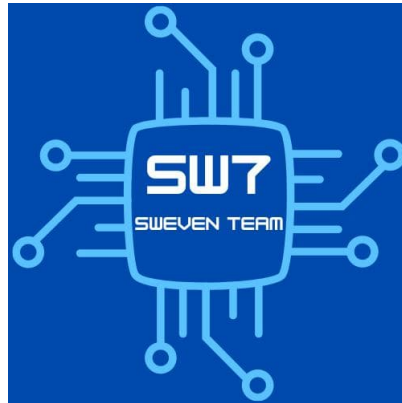


NORME DI PROGETTO



SWEVEN TEAM

swe7.team@gmail.com

INFORMAZIONI SUL DOCUMENTO

| | |
|---------------------|--|
| Versione | 0.1.3 |
| Uso | Interno |
| Destinatari | Gruppo Sweven Team Prof. Tullio Vardanega Prof. Riccardo Cardin Azienda Imola Informatica |
| Stato | in lavorazione |
| Redattori | Irene Benetazzo Samuele Rizzato Mattia Episcopo Tommaso Berlaffa Pietro Macrì Pan Qi Fan Andrea |
| Verificatori | Tommaso Berlaffa Pietro Macrì |
| Approvatori | |

Sintesi

Documento che norma le regole e le convenzioni del way of working del gruppo Sweven Team

Diario delle modifiche

| Versione | Data | Descrizione | Ruolo | Autore | Verificatore |
|----------|------------|--|----------------|--------------------------------|------------------|
| | 2022-05-30 | Aggiunto \$4.2.7, modificati \$4.1 e \$4.2 | Amministratore | Samuele Rizzato | |
| 0.1.4 | 2022-05-24 | Aggiornamento \$2.2.6 | Amministratore | Tommaso Berlaffa | |
| 0.1.3 | 2022-05-13 | Aggiunto \$4.1.6, aggiornato \$2.2.4 | Amministratore | Samuele Rizzato | Irene Benetazzo |
| 0.1.2 | 2022-05-07 | Aggiornamento \$2 | Amministratore | Tommaso Berlaffa | Matteo Pillon |
| 0.1.1 | 2022-05-02 | Aggiornamento \$1 | Amministratore | Irene Benetazzo | Matteo Pillon |
| 0.1.0 | 2022-04-28 | Verifica dell'intero Documento | Verificatore | Tommaso Berlaffa, Pietro Macrì | |
| 0.0.9 | 2022-04-28 | Scrittura \$3.5, \$3.6 | Amministratore | Pan Qi Fan Andrea | Tommaso Berlaffa |
| 0.0.8 | 2022-04-25 | Scrittura \$3.2, \$3.3, \$3.4 | Amministratore | Pietro Macrì | Tommaso Berlaffa |
| 0.0.7 | 2022-04-24 | Scrittura \$2.1, \$2.2 | Amministratore | Tommaso Berlaffa | Pietro Macrì |
| 0.0.6 | 2022-04-22 | Modifica \$4.3, \$4.4 | Amministratore | Mattia Episcopo | Pietro Macrì |
| 0.0.5 | 2022-04-21 | Continuazione \$4.1, scrittura \$4.2 e \$4.3 | Amministratore | Samuele Rizzato | Pietro Macrì |

| | | | | | |
|-------|------------|---------------------|----------------|-----------------|------------------|
| 0.0.4 | 2022-04-03 | Scrittura \$4.1 | Amministratore | Samuele Rizzato | Pietro Macrì |
| 0.0.3 | 2022-04-02 | Continuazione \$3.1 | Amministratore | Irene Benetazzo | Tommaso Berlaffa |
| 0.0.2 | 2022-03-27 | Scrittura \$3.1 | Amministratore | Irene Benetazzo | Tommaso Berlaffa |
| 0.0.1 | 2022-03-26 | Scrittura \$1.1 | Amministratore | Irene Benetazzo | Pietro Macrì |
| | 2022-03-26 | Creazione documento | Amministratore | Irene Benetazzo | |

Indice

| | | |
|-----------|--|----------|
| 1 | Introduzione | 8 |
| 1.1 | Scopo del documento | 8 |
| 1.2 | Scopo del capitolato | 8 |
| 1.3 | Glossario | 8 |
| 1.4 | Riferimenti | 8 |
| 1.4.1 | Normativi | 8 |
| 1.4.2 | Informativi | 8 |
| 2 | Processi Primari | 9 |
| 2.1 | Processo di Fornitura | 9 |
| 2.1.1 | Descrizione | 9 |
| 2.1.2 | Aspettative | 9 |
| 2.1.3 | Documenti | 9 |
| 2.1.3.1 | Studio di Fattibilità | 10 |
| 2.1.3.2 | Piano di Progetto | 10 |
| 2.1.3.3 | Piano di Qualifica | 10 |
| 2.1.4 | Strumenti | 11 |
| 2.1.4.1 | Gantt Project Manager | 11 |
| 2.1.4.2 | Google Sheet _G | 11 |
| 2.2 | Processo di Sviluppo | 11 |
| 2.2.1 | Descrizione | 11 |
| 2.2.2 | Scopo | 11 |
| 2.2.3 | Aspettative | 11 |
| 2.2.4 | Analisi dei Requisiti | 11 |
| 2.2.4.1 | Descrizione | 11 |
| 2.2.4.2 | Scopo | 12 |
| 2.2.4.3 | Classificazione dei Requisiti | 12 |
| 2.2.4.4 | Classificazione dei Casi d'Uso | 12 |
| 2.2.4.5 | Composizione dei Casi d'Uso | 13 |
| 2.2.4.6 | Strumenti | 13 |
| 2.2.4.7 | Metriche | 13 |
| 2.2.5 | Progettazione | 14 |
| 2.2.5.1 | Descrizione | 14 |
| 2.2.5.2 | Scopo | 14 |
| 2.2.5.3 | Technology Baseline | 15 |
| 2.2.5.3.1 | Diagrammi UML | 15 |
| 2.2.5.3.2 | Design Pattern | 15 |
| 2.2.5.4 | Product Baseline | 15 |
| 2.2.5.5 | Metriche | 16 |
| 2.2.6 | Codifica | 16 |
| 2.2.6.1 | Descrizione e Scopo | 16 |
| 2.2.6.2 | Convenzioni per i nomi | 16 |
| 2.2.6.3 | Convenzioni sulle pratiche | 17 |
| 2.2.6.4 | Strumenti | 17 |

| | | |
|----------|---|-----------|
| 2.2.6.5 | Metriche | 17 |
| 3 | Processi di Supporto | 19 |
| 3.1 | Documentazione | 19 |
| 3.1.1 | Scopo | 19 |
| 3.1.2 | Ciclo di vita dei documenti | 19 |
| 3.1.3 | Versionamento dei documenti | 19 |
| 3.1.4 | Suddivisione documenti interni ed esterni | 19 |
| 3.1.5 | Documenti e template | 20 |
| 3.1.5.1 | Template comune a tutti i documenti | 20 |
| 3.1.5.2 | Verbali | 21 |
| 3.1.5.3 | Glossario | 22 |
| 3.1.5.4 | Norme di Progetto | 22 |
| 3.1.5.5 | Studio di Fattibilità | 22 |
| 3.1.5.6 | Candidatura | 22 |
| 3.1.5.7 | Analisi dei Requisiti | 23 |
| 3.1.5.8 | Piano di Progetto | 23 |
| 3.1.5.9 | Piano di Qualifica | 23 |
| 3.1.5.10 | Specifica Architetture | 23 |
| 3.1.5.11 | Manuale Utente | 23 |
| 3.1.6 | Convenzioni tipografiche | 23 |
| 3.1.6.1 | Nome del file | 23 |
| 3.1.6.2 | Stile del testo | 23 |
| 3.1.6.3 | Scrittura della data e dell'ora | 24 |
| 3.1.6.4 | Sigle | 24 |
| 3.1.7 | Metriche | 24 |
| 3.2 | Gestione dello spazio di lavoro | 25 |
| 3.2.1 | Stato dei documenti | 25 |
| 3.2.2 | Versionamento dei software | 25 |
| 3.2.3 | Organizzazione della repository | 26 |
| 3.2.4 | Task | 26 |
| 3.3 | Gestione degli errori | 26 |
| 3.3.1 | Individuazione | 27 |
| 3.3.2 | Identificazione | 27 |
| 3.3.3 | Modifica del sistema di verifica | 27 |
| 3.4 | Controllo della qualità | 28 |
| 3.4.1 | Definizione di qualità | 28 |
| 3.4.2 | Misure di qualità | 28 |
| 3.4.3 | Classificazione dei processi | 28 |
| 3.4.4 | Classificazione dei prodotti | 29 |
| 3.5 | Verifica | 29 |
| 3.5.1 | Analisi | 29 |
| 3.5.2 | Metodi di lettura | 29 |
| 3.5.3 | Test | 29 |
| 3.5.3.1 | Test di unità | 30 |
| 3.5.3.2 | Test di integrazione | 30 |

| | | |
|----------|---|-----------|
| 3.5.3.3 | Test di sistema | 30 |
| 3.5.3.4 | Test di regressione | 31 |
| 3.6 | Validazione | 31 |
| 3.6.1 | Attività | 31 |
| 4 | Processi Organizzativi | 32 |
| 4.1 | Gestione Organizzativa | 32 |
| 4.1.1 | Scopo | 32 |
| 4.1.2 | Aspettative | 32 |
| 4.1.3 | Descrizione | 32 |
| 4.1.4 | Ruoli di progetto | 32 |
| 4.1.4.1 | Responsabile | 33 |
| 4.1.4.2 | Amministratore | 33 |
| 4.1.4.3 | Analista | 33 |
| 4.1.4.4 | Progettista | 33 |
| 4.1.4.5 | Programmatore | 34 |
| 4.1.4.6 | Verificatore | 34 |
| 4.1.5 | Gestione delle riunioni | 34 |
| 4.1.5.1 | Riunioni interne | 34 |
| 4.1.5.2 | Riunioni esterne | 34 |
| 4.1.5.3 | Verbali | 34 |
| 4.1.6 | Metriche | 34 |
| 4.2 | Gestione dell'infrastruttura | 35 |
| 4.2.1 | Scopo | 35 |
| 4.2.2 | Aspettative | 35 |
| 4.2.3 | Descrizione | 35 |
| 4.2.4 | Gestione delle comunicazioni | 36 |
| 4.2.4.1 | Comunicazioni interne | 36 |
| 4.2.4.2 | Comunicazioni esterne | 36 |
| 4.2.5 | Strumenti per la comunicazione | 36 |
| 4.2.5.1 | <i>Telegram_G</i> | 36 |
| 4.2.5.2 | <i>Zoom Meetings_G</i> | 36 |
| 4.2.5.3 | Gmail | 36 |
| 4.2.6 | Gestione degli strumenti di coordinamento | 37 |
| 4.2.6.1 | Ticketing | 37 |
| 4.2.7 | Gestione dei rischi | 37 |
| 4.2.7.1 | Struttura dei rischi | 37 |
| 4.2.8 | Strumenti per la gestione del lavoro | 38 |
| 4.2.8.1 | Git | 38 |
| 4.2.8.2 | GitHub | 38 |
| 4.3 | Formazione | 38 |
| 4.3.1 | Scopo | 38 |
| 4.3.2 | Aspettative | 38 |
| 4.3.3 | Attività | 39 |
| 4.4 | Miglioramento | 39 |
| 4.4.1 | Scopo | 39 |

4.4.2 Aspettative 39

4.4.3 Attività 39

1 Introduzione

1.1 Scopo del documento

Lo scopo di questo documento è di redigere tutte le norme, regole, convenzioni, decisioni prese dai componenti del gruppo Sweven Team, al fine di migliorare la qualità e l'organizzazione per il lavoro del gruppo e individuale.

La stesura del documento è incrementale durante il progetto così da riportare le decisioni prese durante le riunioni. Inoltre le norme possono subire modifiche e aggiornamenti, se ciò avviene sarà premura del responsabile che ha dato l'approvazione alla modifica avvisare tutti i componenti di prenderne visione al più presto.

Tutti i membri del gruppo sono tenuti a rispettare le norme di progetto che costituiscono il way of working del gruppo Sweven Team.

1.2 Scopo del capitolato

Lo scopo di tale progetto è quello di sviluppare un Chatbot che interfacciandosi con software aziendali spesso complessi e dispersivi, semplifichi i compiti che i dipendenti devono svolgere. In particolare vengono individuate le seguenti operazioni:

- Tracciamento della presenza in sede (**EMT_G**)
- Rendiconto attività svolte quotidianamente (**EMT_G**)
- Apertura del cancello aziendale (**MQTT_G**)
- Creazione di una riunione in un servizio esterno
- Servizio di ricerca documentale (**CMIS_G**)
- Creazione e tracciamento di bug (**Redmine_G**)

1.3 Glossario

Per assicurare la massima fruibilità e leggibilità del documento, il team SWEven ha deciso di creare un documento denominato *Glossario* il cui scopo sarà quello di contenere le definizioni dei termini ambigui o specifici del progetto. Sarà possibile riconoscere i termini presenti al suo interno in quanto terminanti con la lettera *G* posta come pedice della parola stessa.

1.4 Riferimenti

1.4.1 Normativi

- standard ISO/IEC 12207

1.4.2 Informativi

- [Capitolato di appalto C1 - BOT4ME](#) (non so se sia corretto)

2 Processi Primari

2.1 Processo di Fornitura

2.1.1 Descrizione

Il processo di fornitura descrive le principali attività ed i compiti necessari alla realizzazione del progetto, al fine di comprendere al meglio le richieste del proponente.

Le varie attività da svolgere vengono in seguito ripartite nel documento *Piano di Progetto*.

Il processo di fornitura è formato dalle seguenti fasi:

- avvio;
- contrattazione;
- pianificazione;
- esecuzione e controllo;
- revisione e valutazione;
- consegna e completamento.

2.1.2 Aspettative

Per tutta la durata del progetto, il gruppo *SWEven Team* manterrà un dialogo costante con l'azienda *Imola Informatica*. Per assolvere al meglio a questa aspettativa, viene creata una chat sull'applicazione *Telegram_G* con la quale comunicare direttamente con i referenti dell'azienda proponente. Grazie ad un dialogo continuo, sarà possibile:

- comprendere e soddisfare al meglio le richieste del proponente ed i requisiti che il prodotto dovrà soddisfare;
- definire al meglio le tempistiche di lavoro ed i costi associati ad esse;
- effettuare una verifica costante e continua tramite il feedback fornitoci direttamente dai referenti aziendali;
- chiarire eventuali dubbi con il proponente.

2.1.3 Documenti

Per poter redigere i documenti nel modo più completo possibile, vengono utilizzati a pieno le documentazioni fornite per la candidatura e vengono organizzate riunioni informative con l'azienda *Imola Informatica*.

2.1.3.1 Studio di Fattibilità

Documento contenente un'attenta analisi sui capitolati proposti, al fine di dare più informazione possibili al gruppo per poter compiere una scelta adeguata nella candidatura per l'appalto.

L'analisi di ogni capitolato è composta da:

- **Informazioni Generali**
- **Descrizione del capitolato:** descrizione riassuntiva sul capitolato;
- **Finalità del progetto:** descrizione degli obiettivi fissati per la realizzazione del capitolato;
- **Tecnologie Interessate:** elenco di tecnologie, spesso consigliate dal proponente stesso, di interesse per la realizzazione del progetto;
- **Aspetti positivi e Criticità:** elementi positivi e criticità riscontrate da una prima analisi sul capitolato;
- **Conclusione:** nella quale si motiva la scelta sulla candidatura, positiva o negativa, per lo specifico capitolato.

2.1.3.2 Piano di Progetto

Documento redatto da amministratori e responsabile di progetto.

Questo sarà composto da:

- **Analisi dei rischi:** contiene un'analisi sui rischi riscontrabili durante il progetto. Ad ogni possibile rischio viene assegnato un livello di gravità, delle precauzioni per evitare il rischio e delle contromisure per diminuire l'entità del problema in caso questo avvenga;
- **Modello di sviluppo:** contiene descrizioni e motivazioni sulla scelta del modello per l'organizzazione del progetto;
- **Pianificazione:** descrive una prima versione di suddivisione del lavoro del gruppo nel corso del tempo, insieme ad una descrizione degli strumenti che verranno utilizzati;
- **Preventivo e Consuntivo:** vengono riportati rispettivamente preventivi e consuntivi.

2.1.3.3 Piano di Qualifica

Documento è necessario affinché il gruppo possa produrre materiale di qualità.

Questo documento viene redatto dagli amministratori e contiene:

- **Qualità di processo:** definisce i valori delle metriche necessari al fine di ottenere processi di qualità;
- **Qualità di prodotto:** definisce i valori delle metriche necessari al fine di ottenere un prodotto di qualità;
- **Test**
- **Resoconto attività di verifica**

2.1.4 Strumenti

2.1.4.1 Gantt Project Manager

Strumento utilizzato per la realizzazione di diagrammi di Gantt. Questi sono utili per tenere traccia delle attività coinvolte nella realizzazione del prodotto e delle relazioni tra loro, offrendo inoltre una scansione temporale delle attività.

2.1.4.2 Google Sheet_G

Strumento che consiste in fogli di calcolo. Viene utilizzato per il calcolo dei preventivi e la suddivisione delle ore per il *Piano di Progetto*.

2.2 Processo di Sviluppo

2.2.1 Descrizione

Le attività principali che definisce questo processo sono:

- analisi dei requisiti;
- progettazione architettuale;
- codifica software.

2.2.2 Scopo

Lo scopo del processo di Sviluppo è quello di descrivere e normare i compiti e le attività di analisi, progettazione, codifica del prodotto da realizzare.

2.2.3 Aspettative

Le aspettative del gruppo possono riassumersi in:

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

2.2.4 Analisi dei Requisiti

2.2.4.1 Descrizione

L'analisi dei requisiti è una attività che viene eseguita dagli analisti e porta alla creazione dell'omonimo documento *Analisi dei requisiti*. Questa sarà composta da:

- descrizione generale del prodotto;
- elenco dei Casi d'Uso;
- elenco dei requisiti, divisi per categoria.

2.2.4.2 Scopo

Lo scopo dell'analisi dei requisiti è quello di:

- fornire requisiti semplici e precisi;
- fissare con il proponente le funzionalità che il prodotto dovrà possedere;
- assistere l'attività dei verificatori, rendendo l'attività di tracciamento dei requisiti più semplice;
- fornire informazioni utili per la stima del calcolo della mole del lavoro.

2.2.4.3 Classificazione dei Requisiti

La classificazione dei requisiti verrà identificata tramite il seguente codice :

R [PRIORITY] - [TYPE] - [NUMBER]

| Nome | Descrizione |
|----------|--|
| R | Acronimo di Requisito |
| PRIORITY | Indica il tipo di priorità : O : Requisito obbligatorio D : Requisito desiderabile F : Requisito Facoltativo |
| TYPE | Indica il tipo di requisito : F : Requisito Funzionale; definizione di una caratteristica necessaria nel software V : Requisito di Vincolo; rappresenta un vincolo avanzato Q : Requisito di Qualità; inerente le regole di qualità |
| NUMBER | Codice Numerico Identificativo |

2.2.4.4 Classificazione dei Casi d'Uso

La classificazione dei casi d'uso verrà identificata tramite il seguente codice :

UC [CASECODE] . [CASESUBCODE]

| Nome | Descrizione |
|-------------|--|
| UC | Acronimo di Caso d'Uso |
| CASECODE | Identifica un'istanza generica del Caso d'Uso |
| CASESUBCODE | Identifica un'istanza specifica del Caso d'Uso |

2.2.4.5 Composizione dei Casi d'Uso

Un Caso d'Uso è definito dalle seguenti componenti :

- Identificativo
- Nome
- Descrizione Grafica, creata con UML 2.0
- Attori :
 - Primari, interagisce direttamente col sistema per ottenere un obiettivo
 - Secondari, aiuta l'Attore Primario nel raggiungimento dell'obiettivo
- Precondizione, ovvero lo stato del sistema affinché la funzione sia disponibile;
- Post-condizione, ovvero lo stato del sistema dopo il Caso d'Uso;
- Scenario Principale, ovvero una rappresentazione del flusso degli eventi;
- Estensione, opzionale; usato per casi alternativi al verificarsi di una specifica condizione.

2.2.4.6 Strumenti

Tutti i diagrammi UML vengono realizzati utilizzando *StarUML_G* nella versione 2.0.

2.2.4.7 Metriche

M1PRR:

- Nome: Percentuale di requisiti obbligatori soddisfatti;
- Descrizione: percentuale che rappresenta i requisiti obbligatori soddisfatti rispetto al totale dei requisiti obbligatori;
- Scopo: assicurarsi che il prodotto soddisfi tutte le richieste obbligatorie del proponente;
- Formula:

$$\frac{\text{requisiti obbligatori soddisfatti}}{\text{requisiti obbligatori totali}} \cdot 100$$

M2PDR:

- Nome: Percentuale di requisiti desiderabili soddisfatti
- Descrizione: percentuale che rappresenta i requisiti desiderabili soddisfatti rispetto al totale dei requisiti desiderabili;
- Scopo: mostrare ulteriori elementi che portano valore aggiunto al progetto;
- Formula:

$$\frac{\text{requisiti desiderabili soddisfatti}}{\text{requisiti desiderabili totali}} \cdot 100$$

M3POR:

- Nome: Percentuale di requisiti opzionali soddisfatti
- Descrizione: percentuale che rappresenta i requisiti opzionali soddisfatti rispetto al totale dei requisiti opzionali;
- Scopo: assicurarsi che il prodotto soddisfi le richieste del proponente;
- Formula:

$$\frac{\text{requisiti opzionali soddisfatti}}{\text{requisiti opzionali totali}} \cdot 100$$

M4VR:

- Nome: Variazione dei Requisiti
- Descrizione: numero che indica quanti requisiti sono cambiati nel tempo;
- Scopo: controllare quanti requisiti sono cambiati al fine di supportare le metriche precedentemente descritte;
- Formula: $RA + RE + RM$
 - RA: numero di requisiti aggiunti dall'ultimo incremento;
 - RE: numero di requisiti eliminati dall'ultimo incremento;
 - RM: numero di requisiti modificati dall'ultimo incremento;

2.2.5 Progettazione**2.2.5.1 Descrizione**

La fase di Progettazione è divisa in due parti:

- **Technology Baseline_G**: descrive le specifiche della progettazione del prodotto ad alto livello; contiene inoltre diagrammi UML;
- **Product Baseline_G**: descrive in maniera dettagliata l'attività di progettazione integrando i contenuti presenti in Technology Baseline; definisce inoltre i test necessari alla verifica.

2.2.5.2 Scopo

Definisce le caratteristiche necessarie per la soddisfazione dei requisiti presentati nell'Analisi dei Requisiti. Questo avviene affinché si possa ottenere la miglior soluzione in grado di soddisfare a pieno gli *stakeholder_G*.

2.2.5.3 Technology Baseline

Viene redatto dai progettisti e contiene:

- **diagrammi UML;**
- **tecnologie utilizzate:** descrizione sulle tecnologie utilizzate e scelte che hanno portato alla specifica tecnologia;
- **tracciamento delle componenti:** ogni requisito soddisfatto da un componente dovrà fare riferimento a quest'ultimo;
- **design pattern;**
- ***Proof of Concept_G*:** prototipo creato per dimostrare l'effettiva fattibilità e funzione del prodotto.

2.2.5.3.1 Diagrammi UML

- **Diagramma delle attività:** descrive le operazioni di una attività;
- **Diagramma delle classi:** elenca attributi, metodi di classi, tipi;
- **Diagramma del package:** mostra raggruppamenti tra classi;
- **Diagramma di sequenza:** descrive sequenze di azioni.

2.2.5.3.2 Design Pattern

Ogni design pattern utilizzato per realizzare l'architettura deve essere descritto e accompagnato da un diagramma che ne esponga significato e struttura.

2.2.5.4 Product Baseline

Viene redatta dal progettista e contiene:

- **Test:**
 - di Unità: verificano il funzionamento di una unità;
 - di Integrazione: verificano l'interazione tra unità;
 - di Sistema: verificano il funzionamento del sistema;
- **Definizione delle classi;**
- **tracciamento delle classi:** ogni requisito soddisfatto da un classe dovrà fare riferimento a quest'ultima.

2.2.5.5 Metriche

M5ATC:

- Nome: Accoppiamento tra Classe;
- Descrizione: indica il numero di classi da cui una classe dipende;
- Scopo: ridurre il numero di dipendenze per rendere il prodotto più semplice e più facilmente modificabile nel tempo;

M6PDG:

- Nome: Profondità delle Gerarchie;
- Descrizione: numero che indica, data una gerarchia, il massimo numero di classi parenti;
- Scopo: limitare la possibilità di avere gerarchie troppo ampie, poiché renderebbe il prodotto troppo complesse e più difficilmente modificabile;

M7FDU:

- Nome: Facilità di Utilizzo;
- Descrizione: numero di click che un utente deve compiere per ottenere il voluto risultato;
- Scopo: limitare il numero di azioni non necessarie al fine di ottenere un'esperienza utente il più semplice e piacevole possibile;

2.2.6 Codifica

2.2.6.1 Descrizione e Scopo

La Codifica fornisce una base comune sulla realizzazione del prodotto software, con lo scopo di poter creare codice uniforme e agevolare manutenzione ed eventuali modifiche future. I programmatori si dovranno attenere a queste norme durante le fasi di progettazione ed implementazione.

Per agevolare la lettura e l'integrazione del codice, verrà utilizzato il sistema di *Continuous Integration_G* fornito da Github, ovvero *Github Action_G*: grazie a quest'ultimo, verranno eseguiti dei controlli nel codice che viene inviato alla repository e, in caso il codice in questione non soddisfi i requisiti di seguito riportati, verrà chiesto di eseguire un'operazione di merge con una versione corretta del codice.

Verranno quindi implementate le convenzioni introdotte da PEP 8, che forniscono uno standard per lo stile da seguire nella scrittura del codice Python.

2.2.6.2 Convenzioni per i nomi

- i nomi dovranno essere univoci;
- dovrà essere utilizzato il PascalCase o lo snake_case per le variabili;
- dovrà essere utilizzato lo Snake Case per metodi e classi.

2.2.6.3 Convenzioni sulle pratiche

| Pratica | Descrizione |
|---------------------------------|--|
| Variabili Utili | Non definire variabili se queste non sono poi utilizzate. |
| Funzioni Ricorsive | Evitare quando possibile funzioni ricorsive. |
| Delimitatori Stringhe | Ogni stringa deve essere delimitata dal singolo apice. |
| Operatori | Gli operatori, quando utilizzati, dovranno essere preceduti e seguiti da spazi. |
| Indentazione | Il codice, per una maggiore leggibilità, dovrà essere indentato con 4 spazi. |
| Commenti | Il codice dovrà essere commentato. I commenti devono essere separati dall'identificatore di commento da uno spazio. Se il commento è inline, dovrà essere presente uno spazio prima dell'identificatore di commento. |
| Complessità Ciclomantica | Viene evitato il più possibile l'annidamento dei cicli, poiché questo renderebbe le funzioni troppo onerose. |

2.2.6.4 Strumenti

- **PyCharm_G**: utilizzato per la scrittura di codice in linguaggio *PYTHON_G*;
- **Visual Studio Code_G**: utilizzato per la scrittura di codice in linguaggio *PYTHON_G* e di codice *HTML_G/CSS_G*;
- **Flask_G**: framework python per la visualizzazione e creazione di semplici applicazioni web.

2.2.6.5 Metriche

M8CC:

- Nome: Code Coverage;
- Descrizione: determina il numero di linee di codice che vengono validate correttamente;
- Scopo: aiuta ad analizzare quanto codice è verificato correttamente.

M9NAC:

- Nome: Numero di Attributi per Classe;
- Descrizione: indica il numero di attributi dichiarati all'interno di una classe;

- Scopo: limitare il numero di attributi utilizzati, per mantenere il codice il più semplice possibile al fine di renderlo il più facilmente modificabile e manutentibile successivamente.

M10PF:

- Nome: Parametri per Funzione;
- Descrizione: indica il numero di parametri che vengono forniti ad una funzione;
- Scopo: limitare il numero di parametri inviati ad una singola funzione, per mantere il codice il più semplice possibile.

M11LCF:

- Nome: Linee di Codice per Funzione;
- Descrizione: indica il numero di righe di codice all'interno di una funzione;
- Scopo: mantenere le funzione le più semplici possibili, al fine di avere del codice più semplice e più facilmente gestibile.

M12PI:

- Nome: Profondità di Innestamento;
- Descrizione: numero intero che dato una funzione indica la quantità di blocchi condizionali e di cicli innestati al suo interno;
- Scopo: diminuire questo valore al fine di rendere il codice più semplice possibile e gestibile nel corso del tempo.

M13CPC:

- Nome: Linee di Commento per Linee di Codice;
- Descrizione: indica il numero di linee di commento a confronto con le effettive righe totali di codice;
- Scopo: il codice deve essere commentato, per permettere una comprensione più facile da membri del gruppo diversi dal creatore del codice.

3 Processi di Supporto

3.1 Documentazione

3.1.1 Scopo

Lo scopo di questa sezione è di riportare tutte le regole, template e decisioni per la stesura di tutti i documenti del gruppo Sweven.

Tutti i documenti verranno scritti in \LaTeX .

3.1.2 Ciclo di vita dei documenti

Il documento viene innanzitutto pianificato, cioè ci si chiede perchè è necessario e si pensa al suo contenuto. Gli amministratori nel ruolo di redattori, creano e scrivono il documento, poi i verificatori lo controllano e il responsabile approva il documento finito e pronto ad essere ufficialmente pubblicato.

Se i verificatori trovano errori superficiali e/o di ortografia correggono immediatamente, mentre se l'errore è più profondo o è necessario riscrivere delle parti allora si incarica nuovamente il redattore di sistemare e poi il verificatore controllerà nuovamente.

3.1.3 Versionamento dei documenti

I documenti sono redatti in maniera incrementale e in più momenti, quindi per una migliore gestione si tengono versionati.

Versione: x.y.z

x è il valore più grande e indica quale versione è stata ufficialmente pubblicata. Viene aggiornato solo dal responsabile quando l'approvazione del documento ha esito positivo e si riportano a zero sia y che z;

y è il valore che indica lo stato di verifica globale del documento. Viene aggiornato dal verificatore quando la verifica globale del documento ha esito positivo, e si riporta a zero z;

z è il valore che indica lo stato di stesura e relativa verifica di singole parti del documento. Viene aggiornato dal verificatore quando verifica positivamente quella sezione scritta dal redattore.

A fine progetto è necessario che tutti i documenti saranno alla versione x.0.0.

3.1.4 Suddivisione documenti interni ed esterni

I documenti si suddividono in interni ed esterni in base allo scopo del documento e ai destinatari. Nei documenti interni i destinatari sono i membri del gruppo Sweven e i committenti, nei documenti esterni anche l'azienda proponente.

Interni:

- **Verbali** di riunioni interne tra i componenti del gruppo Sweven

- **Glossario** nei destinatari si è deciso di mettere tutti per conoscenza
- **Norme di Progetto** nei destinatari si è deciso di mettere tutti per conoscenza

Esterni:

- **Verbali** di riunioni con il gruppo Sweven e l'azienda proponente
- **Studio di Fattibilità**
- **Candidatura**
- **Analisi dei Requisiti**
- **Piano di Progetto**
- **Piano di Qualifica**
- **Specifica Architettuale**
- **Manuale Utente**

3.1.5 Documenti e template

La prima sottosezione illustra il template comune a tutti i documenti.

Nelle sottosezioni successive, una per ogni documento, è scritto lo scopo del documento ed eventuali particolarità di template.

3.1.5.1 Template comune a tutti i documenti

Tutti i documenti vengono suddivisi in varie sezioni e ad ognuna corrisponde un file, in modo tale da rendere più agevole l'aggiornamento e la revisione dei file, oltre a garantire la possibilità di lavorare in contemporanea in maniera asincrona allo stesso file (è sufficiente lavorare in sezioni diverse).

1. **configuration:** costituisce il file principale del documento, contiene tutti i comandi, i pacchetti necessari e riporta le regole generali del documento come i margini, lo stile della pagina, l'intestazione, la numerazione. Inoltre contiene il link a tutti gli altri file che costituiscono le varie parti del documento.
2. **frontespizio:** il file rappresenta la prima pagina del documento, e quindi in alto si è lasciato un notevole spazio bianco, il contenuto è tutto centrato e inizia con il nome del documento, poi l'immagine del logo con sotto nome ed email del gruppo Sweven Team. Poi c'è una tabella centrata con due colonne (visibile solo riga superiore e bordo centrale delle colonne), in cui vengono riportate le varie informazioni, che sono state settate all'inizio del file configuration:
 - Versione
 - Uso
 - Destinatari

- Stato
- Redattori
- Verificatori
- Approvatori

L'elenco dei destinatari, redattori, verificatori e approvatori può essere più di una persona e affinché i nomi siano scritti uno sotto l'altro, nel comando si può scrivere :

Nome\\& Nome\\& Nome\\.

Dopo altro spazio si trova la sintesi del documento, questa frase ha lo scopo di rappresentare molto sinteticamente il contenuto del documento così da permettere al lettore di capire dal frontespizio se è di suo interesse o meno.

3. **diario delle modifiche:** il diario delle modifiche è costituito da una tabella con 5 colonne e tutti i bordi anche delle righe visibili. Le 3 colonne Versione, Data, Ruolo (ultima colonna) sono state impostate center mentre per le altre due Descrizione e Autore è rispettivamente stata data la dimensione di 12em e 7em, inoltre il testo è allineato a sinistra. Viene aumentata l'ampiezza righe ad 1.8 così da non avere tutte le righe attaccate.

La nuova riga la si aggiunge sempre ad inizio tabella così da ottenere che la prima riga comunica quale è stata l'ultima modifica al documento e la versione qui riportata deve corrispondere a quella scritta nel frontespizio.

La data viene scritta in formato americano aaaa-mm-gg Nella colonna autore non si suddividono in sillabe i nomi o cognomi, se capita usare Nome \newline Cognome.

Dopo il diario delle modifiche, nel file configuration c'è il comando di creare l'indice del documento.

4. **contenuto:** le pagine successive contengono il contenuto vero e proprio del documento seguendo l'indice (possono essere anche più file). In tutte le pagine diverse dal frontespizio è prevista un'intestazione in grigio in cui a sinistra c'è il nome del gruppo e a destra si riporta il nome del documento. Mentre nel piè di pagina viene riportato il numero della pagina rispetto alle pagine totali.

3.1.5.2 Verbalì

Rispetto a quanto scritto nella sezione 3.1.5, sono inoltre presenti altri due file template per la stesura dei verbalì: "informazioni" e "conclusioni-decisioni".

L'ordine delle varie parti sarà il seguente:

1. **configuration**
2. **frontespizio**
3. **diario delle modifiche**

4. **informazioni:** questa pagina contiene le informazioni della riunione e l'ordine del giorno previsto. Le informazioni prevedono data, ora, luogo, lista partecipanti ed eventuali assenti.
5. **svolgimento:** l'equivalente del "contenuto" degli altri documenti, sarà un unico file e si svilupperanno i punti scritti nell'ordine del giorno.
6. **conclusioni-decisioni:** la sezione conclusioni riassume in maniera sintetica quanto detto durante la riunione, riassume ciò che è da fare nel breve futuro e se già stabilita si indica la data della prossima riunione.
La tabella del tracciamento delle decisioni, costituita da due colonne, serve per riportare in maniera schematica le decisioni prese e assegnare loro un codice VI_aaaa-mm-gg oppure VE_aaaa-mm-gg in base a verbale interno o esterno, così se necessario questo codice può essere usato per riferirsi alla decisione presa e leggendo il verbale ne troverà la spiegazione.

Il verbale è un documento che viene scritto, verificato e approvato una sola volta, in quanto nel tempo non si aggiorna il verbale di una vecchia riunione. Inoltre il redattore non verrà affiancato dal verificatore durante la scrittura ma avverrà solo al termine. Quindi all'esito positivo della verifica si userà direttamente il codice 0.1.0 e poi 1.0.0 con l'approvazione.

3.1.5.3 Glossario

Il glossario è un documento che ha lo scopo di spiegare il significato di alcuni termini usati all'interno degli altri documenti, così da facilitare la comprensione di essi.

Il glossario non prevede la sezione di introduzione, si è deciso che si userà l'ordine alfabetico così da facilitarne la ricerca, nell'indice compariranno tutte le parole presenti nel documento. Ogni lettera dell'alfabeto costituirà una nuova sezione e ogni parola avrà la sua sottosezione con scritto il significato.

Quando nei documenti si usa una parola e si vuole rimandare al glossario per la spiegazione avrà la G al pedice, ciò si inserisce usando il comando `\glossario`. Per praticità e averne lo stile già impostato, questo comando è stato definito nel file configuration con la scrittura in corsivo del termine e l'aggiunta del pedice.

3.1.5.4 Norme di Progetto

Le norme di progetto, cioè questo documento, hanno lo scopo di redigere e riportare tutte le norme e le decisioni del gruppo Sweven. Il documento sarà composto da tre grandi sezioni: processi primari, processi di supporto e processi organizzativi.

3.1.5.5 Studio di Fattibilità

Lo studio di fattibilità è un documento scritto durante la scelta del capitolato per l'appalto con lo scopo di analizzare ogni capitolato rimasto scrivendo pregi, difetti e le criticità rilevate dai membri del gruppo.

3.1.5.6 Candidatura

La candidatura è il documento ufficiale mediante il quale il gruppo si candida alla gara d'appalto per l'assegnazione del capitolato, dichiarando anche il preventivo dei costi e della data di consegna.

3.1.5.7 Analisi dei Requisiti

L'analisi dei requisiti è un documento scritto durante la fase di analisi del progetto scrivendo in dettaglio i casi d'uso e poi poter mappare i requisiti del prodotto.

3.1.5.8 Piano di Progetto

Il piano di progetto è un documento in cui il gruppo dichiara come verrà gestito lo sviluppo del progetto, scrivendo in dettaglio la pianificazione con relativo preventivo e infine viene riportato il consuntivo effettivo.

3.1.5.9 Piano di Qualifica

Il piano di qualifica è un documento in cui il gruppo dichiara come analizza e verifica che i documenti e il prodotto siano di qualità ponendosi sia soglie minime che soglie desiderabili da soddisfare nelle metriche.

3.1.5.10 Specifica Architettuale

La specifica architettuale è un documento scritto durante le fasi di progettazione per illustrare l'architettura del prodotto *Bot4Me*

3.1.5.11 Manuale Utente

Il manuale utente è un documento redatto per agevolare e fornire tutte le informazioni utili all'utente che utilizzerà il *Bot4Me*

3.1.6 Convenzioni tipografiche

In questa parte si riportano le varie convenzioni per i documenti decise all'interno del gruppo.

3.1.6.1 Nome del file

I documenti pdf vanno nominati con il nome del documento iniziando con la lettera maiuscola e ogni nuova parola inizia con la lettera maiuscola (comprese le proposizioni). Per i verbali in aggiunta va scritta anche la data all'americana usando i trattini e un trattino stacca la data dal nome; esempi: NormeDiProgetto, VerbaleInterno-2022-03-31.

Il gruppo dà importanza ai nomi dei file che vengono ufficialmente pubblicati, mentre per gli altri documenti viene data più libertà al creatore del file, invitando comunque all'utilizzo dell'underscore (trattino basso) o delle maiuscole per separare diverse parole.

3.1.6.2 Stile del testo

- **Normale 12 pt** in generale tutto il testo dei documenti;
- **Grassetto** tutti i titoli che si differenziano per la grandezza automatica di \LaTeX ;
- **Corsivo** i riferimenti ad altri documenti, alle parole del glossario;
- **Elenchi** gli elementi dell'elenco vanno in grassetto solo se è prevista una descrizione. Si usa un ; al termine di ogni voce dell'elenco, tranne all'ultimo.
Nel caso di elenchi annidati, il grassetto si usa solo nell'elenco principale.

3.1.6.3 Scrittura della data e dell'ora

La data va sempre scritta nel formato americano aaaa-mm-gg.
Per l'ora si usa il formato a 24 ore hh:mm.

3.1.6.4 Sigle

- **Nomi dei documenti**

- VI: Verbale Interno;
- VE: Verbale Esterno;
- G: Glossario;
- NdP: Norme di Progetto;
- AdR: Analisi dei Requisiti;
- PdP: Piano di Progetto;
- PdQ: Piano di Qualifica;
- SA: Specifica Architetture;
- MU: Manuale Utente.

- **Revisioni di avanzamento**

- RTB: Requirements and Technology Baseline;
- PB: Product Baseline;
- CA: Customer Acceptance.

- **Ruoli**

- Re: Responsabile;
- Am: Amministratore;
- An: Analista;
- Pt: Progettista;
- Pg: Programmatore;
- Ve: Verificatore;

3.1.7 Metriche

M14IG:

- Nome: Indice di Gulpease;
- Descrizione: numero intero positivo rappresentante la leggibilità di un testo italiano. Il valore è compreso tra 0 e 100:
 - < 80 il testo è difficile da leggere per chi ha la licenza elementare;
 - < 60 il testo è difficile da leggere per chi ha la licenza media;

- < 40 il testo è difficile da leggere per chi ha la licenza superiore.
- Scopo: rendere il documento leggibile;
- Formula:

$$89 + \frac{300 * (\#frasi) - 10 * (\#parole)}{\#lettere}$$

3.2 Gestione dello spazio di lavoro

Questa sezione serve a dare un modello alle procedure di lavoro e a normalizzare i comportamenti necessari per favorire lo svolgersi ordinato del processo nella sua totalità

3.2.1 Stato dei documenti

In precedenza è stato normalizzato il versionamento dei documenti. Un'altra caratteristica altrettanto importante che essi presentano è lo stato. Lo stato di un documento è in una e una sola delle seguenti due alternative:

- In lavorazione;
- Approvato.

Ogni documento parte dalla fase “In lavorazione”; solo l’approvatore può, dopo aver analizzato il documento, decidere se renderlo “Approvato” o meno.

Per scelta del gruppo, ogni documento può essere aggiunto al ramo principale della repository di GitHub in uso solo se presenta lo stato “Approvato”.

3.2.2 Versionamento dei software

Similmente ai documenti, anche i software sono versionati. Ciò avviene secondo il formato:

[x].[y].[z]

x: Indica la versione del software. Viene aggiornato solo quando si soddisfano tutti i requisiti obbligatori.

y: Viene incrementato ad ogni requisito aggiornato.

z: Viene incrementato quando si modifica il software senza raggiungere requisiti.

Come nei documenti, anche per i software la modifica di un qualsiasi campo della versione resetta tutti gli altri campi presenti alla destra di esso.

In sintesi x indica il raggiungimento dell’obiettivo primario, y gli obiettivi raggiunti dall’ultimo cambiamento di x, z il numero di piccole modifiche dall’ultimo cambiamento di x e y.

Quindi il software alla sua creazione è già alla versione 0.0.1 e a fine progetto è necessario che tutti i software siano alla versione 1.y.z.

3.2.3 Organizzazione della repository

I partecipanti del gruppo lavorano singolarmente in locale. Al termine della loro task, condividono il lavoro effettuato tramite un commit del file, o dei file, nel ramo adeguato della repository. I suffissi dei file ricadono nei seguenti casi:

- **.tex**: Il codice \LaTeX ;
- **.jpg .png**: Immagini necessarie e sono sempre contenute nella cartella “images”;
- **.pdf**: Il formato PDF del documento da visionare.

3.2.4 Task

Le task da svolgere singolarmente o in sottogruppi vengono assegnate ai membri del gruppo Sweven tramite GitHub. La piattaforma infatti mette a disposizione degli utenti uno strumento di creazione e gestione delle issues. Tali issues verranno create con i seguenti requisiti:

- **Titolo**: Un titolo semplice, immediato e in grado di riassumere la descrizione;
- **Descrizione**: La descrizione dell’obiettivo da raggiungere, sia in complessivo sia singolarmente. Vengono quindi elencate le task che dovranno seguire i partecipanti;
- **Label**: Un’etichetta che descrive il tipo di lavoro da effettuare;
- **Partecipanti**: La lista di tutti i partecipanti alla issue;
- **Milestone**: Il macrogruppo di obiettivi cui la issue appartiene;
- **Project**: Indica a quale revisione la issue appartiene.

3.3 Gestione degli errori

Questa sezione ha lo scopo di gestire le modalità di approccio e di soluzione ai problemi che possono verificarsi nell’arco dello svolgimento dell’intero progetto. Per evitare perdite di tempo eccessive a causa dei vari ed eventuali errori, serve limitare al massimo due fattori: il ripetersi del medesimo errore e il propagarsi dell’errore su altre entità.

Il percorso per il controllo e l’eliminazione degli errori è il seguente:

1. Individuazione
 2. Analisi possibili conseguenze su entità esterne
 3. Identificazione
 4. Creazione possibili soluzioni
 5. Scelta soluzione
 6. Verifica soluzione scelta
- In caso di verifica con esito positivo, procedere dal punto 7.

- In caso di verifica con esito negativo, riprendere dal punto 5.

7. Modifica del sistema di verifica per evitare che l'errore si presenti nuovamente

Seguono approfondimenti su alcune delle parti sovraelencate.

3.3.1 Individuazione

Non appena individuato un problema si contattano immediatamente le persone coinvolte direttamente o indirettamente con esso. Si trovano quindi le possibili cause.

3.3.2 Identificazione

Al problema viene assegnato un codice univoco che viene successivamente memorizzato. Il codice ha il seguente formato:

PROB [PRIORITY] - [TYPE] - [NUMBER]

| Nome | Descrizione |
|----------|---|
| PROB | Indica un problema |
| PRIORITY | Indica il tipo di priorità : MAX : Priorità totale, problema urgente MED : Priorità media, problema non urgente ma importante MIN : Priorità bassa, problema non urgente e poco importante |
| TYPE | Indica il tipo di problema : G : Problema di grammatica (ortografico, sintattico, etc.) C : Problema di contenuto (dati mancanti o errati) F : Problema di funzionalità (qualcosa non funziona) |
| NUMBER | Codice Numerico Identificativo |

La risoluzione di una serie di problemi va eseguita prima secondo l'ordine di priorità e poi secondo l'ordine degli identificativi.

La priorità viene calcolata in base alla gestione dei tempi e in base all'impatto che il problema ha sul resto del progetto, il che comprende l'eventuale propagazione valutata nel punto 2. del processo.

3.3.3 Modifica del sistema di verifica

Il codice del problema viene aggiunto, insieme ad una breve descrizione e alla soluzione, nel sistema di gestione delle issues offerto da GitHub accennato in precedenza. In questo modo si riesce a tenere meglio sotto controllo l'evolversi della situazione.

3.4 Controllo della qualità

Questa sezione serve a normalizzare le procedure atte a controllare la qualità del prodotto. Tale qualità viene approfondita maggiormente nel PdQ.

La qualità va verificata sia sul software che sulla documentazione.

Risulta dunque necessario non solo assicurarsi che tutti i prodotti soddisfino i requisiti richiesti, ma anche trovare misure per verificare con oggettività una simile richiesta.

3.4.1 Definizione di qualità

Per qualità si intende la capacità di superare processi di verifica e di validazione con esiti positivi.

La verifica certifica la correttezza del processo, mentre la validazione certifica il raggiungimento dei requisiti.

Per ogni processo bisogna garantirne efficacia, ovvero la capacità di raggiungimento degli obiettivi prefissati, ed efficienza, ovvero la minimizzazione dei costi (che siano in termini di peso, di tempi o di soldi).

3.4.2 Misure di qualità

Per mantenere l'oggettività nella misurazione della qualità di un ente, è necessario introdurre delle metriche. Tali metriche saranno dipendenti dall'ente cui sono legate.

Per identificarle si utilizza il seguente formato:

M [NUMBER] [INITIALS]

| Nome | Descrizione |
|----------|--------------------------------|
| M | Indica una metrica di qualità |
| NUMBER | Codice Numerico Identificativo |
| INITIALS | Lettere iniziali del nome |

Per ogni metrica è bene definire su cosa si basa, lo scopo e l'eventuale formula di calcolo. Ulteriori dati sono da aggiungere a discrezione.

3.4.3 Classificazione dei processi

Per tenere sotto controllo la qualità dei processi è bene identificarli. Ciò viene eseguito secondo il seguente formato:

Q_PRC [NUMBER]

| Nome | Descrizione |
|--------|---|
| Q_PRC | Indica la memorizzazione dei processi secondo qualità |
| NUMBER | Codice Numerico Identificativo |

3.4.4 Classificazione dei prodotti

Analogamente alla sottosezione precedente, anche i prodotti sono identificati, secondo il formato:

Q_PRD [NUMBER]

| Nome | Descrizione |
|--------|---|
| Q_PRD | Indica la memorizzazione dei prodotti secondo qualità |
| NUMBER | Codice Numerico Identificativo |

3.5 Verifica

Il processo di verifica è la fase obbligatoria per il procedimento alla validazione, applicata ogni volta che c'è un rischio di errore.

3.5.1 Analisi

Ci sono due tipologie di analisi del prodotto software disponibile, vengono effettuate entrambi se il gruppo lo ritiene necessario:

- **Analisi Statica:** non richiede l'oggetto di verifica, viene studiato la documentazione e il codice non eseguibile, accerta conformità a regola, assenza di difetti, presenza di proprietà desiderate descritte nel Piano di Qualifica;
- **Analisi Dinamica:** serve l'oggetto testabile, vengono effettuate dei test su di esso.

3.5.2 Metodi di lettura

Vengono definite le due modalità di lettura per l'analisi statica:

- **Walkthrough:** la lettura parte da zero, analizza tutto il documento, il verificatore deve presupporre di cercare i difetti del codice o del documento;
- **Inspection:** viene fatto la lettura mirata, con una lista di checklist.

3.5.3 Test

Il test serve per confermare che il prodotto software funzioni effettivamente come desiderato, per questo serve l'oggetto testabile; viene effettuato qualvolta esiste un minimo componente funzionante testabile. Il test deve essere:

- **Ripetibile:** il test deve dare lo stesso risultato con lo stesso input;
- **Automatizzabile:** il test deve essere eseguibile da un processo automatizzato, minimizzare il costo delle persone e velocizzare il processo.

Il test è composto da tre elementi:

- Ambiente d'esecuzione: hardware/software in cui è stato eseguito il test e lo stato iniziale del sistema;
- Attesa: input richiesti per l'esecuzione del test ed il suo corrispondente output atteso;
- Procedura: il procedimento in cui l'oggetto viene analizzato.

3.5.3.1 Test di unità

Un test automatico effettuato sul minimo componente che ha bisogno di un test specifico. Per l'identificazione si usa il seguente formato:

T_U [NUMBER]

| Nome | Descrizione |
|--------|--------------------------------|
| T_U | Indica il test di unità |
| NUMBER | Codice Numerico Identificativo |

3.5.3.2 Test di integrazione

Test da eseguire quando viene effettuato un'unione di componenti verificati, assicurare che l'integrazione avviene correttamente. Per l'identificazione si usa il seguente formato:

T_I [NUMBER]

| Nome | Descrizione |
|--------|--------------------------------|
| T_I | Indica il test di integrazione |
| NUMBER | Codice Numerico Identificativo |

3.5.3.3 Test di sistema

Il test da eseguire alla fine del programma, agisce su tutti i requisiti, testare tutti i requisiti sono stati soddisfatti. Per l'identificazione si usa il seguente formato:

T_S [NUMBER]

| Nome | Descrizione |
|--------|--------------------------------|
| T_I | Indica il test di sistema |
| NUMBER | Codice Numerico Identificativo |

3.5.3.4 Test di regressione

Test da eseguire dopo ogni aggiustamento di un errore, per evitare che la modifica comporta un errore in un'altra parte, dove già effettuato la verifica.

Per l'identificazione si usa il seguente formato:

T_R [NUMBER]

| Nome | Descrizione |
|--------|--------------------------------|
| T_I | Indica il test di regressione |
| NUMBER | Codice Numerico Identificativo |

3.6 Validazione

Il processo di validazione è il processo successivo alla verifica. Quando gli sviluppatori e i verificatori si ritengono soddisfatti del prodotto finale e si ritiene che sia pronto alla consegna, viene richiesta una riunione del gruppo insieme con il responsabile nella quale vengono rivalutati tutte le aspettative e le richieste del proponente. Nel caso questa fase abbia esito positivo, questo viene approvato e rilasciato come prodotto finale.

3.6.1 Attività

La validazione è composta da due fasi:

- **Test di accettazione:** vengono riefettuati tutti i test, assicurandosi che vengano soddisfatti tutti i requisiti necessari;
- **Collaudo:** insieme al proponente viene effettuata un riunione dove viene illustrato il corretto funzionamento dei requisiti richiesti.

4 Processi Organizzativi

4.1 Gestione Organizzativa

4.1.1 Scopo

Lo scopo di questa parte è quello di fornire un insieme organizzato di attività al fine di:

- fare proprio un modello organizzativo per il tracciamento dei rischi;
- stabilire ruoli per pianificare il lavoro e rispettare le scadenze;
- scegliere gli strumenti per le comunicazioni interne e esterne;
- decidere un modello di sviluppo.

4.1.2 Aspettative

Le aspettative per questa parte sono:

- ottenere un piano di schemi da seguire;
- definire i ruoli all'interno del gruppo;
- agevolare le comunicazioni interne e esterne al gruppo;
- controllare il progetto e le attività del gruppo.

4.1.3 Descrizione

Di seguito vengono riportati gli argomenti delle attività di organizzazione:

- definizione dei ruoli e dei compiti assegnati ai componenti del gruppo;
- gestione delle comunicazioni;
- modalità di esecuzione delle attività;
- esame dei progressi delle attività;
- stima dei tempi, risorse e costi.

4.1.4 Ruoli di progetto

I ruoli del progetto saranno ricoperti da ogni membro del gruppo in rotazione per permettere una equa distribuzione delle mansioni da svolgere. L'organizzazione delle attività è esposta nel *Piano di Progetto* e deve essere seguita dalle varie figure progettuali, ovvero:

4.1.4.1 Responsabile

Il responsabile è quella figura che si occupa della parte di coordinamento, di pianificazione del progetto e di mediazione con i soggetti esterni al gruppo. Si occupa di:

- controllare le attività del team;
- pianificare le attività del team;
- coordinare i membri del gruppo;
- approvare i documenti;
- gestire le relazioni esterne.

4.1.4.2 Amministratore

L'amministratore è colui che gestisce l'ambiente di lavoro all'interno del gruppo e le attività che svolge sono:

- controllare le infrastrutture di supporto;
- documentare le regole e gli strumenti utilizzati;
- attuare le scelte tecnologiche fissate dal gruppo;
- controllare le configurazioni e le versioni.

4.1.4.3 Analista

L'analista ha il dovere di identificare e comprendere il dominio del problema per consentire, successivamente, una corretta progettazione. I suoi compiti sono:

- analizzare il dominio del problema;
- scrivere l'*Analisi dei Requisiti*.

L'analista partecipa al progetto fino a quando non si conclude l'analisi del problema.

4.1.4.4 Progettista

Il progettista concorre alla ricerca di una soluzione per il prodotto e alle scelte tecniche e tecnologiche. Partecipa allo sviluppo software ma non alla manutenzione. Le sue attività sono:

- progettare l'architettura dell'applicativo in modo che sia mantenibile e affidabile;
- trovare soluzioni efficienti ai problemi tecnici e tecnologici del progetto;
- controllare la fase di sviluppo.

4.1.4.5 Programmatore

Il programmatore copre la parte di codifica del progetto usando le soluzioni e tecnologie stabilite dal team, inoltre spetta a questa figura la scrittura dei test per la validazione. Ricapitolando si occupa di:

- scrivere il codice che implementi le soluzioni trovate dal progettista;
- realizzare i test per la verifica e validazione del software;
- stilare il *Manuale Utente*.

4.1.4.6 Verificatore

Il verificatore è tenuto a esaminare i progressi del lavoro compiuto dagli altri membri del gruppo. L'attività di verifica viene condotta sul codice e sui documenti col fine di far rispettare le *Norme di Progetto*.

4.1.5 Gestione delle riunioni

4.1.5.1 Riunioni interne

Le riunioni interne avvengono su *Zoom_G*: il link per collegarsi viene inviato, generalmente, qualche minuto prima dell'inizio. Ad ogni riunione i membri del gruppo si aggiornano sul lavoro svolto, discutono dei problemi e/o dubbi incontrati e, a seguire, stabiliscono le attività da svolgere per il prossimo incontro. Tramite un foglio Excel in cui segnare i propri impegni, i componenti del team hanno scelto un giorno fisso in cui effettuare la riunione settimanale.

4.1.5.2 Riunioni esterne

Le riunioni esterne avvengono tramite *Zoom_G* sia con il proponente che con il committente. Prima di ogni incontro con l'azienda Imola, vengono inviate delle mail per concordare la data e le coordinate della riunione. Generalmente si scelgono degli argomenti su cui verterà la riunione e successivamente vengono anticipati all'azienda tramite *Telegram_G* o email.

4.1.5.3 Verbali

In tutte le riunioni viene stabilito un Redattore, il quale si occupa di riassumere tutto ciò che viene detto durante il meeting, un Verificatore, il quale si assicura che non vi siano errori, e un Approvatore, a cui spetta il compito di confermare o respingere, per ulteriori modifiche, il documento. La struttura del verbale viene largamente approfondita nei processi di supporto al punto §3.1.5.2.

4.1.6 Metriche

M15VC:

- Nome: Variazione di Costo;
- Descrizione: numero che indica di quanto si è sopra o sotto al budget preventivato;
 - se è > 0 indica che si è sotto al budget;

- se $\epsilon = 0$ indica che si sta spendendo esattamente il budget;
 - se $\epsilon < 0$ indica che si è sopra al budget;
- Scopo: controllare quanto si sta spendendo al fine di rispettare il preventivo;
- Formula: $CP - CE$
 - CP: indica il Costo Pianificato per svolgere le attività del progetto fino a un determinato momento;
 - CE: indica il Costo Effettivo per svolgere le attività del progetto fino a un determinato momento.

M16VP:

- Nome: Variazione di Piano;
- Descrizione: numero che indica di quanti giorni il lavoro è avanti o indietro rispetto alla pianificazione;
 - se $\epsilon > 0$ si è in anticipo rispetto alla schedule;
 - se $\epsilon = 0$ si è rispettata la schedule;
 - se $\epsilon < 0$ si è in ritardo rispetto alla schedule;
- Scopo: controllare quanto tempo si sta impiegando al fine di rispettare la pianificazione;
- Formula: $GP - GC$
 - GP: indica i Giorni Pianificati per completare il lavoro;
 - GC: indica i Giorni Consuntivati per completare il lavoro;

4.2 Gestione dell'infrastruttura

4.2.1 Scopo

Lo scopo di questa sezione del documento è quello di definire gli strumenti con cui vengono gestite le comunicazioni e l'ambiente di lavoro.

4.2.2 Aspettative

L'aspettativa principale è quella di avere un insieme di strumenti e modalità che i membri del gruppo dovranno utilizzare e rispettare.

4.2.3 Descrizione

In questa parte vengono stabiliti:

- le modalità da seguire per le comunicazioni;
- gli strumenti da adottare per le comunicazioni;
- gli strumenti per la gestione del lavoro.

4.2.4 Gestione delle comunicazioni

4.2.4.1 Comunicazioni interne

Per le comunicazioni interne il gruppo si avvale di due applicazioni, ovvero *Zoom Meetings_G* e *Telegram_G*. *Zoom_G* è una piattaforma di videoconferenza conosciuta da tutti i membri del team e adatta per le riunioni, mentre *Telegram_G* è una applicazione di messaggistica utile per le conversazioni più rapide.

4.2.4.2 Comunicazioni esterne

Le comunicazioni esterne avvengono attraverso:

- Email con l'indirizzo di posta elettronica del gruppo (swe7.team@gmail.com).
- *Telegram_G* per comunicazioni veloci con Imola Informatica.
- *Zoom_G* per le riunioni con il proponente.

4.2.5 Strumenti per la comunicazione

4.2.5.1 *Telegram_G*

Telegram viene usato sia per le comunicazioni interne al gruppo sia per le comunicazioni con il proponente: sono stati creati due distinti canali di comunicazione. L'applicazione viene usata anche con lo scopo di:

- scambiare brevi messaggi per coordinare il gruppo;
- scambiare link, documenti e altre risorse informative;
- comunicare tra i singoli membri.

4.2.5.2 *Zoom Meetings_G*

Programma per le videoconferenze che viene utilizzato per le riunioni, sia all'interno del team che con l'azienda. L'applicazione è ritenuta semplice da usare, è conosciuta da tutti i membri ed è adatta per gli incontri, in quanto offre tutte le funzionalità di cui si necessita, tra le quali:

- comunicazione video e audio;
- servizio di chat e possibilità di salvare quest'ultima;
- condivisione dello schermo;
- registrazione del meeting.

4.2.5.3 Gmail

Servizio di posta elettronica usato per le comunicazioni esterne con il proponente e il committente. La email ufficiale del gruppo è swe7.team@gmail.com.

4.2.6 Gestione degli strumenti di coordinamento

4.2.6.1 Ticketing

Il sistema di ticketing consente di sapere quali siano le attività da svolgere e quelle che stanno svolgendo i componenti. Il Responsabile definisce i compiti e li assegna usando l'*issue tracking system*_G fornito da GitHub. Un'attività quindi viene:

- creata dal Responsabile;
- assegnata dal Responsabile ai vari membri;
- assegnata ad una milestone;
- verificata dopo essere stata svolta;
- chiusa in caso di esito positivo da parte dei verificatori, altrimenti modificata fino a quando passa l'attività di verifica.

Per ogni issue vengono stabiliti:

- un titolo che contiene il nome della issue e un codice che la identifica;
- una descrizione dello scopo della issue;
- eventuali etichette per specificarne la categoria.

4.2.7 Gestione dei rischi

Nel corso di un progetto possono esserci eventuali rischi, questi vanno individuati e riportati nel documento *Piano Di Progetto*.

4.2.7.1 Struttura dei rischi

La classificazione dei rischi verrà effettuata tramite il seguente codice:

R [TYPE] [NUMBER]

| Nome | Descrizione |
|--------|---|
| TYPE | Indica il tipo di rischio : T : rischio tecnologico P : rischio personale O : rischio organizzativo R : rischio per requisito |
| NUMBER | Codice Numerico Identificativo |

Ogni rischio è strutturato come segue:

- codice del rischio;
- descrizione del rischio;
- livello di gravità;
- conseguenze nel caso si manifesti;
- precauzioni;
- contromisure da prendere.

4.2.8 Strumenti per la gestione del lavoro

4.2.8.1 Git

Version control system_G gratuito e open source che permette di mantenere una cronologia di tutte le modifiche dei file e consente di fare la maggior parte delle operazioni in locale.

4.2.8.2 GitHub

Piattaforma di hosting su cui verranno creati i repository del progetto la quale si integra con Git per la parte di versionamento. GitHub mette a disposizione un *issue tracking system_G* fondamentale per gestire le attività da svolgere e creare le milestone del progetto.

4.3 Formazione

4.3.1 Scopo

Quello di formazione è un processo atto a fornire e mantenere le competenze per il personale. Ciò che ci si attende da questo processo è l'assicurare la qualità del lavoro tramite le competenze dei membri del gruppo. In particolar modo si cerca di garantire le competenze di tutti i membri del gruppo per quanto riguarda:

- le tecnologie per la stesura della documentazione;
- gli strumenti da utilizzare per l'organizzazione;
- i linguaggi di programmazione e gli ambienti di sviluppo.

4.3.2 Aspettative

Riguardo la formazione, ogni membro del gruppo è tenuto a provvedere alla propria, approfondendo le tecnologie necessarie. Tutti i membri del gruppo devono cercare di rimanere allineati sulle competenze necessarie ai vari ambiti del progetto. E dunque importante che i membri più esperti in determinati ambiti, condividano le loro competenze con chi ha qualche mancanza.

4.3.3 Attività

Le attività di questo processo prevedono la formazione individuale, la condivisione delle competenze e se necessario la formazione di gruppo. È necessario reperire le giuste fonti per la formazione, dando priorità al materiale fornito dal proponente e dai docenti. Le competenze per le quali è richiesta la formazione e le relative fonti sono di seguito elencate:

- **Organizzazione**
 - GitHub: GitHub documentation
- **Documentazione**
 - Latex: latex documentation
- **Sviluppo**
 - Python: python guide html.it
 - API Rest, Articolo API Rest RedHat
 - Chatterbot, chatterbot documentation

4.4 Miglioramento

4.4.1 Scopo

Il processo di miglioramento serve a controllare tutto ciò che riguarda il progetto per poterlo migliorare in qualche suo aspetto, come efficienza ed efficacia.

4.4.2 Aspettative

Con il processo di miglioramento si cerca di tenere sotto controllo tutto quello che viene eseguito nell'ambito del progetto, per ottenere da ogni controllo dei feedback; di questi si continuerà a mantenere gli aspetti positivi e si cercherà di migliorare quelli negativi. Particolare attenzione verrà posta sull'efficacia e sull'efficienza dei processi: esse andranno costantemente monitorate per essere, se serve, migliorate.

4.4.3 Attività

Per avere un miglioramento continuo, il gruppo si confronterà ad intervalli opportuni su ogni processo, in modo da poter revisionare il lavoro svolto. Ogni controllo sarà atto a confermare gli aspetti positivi e definire i miglioramenti da apportare al processo in esame.