



**Università degli Studi di Padova**

Laurea in Informatica

Corso di Ingegneria del Software

Anno Accademico 2023/2024



**Gruppo: SWEet16**

**Email: [sweet16.unipd@gmail.com](mailto:sweet16.unipd@gmail.com)**

# Norme di Progetto

Redattori: Alberto M., Bilal E., Alex S.

Verificatori: Iulius S., Alex S., Alberto M., Bilal E.

Amministratore: Alex S.

Destinatari: T. Vardanega

R. Cardin

Versione: 1.0.0

# Registro delle modifiche

Versione	Data	Autore	Verificatore	Descrizione
1.0.0	2024/04/15	Alex S.		Approvazione per il rilascio
0.12.0	2024/03/29	Alberto M.	Alex S.	Stesura gestione dei processi
0.11.0	2024/03/24	Bilal E. M.	Alberto M.	Stesura verifica
0.10.0	2024/03/19	Alberto M.	Bilal E. M.	Stesura validazione
0.9.0	2024/03/15	Alberto M.	Alex S.	Stesura gestione qualità
0.8.1	2024/03/12	Alex S.	Alberto M.	Modifica struttura UC
0.8.0	2024/03/11	Alberto M.	Alex S.	Stesura progettazione
0.7.0	2024/03/07	Bilal E. M.	Alberto M.	Stesura gestione configurazione
0.6.0	2024/03/03	Alex S.	Alberto M.	Stesura codifica
0.5.1	2024/02/27	Alberto M.	Iulius S.	Aggiunte norme tipografiche
0.5.0	2024/02/23	Alex S.	Alberto M.	Stesura documentazione
0.4.0	2024/02/18	Alberto M.	Alex S.	Stesura sviluppo
0.3.0	2024/02/15	Alberto M.	Alex S.	Stesura fornitura
0.2.0	2024/02/13	Alberto M.	Alex S.	Stesura introduzione
0.1.0	2024/02/10	Alberto M.	Iulius S.	Stesura scheletro

# Indice

<b>1</b>	<b>Introduzione e scopo del documento</b>	<b>7</b>
1.1	Scopo del documento . . . . .	7
1.2	Scopo del prodotto . . . . .	7
1.3	Glossario . . . . .	7
1.4	Maturità del documento . . . . .	7
1.5	Riferimenti . . . . .	8
1.5.1	Riferimenti normativi . . . . .	8
1.5.2	Riferimenti informativi . . . . .	8
<b>2</b>	<b>Processi Primari</b>	<b>9</b>
2.1	Fornitura . . . . .	9
2.1.1	Descrizione, scopo ed aspettative . . . . .	9
2.1.2	Proponente . . . . .	9
2.1.3	Documentazione . . . . .	9
2.1.3.1	Analisi dei Requisiti . . . . .	9
2.1.3.2	Piano di Progetto . . . . .	10
2.1.3.3	Piano di Qualifica . . . . .	10
2.1.3.4	Glossario . . . . .	11
2.1.3.5	Lettera di Presentazione . . . . .	11
2.2	Sviluppo . . . . .	11
2.2.1	Descrizione, scopo ed aspettative . . . . .	11
2.2.2	Analisi dei Requisiti . . . . .	11
2.2.2.1	Denominazione e Legenda . . . . .	12
2.2.2.1.1	Struttura Casi d'Uso . . . . .	12
2.2.2.1.2	Denominazione Casi d'Uso . . . . .	12
2.2.2.1.3	Struttura dei Requisiti . . . . .	12
2.2.2.1.4	Denominazione dei Requisiti . . . . .	13
2.3	Progettazione . . . . .	13
2.3.1	Descrizione, scopo ed aspettative . . . . .	13

2.4	Codifica . . . . .	14
2.4.1	Strumenti utilizzati . . . . .	14
<b>3</b>	<b>Processi di Supporto</b>	<b>15</b>
3.1	Documentazione . . . . .	15
3.1.1	Scopo . . . . .	15
3.1.2	Ciclo di vita del documento . . . . .	15
3.1.3	Template . . . . .	15
3.1.4	Documenti prodotti . . . . .	15
3.1.4.1	Esterni . . . . .	15
3.1.4.2	Interni . . . . .	16
3.1.5	Struttura del documento . . . . .	16
3.1.5.1	Prima pagina . . . . .	16
3.1.5.2	Registro delle modifiche - Changelog . . . . .	16
3.1.5.3	Indice . . . . .	16
3.1.5.4	Verbali . . . . .	17
3.1.6	Norme tipografiche . . . . .	17
3.1.6.1	Nome del file . . . . .	17
3.1.6.2	Stile del testo . . . . .	17
3.1.6.2.1	Elenchi puntati e numerati . . . . .	18
3.1.6.3	Glossario . . . . .	18
3.1.6.4	Altre norme tipografiche . . . . .	18
3.1.7	Elementi grafici . . . . .	18
3.1.7.1	Immagini . . . . .	18
3.1.7.2	Tabelle . . . . .	18
3.1.8	Strumenti . . . . .	19
3.2	Gestione della Configurazione . . . . .	20
3.2.1	Descrizione, scopo ed aspettative . . . . .	20
3.2.2	Versionamento . . . . .	20
3.2.3	Repository . . . . .	20

3.2.3.1	Tecnologie e strumenti . . . . .	20
3.2.3.2	Struttura Repository . . . . .	20
3.2.3.3	Modifiche alla repository . . . . .	21
3.3	Gestione della Qualità . . . . .	22
3.3.1	Descrizione, scopo ed aspettative . . . . .	22
3.3.2	Piano di Qualifica . . . . .	22
3.3.3	Denominazione metriche . . . . .	22
3.4	Verifica . . . . .	23
3.4.1	Scopo aspettative e descrizione . . . . .	23
3.4.2	Analisi statica . . . . .	23
3.4.3	Analisi dinamica . . . . .	23
3.4.4	Verifica della documentazione . . . . .	24
3.5	Validazione . . . . .	24
3.5.1	Descrizione, scopo ed aspettative . . . . .	24
<b>4</b>	<b>Processi organizzativi</b>	<b>25</b>
4.1	Gestione dei Processi . . . . .	25
4.1.1	Descrizione, scopo ed aspettative . . . . .	25
4.1.2	Ruoli di progetto . . . . .	25
4.1.2.1	Responsabile di progetto . . . . .	25
4.1.2.2	Amministratore di progetto . . . . .	26
4.1.2.3	Analista . . . . .	26
4.1.2.4	Progettista . . . . .	26
4.1.2.5	Programmatore . . . . .	26
4.1.2.6	Verificatore . . . . .	27
4.1.3	Procedure . . . . .	27
4.1.3.1	Gestione delle comunicazioni . . . . .	27
4.1.3.1.1	Comunicazioni interne . . . . .	27
4.1.3.1.2	Comunicazioni esterne . . . . .	27
4.1.3.2	Gestione degli incontri . . . . .	27

4.1.3.2.1	Incontri interni . . . . .	27
4.1.3.2.2	Incontri esterni . . . . .	28
4.1.3.3	Rotazione dei ruoli . . . . .	29
4.1.3.4	Gestione dei compiti e delle task . . . . .	29
4.2	Strumenti . . . . .	29
4.3	Formazione . . . . .	30

# 1 Introduzione e scopo del documento

## 1.1 Scopo del documento

Il presente documento si pone lo scopo di individuare e definire le *best practices*<sup>G</sup> e il *Way of Working*<sup>G</sup> del progetto che ogni componente del gruppo SWEet16 si impegna a rispettare durante l'intero svolgimento del progetto Easy Meal.

In questo modo si cercherà di garantire omogeneità e coesione in ogni aspetto del suddetto progetto.

## 1.2 Scopo del prodotto

Lo scopo dell'applicazione è quello di creare una piattaforma che permetta di gestire e semplificare il processo di *prenotazione*<sup>G</sup> di tavoli all'interno dei ristoranti.

Sarà inoltre possibile anticipare l'esperienza culinaria visionando prima il menù ed andando ad effettuare la propria *ordinazione*<sup>G</sup> prima di arrivare al ristorante.

Il prodotto offre inoltre un'esperienza di ordinazione delle *pietanze*<sup>G</sup> collaborativa e coinvolgente, permettendo di condividerla con amici ed, in caso di dubbi, interagire direttamente con lo staff del ristorante.

L'idea è una piattaforma *SaaS (Software as a Service)*<sup>G</sup>, in cui i saranno presenti due tipi di utenti:

- *Cliente*<sup>G</sup>: Utente registrato all'interno dell'applicazione, può cercare ristoranti, effettuare prenotazioni, ordinazioni e inserire feedback e recensioni;
- *Ristoratore*<sup>G</sup>: Utente registrato all'interno dell'applicazione, può gestire uno o più ristoranti, controllando le prenotazioni e le ordinazioni dei clienti ed i menù del/i ristorante/i.

La piattaforma dovrà essere disponibile attraverso una *Webapp*<sup>G</sup> accessibile da qualsiasi dispositivo, esso sia *Desktop*<sup>G</sup> o *Mobile*<sup>G</sup>.

## 1.3 Glossario

Al fine di evitare possibili ambiguità o incomprensioni riguardanti la terminologia usata nel documento, è stato deciso di adottare un glossario in cui vengono riportate le varie definizioni. In questa maniera in esso verranno posti tutti i termini specifici del dominio d'uso con relativi significati.

La presenza di un termine all'interno del glossario viene indicata applicando una " <sup>G</sup> " ad apice della parola.

## 1.4 Maturità del documento

Il presente documento è redatto con un approccio incrementale al fine di poter trattare nuove o ricorrenti questioni in modo rapido ed efficiente, sulla base di decisioni concordate tra tutti i membri del gruppo.

Non può pertanto essere considerato definitivo nella sua attuale versione.

## 1.5 Riferimenti

### 1.5.1 Riferimenti normativi

- Regolamento del progetto didattico:  
<https://www.math.unipd.it/~tullio/IS-1/2023/Dispense/PD2.pdf>
- *Capitolato d'appalto*<sup>G</sup> C3 - Easy Meal:  
<https://www.math.unipd.it/~tullio/IS-1/2023/Progetto/C3.pdf>

### 1.5.2 Riferimenti informativi

- I processi di ciclo di vita del software:  
<https://www.math.unipd.it/~tullio/IS-1/2023/Dispense/T2.pdf>
- Glossario:  
<https://github.com/SWEet16-SWE-Group/docs/blob/main/RTB/Documentazione%20Esterna/Glossario.pdf>
- ISO/IEC 12207:  
[http://www.colonese.it/SviluppoSw\\_Standard\\_ISO12207.html](http://www.colonese.it/SviluppoSw_Standard_ISO12207.html)

Tutti i riferimenti (normativi e informativi) a risorse web soggette a variazione sono stati consultati il 2024/02/19.



## 2 Processi Primari

### 2.1 Fornitura

#### 2.1.1 Descrizione, scopo ed aspettative

Il processo di fornitura determina ogni compito, attività e risorsa necessaria al corretto svolgimento del progetto.

Tale processo verrà avviato solo in seguito ad un'attenta analisi preliminare delle richieste del *proponente*<sup>G</sup>, seguito da uno studio di fattibilità delle menzionate richieste e concluso dalla definizione di un accordo contrattuale con il proponente.

Questa sezione ha lo scopo di elencare tutte le metriche, gli strumenti e i documenti utilizzati al fine di realizzare il processo di fornitura.

Le aspettative sono:

- Avere una chiara struttura dei documenti;
- Definire in modo chiaro le tempistiche di lavoro;
- Chiarire eventuali dubbi e stabilire vincoli con il proponente.

#### 2.1.2 Proponente

Il team si è accordato con l'azienda proponente Imola Informatica per avere un continuo riscontro sul prosieguo del progetto, mantenendosi in contatto asincrono tramite un gruppo *Telegram*<sup>G</sup>. Questo ci permetterà di poter chiarire velocemente eventuali dubbi, ed avere un riscontro sulla documentazione redatta, oltre che alla verifica dei verbali esterni.

Sono inoltre programmati degli incontri sincroni a cadenza variabile, a seguito di una richiesta da parte del gruppo o dall'azienda, utili per chiarimenti sui vincoli o necessità del *capitolato*<sup>G</sup>, oltre che per avere un feedback su quanto prodotto fino a quel momento.

#### 2.1.3 Documentazione

Vengono di seguito elencati i documenti che il gruppo SWEet16 consegnerà all'azienda proponente *Imola Informatica* e ai committenti *Prof. Tullio Vardanega* e *Prof. Riccardo Cardin*.

##### 2.1.3.1 Analisi dei Requisiti

L'Analisi dei Requisiti è un documento che definisce le funzionalità che il prodotto offre ed i requisiti da soddisfare in modo tale che il software sviluppato sia conforme alle richieste del proponente.

Viene così suddiviso:

- Descrizione del prodotto, contenente gli obiettivi del prodotto e le sue funzionalità principali;

- Casi d'uso, i quali descrivono tutti gli scenari di utilizzo del sistema da parte degli utenti;
- Lista dei requisiti, cioè tutte le richieste e vincoli definiti dal proponente o ricavati dal gruppo. I requisiti sono divisi in obbligatori, desiderabili o opzionali.

#### **2.1.3.2 Piano di Progetto**

Il Piano di Progetto verrà redatto durante tutta la durata del progetto, ed andrà a costituire uno strumento utile alla pianificazione di tutte le attività da svolgere, le risorse necessarie e la scadenza del progetto stesso.

È un documento ufficiale soggetto a versionamento ed approvazione, che viene utilizzato per descrivere in modo chiaro e conciso gli obiettivi di progetto e gli elementi necessari per il loro compimento;

È suddiviso nelle seguenti parti:

- Analisi dei rischi: Analisi delle problematiche e difficoltà che potrebbero venire a crearsi durante il progetto e metodologie per prevenire e/o risolvere queste problematiche, in modo da minimizzare questi rallentamenti;
- Modello di sviluppo: Descrizione dell'approccio metodologico e strutturato utilizzato durante lo sviluppo del prodotto;
- Pianificazione: Definizione dei periodi, con le relative attività di progetto da svolgere;
- Preventivo: Riepilogo del prospetto economico ed orario totale, diviso per ogni periodo;
- Consuntivo di periodo: Tracciamento dell'andamento rispetto al preventivo;
- Attualizzazione dei rischi.

#### **2.1.3.3 Piano di Qualifica**

Il Piano di Qualifica rappresenta i compiti e le attività relative al progetto che dovranno essere svolto dal Verificatore all'interno del progetto, utili a garantire qualità al prodotto software che si andrà a sviluppare.

Contiene inoltre le metriche e le misure necessarie a garantire che il prodotto finale sia conforme alle specifiche richieste e alle aspettative del committente.

Nello specifico, esso è formato dalle seguenti parti:

- Qualità di processo: Definizione dei parametri e delle metriche utili a garantire processi di elevata qualità;
- Qualità di prodotto: Definizione dei parametri e delle metriche per garantire un prodotto di elevata qualità;
- Test e specifiche: Descrizione dei test necessari per assicurare che i requisiti stabiliti vengano soddisfatti;
- Resoconto delle attività di verifica ed eventuali criticità riscontrate.

#### 2.1.3.4 Glossario

Il Glossario è un documento di supporto concepito per i membri del gruppo, ma anche per i committenti e l'azienda proponente.

Permette di evitare ambiguità o confusione riguardanti l'utilizzo di terminologia di dominio utilizzata in tutta la documentazione prodotta dal gruppo.

#### 2.1.3.5 Lettera di Presentazione

La Lettera di Presentazione è il documento con cui il gruppo *SWEet16* esprime la propria volontà di partecipare alla fase di revisione del prodotto software.

Al suo interno vengono elencati i documenti prodotti e messi a disposizione per la visione ai committenti e all'azienda proponente.

### 2.2 Sviluppo

#### 2.2.1 Descrizione, scopo ed aspettative

Il processo di sviluppo contiene le attività e i compiti dello sviluppatore, tra cui le attività per l'Analisi dei Requisiti, la progettazione, la codifica ed gli strumenti utilizzati.

Lo scopo del processo di sviluppo è quello di descrivere i compiti e le attività da svolgere per la codifica del prodotto software richiesto.

In questa sezione vengono dunque descritte le attività, le norme e le convenzioni adottate per questo processo.

Le aspettative per una corretta applicazione del processo di sviluppo sono:

- Realizzare un prodotto finale conforme alle richieste del proponente illustrate dall'Analisi dei Requisiti;
- Determinare i vincoli tecnologici;
- Determinare gli obiettivi di sviluppo;
- Determinare i vincoli di *design*<sup>G</sup>.

#### 2.2.2 Analisi dei Requisiti

L'Analisi dei Requisiti è l'attività preliminare che permette di definire chiaramente, grazie al lavoro di Analisti, i requisiti diretti ed indiretti, impliciti ed espliciti che il proponente richiede per la realizzazione del prodotto, ed i vari *casi d'uso*<sup>G</sup> del prodotto stesso.

In questa attività è importante suddividere il problema iniziale in *requisiti*<sup>G</sup> quanto più elementari possibile, andando a facilitare il lavoro durante la fase di sviluppo.

Al fine di poter fornire una corretta interpretazione e realizzazione del prodotto, l'attività ha lo scopo di comprendere le specificità del *capitolato*<sup>G</sup>, sulla base di un confronto mirato con il proponente, interpretando e ampliando la relativa realizzazione.

### 2.2.2.1 Denominazione e Legenda

#### 2.2.2.1.1 Struttura Casi d'Uso

I casi d'uso esprimono un comportamento o un modo di utilizzare il prodotto per raggiungere un determinato obiettivo.

Vengono descritti graficamente mediante l'ausilio di diagrammi *UML<sup>G</sup>*.

Ciascun caso d'uso è costituito da:

- Codice identificativo;
- Attore Primario;
- Precondizioni;
- Postcondizioni;
- Scenario principale;
- Generalizzazioni (se esistono);
- Specializzazioni (se esistono);
- Estensioni (se esistono).

#### 2.2.2.1.2 Denominazione Casi d'Uso

Ciascun caso d'uso viene classificato univocamente mediante l'utilizzo del seguente schema:

UC[Numero].[Sottocaso]-[Titolo]

Dove:

- UC: Acronimo di "Use Case";
- Numero: Numero associato al caso d'uso principale;
- Sottocaso d'uso: Numero associato al sottocaso d'uso (opzionale);
- Titolo: Titolo assegnato al caso d'uso.

#### 2.2.2.1.3 Struttura dei Requisiti

I requisiti individuati nell'Analisi dei Requisiti sono stati strutturati nel modo seguente:

- Codice;
- Descrizione;
- Classificazione in:
  - Requisito obbligatorio;
  - Requisito facoltativo;
  - Requisito desiderabile.
- Fonte: L'origine del requisito.

#### 2.2.2.1.4 Denominazione dei Requisiti

Ogni requisito è identificato univocamente secondo il seguente schema:

**R[Tipo].[Codice].[SottoCodice]**

Dove:

- Tipo: Indica la tipologia del Requisito e può essere:
  - F: Requisito funzionale;
  - Q: Requisito di qualità;
  - S: Requisito di sistema;
  - P: Requisito prestazionale e di sicurezza.
- Codice: Numero identificativo univoco ed incrementale in base alla tipologia di requisito;
- SottoCodice: Numero associato al sottorequisito (opzionale).

### 2.3 Progettazione

#### 2.3.1 Descrizione, scopo ed aspettative

La fase di progettazione ha lo scopo di definire le linee essenziali della struttura del progetto in funzione dei requisiti individuati durante la fase di analisi e specificati nell'Analisi dei Requisiti. Questa attività viene svolta dai progettisti che si occupano di specificare le funzionalità dei sottosistemi, per riuscire a implementare tutti i requisiti specificati all'interno di un unico sistema.

Gli obiettivi che il gruppo si pone durante questa fase del ciclo di vita del software sono:

- Trasformazione di tutti i requisiti in specifiche dettagliate che vadano a coprire tutti gli aspetti del sistema;
- Creazione di una demo prototipale del sistema per testare le tecnologie e la loro compatibilità, detta *PoC (Proof of Concept)*<sup>G</sup> (da qui in poi "PoC");
- Approvazione del passaggio alla fase di sviluppo.

Questo processo può essere suddiviso in tre fasi distinte:

1. Design dell'interfaccia: In questa fase si procede ad un alto livello di astrazione rispetto al funzionamento interno del sistema. L'attenzione è focalizzata sulle tecnologie che verranno utilizzate nella fase di sviluppo software, l'obiettivo di questa fase è la codifica del sopracitato PoC;
2. Progettazione architetturale: In questa fase si definisce la struttura generale del sistema, andando ad ignorare i dettagli interni dei componenti principali. Vengono inoltre definiti i test di integrazione;
3. Progettazione dettagliata: In questa fase si vanno a specificare i dettagli di tutti i componenti del prodotto e le specifiche architetture. Vengono definiti i diagrammi delle classi e i test di unità per ogni componente del prodotto.

## 2.4 Codifica

Superata la fase di Progettazione si passa alla fase di Codifica dove i programmatori realizzano in concreto ciò che è stato deciso nella fase precedente. Tale attività viene svolta seguendo le norme e le convenzioni descritte qui sotto durante la fase di progettazione, allo scopo di garantire codice di qualità che soddisfi le richieste del proponente e sia mantenibile nel tempo.

### 2.4.1 Strumenti utilizzati

- **Git:** Strumento di versionamento per lo sviluppo incrementale e asincrono del codice:

<https://git-scm.com/>

- **GitHub:** Piattaforma di condivisione del codice dotata di ITS per l'assegnazione dei compiti:

<https://github.com/>

- **Docker:** Strumento di sviluppo del software che garantisce delle fondamenta sempre uguali e comuni a tutti i membri del gruppo indipendentemente dal sistema operativo e dalle librerie installate.

Viene utilizzato nella fase di *deployment*<sup>G</sup> del software;

<https://www.docker.com/>

- **Visual Studio Code:** *IDE*<sup>G</sup> multiplatforma versatile e gratuito:

<https://code.visualstudio.com>

## 3 Processi di Supporto

### 3.1 Documentazione

#### 3.1.1 Scopo

Lo scopo di questo documento è fornire uno standard da seguire durante il processo di documentazione.

#### 3.1.2 Ciclo di vita del documento

Tappe fondamentali del ciclo di vita di ogni documento sono:

- **Creazione:** Creazione del documento partendo da un template;
- **Strutturazione:** Creazione di file distinti rappresentanti le singole sezioni dell'indice dei contenuti;
- **Stesura:** Fase di scrittura dei contenuti nei documenti. Può essere fatta da uno o più redattori, lavorando asincronicamente su diverse sezioni;
- **Verifica:** Una volta che una sezione viene completata entra in fase di verifica, dove uno o più verificatori confermeranno le modifiche o rilasceranno dei commenti di correzione, in questo secondo caso il documento torna nella fase di stesura;
- **Approvazione:** Rilascio del documento.

Le prime due fasi sono svolte da tutti i membri del gruppo mentre la terza è affidata esclusivamente al responsabile.

#### 3.1.3 Template

È stato deciso di utilizzare un template  $\text{\textit{L}AT\text{\textit{E}}X}^G$  come base di partenza per la stesura di ogni documento.

#### 3.1.4 Documenti prodotti

I documenti prodotti si possono suddividere in due macrosezioni.

##### 3.1.4.1 Esterni

Di interesse principale per il proponente e i committenti.

- Analisi dei requisiti;
- Piano di progetto;
- Piano di qualifica;
- Verbali esterni.

### 3.1.4.2 Interni

Di interesse principale solo per membri del gruppo.

- Norme di progetto;
- Verbali interni.

### 3.1.5 Struttura del documento

#### 3.1.5.1 Prima pagina

La prima pagina di ogni documento dovrà contenere:

- Logo di Unipd accompagnato dal nome dell'università, corso di laurea, materia e anno accademico;
- Logo del gruppo SWEet16 accompagnato dal nome del gruppo e la rispettiva email;
- Titolo del documento;
- Informazioni aggiuntive:
  - **Redattori:** Indica chi si è occupato di redigere il documento;
  - **Verificatori:** Indica chi sono stati i verificatori in quel documento;
  - **Destinatari:** Indica a chi è destinato il documento;
  - **Versione:** Indica l'attuale versione del documento.

#### 3.1.5.2 Registro delle modifiche - Changelog

Il file `registro-modifiche.tex` contiene il registro delle modifiche, presente in ogni singolo documento, ad eccezione dei verbali.

Esso corrisponde ad una tabella nella quale si tiene traccia di ogni attività svolta sulla documentazione. In particolare la tabella è composta dalle seguenti voci:

- **Versione:** L'attuale versione del documento;
- **Data:** La data in cui avviene un'attività;
- **Autore:** Nome della persona che ha apportato i cambiamenti al documento;
- **Verificatore:** Nome della persona che ha verificato i cambiamenti apportati al documento;
- **Descrizione:** Breve descrizione della modifica effettuata.

#### 3.1.5.3 Indice

Ogni documento presenta un indice di navigazione allo scopo di facilitare quest'ultima e dare un'anteprima di ciò che esso contiene.



#### 3.1.5.4 Verball

I verball sono documenti informali non troppo diversi dagli altri documenti, con l'unica differenza di non essere soggetti a versionamento. La prima pagina è la stessa fornita da template. Il resto della struttura è divisa in due:

- Partecipanti: Coloro che hanno effettivamente partecipato all'incontro;
- Orario di inizio e fine;
- Sintesi ed elaborazione dell'incontro.

#### 3.1.6 Norme tipografiche

##### 3.1.6.1 Nome del file

Di seguito viene descritta la convenzione per la rappresentazione dei nomi dei file, validi per tutti i documenti:

- Il nome di ogni file inizia con la lettera maiuscola;
- Se il nome comprende più parole queste iniziano per lettera minuscola e sono separate da ' ';
- Segue poi il numero di versione composto da una 'v' e tre numeri separati da un '.';
- Un esempio corretto è `Norme_di_progetto_v0.1.0`.

Per i verball il nome del documento sarà semplicemente la data in cui si è tenuto l'incontro nel formato `AAAA-MM-GG`.

##### 3.1.6.2 Stile del testo

Gli stili del testo che vengono apportati all'interno di tutti i documenti sono:

- **Grassetto:** Lo stile grassetto viene utilizzato per indicare i termini degli elenchi puntati ed i titoli delle sezioni;
- **Corsivo:** Lo stile corsivo viene utilizzato per indicare il nome del gruppo, il nome dell'azienda proponente e per le parole di particolare rilevanza;
- **Sottolineato:** Utilizzato per i link presenti nei documenti;
- **Monospazio:** Utilizzato per riferimenti a codice.

### **3.1.6.2.1 Elenchi puntati e numerati**

Di seguito lo stile utilizzato dal team di sviluppo per gli elenchi puntati e numerati:

- Ogni punto dell'elenco inizia con la lettera maiuscola;
- Alla fine di ogni punto vi è un ',';
- Dopo l'ultima voce vi è un ',';
- Se vi è un concetto da spiegare esso viene scritto in grassetto seguito da ':' e segue la spiegazione di esso che comincia per lettera maiuscola.

### **3.1.6.3 Glossario**

Il Glossario è un documento che contiene tutte le parole ritenute ambigue nel contesto del progetto.

I termini presenti sono separati in sezioni indicanti la loro prima lettera, in ordine alfabetico, affiancati da una breve descrizione del significato con cui sono usati e eventuali sinonimi. La definizione di questi ultimi non sarà riportata all'interno.

I termini e i loro eventuali sinonimi riporteranno una 'G' in apice all'interno dei documenti. La G sarà presente solo nella prima occorrenza di quella parola in quel documento.

### **3.1.6.4 Altre norme tipografiche**

- Data in formato AAAA/MM/GG.

## **3.1.7 Elementi grafici**

### **3.1.7.1 Immagini**

Le immagini vanno centrate orizzontalmente nella pagina e devono rispettare i margini previsti dal foglio qual ora fossero molto grandi.

Grafici, diagrammi e schemi contano come immagini.

### **3.1.7.2 Tabelle**

Le tabelle devono occupare tutta la larghezza del foglio indipendentemente dal numero di colonne. Inoltre per tabelle troppo grandi, queste vanno spezzate su più pagine e l'intestazione deve essere ripetuta per ognuno dei seguenti pezzi.

Le righe devono essere a colori alternati per facilitarne la lettura.

### 3.1.8 Strumenti

Di seguito sono riportati gli strumenti utilizzati dal team di sviluppo in fase di redazione dei documenti:

- **L<sup>A</sup>T<sub>E</sub>X**: Linguaggio di markup per la preparazione di testi, basato sul programma di composizione tipografica TEX:

<https://latex-project.org>

- **L<sup>A</sup>T<sub>E</sub>XWorkShop**: Estensione per VSCode che include un compilatore integrato, l'auto-completamento dei comandi L<sup>A</sup>T<sub>E</sub>X e la segnalazione di errori di sintassi. L'estensione è poi configurata in sinergia con *Git*<sup>G</sup> per ignorare la presenza di eventuali artefatti binari risultanti dalla compilazione:

<https://github.com/James-Yu/LaTeX-Workshop>

- **StarUML**: Software di disegno grafico multiplatforma per la produzione di diagrammi *UML*<sup>G</sup>. Il gruppo ha scelto questo strumento perché gratuito (anche se con qualche limitazione) e perché comprende tutte le funzionalità necessarie al nostro utilizzo:

<https://staruml.io/>

## 3.2 Gestione della Configurazione

### 3.2.1 Descrizione, scopo ed aspettative

Lo scopo è di supervisionare e regolare in modo organizzato la creazione di documenti e codice. Ogni elemento del quale dovrà essere gestita configurazione, sarà soggetto a versionamento e controllo delle modifiche, al fine di assicurare l'integrità del prodotto nel tempo.

Tutti gli strumenti dedicati alla configurazione utilizzati per la produzione di documenti e codice vengono raccolti, organizzati e coordinati. Questo include la gestione della struttura e della disposizione dei file all'interno del repository, nonché gli strumenti per il versionamento e il coordinamento.

### 3.2.2 Versionamento

Ogni modifica apportata a un documento genera una nuova versione seguendo il formato  $v[X].[Y].[Z]$ , dove:

- **X:** Rappresenta la versione approvata dal *Responsabile*, l'unico autorizzato ad incrementarla;
- **Y:** Rappresenta la versione approvata dal *Verificatore*, l'unico autorizzato ad incrementarla;
- **Z:** Rappresenta la versione dell'ultima modifica.

Ogni parte del codice di versione inizia da 0 e ritorna a 0 ogni volta che la componente alla sua sinistra viene incrementata.

### 3.2.3 Repository

#### 3.2.3.1 Tecnologie e strumenti

Per il progetto viene usato *GitHub<sup>G</sup>*, che a sua volta è basato sul sistema di versionamento *Git<sup>G</sup>*. Il gruppo fa pieno uso delle funzionalità di *GitHub* tra cui:

- **Issue:** Permette di discutere, tracciare e risolvere un problema nel progetto;
- **Milestone:** Permette di raggruppare e monitorare contemporaneamente una serie di issues aiutando a tracciare il progresso verso obiettivi del progetto;
- **Pull request:** Consente di proporre delle modifiche al codice e di richiederne la revisione da un altro membro.

#### 3.2.3.2 Struttura Repository

Si è deciso di utilizzare un'unica repository contenente il codice sorgente per la documentazione e per il *PoC<sup>G</sup>*. La repository è composta da 3 cartelle così divise:

- **Template:** Contenente il template per il documento  $\text{\LaTeX}$ ;

- **Candidatura:** Contenente i documenti relativi alla candidatura, come l'analisi dei capitoli, la lettera di presentazione, il preventivo dei costi e infine i verbali;
- **RTB:** Contenente i documenti relativi alla revisione ed è suddivisa in tre sotto cartelle:
  - **Documentazione interna**, all'interno della quale sono contenute le norme di progetto;
  - **Documentazione esterna**, che comprende i documenti tra cui Analisi dei requisiti, Piano di progetto, Piano di qualifica e il Glossario;
  - **Verbali**, che a loro volta sono suddivisi in interni ed esterni.

Ciascun documento segue la convenzione di nomenclatura *Nome\_documento*, ad eccezione dei verbali che vengono denominati con la data della loro stesura nel formato *AAAA\_MM\_GG*.

Inoltre la repository è suddivisa in 3 branch principali:

- **Main:** Il branch principale che contiene l'ultima versione dei documenti compilati in formato *PDF*;
- **Poc-main:** Contiene il codice sorgente del *PoC*;
- **Rtb-documentazione-main:** Il branch dedicato alla creazione/modifica dei documenti relativi alla revisione RTB.

### 3.2.3.3 Modifiche alla repository

Le modifiche ai vari documenti non vanno fatte direttamente nel branch *rtb-documentazione-main*, poiché questo porterebbe ad un elevato rischio di incongruenze e conflitti. Per ogni sezione di un documento, vengono creati appositi branch dove è possibile apportare modifiche. Successivamente, una volta terminate le modifiche, è necessario effettuare una *pull request*<sup>G</sup> con verifica obbligatoria per integrare le modifiche nel branch. Se la verifica non è soddisfatta bisogna apportare le giuste modifiche e richiedere un'altra verifica. Se la verifica è soddisfatta le modifiche verranno integrate con il branch *rtb-documentazione-main*, viene quindi effettuato il *merge*.

### 3.3 Gestione della Qualità

#### 3.3.1 Descrizione, scopo ed aspettative

La gestione della qualità di progetto è l'insieme dei processi e delle attività che vengono eseguite per garantire che la qualità del prodotto finale segua standard e metriche rigorose descritte all'interno del Piano di Qualifica.

I suoi obiettivi sono:

- Comprendere, valutare e gestire le aspettative in modo tale che tutti i requisiti del proponente vengano rispettati;
- Riuscire ad impostare chiare metriche di qualità e documentare tutte le procedure necessarie al completamento del progetto secondo le aspettative richieste;
- Sviluppare il prodotto seguendo queste metriche, così da consegnarlo entro tempistiche prestabilite, rispettando il budget scelto e secondo i requisiti e le aspettative del proponente.

#### 3.3.2 Piano di Qualifica

Per rispettare tutti gli obiettivi di questo processo si utilizza il Piano di Qualifica, un documento dove vengono descritte le metriche e le modalità utilizzate per valutare la qualità dei prodotti e dei processi, al cui interno sono presenti:

- Definizione delle metriche per analizzare la qualità del prodotto;
- Definizione di un sistema per il controllo della qualità durante tutto il ciclo di vita del progetto;
- Definizione di un piano di miglioramento per analizzare le prestazioni di qualità e identificare attività per migliorare le stesse.

#### 3.3.3 Denominazione metriche

Le metriche utilizzate vengono definite come segue:

**M[Tipologia].[Titolo]**

Dove:

- Tipologia:
  - PC: Per processo;
  - PD: Per prodotto.
- Titolo: Nome della metrica utilizzata.

## 3.4 Verifica

### 3.4.1 Scopo aspettative e descrizione

L'obiettivo è stabilire il metodo di verifica per garantire l'assenza di errori durante lo sviluppo del prodotto, la redazione della documentazione e il rispetto dei requisiti specificati. Le aspettative del gruppo *SWEt16* riguardo all'utilizzo di questo processo includono:

- Verificare ogni fase seguendo criteri definiti, coerenti e adattabili in caso di necessità;
- Condurre una verifica attenta per garantire il successo durante la fase di validazione;
- Automatizzare il più possibile le attività svolte durante il processo di verifica;
- Rispettare gli obiettivi di copertura indicati nel piano di qualifica.

Il processo di verifica si svolge in due forme:

- **Analisi statica:** Non richiede l'esecuzione dell'oggetto di verifica;
- **Analisi dinamica:** Richiede l'esecuzione dell'oggetto di verifica.

### 3.4.2 Analisi statica

L'analisi statica implica l'esame del codice prima della sua esecuzione e assicura che il software soddisfi i requisiti e le specifiche indicate.

Poiché non richiede l'esecuzione dell'oggetto in esame, questo tipo di analisi non si applica solo al codice, ma anche alla documentazione. L'analisi statica comprende due tecniche principali:

- **Walkthrough:** Questa tecnica coinvolge una revisione generale alla ricerca di errori senza presupposti specifici. Si articola nelle seguenti attività:
  - Pianificazione;
  - Lettura;
  - Discussione;
  - Correzione degli errori.
- **Inspection:** Questa tecnica comporta una revisione mirata alla ricerca di errori conosciuti e si suddivide nelle seguenti attività:
  - Pianificazione;
  - Definizione di una lista di controllo;
  - Lettura;
  - Correzione degli errori.

### 3.4.3 Analisi dinamica

L'analisi dinamica è applicabile principalmente al prodotto software poiché implica l'esecuzione di test, cioè prove sul codice in esecuzione. Un test ben definito deve:

- Essere ripetibile, cioè, dato un determinato input, si deve ottenere sempre lo stesso output per ogni prova eseguita;
- Specificare l'ambiente di esecuzione;
- Identificare input e output richiesti;
- Fornire informazioni utili sui risultati dell'esecuzione.

L'automatizzazione sarà realizzata mediante strumenti dedicati non ancora definiti dal gruppo.

#### **3.4.4 Verifica della documentazione**

La revisione della documentazione comprende:

- Verifica dell'ortografia e della grammatica;
- Accertamento dell'adeguato rispetto delle norme tipografiche e di formattazione stabilite, descritte nella sezione 3.1.6 di questo documento;
- Valutazione della coerenza e della rilevanza dei contenuti redatti.

### **3.5 Validazione**

#### **3.5.1 Descrizione, scopo ed aspettative**

La validazione del software è un processo successivo alla fase di verifica che permette di assicurarsi che il software soddisfi i requisiti del committente predefiniti e specificati, e le richieste e aspettative del proponente.

In caso di buona verifica durante tutta la fase di sviluppo, allora il processo di validazione avrà esito positivo.

Questo garantisce che il prodotto finale sia in linea rispetto alle aspettative.

Le aspettative del gruppo nell'applicazione di questo processo sono:

- Rilevare possibili errori ignorati o trascurati durante la fase di verifica;
- Assicurarsi che il prodotto finale soddisfi i requisiti specificati all'interno dell'Analisi dei Requisiti.



## 4 Processi organizzativi

### 4.1 Gestione dei Processi

#### 4.1.1 Descrizione, scopo ed aspettative

Lo scopo del processo è la stesura del documento denominato Piano di Progetto, utilizzato dal gruppo per l'organizzazione e la gestione dei ruoli.

In particolare le attività previste dal processo organizzativo di gestione dei processi sono:

- Assegnazioni dei ruoli e dei compiti;
- Comunicazione interne/esterne;
- Incontri interni/esterni;
- Strumenti di coordinamento;
- Strumenti di versionamento;
- Analisi dei Rischi e loro mitigazione.

Le aspettative sono invece:

- Ottenere un'organizzazione efficace tra i membri del gruppo;
- Adottare un buon *Way of Working*;
- Ottenere un'equa distribuzione dei ruoli, attraverso un'efficace rotazione degli stessi.

#### 4.1.2 Ruoli di progetto

Il Responsabile di Progetto si occupa di suddividere i ruoli tra i membri del gruppo, garantendo che ognuno di essi assuma nel corso del progetto almeno una volta ogni ruolo.

Vengono di seguito descritti i singoli ruoli.

##### 4.1.2.1 Responsabile di progetto

Il suo compito consiste nel garantire lo svolgimento delle attività pianificate entro i tempi e le modalità previste dal gruppo.

Si occupa anche delle comunicazioni esterne con committenti e proponenti.

Le sue responsabilità sono:

- Approvare la documentazione nella fase finale del processo di verifica;
- Gestire l'assegnazione dei compiti agli altri membri del gruppo;
- Organizzare il lavoro in modo da minimizzare la probabilità che si verifichino problemi.

#### **4.1.2.2 Amministratore di progetto**

È la figura che si occupa della gestione degli strumenti necessari all'ambiente di lavoro, le sue responsabilità sono:

- Gestire il sistema di archiviazione e versionamento di documentazione e codice;
- Gestire il sistema di configurazione del prodotto;
- Mantenere un ambiente di sviluppo efficiente, fornendo strumenti adeguati ai componenti del gruppo;
- Salvaguardare la documentazione di progetto;
- Redigere ed attuare i piani e le procedure per la gestione della qualità.

#### **4.1.2.3 Analista**

Il suo compito consiste nell'individuare, analizzare e documentare i servizi che il sistema deve fornire.

Le sue responsabilità sono:

- Studiare il problema ed il relativo contesto applicativo;
- Comprendere il problema e definire la complessità ed i requisiti;
- Redigere il documento denominato Analisi dei Requisiti.

#### **4.1.2.4 Progettista**

Il suo compito consiste nel definire la struttura architeturale del sistema. È responsabile delle attività di progettazione, le quali devono portare alla realizzazione di un prodotto che soddisfa i requisiti individuati dagli analisti.

Le sue responsabilità sono:

- Studiare l'architettura più adatta al prodotto da realizzare;
- Garantire la qualità del prodotto;
- Identificare l'architettura ad alto livello del sistema e l'architettura a livello di componenti del sistema.

#### **4.1.2.5 Programmatore**

Il suo compito consiste nella scrittura del codice richiesto dallo svolgimento del progetto.

Le sue responsabilità sono:

- Scrivere codice pulito e facile da mantenere, rispettando le Norme di Progetto;
- Realizzare gli strumenti per la verifica e la validazione del software.

#### 4.1.2.6 Verificatore

Il suo compito consiste nella verifica del lavoro svolto dagli altri componenti del progetto, sulla base delle proprie conoscenze tecniche, esperienza e conoscenza delle norme.

Le sue responsabilità sono:

- Controllare che vengano rispettate le norme di progetto, inserite all'interno di questo documento;
- Verificare la conformità dei prodotti ai requisiti funzionali e di qualità;
- Segnalare eventuali anomalie riscontrate nella verifica in modo che vengano corrette tempestivamente.

#### 4.1.3 Procedure

##### 4.1.3.1 Gestione delle comunicazioni

Per il coordinamento e le comunicazioni durante l'intera durata del progetto il gruppo *SWEet16* adotterà le seguenti procedure:

- **Comunicazione interna:** Coinvolge tutti i membri del gruppo;
- **Comunicazione esterna:** Coinvolge proponente e/o committenti.

##### 4.1.3.1.1 Comunicazioni interne

Le comunicazioni interne avvengono tramite le applicazioni:

- **Telegram<sup>G</sup>:** Permette una rapida comunicazione tra i membri del gruppo e viene utilizzato per risolvere veloci dubbi o organizzare incontri interni;
- **Discord<sup>G</sup>:** Utilizzato per comunicazione tramite canale vocale.

##### 4.1.3.1.2 Comunicazioni esterne

Le comunicazioni esterne vengono gestite dal Responsabile di Progetto tramite l'utilizzo dei seguenti canali:

- **Posta elettronica:** Tramite l'indirizzo del gruppo `sweet16.unipd@gmail.com`;
- **Telegram<sup>G</sup>:** Utilizzato per avere un canale di comunicazione diretto con il proponente per organizzare incontri periodici.

##### 4.1.3.2 Gestione degli incontri

##### 4.1.3.2.1 Incontri interni

Negli incontri interni i partecipanti sono solo i membri che compongono il gruppo.

Questi incontri vengono svolti almeno una volta a settimana, il Martedì sera o il Giovedì sera,

in base alla disponibilità dei membri del gruppo.

In caso servisse più di un incontro per quella settimana questi possono anche essere fatti nel fine settimana.

Un incontro può essere richiesto da un qualsiasi membro del gruppo al Responsabile di Progetto che poi l'organizzerà in base alla disponibilità degli altri membri.

Una volta fissata la data e l'ora queste vengono comunicate tramite il gruppo *Telegram*<sup>G</sup>.

La modalità d'incontro è sempre virtuale e non fisica date le non poche difficoltà che ci sarebbero, sia in termini di distanza fisica che di orario.

Come piattaforma per gli incontri viene utilizzato *Discord*<sup>G</sup> che permette di avere un canale di comunicazione vocale affidabile e veloce.

Ogni riunione sincrona viene così svolta:

- Precedentemente all'incontro il Responsabile stila un ordine del giorno sui principali punti da vedere durante la riunione;
- Il Responsabile decide chi scriverà il Verbale Interno per l'incontro, questa persona si occuperà di prendere appunti sugli argomenti trattati;
- Viene fatta una discussione sul lavoro effettuato da ogni membro del gruppo dall'ultimo incontro sincrono;
- Viene fatta una discussione sugli eventuali dubbi che sono emersi dall'ultimo incontro;
- Si effettua una pianificazione delle prossime attività da svolgere entro il prossimo incontro;
- Se l'incontro viene effettuato il Giovedì sera, si scrive anche il Diario di Bordo per il giorno successivo.

Alla fine di ogni incontro la persona designata si occupa della stesura del Verbale Interno fornendo una breve descrizione sui punti trattati nell'incontro appena svolto. Quest'ultimo dovrà poi essere approvato dal Responsabile.

#### **4.1.3.2.2 Incontri esterni**

Negli incontri esterni i partecipanti sono i membri del gruppo ed il proponente o il committente. Questi incontri vengono richiesti dal Responsabile di Progetto se vi è la necessità di chiarimento di una problematica riscontrata da parte del gruppo.

Il Responsabile provvede a formalizzare una richiesta all'entità esterna tramite i canali di comunicazione preposti (canale *Telegram*<sup>G</sup> per il proponente ed email per il committente).

In caso di richiesta accettata, verranno comunicate ai membri del gruppo, tramite canali di comunicazione interni, le informazioni riguardanti data, ora e le modalità dell'incontro.

In occasione di ogni incontro esterno viene redatto un verbale da un redattore scelto dal Responsabile.

Il contenuto della riunione deve essere riportato nel Verbale Esterno corrispondente ed approvato dal responsabile e dalla figura esterna (proponente o committente).

#### 4.1.3.3 Rotazione dei ruoli

È prevista una rotazione dei ruoli a cadenza periodica da parte del gruppo *SWEet16*. L'attribuzione dei ruoli viene svolta secondo i seguenti criteri:

- Equità;
- Assenza di conflitti;
- Continuità.

#### 4.1.3.4 Gestione dei compiti e delle task

Per la suddivisione dei compiti viene utilizzato GitHub come *ITS<sup>G</sup>*, in particolare questo è il ciclo di vita di una *issue<sup>G</sup>*:

- **Individuazione:** Viene individuata un'attività a seguito di un verbale interno o esterno, o da un commento su una pull request;
- **Creazione:** La *issue<sup>G</sup>* viene creata fornendo una descrizione testuale e applicando le *label<sup>G</sup>* corrette;
- **Assegnazione:** Il Responsabile di Progetto assegna la *issue<sup>G</sup>* ad uno o più componenti del gruppo;
- **Completamento:** L'attività viene completata da chi la sta svolgendo, vengono inoltre assegnati uno o più Verificatori per il punto successivo;
- **Verifica:** Un Verificatore effettua un controllo di qualità sulle azioni effettuate, può eventualmente lasciare dei commenti se non ritiene l'attività completata;
- **Accettazione:** A seguito della conferma da parte di un Verificatore il Responsabile effettua a sua volta un controllo e comunica se l'esito dell'attività è positivo o meno.

## 4.2 Strumenti

Sezione contenente gli strumenti utilizzati per i processi organizzativi che hanno permesso di attuare in modo pratico ed efficace il lavoro, questi sono:

- **Github:** Piattaforma di condivisione di codice, utilizzata per il versionamento del software e per la gestione delle issue.  
L'*ITS<sup>G</sup>* e il meccanismo di versionamento permettono di tener traccia delle modifiche apportate nel tempo in modo automatico e con una grande quantità di informazioni utili.

<https://github.com>

- **Discord:** Principale strumento utilizzato dal gruppo per le comunicazioni interne sincrone. Il gruppo ha adibito un server suddiviso in due canali:
  - **Canale Vocale:** Canale principale utilizzato durante gli incontri interni;
  - **Canale Testuale:** Canale utilizzato per la condivisione di risorse durante gli incontri interni.

<https://discord.com>

- **Telegram:** Strumento di comunicazione asincrona utilizzato da tutti i membri del gruppo per comunicazioni di varia natura. Utilizzato inoltre per comunicazioni veloci con il *proponente*<sup>G</sup>;

<https://telegram.org/>

- **Google Drive:** Strumento utilizzato come directory condivisa per stesura di bozze o veloci appunti. Vengono inoltre salvati al suo interno di Diari di Bordo:

<https://drive.google.com/>

- **Microsoft Teams:** Piattaforma di videochiamata utilizzato per comunicazione esterne con il proponente o per formazione:

<https://teams.microsoft.com/>

### 4.3 Formazione

Ogni membro del gruppo è responsabile della propria formazione circa l'utilizzo degli strumenti decisi per lo sviluppo.