valore numerico tra 1 e 3 e ha il seguente significato:

- 1 - Requisito obbligatorio;
- 2 - Requisito desiderabile;
- 3 - Requisito opzionale.

2. **Tipologia:** rappresenta la natura del requisito e può assumere uno dei seguenti valori:

**V** : Di vincolo;  
**P** : Prestazionale;  
**Q** : Qualitativo;  
**F** : Funzionale.

3. **Codice:** un identificativo univoco del requisito.

Ogni requisito è inoltre corredato dalle seguenti informazioni:

- **Descrizione:** breve descrizione del requisito;
- **Fonti:** elenco delle fonti da cui deriva il requisito, che possono essere:
  - Capitolato;
  - Interna;
  - Caso d'uso.

#### 2.2.4.1.5 Classificazione dei casi d'uso

Ogni caso d'uso viene identificato da un codice univoco con la seguente sintassi:

**UC[CodiceCaso].[CodiceSottocaso]**

**CodiceCaso:** numero progressivo che identifica il caso d'uso generico;

**CodiceSottocaso:** numero progressivo che identifica il sotto caso.

Ogni caso d'uso è inoltre corredato dalle seguenti informazioni:

- **Nome:** nome del caso d'uso;
- **Diagramma UML<sub>G</sub>:** rappresentazione grafica opzionale;
- **Attori:**<sub>G</sub> principali e secondari;
- **Descrizione:** descrizione schematica del caso d'uso;
- **Precondizione:** stato del sistema prima del verificarsi degli eventi del caso d'uso;
- **Postcondizione:** stato del sistema dopo che si sono verificati gli eventi del caso d'uso;
- **Scenario principale:** flusso degli eventi;
- **Estensioni:** scenari alternativi a quello principale.

#### 2.2.4.1.6 Strumenti

Per la creazione dei diagrammi UML<sub>G</sub> viene utilizzato l'editor [StarUML](#).

#### 2.2.4.2 Progettazione

##### 2.2.4.2.1 Scopo

Lo scopo dell'attività di progettazione è quello di definire le caratteristiche del prodotto software in base ai requisiti definiti durante l'attività di analisi dei requisiti.

#### 2.2.4.2.2 Descrizione

L'attività di progettazione definisce l'architettura del prodotto software ricercando una soluzione soddisfacente per tutti gli stakeholder senza usare codice e precede l'attività di codifica. La progettazione si divide in due attività:

- Progettazione architetturale: a partire dall'*Analisi dei requisiti<sub>G</sub>*, produce la *Technology Baseline<sub>G</sub>*;
- Progettazione di dettaglio: segue la progettazione architetturale e produce la *Product Baseline<sub>G</sub>*.

#### 2.2.4.2.3 Procedure

- *Technology Baseline<sub>G</sub>*;
- *Product Baseline<sub>G</sub>*.

#### 2.2.4.2.4 Qualità dell'architettura

Per garantire la qualità del prodotto, l'architettura logica dovrà presentare le seguenti caratteristiche:

- Soddisfare tutti i requisiti definiti nel documento *Analisi dei Requisiti v3.0.0<sub>D</sub>*;
- Essere comprensibile per il proponente;
- Definire componenti semplici, modulari e riusabili;
- Essere flessibile in caso di modifiche ai requisiti;
- Essere affidabile, quindi svolgere in maniera corretta le sue funzioni quando richiesto;
- Risultare sicura e robusta in caso di malfunzionamenti o intrusioni;
- Essere disponibile in tempi brevi in caso di manutenzione;
- Utilizzare in maniera efficiente le risorse disponibili;
- Utilizzare l'incapsulamento per ridurre le dipendenze tra componenti.

#### 2.2.4.2.5 Technology Baseline<sub>G</sub>

Definisce le specifiche della progettazione ad alto livello del prodotto e delle sue componenti, individua le interazioni tra le componenti e fornisce una definizione dei test di integrazione. Deve includere:

- **Tecnologie utilizzate:** elenco e descrizione delle tecnologie, librerie e *framework<sub>G</sub>* utilizzati;
- **Proof of Concept<sub>G</sub>:** bozza eseguibile del progetto che funge da *baseline<sub>G</sub>* per lo sviluppo successivo del progetto;
- **Tracciamento componenti:** mappatura per tracciare la corrispondenza tra le componenti e i requisiti;
- **Test di integrazione:** test per verificare la corretta interazione tra le componenti.

#### 2.2.4.2.6 Product Baseline<sub>G</sub>

Definisce le scelte di progettazione di dettaglio, successiva alla *Technology Baseline<sub>G</sub>*, e contiene:

1. **Design Pattern<sub>G</sub>:** descrizione dei *design pattern<sub>G</sub>* utilizzati;
2. **Specifica architetturale:** insieme di diagrammi UML<sub>G</sub>. Vengono utilizzati:
  - Diagrammi delle classi:** rappresentano le classi del sistema, con attributi, operazioni e relazioni;
  - Diagrammi di sequenza:** descrivono le procedure e le interazioni tra le classi per svolgere una determinata operazione;
  - Diagrammi di attività:** descrivono il flusso di operazioni di un processo.
3. **Test:** definizione dei test di verifica. Vengono definiti test di unità, di integrazione e di sistema;
4. **Tracciamento classi:** mappatura per tracciare la corrispondenza tra classi e requisiti.



#### 2.2.4.2.7 Diagrammi delle classi

I diagrammi delle classi consentono di descrivere i tipi di oggetti presenti nel sistema, con le proprietà e le relazioni tra essi, astruendo il linguaggio di programmazione.

Per la realizzazione di tali diagrammi il gruppo si attiene alle norme e linee guida definite da:

- [OMG - UML<sub>G</sub> 2.0](#).

#### 2.2.4.2.8 Diagrammi di sequenza

I diagrammi di sequenza descrivono la sequenza di realizzazione di un determinato comportamento tramite la collaborazione degli oggetti rappresentati tramite i diagrammi delle classi, tenendo conto anche dell'ordine in cui interagiscono.

Per la realizzazione di tali diagrammi il gruppo si attiene alle norme e linee guida definite da:

- [OMG - UML<sub>G</sub> 2.0](#).

#### 2.2.4.3 Codifica

##### 2.2.4.3.1 Scopo

L'attività di codifica ha lo scopo di definire le norme che il gruppo deve rispettare nella stesura del codice. Gli obiettivi di questa attività sono:

- scrivere codice leggibile e uniforme;
- agevolare le fasi di verifica e manutenzione;
- fornire un prodotto che soddisfi le aspettative del proponente;
- realizzare un prodotto di qualità facilmente mantenibile ed estendibile.

##### 2.2.4.3.2 Descrizione

L'attività di codifica deve seguire gli obiettivi qualitativi definiti nel documento *Piano di Qualifica v2.0.0<sub>D</sub>*.

##### 2.2.4.3.3 Procedure

Ogni programmatore si deve attenere alle seguenti linee guida, a seconda del linguaggio di programmazione utilizzato:

- **Python<sub>G</sub>**: linee guida fornite dal [sito web](#) del linguaggio;
- **HTML 5<sub>G</sub>**: linee guida fornite dal [sito web](#) W3C;
- **CSS 3<sub>G</sub>**: linee guida fornite dal [sito web](#) W3C;
- **Javascript<sub>G</sub>**: linee guida fornite da [Google](#).

##### 2.2.4.3.4 Strumenti

Per agevolare il rispetto delle linee guida sopracitate, vengono utilizzati i seguenti strumenti:

- Visual Studio Code
- [HTML Validator](#);
- [CSS Validator](#);
- <https://browserslist>.

## 2.2.5 Metriche di qualità

### 2.2.5.1 MQPC09 - Requirements Stability Index (RSI)

Indice che definisce la percentuale di stabilità dei requisiti tramite la seguente formula:

$$RSI = 1 - \left( \frac{requisitiCambiati + requisitiRimossi + requisitiAggiunti}{requisitiIniziali} \right) * 100$$

Il risultato ottenuto può indicare le seguenti deduzioni:

- 0%: nessun requisito iniziale non ha subito cambiamenti;
- $0\% < x < 100\%$ : solo alcuni requisiti iniziali non hanno subito cambiamenti;
- 100%: tutti i requisiti iniziali non hanno subito cambiamenti.

### 2.2.5.2 MQPD03 - Copertura requisiti obbligatori

Percentuale dei requisiti obbligatori soddisfatti, definita dalla formula:

$$\frac{requisitiObbligatoriSoddisfatti}{requisitiObbligatori}$$

### 2.2.5.3 MQPD04 - Copertura requisiti desiderabili

Percentuale dei requisiti desiderabili soddisfatti, definita dalla formula:

$$\frac{requisitiDesiderabiliSoddisfatti}{requisitiDesiderabili}$$

### 2.2.5.4 MQPD05 - Copertura requisiti opzionali

Percentuale dei requisiti opzionali soddisfatti, definita dalla formula:

$$\frac{requisitiOpzionaliSoddisfatti}{requisitiOpzionali}$$

### 2.2.5.5 MQPD06 - Facilità di utilizzo

Valore che indica la facilità di utilizzo dell'interfaccia utente, valutando il numero di iterazioni da compiere per raggiungere la funzionalità desiderata.

### 2.2.5.6 MQPD07 - Numero di browser<sub>G</sub> supportati

Metriche che serve a verificare quanti browser<sub>G</sub> vengono supportati dall'applicativo.

Per verificare quali browser<sub>G</sub> siano supportati dall'applicativo prodotto, viene utilizzato lo strumento *Browserslist* disponibile al seguente link: <https://browserslist>

## 2.2.6 Strumenti

- **Browserslist:** utilizzato per verificare quali browser<sub>G</sub> vengono supportati dall'applicativo e in quali versioni.

Per gli altri strumenti utilizzati si riporta all'elenco presente nella sezione §2.2.4.3.4 .

## 3 Processi di supporto

### 3.1 Documentazione

#### 3.1.1 Scopo

Scopo di questa sezione è quello di definire le norme per redigere e verificare la documentazione.

#### 3.1.2 Descrizione

Questo processo serve a definire uno standard per la stesura dei documenti, in modo da garantire coerenza e uniformità sia per quelli interni che per quelli esterni.

#### 3.1.3 Aspettative

Le aspettative del gruppo per questo processo sono:

- Avere una chiara struttura per i documenti, in modo da ottenere un risultato uniforme;
- Avere delle norme e convenzioni ben precise che coprano tutti gli aspetti della stesura di un documento.

#### 3.1.4 Attività

##### 3.1.4.1 Stesura dei documenti

###### 3.1.4.1.1 Descrizione

Normare la stesura dei documenti, definendo le fasi del ciclo di vita degli stessi.

###### 3.1.4.1.2 Ciclo di vita del documento

Le fasi del ciclo di vita di un documento sono le seguenti:

1. **Stesura:** viene redatto il contenuto del documento;
2. **Verifica:** le diverse sezioni del documento vengono verificate, aggiornate con ulteriori informazioni, corrette se necessario;
3. **Approvazione:** il responsabile del documento approva il suo contenuto per il rilascio.

##### 3.1.4.2 Strutturazione dei documenti

###### 3.1.4.2.1 Descrizione

Definire la struttura per il *layout* dei diversi documenti redatti.

###### 3.1.4.2.2 Template

Il team di sviluppo ha deciso di realizzare un template mediante l'utilizzo di  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}_G$  in modo da avere una struttura uniforme di tutte le pagine nei vari documenti. Questo velocizza inoltre la stesura dei documenti stessi, in quanto quando ciascun membro del gruppo dovrà redigere alcune sezioni, dovrà solamente occuparsi della parte di contenuto e non della struttura grafica.

##### 3.1.4.3 Classificazione dei documenti

###### 3.1.4.3.1 Descrizione

Classificare i documenti in base alle loro caratteristiche primarie e al loro scopo.

#### 3.1.4.3.2 Documenti informali

Vengono definiti documenti informali tutti i documenti che:

- non sono ancora stati approvati dal responsabile di progetto;
- non sono soggetti a *versionamento<sub>G</sub>*, come ad esempio i verbali.

Tali documenti possono inoltre essere ad uso:

- **Esterno** se sono di interesse per il gruppo, il committente e il proponente;
- **Interno** se sono solo di interesse per il gruppo.

#### 3.1.4.3.3 Documenti formali

Vengono definiti documenti formali tutti i documenti che:

- sono stati approvati dal responsabile di progetto;
- sono soggetti a *versionamento<sub>G</sub>*;
- sono rilasciabili all'esterno del gruppo.

Tali documenti possono inoltre essere ad uso:

- **Esterno** se sono di interesse per il gruppo, il committente e il proponente;
- **Interno** se sono solo di interesse per il gruppo.

#### 3.1.4.4 Redazione Verbali

##### 3.1.4.4.1 Descrizione

I verbali sono documenti redatti a seguito di una riunione interna o con il proponente di progetto, e ne contengono un resoconto dettagliato. Prevedono una singola versione, in quanto non subiscono modifiche una volta redatti.

##### 3.1.4.4.2 Prima pagina

La struttura della prima pagina di ogni verbale è la seguente:

- **Logo:** il logo del gruppo è collocato in alto al centro;
- **Titolo:** il titolo del documento specifica se si tratta di un verbale interno o esterno, e lo identifica con un numero progressivo;
- **E-mail del gruppo:** indirizzo email del gruppo;
- **Data incontro:** data in cui si è svolto l'incontro;
- **Informazioni documento:**
  - Responsabile: indica il nome del membro con il ruolo di responsabile;
  - Redattori: indica il nome dei membri che hanno redatto il documento;
  - Verificatori: indica il nome dei membri che hanno verificato il documento;
  - Stato: Indica lo stato attuale del documento (Approvato, In attesa di approvazione).

##### 3.1.4.4.3 Indice

Indice che racchiude il contenuto del documento.

#### 3.1.4.4.4 Informazioni sull'incontro

Questa sezione contiene le seguenti informazioni sull'incontro tenutosi:

- **Data incontro:** data in cui si è svolto l'incontro;
- **Data approvazione:** data in cui il verbale è stato approvato;
- **Ora:** orario dell'incontro;
- **Luogo:** luogo in cui si è tenuto l'incontro;
- **Presenti:** elenco dei membri del gruppo presenti all'incontro;
- **Partecipanti esterni:** elenco dei nomi di partecipanti esterni al gruppo, se presenti;
- **Motivazione dell'incontro:** breve descrizione della motivazione per cui è stato tenuto l'incontro.

#### 3.1.4.4.5 Punti discussi

Riporta un elenco delle tematiche affrontate durante l'incontro.

#### 3.1.4.4.6 Esito incontro

Riporta in maniera riassuntiva le decisioni prese durante l'incontro.

#### 3.1.4.4.7 Tracciamento delle decisioni

Riporta in forma tabulare le decisioni prese durante l'incontro, in modo da poterle tracciare ed identificare i corrispondenti *issue* gestiti tramite la piattaforma *AsanaG*.

Ogni decisione è identificata tramite un codice univoco che segue il seguente pattern:

VX-MM-DD-Y

dove:

**X:** uso del verbale, **I** se interno, **E** se esterno;

**MM:** mese in cui è stato redatto il verbale;

**DD:** giorno in cui è stato redatto il verbale;

**Y:** numero progressivo che identifica la decisione.

#### 3.1.4.4.8 Struttura delle pagine

Ogni pagina di contenuto è strutturata come segue:

- **Intestazione** composta da:
  - **Logo del gruppo** a sinistra;
  - **Nome del gruppo** a destra.
- **Piè di pagina** in cui si trova il **numero di pagina** rispetto alle pagine totali, a destra.

#### 3.1.4.5 Redazione altri documenti

##### 3.1.4.5.1 Descrizione

In questa sezione viene descritta la struttura di tutti i documenti che non sono verbali.

#### 3.1.4.5.2 Documenti

- *Analisi dei Requisiti*;
- *Piano di Progetto*;
- *Piano di Qualifica*;
- *Glossario*;
- *Norme di Progetto*.

#### 3.1.4.5.3 Prima pagina

- **Logo:** il logo del gruppo è collocato in alto al centro;
- **Titolo:** il titolo del documento;
- **Nome progetto:** il nome del progetto;
- **E-mail del gruppo:** indirizzo email di gruppo;
- **Data di rilascio:** data in cui viene rilasciato il documento;
- **Informazioni documento:**
  - Responsabile: indica il nome del membro con il ruolo di responsabile;
  - Redattori: indica il nome dei membri che hanno redatto il documento;
  - Verificatori: indica il nome dei membri che hanno verificato il documento;
  - Destinatari: indica i destinatari del documento;
  - Uso: specifica se il documento sia per uso interno o esterno;
  - Stato: Indica lo stato attuale del documento (Approvato, In attesa di approvazione).
  - Versione: indica la versione corrente del documento.

#### 3.1.4.5.4 Registro delle modifiche

In questa sezione viene riportata una tabella che descrive il registro delle modifiche effettuate al documento in oggetto, con i seguenti contenuti:

- **Versione:** versione del documento a seguito della modifica;
- **Autore:** nome di chi ha effettuato la modifica;
- **Data:** data di modifica del documento;
- **Ruolo:** ruolo dell'autore della modifica;
- **Descrizione:** breve descrizione della modifica apportata al documento.

#### 3.1.4.5.5 Indice

Indice che racchiude il contenuto del documento.

#### 3.1.4.5.6 Elenco delle tabelle e delle figure

Elenco delle tabelle e delle figure contenute nel documento, identificate da un codice numerico.

### 3.1.4.5.7 Struttura delle pagine

Ogni pagina di contenuto è strutturata come segue:

- **Intestazione** composta da:
  - **Logo del gruppo** a sinistra;
  - **Nome del gruppo** a destra.
- **Piè di pagina** in cui si trova il **numero di pagina** rispetto alle pagine totali, a destra.

### 3.1.4.6 Glossario

#### 3.1.4.6.1 Descrizione

Questo documento serve a fornire una breve definizione di tutte le parole presenti nei vari documenti che possono risultare ambigue o poco chiare. I termini sono elencati seguendo l'ordine lessicografico. Le norme relative al glossario sono:

- Ogni parola presente nel glossario, verrà contrassegnata con una "G" maiuscola a pedice della parole stessa;
- Se un termine già presente nel glossario appare nuovamente nel testo, ad esso verrà apposta la "G" maiuscola a pedice.

### 3.1.4.7 Definizione norme tipografiche

#### 3.1.4.7.1 Descrizione

Descrivere le convenzioni da adottare per rendere uniformi i documenti.

#### 3.1.4.7.2 Nomi dei file e delle *directory*

I nomi dei documenti e delle *directory* adottano la convenzione *CamelCase*:

- Ogni parola inizia con una lettera maiuscola;
- Le restanti lettere della parola sono in minuscolo;
- Non sono presenti spaziature tra le parole (se prevista più di una);
- Non vengono omesse le preposizioni;
- Nel caso nel documento sia previsto un *changelog<sub>G</sub>*, l'ultima versione del *changelog<sub>G</sub>* viene riportata anche nel nome del documento, concatenata al nome da un carattere di *underscore* "\_".

In particolare possiamo identificare:

- **Verbali:** i verbali dovranno rispettare la seguente forme di nomenclatura:

Verbale\_X\_YY-ZZ.pdf

- **X:** Indica la destinazione, se Interno o Esterno;
- **YY:** Indica il mese in doppia cifra;
- **ZZ:** Indica il giorno in doppia cifra.

- **Documenti formali:** la denominazione seguirà le norme CamelCase e di revisione documentale:

NomeDocumento\_vX.Y.Z

- **vX.Y.Z:** Indicano la versione secondo quanto descritto al paragrafo §3.2.4.1.2 denominato **Codice di versione**.

Le estensioni dei documenti sono escluse da queste norme.

### 3.1.4.7.3 Stile di testo

I differenti stili del testo adottati indicano specifiche funzioni semantiche:

- **Grassetto**: utilizzato per esaltare una parola seguita da una spiegazione, per le sezioni o i titoli;
- *Corsivo*: utilizzato per tecnicismi e per indicare parole rilevanti nel testo in lingua straniera.

### 3.1.4.7.4 Sigle

In questa sezione verranno elencate le sigle presenti nei vari documenti.

1. Sigle che rappresentano i ruoli ricoperti dai vari membri del gruppo:

- **Re**: Responsabile;
- **Am**: Amministratore;
- **An**: Analista;
- **Pg**: Progettista;
- **Pr**: Programmatore;
- **Ve**: Verificatore.

### 3.1.4.7.5 Formato della data

Il gruppo ha deciso di adottare il seguente formato per rappresentare le date che vengono inserite nei nomi dei file dei diversi documenti:

**MM-DD**

in modo da creare un ordine cronologico dei documenti in maniera automatica nel sistema di *versionamento<sub>G</sub>* scelto (*GitHub<sub>G</sub>*). Si è deciso di non inserire l'anno poiché l'intero progetto si svolgerà durante l'anno 2022, quindi è stato ritenuto ridondante aggiungerlo.

Il formato della data all'interno dei documenti sarà invece il seguente:

**DD-MM-YYYY**

perché ritenuto di più facile lettura.

### 3.1.4.7.6 Verifica ortografica

Per il controllo ortografico viene utilizzata la correzione in tempo reale integrata nell'editor *Texmaker*, che permette la visualizzazione immediata degli errori grazie alla sottolineatura delle parole non corrette secondo la lingua italiana; altri strumenti verranno integrati nel paragrafo durante l'avanzamento del progetto.

### 3.1.4.7.7 Indice di Gulpease<sub>G</sub>

Per garantire una buona leggibilità dei documenti viene utilizzato l'*Indice di Gulpease<sub>G</sub>*.

## 3.1.4.8 Elementi grafici

### 3.1.4.8.1 Descrizione

Definire le norme riguardanti gli elementi grafici presenti nei documenti.

### 3.1.4.8.2 Tabelle

Le tabelle di ogni documento, fatta eccezione per il registro delle modifiche, seguono le seguenti convenzioni:

- Ogni tabella contiene al di sotto di essa una breve didascalia descrittiva della tabella stessa;
- Ogni tabella viene identificata con un numero progressivo a partire da 1, per ogni documento.



### 3.1.4.8.3 Immagini

Le immagini sono collocate al centro della pagina e al di sotto di esse è presente una breve descrizione dell'immagine. Sono inoltre identificate con un numero progressivo a partire da 1, per ogni documento.

### 3.1.4.8.4 Grafici

I grafici vengono riportati come immagini, quindi seguono le regole della sezione precedentemente riportate.

I grafici possono essere Diagrammi UML<sub>G</sub>, Diagrammi di Gantt, torte o istogrammi.

### 3.1.4.9 Metriche di qualità

#### 3.1.4.9.1 MQPD01 - Indice di Gulpease<sub>G</sub>

Indice che definisce il grado di leggibilità di un testo in lingua italiana mediante la formula:

$$89 + \frac{300(\text{numeroFrase}) - 10(\text{numeroLettere})}{\text{numeroParole}}$$

#### 3.1.4.9.2 MQPD02 - Errori ortografici

Metrica che permette di misurare la correttezza lessicografica dei documenti contando il numero di errori ortografici presenti.

#### 3.1.4.10 Strumenti

- **L<sup>A</sup>T<sub>E</sub>X<sub>G</sub>**: Linguaggio di *markup* basato sul sistema di composizione *Tex* con approccio al **WYSIWYM (What You See Is What You Mean)**: si possono scrivere documenti concentrandosi nel contenuto e non nella forma. Questo ci viene in aiuto per scrivere esempi di codice e formule. **L<sup>A</sup>T<sub>E</sub>X<sub>G</sub>** necessita di un compilatore, in particolare ogni componente utilizza editor differenti, in base alla familiarità e alle competenze. Per citarne alcuni: *TeXworks* (Windows), *Overleaf* (cross-platform, online);
- Editor di testo: nessuno predefinito, prevalentemente viene usato *Visual Studio Code*;
- *Google Spreadsheet*: usato per la creazione di grafici;
- *StarUML*: usato per la creazione dei diagrammi UML<sub>G</sub>;
- *TeamGantt*: usato per la creazione dei Diagrammi di Gantt<sub>G</sub>;
- *Google Drive*: usato per condividere i grafici;
- [https://farfalla-project.org/readability\\_static/](https://farfalla-project.org/readability_static/): usato per calcolare l'indice di Gulpease<sub>G</sub>.

## 3.2 Gestione della configurazione

### 3.2.1 Scopo

Lo scopo di questo processo è quello di identificare, registrare e verificare le modifiche apportate a documenti e codice.

### 3.2.2 Descrizione

Questo processo definisce l'insieme di norme per rendere mantenibili i documenti e il codice sorgente, definendo le modalità di versionamento<sub>G</sub> e coordinamento per il progetto.

### 3.2.3 Aspettative

Le aspettative del gruppo per questo processo sono le seguenti:

- Condividere i file tra tutti i membri del gruppo in modo semplice ed efficace;
- Possibilità di registrare e verificare una modifica;
- Individuare e correggere eventuali errori e/o conflitti in tempi brevi;
- Possibilità di effettuare il ripristino a una versione precedente del prodotto;
- Possedere un tracciamento di tutte le modifiche effettuate ad un prodotto.

### 3.2.4 Attività

#### 3.2.4.1 Versionamento<sub>G</sub>

##### 3.2.4.1.1 Descrizione

Definire le norme per il *versionamento<sub>G</sub>* della documentazione e delle componenti software.

##### 3.2.4.1.2 Codice di versione

Il codice di versione si deve attenere alla seguente sintassi:

**X.Y.Z**

- **X**: il numero di approvazione progressiva del documento. Parte da zero e non si resetta mai;
- **Y**: il numero di verifica progressivo del documento. Parte da zero e si resetta quando X viene incrementato;
- **Z**: il numero di modifica progressivo del documento. Parte da zero e si resetta quando Y viene incrementato o resettato.

La registrazione delle modifiche avviene tramite apposito *changelog<sub>G</sub>*, se previsto all'inizio del documento. In questa sezione vengono segnate le modifiche comprese di dettagli informativi, maggiori informazioni sono contenute nella sezione §3.1.4.5.4 denominata **Registro delle modifiche**.

##### 3.2.4.1.3 Sistemi software utilizzati

Il versionamento<sub>G</sub> viene gestito tramite apposita piattaforma *GitHub<sub>G</sub>* (desktop<sub>G</sub> e cli), la quale permette la creazione di differenti *repository<sub>G</sub>*. L'amministratore ha il compito di gestire ed organizzare le *repository<sub>G</sub>*, in modo che siano accessibili a tutti i componenti del gruppo e ad eventuali membri esterni abilitati. Sono previste le seguenti *repository<sub>G</sub>*:

- Docs
- Documentation
- PoC
- ChatBot
- Dev

#### 3.2.4.1.4 Formato dei file

Tutte le tipologie di file di testo sono attualmente consentite all'interno della *repository<sub>G</sub>* documentale, con ulteriore estensione verso file di tipo immagine e file di progetto.

- .tex: Contiene il sorgente del documento;
- .pdf: E' il .tex compilato;
- .png: Formato file immagine;
- .md: Formato markdown;
- .gitignore: Contiene le regole per "ignorare" file dal caricamento in *GitHub<sub>G</sub>*.

#### 3.2.4.1.5 Norme sui commit

Ogni commit deve avere una descrizione sensata ed esaustiva, in modo tale da rendere immediata l'identificazione da parte di tutti i membri del gruppo della motivazione che ha spinto ad apportare determinate modifiche.

#### 3.2.4.2 Struttura del *repository<sub>G</sub>*

##### 3.2.4.2.1 Descrizione

Definire la struttura delle *repository* per la documentazione e per il codice sorgente e normare il loro utilizzo.

##### 3.2.4.2.2 SkyNet-UniPd/Docs

Contiene tutta la documentazione ancora in sviluppo, con un *branch<sub>G</sub>* per ogni differente tipologia di documento. Inoltre ogni qualvolta un documento risulta approvato, viene reso disponibile anche nel *branch<sub>G</sub>* principale denominato "*main*".

##### 3.2.4.2.3 SkyNet-UniPd/Documentation

Contiene tutta la documentazione del progetto approvata e resa disponibile all'esterno.

##### 3.2.4.2.4 SkyNet-UniPd/PoC

Contiene il codice sorgente del *Proof Of Concept<sub>G</sub>*.

##### 3.2.4.2.5 SkyNet-UniPd/ChatBot

Contiene tutto il codice sorgente del progetto approvato e reso disponibile all'esterno

##### 3.2.4.2.6 SkyNet-UniPd/Dev

Contiene tutto il codice sorgente del progetto ancora nella fase di sviluppo.

#### 3.2.5 Metriche di qualità

Non sono state definite delle metriche di qualità per questo processo.

#### 3.2.6 Strumenti

- *GitHub<sub>G</sub>*: piattaforma per gestire il *versionamento<sub>G</sub>* della documentazione e del codice.

### 3.3 Gestione della qualità

#### 3.3.1 Scopo

Questa sezione definisce le norme per la gestione della qualità del progetto, al fine di soddisfare i requisiti richiesti dal proponente. Affinché il prodotto software e la documentazione siano di buona qualità, ogni membro del gruppo dovrà:

- Attenersi al *Piano di Qualifica v2.0.0<sub>D</sub>*;
- Porsi obiettivi incrementali;
- Ricerare auto miglioramento continuo e informare il resto del gruppo di software o servizi utili a tutti se trovati in autonomia.

#### 3.3.2 Descrizione

La gestione della qualità è un processo che viene descritto più approfonditamente nel documento *Piano di Qualifica v2.0.0<sub>D</sub>*. In esso vi si troveranno informazioni riguardo le metriche e le modalità usate per valutare la qualità dei prodotti e dei processi. Tali metriche vengono riportate in questo documento nella sezione *metriche di qualità* dei processi.

#### 3.3.3 Aspettative

Le aspettative del gruppo riguardo questo aspetto sono:

- Conseguimento della qualità di prodotto secondo quanto richiesto dal proponente;
- Buona qualità di organizzazione interna del gruppo;
- Prova oggettiva della qualità del prodotto finale.

#### 3.3.4 Attività

##### 3.3.4.1 Definizione obiettivi di qualità

###### 3.3.4.1.1 Descrizione

Definizione delle attività di verifica e validazione del software.

###### 3.3.4.1.2 Procedure

Il processo di gestione della qualità si articola nelle seguenti attività:

- **Pianificazione** degli obiettivi da perseguire, tenendo conto le risorse disponibili;
- **Esecuzione** di quanto è stato pianificato;
- **Controllo** dei risultati ottenuti e confronto con i risultati attesi;
- **Correzione** delle problematiche eventualmente riscontrate, adeguando strategie e metriche in base ai risultati ottenuti.

###### 3.3.4.1.3 Denominazione metriche di qualità

Ogni metrica viene identificata da un codice univoco che deve seguire il seguente pattern:

**MQ[Tipo][X]**

- **Tipo:** tipologia di metrica, può essere:
  - PD:** Prodotto;
  - PC:** Processo;
  - T:** Test.
- **X:** numero progressivo identificativo della metrica

### 3.3.5 Metriche di qualità

#### 3.3.5.1 MQPC013 - Metrics Satisfied (MS)

Metrica che indica la percentuale delle metriche soddisfatte rispetto alle metriche disponibili durante la misurazione, definita dalla formula:

$$\frac{metricheSoddisfatte}{metricheTotali} * 100$$

### 3.3.6 Strumenti

Non sono stati utilizzati strumenti in particolare per la gestione di questo processo.

## 3.4 Gestione dei cambiamenti

### 3.4.1 Scopo

Lo scopo del processo di gestione dei cambiamenti è quello di assicurare una gestione appropriata e tracciabile delle problematiche che si verificano durante l'esecuzione dei processi.

### 3.4.2 Descrizione

Il processo di gestione dei cambiamenti identifica, documenta, analizza e risolve le problematiche riscontrate. In particolare serve per:

- evitare che si ripetano i medesimi errori in futuro;
- evitare che si propaghino gli errori;
- individuare gli errori più comuni per potervi focalizzare la verifica.

### 3.4.3 Aspettative

Le aspettative del processo di gestione dei cambiamenti sono le seguenti:

- riuscire a individuare le cause di un problema;
- determinare le possibili soluzioni a un problema riscontrato;
- riuscire a risolvere il problema riscontrato;
- evitare di commettere gli stessi errori in futuro.

### 3.4.4 Attività

#### 3.4.4.1 Ciclo risolutivo

Per gestire e risolvere i problemi riscontrati, si eseguono i seguenti passi:

1. **Individuazione** del problema e delle possibili cause e conseguenze;
2. **Categorizzazione** del problema seguendo le linee guida per la denominazione riportate nel paragrafo seguente;
3. **Valutazione** delle possibili soluzioni al problema e scelta della soluzione effettiva;
4. **Risoluzione** del problema;
5. **Verifica** della risoluzione effettiva del problema;
6. **Miglioramento** dei sistemi di verifica in modo da evitare la ricomparsa del problema in futuro.

### 3.4.4.2 Denominazione cambiamenti

Per agevolare il tracciamento, ogni problema è identificato con un codice univoco che deve seguire il seguente pattern:

**P[Priorità][Tipo][X]**

- **Priorità:** livello di priorità del problema, può assumere i valori:
  - 1:** priorità alta, da trattare subito;
  - 2:** priorità media, non urgente ma comunque rilevante;
  - 3:** priorità bassa, non urgente.
- **Tipo:** tipo di problema, può assumere i valori:
  - O:** ortografico;
  - C:** contenuto.
- **X:** numero progressivo identificativo del problema.

### 3.4.5 Metriche di qualità

Non sono state definite delle metriche di qualità per questo processo.

### 3.4.6 Strumenti

Non sono stati utilizzati strumenti in particolare per la gestione di questo processo.

## 3.5 Verifica

### 3.5.1 Scopo

Lo scopo del processo di verifica è quello di Identificare eventuali errori per consentire un corretto sviluppo e presentare il prodotto finale garantendo completezza. Sia la documentazione che il software devono essere sottoposti al processo di verifica.

### 3.5.2 Descrizione

Il processo di verifica è composto da due diverse attività:

- **Analisi**, ossia un controllo del codice sorgente con successiva esecuzione. Si compone di analisi statica e analisi dinamica;
- **Test**, vari test eseguiti sul software prodotto.

### 3.5.3 Aspettative

Un corretto utilizzo del processo di verifica si svolge con:

- individuazione di una valida procedura di verifica da utilizzare;
- definizione di criteri di verifica del prodotto;
- l'esecuzione dopo ogni fase di produzione;
- definizione di un sistema di catalogazione degli errori e difetti del prodotto che dovranno essere segnalati e corretti.

### 3.5.4 Attività

#### 3.5.4.1 Analisi

##### 3.5.4.1.1 Descrizione

Normare le attività di analisi statica e dinamica.

##### 3.5.4.1.2 Analisi statica

L'analisi statica permette di rilevare errori all'interno di documentazione e codice e non necessita della loro esecuzione. Si può eseguire tramite due modalità:

- **Walkthrough:** un metodo oneroso in termini di efficienza e risorse, in quanto richiede che tutti i componenti del team revisionino a tappeto tutta la parte interessata. Questo metodo è il più semplice da imparare e sarà essenziale nelle prime fasi del progetto;
- **Inspection:** un metodo assai meno oneroso, in quanto richiede che sia un singolo componente ad analizzare la lista di controllo degli elementi più frequenti, aumentando l'efficienza e tralasciando così tutte le sezioni senza problematiche.

Tale verifica ricerca nel codice dei *pattern* che, solitamente, rendono più difficoltoso la ricerca di un *bug*. Questa ricerca non si riflette sul comportamento del codice a differenza di altre pratiche, come i test di unità, ma analizza lo stile di scrittura del codice.

##### 3.5.4.1.3 Analisi dinamica

L'analisi dinamica necessita del codice e della sua compilazione/esecuzione in quanto permette di identificare errori e controlla il corretto funzionamento del prodotto. Viene eseguita al termine di ogni parte tramite test.

#### 3.5.4.2 Test

##### 3.5.4.2.1 Descrizione

I test determinano che una componente svolga correttamente le sue attività e rispetti i vincoli previsti, per fare ciò devono essere quindi rispettati i seguenti parametri:

- **Ambiente:** hardware o software sul quale viene eseguito il test;
- **Stato iniziale:** lo stato di partenza del test;
- **Input:** dati in ingresso;
- **Output<sub>G</sub>:** dati in uscita;
- **Istruzioni aggiuntive:** ulteriori informazioni utili quali istruzioni di esecuzione dei test, per l'interpretazione dell'*output<sub>G</sub>*, ecc.

Affinché questi test si possano definire "buoni", devono essere ripetibili, provvisti di ambiente di esecuzione, avere *input* ed *output<sub>G</sub>* ben identificati, avvertire in caso di eccezioni e fornire un *log<sub>G</sub>* con i risultati. Per mantenere il tracciamento dei test effettuati, essi vengono elencati in una tabella dove viene indicato:

- **ID:** codice identificativo del test;
- **Descrizione:** breve descrizione del caso specifico per cui viene effettuato il test;
- **Stato:** stato del test, può assumere i valori:
  - DI:** Da Implementare;
  - DNI:** Da Non Implementare;
  - I:** Implementato.
- **Esito:** esito del test, può assumere i valori:
  - S:** Successo;
  - F:** Fallimento.

#### 3.5.4.2.2 Test di unità

Test per una partizione molto piccola di software (unità). Devono essere eseguiti per primi, in modo da garantire il corretto funzionamento dell'unità prima di procedere con la sua integrazione. Il codice identificativo per questi test segue il seguente pattern:

$$\text{TU}[\mathbf{X}]$$

dove  $\mathbf{X}$  è un numero progressivo che identifica il test.

#### 3.5.4.2.3 Test di integrazione

Test che controllano la corretta interazione tra le diverse unità del software. Man mano che un'unità viene testata correttamente, viene aggiunta all'insieme di unità per cui si deve controllare l'interazione, fino a testare l'intero sistema. Il codice identificativo per questi test segue il seguente pattern:

$$\text{TI}[\mathbf{X}]$$

dove  $\mathbf{X}$  è un numero progressivo che identifica il test.

#### 3.5.4.2.4 Test di sistema

Test per verificare il corretto funzionamento dell'intero sistema, di modo che i requisiti definiti nel documento di Analisi dei Requisiti<sub>D</sub> vengano rispettati. Il codice identificativo per questi test segue il seguente pattern:

$$\text{TS}[\text{Priorità}][\text{Tipo}][\mathbf{X}]$$

- **Priorità:** priorità del requisito da verificare, può assumere i valori:
  - 1: priorità alta;
  - 2: priorità media;
  - 3: priorità bassa.
- **Tipo:** tipologia di requisito da verificare, può assumere i valori:
  - F: funzionale;
  - P: prestazionale;
  - Q: qualitativo;
  - V: di vincolo.
- **X:** numero progressivo che identifica il test.

#### 3.5.4.2.5 Test di regressione

Test per verificare il corretto funzionamento del sistema anche a seguito di modifiche o dell'aggiunta di nuove funzionalità. Il codice identificativo per questi test segue il seguente pattern:

$$\text{TR}[\mathbf{X}]$$

dove  $\mathbf{X}$  è un numero progressivo che identifica il test.

### 3.5.5 Metriche di qualità

#### 3.5.5.1 MQPC10 - Tests Passed (TP)

Metrica che indica la percentuale dei test superati mediante la formula:

$$TP = \frac{\text{testSuperati}}{\text{testTotali}} * 100$$



### 3.5.5.2 MQPC11 - Code Coverage (CC)

Metrica che indica la percentuale di linee di codice coperte dai test.

Viene utilizzato lo strumento *Coverage.py*, il quale permette di automatizzare il calcolo mediante l'esecuzione dei test e produce un *report* in cui viene indicata la copertura dei test per ogni singolo file e quella generale del prodotto software.

Per calcolare il risultato si utilizza quindi la seguente formula:

$$CC = \frac{\text{lineeDiCodiceTestate}}{\text{lineeDiCodiceTotali}} * 100$$

### 3.5.5.3 MQPC12 - Static Code Analysis (SCA)

Metrica per garantire la qualità dell'attività di analisi statica del codice.

Per fare ciò viene utilizzato lo strumento *PyLint*. Esso permette, attraverso un file di configurazione, di personalizzare i parametri che fungono da vincoli per un controllo ottimale del codice.

Il risultato sarà una valutazione in decimi, calcolata in base ai requisiti soddisfatti mediante la seguente formula:

$$SCA = \max(0, 0 \text{ if fatal else } 10.0 - ((\text{float}(5 * \text{error} + \text{warning} + \text{refactor} + \text{convention}) / \text{statement}) * 10))$$

dove le variabili 'fatal', 'error', 'warning', 'refactor', 'convention' e 'info' contengono il numero di errori di ciascun tipo, mentre la variabile 'statement' contiene il numero di *statement* analizzati nel codice.

## 3.5.6 Strumenti

- **PyLint:** utilizzato per effettuare l'analisi statica del software;
- **Coverage.py:** utilizzato per effettuare l'analisi di copertura dei test per il prodotto software.

## 3.6 Validazione

### 3.6.1 Scopo

Lo scopo di questo processo è quello di assicurarsi che il prodotto finale rispetti i requisiti e soddisfi le aspettative del proponente.

### 3.6.2 Descrizione

Tale processo consiste nell'esaminare il prodotto ultimato prima del rilascio assicurandosi che rispetti i vincoli concordati con il proponente e le aspettative del committente. Il responsabile di progetto avrà l'onere di controllare il prodotto finale e stabilire se accettarlo o rifiutarlo richiedendo una nuova verifica.

### 3.6.3 Aspettative

Le aspettative dei membri riguardo a questo processo sono:

- Assicurarsi che il prodotto rispetti i requisiti riportati nel documento: *Analisi dei Requisiti v3.0.0<sub>D</sub>*;
- Assicurarsi che il prodotto rispetti i requisiti di qualità attesi secondo quanto riportato nel *Piano di Qualifica v2.0.0<sub>D</sub>*;
- Dimostrare la correttezza del software tramite le attività svolte nella fase di verifica.

### 3.6.4 Attività

#### 3.6.4.1 Testing con il committente

##### 3.6.4.1.1 Descrizione

Consiste nell'esecuzione di una serie di test di accettazione in presenza del proponente. I test valuteranno il soddisfacimento dei casi d'uso, il rispetto dei requisiti obbligatori e di altri requisiti concordati con il committente. I test vengono trattati come un qualsiasi altro tipo di test (paragrafo 3.5.5.1), senza tenere traccia del risultato.

##### 3.6.4.1.2 Test di accettazione

Test che accertano il soddisfacimento dei requisiti utente alla presenza del committente. I test di accettazione vengono svolti una volta completato il prodotto, immediatamente prima del rilascio e dopo aver testato l'intero sistema. Se il proponente li considera superati, allora il prodotto può essere approvato e conseguentemente rilasciato. Il codice identificativo per questi test segue il seguente pattern:

**TA[Priorità][Tipo][X]**

- **Priorità:** priorità del requisito da verificare, può assumere i valori:
  - 1:** priorità alta;
  - 2:** priorità media;
  - 3:** priorità bassa.
- **Tipo:** tipologia del requisito da verificare, può assumere i valori:
  - F:** funzionale;
  - P:** prestazionale;
  - Q:** qualitativo;
  - V:** di vincolo.
- **X:** numero progressivo che identifica il test.

### 3.6.5 Metriche di qualità

Non sono state definite delle metriche di qualità per questo processo.

### 3.6.6 Strumenti

Non sono stati utilizzati strumenti in particolare per la gestione di questo processo.

## 4 Processi Organizzativi

### 4.1 Gestione dei processi

#### 4.1.1 Scopo

Lo scopo di questa sezione è quello di normare le modalità di coordinamento e organizzazione tra i vari membri del gruppo e di pianificazione delle attività da eseguire. Gli obiettivi principali di questo processo sono:

- coordinare la comunicazione all'interno e all'esterno del gruppo;
- pianificare le attività tenendo conto di tempi e risorse;
- individuare e mitigare i rischi.

L'elenco completo delle attività viene definito in dettaglio nel documento *Piano di Progetto v1.0.0D*.

#### 4.1.2 Descrizione

Insieme di attività a supporto della realizzazione degli altri processi.

Le attività di gestione sono le seguenti:

- Assegnazione dei compiti e dei ruoli;
- Analisi dei rischi;
- Istanziamento dei processi;
- Pianificazione con stima dei costi in base alle risorse e ai tempi;
- Revisione e valutazione periodica delle attività.

#### 4.1.3 Aspettative

Le aspettative del processo sono le seguenti:

- Raggiungere un'organizzazione efficace e ragionevole tra i membri del gruppo;
- Assicurarsi un'equa distribuzione ed una rotazione efficiente dei ruoli di progetto;
- Adottare un buon *way of working*<sub>G</sub>;
- Riuscire a controllare le spese.

#### 4.1.4 Attività

##### 4.1.4.1 Assegnazione ruoli di progetto

###### 4.1.4.1.1 Descrizione

Definizione delle figure professionali necessarie per lo sviluppo del progetto.

Devono essere garantite:

- Suddivisione dei ruoli usando la rotazione;
- Equa assegnazione delle ore da svolgere nei vari ruoli.

Per ogni ruolo vengono evidenziati i compiti da svolgere e le responsabilità.

#### 4.1.4.1.2 Responsabile di progetto

Svolge un ruolo fondamentale durante l'intera durata del progetto, ha il compito di rappresentare il gruppo nelle comunicazioni con le figure esterne come il proponente e il committente e di coordinare i vari membri del gruppo.

In particolare si occupa di:

- Approvare e rilasciare i documenti;
- Pianificare le attività lavorative e conseguentemente le scadenze;
- Coordinare le risorse, attività e comunicazioni all'interno del gruppo.

#### 4.1.4.1.3 Amministratore di progetto

Si occupa della gestione dell'ambiente di sviluppo.

In particolare si occupa di:

- Gestire la documentazione, controllando che venga verificata ed eventualmente corretta;
- Redigere il *Piano di Progetto v1.0.0<sub>D</sub>* affiancato dal responsabile di progetto;
- Gestire il sistema di configurazione e *versionamento<sub>G</sub>* del prodotto;
- Realizzare le procedure per la gestione della qualità;
- Mantenere efficiente l'ambiente di sviluppo, fornendo strumenti adeguati ai membri del gruppo;
- Gestire le infrastrutture e i servizi di supporto.

#### 4.1.4.1.4 Analista

Si occupa di individuare, analizzare e documentare i servizi che il sistema deve fornire. E' fondamentale nella fase iniziale per la stesura del documento di *Analisi dei Requisiti v3.0.0<sub>D</sub>*

In particolare si occupa di:

- Determinare i requisiti impliciti ed espliciti richiesti dal proponente;
- Esaminare il dominio applicativo del progetto;
- Analizzare la complessità del sistema e delle sue funzionalità,
- Redigere il documento di *Analisi dei Requisiti v3.0.0<sub>D</sub>*.

#### 4.1.4.1.5 Progettista

Svolge il ruolo di responsabile delle attività di progettazione, assicurandosi di rispettare i requisiti individuati.

In particolare si occupa di:

- Definire la struttura architetturale del sistema in base alle tecnologie scelte e ai requisiti;
- Soddisfare i requisiti individuati dagli analisti;
- Cercare di limitare le dipendenze tra i componenti.

#### 4.1.4.1.6 Programmatore

Si occupa della codifica in base all'architettura fornita dai progettisti.

In particolare si occupa di:

- Implementare le specifiche fornite dai progettisti;
- Scrivere codice versionato, mantenibile e documentato;
- Rendere semplice la verifica e validazione del codice usando le modalità e gli strumenti previsti dalle *Norme di Progetto v1.0.0<sub>D</sub>*;
- Redigere il *Manuale Utente<sub>D</sub>*.

#### 4.1.4.1.7 Verificatore

Si occupa di tutte le attività di verifica del codice e della documentazione per l'intera durata del progetto. In particolare si occupa di:

- Verificare i prodotti nella fase di revisione, controllando che siano rispettate le *Norme di Progetto v1.0.0<sub>D</sub>*;
- Segnalare eventuali errori o anomalie riscontrate nella verifica in modo che vengano corrette in tempi brevi;
- Redigere la parte di retrospettiva del *Piano di Qualifica v2.0.0<sub>D</sub>*.

#### 4.1.4.2 Assegnazione delle attività

##### 4.1.4.2.1 Descrizione

Le attività vengono pianificate dal responsabile di progetto, e vengono auto assegnate dai singoli membri del gruppo preferendo le seguenti condizioni:

- compiti con scadenza più vicina;
- compiti vecchi e non ancora svolti;
- compiti con alta priorità.

Il responsabile di progetto ha la facoltà di assegnare personalmente i compiti qual'ora ci siano *task* urgenti che non sono stati assegnati.

I membri del gruppo si impegnano a svolgere le attività loro assegnate nei tempi di consegna previsti e sono responsabili della chiusura del loro compito.

#### 4.1.4.3 Gestione delle comunicazioni

##### 4.1.4.3.1 Descrizione

Definizione degli strumenti utili alla comunicazione interna ed esterna del gruppo, con lo scopo di gestire il lavoro, risolvere piccole problematiche e organizzare efficientemente le riunioni.

##### 4.1.4.3.2 Comunicazioni interne

Le comunicazioni interne avvengono tramite le applicazioni *Telegram<sub>G</sub>*, usata principalmente per le richieste urgenti e l'organizzazione delle riunioni, e *Discord<sub>G</sub>*, usata per aggiornare gli altri componenti del gruppo sullo sviluppo del codice e della documentazione.

Le discussioni su *Discord<sub>G</sub>* sono organizzate in canali nel seguente modo:

- un canale per ogni documento dove discutere nello specifico del contenuto dello stesso;
- **#ruoli-settimanali** per tenere traccia in maniera rapida dei ruoli assegnati settimanalmente;
- **#codifica** per discutere della parte di codifica del prodotto;
- **#varie** per comunicazioni di carattere generale.

##### 4.1.4.3.3 Comunicazioni esterne

Le comunicazioni esterne avvengono tramite l'applicazione *Telegram<sub>G</sub>* per quanto riguarda il proponente e via mail per quanto riguarda il committente. L'indirizzo email utilizzato per le comunicazioni è il seguente: [skynet.swe@gmail.com](mailto:skynet.swe@gmail.com).

#### 4.1.4.4 Gestione degli incontri

##### 4.1.4.4.1 Descrizione

Definizione degli strumenti e delle regole utili all'organizzazione e gestione degli incontri interni ed esterni. Gli incontri molto spesso vengono svolti da remoto, ma possono essere svolti anche di persona.

Il membro che nella data dell'incontro svolge il ruolo di amministratore avrà il compito di redigere il verbale.

#### 4.1.4.4.2 Incontri interni

Gli incontri interni riguardano solo i membri del gruppo *SkyNet* e vengono svolti tramite lo strumento *Google Meet<sub>G</sub>* sempre nella stessa stanza virtuale.

Il responsabile di progetto deve assicurarsi di:

- Concordare una data in base alle disponibilità di tutti i membri;
- Comunicare il programma e gli argomenti che dovranno essere trattati nell'incontro.

#### 4.1.4.4.3 Incontri esterni

Gli incontri interni riguardano i membri del gruppo *SkyNet* e almeno un membro esterno, che può essere uno dei rappresentanti dell'azienda proponente oppure il committente. Tutte le riunioni vengono svolte tramite lo strumento *Zoom<sub>G</sub>*.

Il responsabile di progetto deve assicurarsi di:

- Concordare una data in base alle disponibilità dei membri del gruppo e dei membri esterni;
- Comunicare gli argomenti che dovranno essere trattati nell'incontro.

#### 4.1.4.5 Gestione degli strumenti di coordinamento

##### 4.1.4.5.1 Descrizione

Definizione degli strumenti utili al coordinamento del gruppo rispetto ai *task* da svolgere.

Il responsabile di progetto deve:

- Assegnare i task in base ai ruoli ed alle disponibilità dei membri del gruppo;
- Controllare lo stato di avanzamento di tutti i task.

I componenti del gruppo devono:

- Essere a conoscenza dei propri compiti da svolgere e delle rispettive scadenze;
- Essere a conoscenza dello stato in cui si trovano tutti i task, in particolare quelli correlati ai propri compiti.

##### 4.1.4.5.2 Software per la gestione dei task

Per gestire lo sviluppo del progetto si è deciso di usare il software *Asana<sub>G</sub>*, che permette di creare una *board di progetto<sub>G</sub>* in cui si possono definire dei semplici *task*, assegnarli ad un membro del gruppo e segnalarli come completati. Abbiamo deciso di dividere la *board di progetto<sub>G</sub>* in 4 sezioni, la prima contiene i *task* di cui non è stato ancora iniziato lo sviluppo, la seconda contiene i *task* in sviluppo, la terza contiene i *task* da verificare, mentre l'ultima sezione contiene tutti i *task* completati e verificati da un verificatore.

Ogni *task/issue* viene identificato tramite il codice descritto nella sezione §3.1.4.4.7 .

#### 4.1.4.6 Gestione degli strumenti di *versionamento<sub>G</sub>*

##### 4.1.4.6.1 Descrizione

##### 4.1.4.6.2 Repository<sub>G</sub>

Per il *versionamento<sub>G</sub>* dei file è stata scelta la piattaforma *GitHub<sub>G</sub>*. L'amministratore ha creato i *repository<sub>G</sub>* e ha aggiunto come collaboratori tutti i membri del gruppo *SkyNet*. La struttura dei *repository<sub>G</sub>* utilizzati è stata già elencata nella sezione §3.2.4.2.

##### 4.1.4.6.3 Tipi di file e *.gitignore*

Nelle cartelle che contengono i documenti, sono presenti solo i file *.tex*, *.png* e *.pdf*.

Nel file *.gitignore* sono stati aggiunti tutti i file e cartelle che devono essere ignorati da *GitHub<sub>G</sub>*, ad esempio quelli generati automaticamente dal IDE, al fine di evitare la presenza di elementi indesiderati o non rilevanti che non devono quindi essere versionati.

#### 4.1.4.7 Gestione dei rischi

##### 4.1.4.7.1 Descrizione

Definizione delle norme per la classificazione, segnalazione e documentazione dei rischi rilevati durante lo sviluppo del prodotto *software*.

##### 4.1.4.7.2 Codifica dei rischi

I rischi vengono codificati nel seguente formato:

**RIS[Tipologia][ID]**

- **Tipologia:** carattere che indica il tipo del rischio:
  - **O:** organizzativo;
  - **R:** requisito;
  - **S:** strumentale;
  - **T:** tecnologico.
- **ID:** identificativo numerico del rischio, il suo valore è incrementale e collegato alla sua tipologia.

#### 4.1.5 Metriche di qualità

##### 4.1.5.1 MQPC14 - Non-Calculated Risks (NCR)

Metrica incrementale che enumera i rischi non previsti per ogni periodo mediante la formula:

$$NCR = NCR + 1$$

Il valore della metrica viene azzerato ad ogni periodo.

#### 4.1.6 Strumenti

Non sono stati utilizzati strumenti in particolare per la gestione di questo processo.

### 4.2 Formazione

#### 4.2.1 Scopo

Definire le norme che riguardano il processo di formazione dei membri del gruppo. Documentarsi quindi sui linguaggi di programmazione, sulle loro librerie e sugli strumenti necessari per lo sviluppo del prodotto *software* assegnato dal proponente.

#### 4.2.2 Descrizione

Processo organizzativo che riguarda la formazione dei membri del gruppo *SkyNet*.

#### 4.2.3 Aspettative

Le aspettative per la formazione dei membri del gruppo sono:

- Raggiungere una buona conoscenza del linguaggio  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}_G$  per la stesura dei documenti;
- Raggiungere un'ottima conoscenza dell'ambiente di sviluppo dell'applicativo;
- Raggiungere una buona conoscenza dei linguaggi di programmazione, librerie e strumenti necessari per lo sviluppo dell'applicativo;
- Raggiungere una buona conoscenza dei software per la gestione dei task e degli strumenti per il *versionamento*<sub>G</sub> del codice;
- Raggiungere una buona conoscenza dei software per la gestione degli incontri remoti.

#### 4.2.4 Attività

##### 4.2.4.1 Modalità di formazione

###### 4.2.4.1.1 Descrizione

La formazione di ogni membro del gruppo avviene autonomamente, tramite la documentazione e il supporto fornito in rete, ma anche grazie al materiale fornito dai docenti e dal proponente.

###### 4.2.4.1.2 Fonti

Di seguito vengono riportate le principali fonti delle documentazioni e materiali di supporto trovati in rete:

- *Versionamento<sub>G</sub>* del codice:
  - *GitHub<sub>G</sub>*
- Gestione dei task:
  - *Asana<sub>G</sub>*
- Redazione dei documenti:
  - *LaTeX<sub>G</sub>*
- Linguaggi di programmazione per lo sviluppo software:
  - *Python<sub>G</sub>*
  - *HTML 5<sub>G</sub>*
  - *CSS 3<sub>G</sub>*
  - *Javascript<sub>G</sub>*
- Librerie per lo sviluppo software:
  - *Chatterbot<sub>G</sub>*
- *Framework<sub>G</sub>* per lo sviluppo software:
  - *Django<sub>G</sub>*
  - *React<sub>G</sub>*
- *Hosting* dell'applicativo:
  - *Render<sub>G</sub>*

##### 4.2.5 Metriche di qualità

Non sono state definite delle metriche di qualità per questo processo.

##### 4.2.6 Strumenti

Non sono stati utilizzati strumenti in particolare per la gestione di questo processo.