



# Norme di Progetto

7DOS - 5 Dicembre 2018

## Informazioni sul documento

<b>Versione</b>	1.0.0
<b>Responsabile</b>	Nicolò Tartaggia
<b>Verifica</b>	Giacomo Barzon Andrea Trevisin
<b>Redazione</b>	Lorenzo Busin Marco Costantino Michele Roverato Giovanni Sorice
<b>Stato</b>	Approvato
<b>Uso</b>	Interno
<b>Destinato a</b>	Prof.Tullio Vardanega Prof.Riccardo Cardin 7DOS
<b>Email</b>	<a href="mailto:7dos.swe@gmail.com">7dos.swe@gmail.com</a>

## Descrizione

Questo documento descrive le regole, gli strumenti e le convenzioni adottate durante la realizzazione del progetto *G&B*.

## Diario delle modifiche

Modifica	Autore	Ruolo	Data	Versione
<i>Approvazione del documento</i>	Nicolò Tartaggia	Responsabile	2018-12-05	1.0.0
<i>Verifica del documento</i>	Andrea Trevisin	Verificatore	2018-12-04	0.9.0
<i>Verifica del documento</i>	Michele Roverato	Verificatore	2018-12-02	0.8.0
<i>Completamento stesura processi organizzativi</i>	Marco Costantino	Analista	2018-12-01	0.7.0
<i>Completamento stesura processi di supporto</i>	Lorenzo Busin	Analista	2018-12-01	0.6.0
<i>Completamento stesura processi primari</i>	Giovanni Sorice	Analista	2018-12-01	0.5.0
<i>Sviluppo stesura processi primari</i>	Giovanni Sorice	Analista	2018-11-29	0.4.3
<i>Sviluppo stesura processi organizzativi</i>	Michele Roverato	Analista	2018-11-23	0.4.2
<i>Sviluppo stesura processi di supporto</i>	Lorenzo Busin	Analista	2018-11-27	0.4.1
<i>Inizio stesura processi di supporto</i>	Lorenzo Busin	Analista	2018-11-26	0.3.0
<i>Inizio stesura processi organizzativi</i>	Marco Costantino	Analista	2018-11-26	0.2.0
<i>Inizio stesura processi primari</i>	Giovanni Sorice	Analista	2018-11-26	0.1.0
<i>Stesura della sezione Introduzione</i>	Giovanni Sorice	Analista	2018-11-26	0.0.2
<i>Stesura dello scheletro del documento</i>	Lorenzo Busin	Analista	2018-11-26	0.0.1

---

# Indice

<b>1</b>	<b>Introduzione</b>	<b>5</b>
1.1	Scopo del documento	5
1.2	Glossario	5
1.3	Riferimenti	5
1.3.1	Normativi	5
1.3.2	Informativi	5
<b>2</b>	<b>Processi primari</b>	<b>6</b>
2.1	Fornitura	6
2.1.1	Attività	6
2.1.1.1	Studio di Fattibilità	6
2.1.1.2	Piano di Progetto	7
2.1.1.3	Piano di Qualifica	7
2.2	Sviluppo	7
2.2.1	Attività	8
2.2.1.1	Analisi dei requisiti	8
2.2.1.1.1	Scopo	8
2.2.1.1.2	Casi d'uso	8
2.2.1.1.3	Requisiti	9
2.2.1.1.4	UML	9
2.2.1.2	Progettazione	10
2.2.1.2.1	Scopo	10
2.2.1.2.2	Attività	10
2.2.1.3	Codifica	10
2.3	Strumenti	11
2.3.1	Creazione diagrammi UML	11
2.3.2	Diagrammi di Gantt e consuntivi	11
2.3.3	Tracciamento dei requisiti	11
<b>3</b>	<b>Processi di supporto</b>	<b>12</b>
3.1	Documentazione	12
3.1.1	Fasi di sviluppo	12
3.1.2	Template	12
3.1.3	Struttura dei documenti	12
3.1.3.1	Frontespizio	12
3.1.3.2	Diario delle modifiche	13
3.1.3.3	Indice	13
3.1.3.4	Intestazione	13
3.1.3.5	Piè di pagina	13
3.1.4	Norme tipografiche	14
3.1.4.1	Stile del testo	14
3.1.4.2	Elenchi puntati	14
3.1.4.3	Note a piè di pagina	14
3.1.4.4	Formati comuni	14

3.1.5	Elementi grafici . . . . .	15
3.1.5.1	Immagini . . . . .	15
3.1.5.2	Tabelle . . . . .	15
3.1.6	Nomenclatura dei documenti . . . . .	15
3.1.7	Classificazione dei documenti . . . . .	15
3.1.7.1	Documenti informali . . . . .	15
3.1.7.2	Documenti formali . . . . .	15
3.1.7.3	Verbalì . . . . .	16
3.1.8	Sigle usate . . . . .	16
3.1.9	Glossario . . . . .	16
3.1.10	Strumenti di supporto . . . . .	17
3.1.10.1	L <sup>A</sup> T <sub>E</sub> X . . . . .	17
3.1.10.2	TexStudio . . . . .	17
3.2	Verifica . . . . .	17
3.2.1	Analisi dei processi . . . . .	17
3.2.1.1	Controllo delle metriche . . . . .	17
3.2.1.2	Analisi grafico PDCA . . . . .	17
3.2.2	Analisi dei documenti . . . . .	18
3.2.2.1	Controllo ortografico . . . . .	18
3.2.2.2	Controllo del periodo . . . . .	18
3.2.2.3	Rispetto delle norme di progetto . . . . .	18
3.2.2.4	Verifica sull'indice Gulpease . . . . .	18
3.2.2.5	Miglioramento del processo di verifica . . . . .	18
<b>4</b>	<b>Processi organizzativi . . . . .</b>	<b>19</b>
4.1	Gestione del progetto . . . . .	19
4.1.1	Ruoli di progetto . . . . .	19
4.1.1.1	Analista . . . . .	19
4.1.1.2	Progettista . . . . .	19
4.1.1.3	Programmatore . . . . .	20
4.1.1.4	Verificatore . . . . .	20
4.1.1.5	Amministratore . . . . .	20
4.1.1.6	Responsabile . . . . .	20
4.1.2	Sistema di ticketing . . . . .	20
4.1.2.1	Piattaforma . . . . .	20
4.1.2.2	Gestione dei task . . . . .	21
4.1.2.3	Norme sui task . . . . .	21
4.2	Gestione delle comunicazioni . . . . .	21
4.2.1	Comunicazioni interne al gruppo . . . . .	21
4.2.2	Comunicazioni esterne . . . . .	21
4.3	Gestione degli incontri . . . . .	22
4.3.1	Incontri interni . . . . .	22
4.3.2	Incontri esterni . . . . .	22
4.4	Strumenti di gestione e coordinamento . . . . .	22
4.4.1	Discord . . . . .	22
4.4.2	Github . . . . .	22

4.4.3	Git . . . . .	23
4.4.4	Client Git . . . . .	23
4.5	Norme di versionamento . . . . .	23
4.5.1	Norme sui file . . . . .	23
4.5.2	Norme sui commit . . . . .	23
4.5.3	Norme su branching e merging . . . . .	23
4.6	Formazione . . . . .	24

# 1 Introduzione

## 1.1 Scopo del documento

Il presente documento descrive e fissa tutte le norme, le convenzioni e gli strumenti che verranno adottati dal nostro team per assicurare un *modus operandi* comune a tutti i membri nello sviluppo del *progetto<sub>g</sub>*. Questo suppone che tutti i componenti del gruppo abbiano preso visione del documento e ne abbiano concordato e accettato i modi per garantire la massima omogeneità e collaborazione per tutto il progetto.

## 1.2 Glossario

Per rendere la lettura del documento più semplice, chiara e comprensibile viene allegato il documento *Glossario v1.0.0* nel quale sono contenute le definizioni dei termini tecnici, dei vocaboli ambigui, degli acronimi e delle abbreviazioni. La presenza di un termine all'interno del Glossario è segnalata con una "g" posta come pedice (esempio: *Glossario<sub>g</sub>*).

## 1.3 Riferimenti

### 1.3.1 Normativi

- **ISO/IEC 12207:**  
[https://www.math.unipd.it/~tullio/IS-1/2009/Approfondimenti/ISO\\_12207-1995.pdf](https://www.math.unipd.it/~tullio/IS-1/2009/Approfondimenti/ISO_12207-1995.pdf);
- Da verbali .

### 1.3.2 Informativi

- **Norme di Progetto:** *Norme di Progetto v1.0.0*.
- **Piano di Progetto:** *Piano di Progetto v1.0.0*.
- **Grafana<sub>g</sub> Code Styleguide:**  
<http://docs.grafana.org/plugins/developing/code-styleguide/>;
- **Angular TypeScript<sub>g</sub> Code Styleguide:**  
<https://angular.io/guide/styleguide>;
- **Software Engineering - Ian Sommerville- Tenth Edition (2015).**

## 2 Processi primari

### 2.1 Fornitura

Il fine di questa sezione è quello di definire le norme che i membri del gruppo 7DOS sono invitati a rispettare con l'obiettivo di proporsi e diventare fornitori nei confronti dell'azienda proponente Zucchetti s.r.l. e dei committenti Prof. Tullio Vardanega e Prof. Riccardo Cardin per quanto concerne il prodotto *G&B*. Per raggiungere questa meta nel miglior modo possibile, abbiamo intenzione di collaborare in modo *efficiente<sub>g</sub>* e *efficace<sub>g</sub>* con i referenti dell'azienda. I punti fondamentali che verranno affrontati insieme al proponente saranno:

- Determinare gli aspetti cruciali al fine di soddisfare l'azienda proponente;
- Concordare la qualifica del prodotto;
- Determinare vincoli sui processi e sui requisiti;
- Stimare i costi del prodotto finale.

#### 2.1.1 Attività

##### 2.1.1.1 Studio di Fattibilità

Successivamente alla presentazione dei Capitolati d'appalto, ogni componente del gruppo ha svolto un'attenta analisi delle proposte presentate per poi decidere durante la riunione del 27 novembre 2017 il capitolato per il quale partecipare alla gara d'appalto. In seguito, gli analisti hanno svolto ulteriori analisi dei rischi e delle opportunità di ogni capitolato che sono sfociate nella redazione del Documento di fattibilità v 1.0.0. Nel documento vi troviamo le motivazioni che hanno portato il nostro gruppo a favorire la scelta del prodotto per cui proporci come fornitori. Inoltre, si riportano, per ogni capitolato, le seguenti informazioni:

- **Descrizione:** riporta una breve sintesi del prodotto da sviluppare;
- **Studio del dominio:** riporta un'analisi del dominio applicativo, in cui vi è una più corposa descrizione del prodotto da sviluppare con l'aggiunta di una generale contestualizzazione, e un'analisi del dominio tecnologico, in cui vengono elencate le maggiori tecnologie coinvolte per ogni prodotto secondo la descrizione del capitolato e dalle esperienze pregresse dei componenti del gruppo;
- **Valutazione finale:** vi si può trovare in breve la motivazione della scelta presa per ogni capitolato in base a ciò che è stato riportato nelle tre sezioni precedenti;
- **Valutazione generale:** composta dagli aspetti positivi e dagli aspetti negativi trovati discussi dal gruppo riguardo al capitolato in esame.

### 2.1.1.2 Piano di Progetto

La redazione di un *piano<sub>g</sub>* da seguire durante la realizzazione del progetto spetta al Responsabile, aiutato nelle scelte dagli Amministratori. Il documento dovrà coprire le seguenti tematiche:

- **Analisi dei rischi:** riporta una dettagliata analisi dei rischi che si potrebbero incontrare durante la realizzazione del progetto, determinandone, in base alle conoscenze pregresse e alle nuove acquisite, la probabilità che essi accadano e la loro gravità. Inoltre, quando possibile, verranno analizzati i possibili metodi per affrontarli;
- **Pianificazione:** viene presentata una pianificazione delle *attività<sub>g</sub>* da svolgere nel corso del progetto, fornendo delle scadenze temporali il più possibile precise e veritiere;
- **Suddivisione risorse:** riporta la suddivisione oraria delle risorse, tenendo conto dei ruoli ricoperti da tutti i membri. In questo modo si ha una visione chiara su come il gruppo sta procedendo e quante ore vengono impiegate per ogni attività;
- **Preventivo:** sulla base della pianificazione, viene stimata la quantità di lavoro necessaria per portare a termine ogni attività (e quindi ogni fase) del progetto, per arrivare infine ad avere una valutazione complessiva per tutto il progetto e proporre un preventivo finale con il costo del lavoro precedentemente stimato.

### 2.1.1.3 Piano di Qualifica

Il *compito<sub>g</sub>* di *verifica<sub>g</sub>* e *validazione<sub>g</sub>* viene svolto da parte dei Verificatori. Questa mansione verrà svolta secondo un preciso metodo che deve coprire le seguenti tematiche:

- **Metodo di verifica:** riporta le procedure di controllo sulla *qualità<sub>g</sub>* di *processo<sub>g</sub>* e di *prodotto<sub>g</sub>*, considerando i mezzi e le risorse a disposizione;
- **Misure e metriche:** presenta criteri oggettivi per i documenti, i processi e il software;
- **Gestione della revisione:** precisa nel dettaglio le metodologie di comunicazione delle procedure di controllo per la qualità di processo e delle anomalie;
- **Pianificazione del collaudo:** definisce dettagliatamente le metodologie di collaudo a cui sarà sottoposto il progetto realizzato;
- **Resoconto dell'attività di verifica:** riporta le metriche calcolate e il resoconto sul collaudo delle attività sottoposte a verifica e validazione.

## 2.2 Sviluppo

Il processo in questione affronta le attività ed i compiti svolti dal gruppo con l'obiettivo di sviluppare il software richiesto dal proponente. Per una corretta implementazione è fondamentale:

- realizzare un prodotto finale conforme alle richieste del proponente;
- realizzare un prodotto finale che soddisfa i test di verifica e validazione;



- fissare gli obiettivi di sviluppo;
- fissare i vincoli tecnologici e di design.

Inoltre, il gruppo ha deciso di seguire le linee guida dettate dallo standard  $ISO_g/IEC_g$  12207. Per questo motivo le attività alla base del progetto di sviluppo saranno le seguenti:

- Analisi dei Requisiti;
- Progettazione;
- Codifica.

## 2.2.1 Attività

### 2.2.1.1 Analisi dei requisiti

#### 2.2.1.1.1 Scopo

Determinare con precisione i requisiti del progetto ed elencarli in modo formale. Essi vengono estrapolati da varie fonti:

- documenti di specifica del *capitolato<sub>g</sub>*;
- incontri con l'azienda proponente;
- verbali interni ed esterni;
- casi d'uso.

#### 2.2.1.1.2 Casi d'uso

Ogni caso d'uso è descritto dalla seguente struttura:

- codice identificativo:

**UC {codice\_padre}.{codice\_figlio}**

- UC specifica che si tratta di un caso d'uso;
  - codice\_padre identifica univocamente i casi d'uso;
  - codice\_figlio è un numero progressivo che identifica i sottocasi.
- titolo;
  - diagramma UML;
  - attori;
  - scopo e descrizione;
  - condizione;
  - flusso base degli eventi;

- postcondizioni;
- inclusioni (se presenti);
- estensioni (se presenti).

#### 2.2.1.1.3 Requisiti

Ogni requisito è descritto dalla seguente struttura:

- Nome;
- Tipo;
- Importanza;
- Stato implementazione;
- Fonti.

Inoltre, a ciascun requisito corrisponde un codice identificativo così composto:

**R {importanza}.{tipo}.{identificativo}**

- R specifica che si tratta di un requisito ;
- importanza identifica la rilevanza del requisito e può assumere 3 valori:
  - 0: indica che il requisito è obbligatorio e il suo soddisfacimento dovrà necessariamente avvenire;
  - 1: indica che il requisito è desiderabile, cioè il suo soddisfacimento può portare maggiore completezza al sistema ma non è fondamentale per lo stesso;
  - 2: indica che il requisito è opzionale, e quindi la decisione di implementarlo o meno verrà presa dopo le dovute considerazioni;
- tipo distingue se si tratta di un requisito funzionale (F), di qualità (Q), di prestazione (P) o di vincolo (V);
- identificativo è un numero progressivo che identifica i sottocasi.

#### 2.2.1.1.4 UML

I diagrammi UML devono essere realizzati usando la versione del linguaggio *v2.0*

### 2.2.1.2 Progettazione

#### 2.2.1.2.1 Scopo

Esplícita concretamente una prima forma ad alto livello del design del software pensata per il progetto, determinandone le caratteristiche più in evidenza in modo da:

- **Comunicare con gli stakeholder** e dare delle informazioni chiare, così da intraprendere le prime discussioni sul progetto;
- **Analizzare il sistema** in modo esplicito nei primi passi del progetto può portare ad importanti considerazioni riguardo a performance, affidabilità e manutenibilità;
- Capire come il sistema è organizzato e come i componenti interoperano tra di loro sottolineando la possibilità di **riutilizzo in larga scala** di parte del codice, dato che si è notata una frequente ripetizione dei requisiti in un sistema complesso come quello affrontato dal nostro team.

#### 2.2.1.2.2 Attività

Il design del software non è composta da una singola attività, bensì da un insieme preciso di sotto-attività progettuali che portano al design finale. Le sotto-attività che il nostro gruppo ha scelto di svolgere sono:

- **Progettazione architetturale:** i progettisti identificheranno una struttura globale del sistema, quindi i *componenti<sub>g</sub>* principali, le loro relazioni e come saranno distribuiti;
- **Progettazione dell'interfaccia:** i progettisti definiranno le interfacce tra i *moduli<sub>g</sub>* di sistema. Ciò deve essere fatto assolutamente in modo non ambiguo dato che, grazie a queste specifiche, ogni componente potrà usare in modo appropriato le funzionalità degli altri componenti del progetto senza sapere l'effettiva implementazione;
- **Selezione e progettazione dei componenti:** i progettisti cercheranno dei componenti riutilizzabili e valuteranno se inserirli nel progetto, specificandone gli appropriati utilizzi e aggiustamenti, o se sarà più opportuno costruire dei nuovi componenti.

#### 2.2.1.3 Codifica

Le convenzioni stilistiche definite in *Grafana Plugin Code Styleguide<sub>g</sub>* e, più in particolare per il linguaggio Typescript, in *Angular TypeScript Styleguide<sub>g</sub>* verranno seguite dai programmatori per lo sviluppo dell'intero progetto.

Soltanto il Responsabile di progetto, dopo un'attenta analisi e valutazione, potrà ammettere modifiche alle convinzioni stabilite.

L'unica lingua ammessa per i nomi di variabili, classi e funzioni è l'inglese.

## 2.3 Strumenti

### 2.3.1 Creazione diagrammi UML

Per quanto concerne la creazione e la modellazione dei diagrammi UML per i casi d'uso, è stato deciso di utilizzare *Astah UML<sub>g</sub>*, data la sua facilità di utilizzo e compatibilità con UML 2.x. Per la successiva fase di modellazione delle classi, potrebbe verificarsi una scelta diversa riguardo al software da utilizzare.

### 2.3.2 Diagrammi di Gantt e consuntivi

Per quanto riguarda la creazione dei diagrammi di Gantt è stato deciso di utilizzare *GanttProject<sub>g</sub>* data la sua completezza, facilità di utilizzo e la possibilità di esportare i diagrammi sottoforma di *png<sub>g</sub>*. Inoltre, i file *.gan<sub>g</sub>* di GanttProject sono salvati sotto forma di *xml<sub>g</sub>*, quindi facilmente versionabili. Infine, per l'attuale realizzazione dei consuntivi è stato deciso di utilizzare *Microsoft Excel<sub>g</sub>* come strumento per creare i grafici.

### 2.3.3 Tracciamento dei requisiti

Per quanto concerne il tracciamento dei requisiti è stato deciso di utilizzare *nTask<sub>g</sub>*, data la sua comoda *board view<sub>g</sub>*, la possibilità di creare più feature anche nella versione gratuita, la possibilità di avere *risk tracking<sub>g</sub>* integrato e la creazione semi-automatica dei diagrammi di Gantt.

## 3 Processi di supporto

### 3.1 Documentazione

Questa sezione descrive in modo dettagliato le procedure adottate dal gruppo in merito alla redazione, verifica e approvazione di tutta la documentazione prodotta. Tali norme devono essere rispettate in modo tassativo, al fine di realizzare documenti formali, coerenti e non ambigui.

#### 3.1.1 Fasi di sviluppo

Ogni documento deve attraversare le seguenti fasi per essere considerato formale:

- **Redazione:** un documento si trova in questa fase dal momento in cui viene creato fino alla sua approvazione. I *Redattori* si occupano della stesura e della modifica delle sezioni che gli sono state assegnate dal *Responsabile di Progetto*;
- **Verifica:** un documento entra in questa fase al termine del lavoro dei *Redattori*. Il *Responsabile di Progetto* deve assegnare ai *Verificatori* la procedura di verifica e validazione del documento. In caso di esito positivo esso passa allo stato *Approvato*, nel caso contrario il *Responsabile di Progetto* affida ai *Redattori* il compito di apportare eventuali correzioni;
- **Approvazione:** un documento entra in questa fase una volta superata la fase di verifica in modo positivo ed è compito del *Responsabile di Progetto* approvarlo in maniera ufficiale.

#### 3.1.2 Template

Per uniformare la struttura dei documenti è stato creato un template *LaTeX<sub>g</sub>* che implementa la formattazione e l'impaginazione degli stessi. Il contenuto di ogni documento è costituito di più file, uno per ogni sezione, la cui stesura è stata incaricata ai *Redattori*.

#### 3.1.3 Struttura dei documenti

Tutta la documentazione deve rispettare la medesima struttura.

##### 3.1.3.1 Frontespizio

Questa sezione contiene tutti gli elementi che dovranno essere presenti nella prima pagina di ogni documento.

- **Logo del gruppo;**
- **Titolo del documento;**
- **Nome del gruppo;**
- **Data di approvazione;**
- **Informazioni sul documento:**

- Versione corrente;
- Nome e cognome del *Responsabile di Progetto*;
- Nome e cognome dei *Verificatori*;
- Nome e cognome dei *Redattori*;
- Destinazione d'uso;
- Destinatari del documento;
- Indirizzo email del gruppo.

- **Descrizione del documento.**

### 3.1.3.2 Diario delle modifiche

Questa sezione contiene le modifiche apportate al documento, organizzate in modo tabulare ed ordinate in modo decrescente dall'alto verso il basso secondo la versione dello stesso. Ogni colonna descrive le seguenti informazioni:

- **Modifica:** tipo e soggetto di ogni modifica;
- **Autore:** nome e cognome dell'autore;
- **Ruolo:** ruolo dell'autore in quel momento;
- **Data:** data di esecuzione;
- **Versione:** versione del documento.

### 3.1.3.3 Indice

Ogni documento, esclusi i verbali, contiene un indice, il quale consente una visione generale del suo contenuto, ordinato e numerato rispetto alle sezioni presenti. Ci possono essere al più tre tipologie: indice delle sezioni, indice delle tabelle e indice delle immagini.

### 3.1.3.4 Intestazione

- Logo del gruppo a sinistra;
- Titolo della sezione corrente a destra.

### 3.1.3.5 Piè di pagina

- Nome del documento e nome del gruppo a sinistra;
- Numero progressivo della pagina corrente a destra.

### 3.1.4 Norme tipografiche

#### 3.1.4.1 Stile del testo

- **Corsivo**: è usato per termini specifici o poco comuni, per indicare ruoli all'interno del progetto, per le citazioni e per i riferimenti ai documenti;
- **Grassetto**: è usato per evidenziare concetti e parole chiave;
- **Maiuscolo**: è usato per indicare gli acronimi;
- **Sottolineato**: è usato nel Glossario per evidenziare link a termini presenti nel documento che vengono impiegati nella descrizione di altri termini;
- **Azzurro**: è usato per indicare collegamenti ipertestuali.

#### 3.1.4.2 Elenchi puntati

Ogni punto dell'elenco deve avere la lettera maiuscola e terminare con il carattere punto e virgola, tranne l'ultimo che deve terminare con il carattere punto.

#### 3.1.4.3 Note a piè di pagina

In caso di presenza in una pagina interna di note da esplicitare, esse vanno indicate nella pagina corrente, in basso a sinistra. Ogni nota deve riportare un numero e una descrizione.

#### 3.1.4.4 Formati comuni

- **Orari: HH:MM**
  - **HH**: indica le ore scritte con due cifre;
  - **MM**: indica i minuti scritti con due cifre.
- **Date: AAAA-MM-GG**
  - **AAAA**: indica l'anno scritto con quattro cifre;
  - **MM**: indica il mese scritto con due cifre;
  - **GG**: indica il giorno scritto con due cifre.
- **Termini ricorrenti**
  - **Nomi propri**: vanno scritti usando il formato **Nome Cognome**;
  - **Ruoli di progetto**: vanno scritti in corsivo e con la lettera maiuscola;
  - **Nomi dei documenti**: vanno scritti con la lettera maiuscola;
  - **Riferimenti ai documenti**: vanno scritti in corsivo e con la lettera maiuscola.

### 3.1.5 Elementi grafici

#### 3.1.5.1 Immagini

Le immagini devono essere separate dal testo lasciando una spaziatura per facilitarne la lettura e vanno centrate orizzontalmente. Possono essere organizzate in modo affiancato e per ogni figura deve essere presente una breve didascalia, oltre ad un identificativo che permetta il relativo inserimento all'interno dell'indice delle immagini. I formati consentiti sono **PNG** e **JPG**.

#### 3.1.5.2 Tabelle

Le tabelle devono essere separate dal testo lasciando una spaziatura per facilitarne la lettura e vanno centrate orizzontalmente. Possono essere organizzate in modo affiancato e per ogni tabella deve essere presente una breve didascalia, oltre ad un identificativo che permetta il relativo inserimento all'interno dell'indice delle tabelle. L'intestazione di ogni colonna deve essere in grassetto ed avere la lettera maiuscola.

### 3.1.6 Nomenclatura dei documenti

Il formato usato per la nomenclatura dei documenti, tranne i *Verbali*, è il seguente:

**NomeDocumento\_vX.Y.Z**

- **NomeDocumento**: indica il nome del documento, scritto senza spazi e con la lettera maiuscola in ogni parola.
- **vX.Y.Z**: indica la versione del documento secondo il seguente criterio:
  - **X**: viene incrementato in seguito ad un'approvazione ufficiale del documento da parte del *Responsabile di Progetto* e ciò comporta l'azzeramento di Y e Z;
  - **Y**: viene incrementato in seguito ad un'azione di verifica o di stesura di una parte corposa del documento e ciò comporta l'azzeramento di Z;
  - **Z**: viene incrementato in seguito ad un'azione di stesura di una parte esigua del documento.

Il formato usato per la nomenclatura dei *Verbali* invece è **Verbale\_AAAA-MM-GG**.

### 3.1.7 Classificazione dei documenti

#### 3.1.7.1 Documenti informali

Un documento viene considerato informale se non è stato approvato dal *Responsabile di Progetto*, pertanto è concesso esclusivamente ad uso interno al gruppo.

#### 3.1.7.2 Documenti formali

Un documento viene considerato formale dopo aver superato con esito positivo l'attività di



verifica e in seguito alla sua approvazione da parte del *Responsabile di Progetto*, pertanto può essere destinato ad una distribuzione esterna al gruppo.

### 3.1.7.3 Verbali

Ogni verbale deve essere redatto dal segretario durante le riunioni, sia interne che esterne, tenute dal gruppo e deve rispettare il seguente contenuto:

- **Informazioni incontro:** informazioni generali riguardo la riunione descritte secondo la seguente struttura:
  - Luogo;
  - Data;
  - Ora;
  - Partecipanti del gruppo;
  - Partecipanti esterni.
- **Argomenti affrontati:** descrizione dei temi discussi dal gruppo ed eventuali decisioni prese a riguardo.

### 3.1.8 Sigle usate

- **AR:** Analisi dei Requisiti;
- **GL:** Glossario;
- **NdP:** Norme di Progetto;
- **PdP:** Piano di Progetto;
- **PdQ:** Piano di Qualifica;
- **SF:** Studio di Fattibilità;
- **RA:** Revisione di Accettazione;
- **RP:** Revisione di Progettazione;
- **RQ:** Revisione di Qualifica;
- **RR:** Revisione dei Requisiti.

### 3.1.9 Glossario

Il Glossario deve contenere tutti quei termini che possono risultare ambigui o che possono essere fraintesi. Le definizioni, elencate in modo alfabetico, devono essere chiare e concise. L'inserimento nel Glossario deve avvenire parallelamente alla stesura della documentazione. Deve essere segnalata soltanto la prima occorrenza del termine in un documento e deve essere impiegato l'apposito comando personalizzato che lo formatta in corsivo aggiungendo una lettera 'g' come pedice(eg. *Termine nel Glossario<sub>g</sub>*).

### 3.1.10 Strumenti di supporto

#### 3.1.10.1 $\text{\LaTeX}$

L'intera documentazione deve essere redatta usando il linguaggio di *markup*<sub>g</sub>  $\text{\LaTeX}$  che offre la possibilità di personalizzare comandi e variabili da usare all'interno dei documenti facilitandone la gestione e l'aggiornamento. Permette anche di dividere il contenuto di un documento dalla rispettiva struttura, suddividendolo in sezioni per facilitarne la stesura da parte dei *Redattori* e di mantenere la stessa formattazione in tutti i documenti mediante l'uso di un template comune.

#### 3.1.10.2 TexStudio

Per redigere i documenti in  $\text{\LaTeX}$  deve essere impiegato l'editor *TexStudio*<sub>g</sub>. L'ambiente mette a disposizione un'interfaccia per la scrittura dei documenti, organizzati secondo una struttura gerarchica e la visualizzazione dell'anteprima del risultato ottenuto in seguito alla compilazione avvenuta con successo e in caso contrario segnalando all'utente eventuali errori.

## 3.2 Verifica

Il processo di verifica ha lo scopo di esaminare un prodotto al fine di accertare che le attività produttive non abbiano introdotto errori sullo stesso, fornendo una prova oggettiva della sua correttezza o nel caso contrario, segnalando le eventuali problematiche riscontrate. Per questa prima parte del progetto il processo di verifica viene eseguito sulla documentazione.

### 3.2.1 Analisi dei processi

Descrizione delle procedure per verificare adeguatamente lo svolgimento dei processi.

#### 3.2.1.1 Controllo delle metriche

Alla conclusione di ogni fase del progetto, per ogni macro-attività , si devono calcolare i relativi indici. Al fine di avere un indice complessivo di fase deve essere inoltre calcolato il valore medio di tali indici.

#### 3.2.1.2 Analisi grafico PDCA

Per mezzo del grafico *PDCA*<sub>g</sub> della fase di progetto in esame, verranno tratte delle considerazioni riguardo lo svolgimento dei processi. Dal grafico si possono identificare in modo visuale, quindi generico e non numericamente *quantificabile*<sub>g</sub>, dei dati sulla fase in esame ed osservarne la tendenza:

- Errori di pianificazione, rappresentati da variazioni delle attività nello stato Plan;
- Velocità con cui il gruppo di lavoro porta avanti i processi tra i vari stati del ciclo PDCA.

### 3.2.2 Analisi dei documenti

Descrizione delle procedure da eseguire per verificare adeguatamente i documenti.

#### 3.2.2.1 Controllo ortografico

Per verificare la presenza di errori ortografici nella documentazione deve essere usato lo strumento di controllo ortografico offerto dall'editor *TexStudio* che integra i dizionari di *OpenOffice<sub>g</sub>* per segnalare potenziali errori presenti nel testo.

#### 3.2.2.2 Controllo del periodo

Per verificare la presenza di errori di sintassi, di parole grammaticalmente corrette ma presenti in un contesto sbagliato e di periodi difficili da comprendere è necessario l'intervento umano. Per questo motivo ogni documento deve essere sottoposto a verifica da parte dei *Verificatori* incaricati, secondo la strategia *walkthrough<sub>g</sub>*.

#### 3.2.2.3 Rispetto delle norme di progetto

La verifica del rispetto delle norme descritte in *Norme di Progetto v1.0.0* è compito dei *Verificatori* incaricati. Non essendo possibile impiegare strumenti automatici per verificare la corretta applicazione di tutte le norme spetta quindi ai *Verificatori* sottoporre i documenti a verifica, secondo la strategia *inspection<sub>g</sub>*.

#### 3.2.2.4 Verifica sull'indice Gulpease

Per ogni documento i *Verificatori* devono calcolare l'indice di leggibilità e se questo dovesse risultare troppo basso, dovrà essere eseguita la verifica del documento con l'obiettivo di ricercare frasi troppo prolisse o complesse. Il calcolo deve essere effettuato attraverso uno *script<sub>g</sub>* scritto in *Perl<sub>g</sub>* che trasforma i documenti dal formato .pdf al formato .txt per effettuare un calcolo più preciso, eliminando le tabelle ma non il loro contenuto.

#### 3.2.2.5 Miglioramento del processo di verifica

Per migliorare e ottimizzare il processo di verifica i *Verificatori* devono riportare gli errori riscontrati più frequentemente al fine di rendere più efficiente ed efficace la verifica di un documento, prestando maggiore attenzione agli errori più comuni.

## 4 Processi organizzativi

### 4.1 Gestione del progetto

I punti principali dell'attività di gestione di progetto sono:

- L'istanziamento dei processi di progetto;
- La pianificazione e la gestione dei compiti e delle attività che compongono i processi. Obiettivi di quest'attività sono:
  - Permettere l'analisi dei rischi;
  - Sviluppare una strategia di lavoro che faccia uso di *best practices*<sub>g</sub>;
  - Permettere una stima dei costi delle attività di progetto.
- La verifica delle attività e la loro eventuale modifica nell'ottica del miglioramento. Per garantire che la verifica sia efficace deve:
  - Essere supportata da strumenti informatici che riducano il carico di verifica sulla persona;
  - Far uso di metriche numeriche ben definite.

#### 4.1.1 Ruoli di progetto

Segue la lista dei ruoli di progetto che ogni membro dovrà ricoprire a rotazione. Le rotazioni andranno effettuate quando arrecheranno meno disagi alle attività di progetto e i ruoli saranno stabiliti casualmente con eventuali aggiustamenti per garantire che ogni membro possa assumere almeno una volta ogni ruolo.

##### 4.1.1.1 Analista

L'Analista si occupa di determinare e descrivere in maniera formale i requisiti del prodotto, siano essi impliciti o espliciti. Deve:

- Comunicare con il committente per determinare i requisiti del prodotto e i requisiti di dominio;
- Produrre uno Studio di Fattibilità;
- Produrre l'Analisi dei Requisiti.

##### 4.1.1.2 Progettista

Il Progettista fa uso dell'Analisi dei Requisiti per produrre un architettura che rispetti i requisiti definiti dall'analista. Deve:

- Produrre un'architettura che faccia uso di *best practices* e che agevoli *manutenzione*<sub>g</sub> e verifica;
- Assicurarsi che l'architettura rispetti i requisiti definiti dall'analista;

- Redigere la documentazione relativa all'architettura del prodotto e un manuale destinato al programmatore per l'implementazione dell'architettura progettata.

#### 4.1.1.3 Programmatore

Il Programmatore ha come compito quello di implementare l'architettura progettata dal progettista. Deve:

- Scrivere codice che sia documentato e che rispetti le direttive del progettista;
- Codificare i moduli necessari al testing del codice;
- Produrre il manuale utente.

#### 4.1.1.4 Verificatore

Il Verificatore si occupa delle attività di verifica. Deve:

- Verificare la qualità dei prodotti delle varie attività;
- Verificare che le attività siano svolte rispettando le norme di progetto stabilite;
- Redigere il Piano di Qualifica.

#### 4.1.1.5 Amministratore

L'Amministratore gestisce l'ambiente di lavoro del team. Deve:

- Determinare quali strumenti verranno usati dal team (sistema di versioning, *sistema di ticketing*<sub>g</sub>, strumenti per la comunicazione, etc.);
- Garantisce l'operatività dei sistemi informatici a supporto delle attività di progetto.

#### 4.1.1.6 Responsabile

Il Responsabile rappresenta il team nelle interazioni con il committente, si occupa di coordinamento interno, si assume la responsabilità delle scelte prese ed approva i prodotti dei processi. Deve:

- Gestire i processi di comunicazione interna ed esterna (incontri, comunicazioni con il committente, riunioni del team);
- Approvare i prodotti risultanti dei processi (documenti, altro);
- Coordinare il gruppo assegnando attività ai vari membri e scadenze e gestendo i rischi.

### 4.1.2 Sistema di ticketing

#### 4.1.2.1 Piattaforma

Per la gestione del sistema di ticketing viene utilizzato *nTask*<sub>g</sub>. La piattaforma permette di

creare  $task_g$  e di assegnarli ai membri del gruppo. Ogni task è caratterizzato da descrizione, data di inizio e data di fine. nTask permette inoltre di avere diverse visualizzazioni dei task assegnati: lista testuale, griglia, calendario.

#### 4.1.2.2 Gestione dei task

Vengono distinti 2 tipi di task:

- Pro-attivi: creati e assegnati dal Responsabile di progetto, organizzano lo svolgimento delle attività e la verifica;
- Reattivi: creati e assegnati dai Verificatori, organizzano la correzione o il *miglioramento* dei prodotti delle attività.

Qualora l'assegnatario di un task dovesse realizzare tale task non è completabile entro la data di fine, o se ci dovessero essere altri problemi con il task, avrà la responsabilità di segnalare il problema al responsabile di progetto che sarà tenuto a ripianificare la risoluzione del task.

#### 4.1.2.3 Norme sui task

Il nome del task deve essere nella forma [attività] nome. Il nome deve essere descrittivo del task stesso. Ogni task dovrà avere almeno un assegnatario.

### 4.2 Gestione delle comunicazioni

#### 4.2.1 Comunicazioni interne al gruppo

Le comunicazioni interne sono gestite con *Discord<sub>g</sub>*, un software di messaggistica istantanea e di *VoIP<sub>g</sub>* con altri membri di uno stesso server. In alternativa, ove necessario, gli utenti possono scambiarsi messaggi privati anche al di fuori di un server. Un server può essere creato gratuitamente, ed è possibile organizzare le conversazione creando vari canali vocali e testuali associati ad uno scopo e ad alcuni membri del team.

Oltre a qualche canale vocale per discutere di vari argomenti, i canali testuali principali sono: #general per le discussioni generali e i canali nella categoria "Notification Center" dove apparirà una notifica inviata da un bot in caso di evento rilevante, per esempio l'arrivo di una nuova e-mail o un commit nel repository.

Il maggior vantaggio rispetto a *Slack<sub>g</sub>* sta nell'essere un software con tutte le funzionalità di base disponibili gratuitamente, rendendo molto vantaggioso avere la possibilità di trovarsi a discutere con tutti gli altri membri del gruppo, cosa non possibile nella versione gratuita di Slack che offre solo chiamate 1-1.

#### 4.2.2 Comunicazioni esterne

Le comunicazioni esterne sono compito del Responsabile di progetto, che utilizzerà l'indirizzo e-mail del gruppo [7dos.swe@gmail.com](mailto:7dos.swe@gmail.com) facendo così apparire una notifica nel server Discord del team una volta inviata la mail.

## 4.3 Gestione degli incontri

### 4.3.1 Incontri interni

Il Responsabile di progetto ha il compito di decidere il giorno e la data di un incontro, discutendone con gli altri membri del team. Per comodità, è stato scelto il martedì come giorno di riferimento da proporre inizialmente ai membri del team.

Una volta raggiunto un accordo su un giorno con un numero consistente di partecipanti (almeno 4), tutti i membri che hanno accettato dovranno presentarsi in orario nel luogo prestabilito per la riunione, comunicando eventuali ritardi o impegni insormontabili. Ad ogni incontro, tutti devono partecipare attivamente alla discussione offrendo la loro opinione sui punti da decidere e discutere, mentre un membro del team scelto dal responsabile prenderà appunti sui contenuti da inserire appena possibile in un verbale.

Se non è possibile trovare un giorno in cui effettuare una riunione in un luogo fisico, si tenterà di scegliere un giorno in cui trovarsi a discutere su Discord.

### 4.3.2 Incontri esterni

Gli incontri esterni saranno anch'essi organizzati dal Responsabile di progetto, che si metterà in contatto con il committente o il proponente tramite email come descritto al punto 4.1.2 per accordarsi con essi sulla data, l'ora e il luogo dell'incontro. Una volta che questi ultimi saranno stati decisi e confermati da entrambe le parti, il responsabile avviserà i membri utilizzando Discord. In caso di impossibilità da una delle due parti ad organizzare un incontro fisico, si considererà l'opzione di un incontro utilizzando un servizio VoIP come *Skype<sub>g</sub>*, *Hangout<sub>g</sub>* o simili.

## 4.4 Strumenti di gestione e coordinamento

In questa sezione si riassumono brevemente i principali strumenti utilizzati nella comunicazione e nella collaborazione del team.

### 4.4.1 Discord

Discord è il software che si è scelto di utilizzare per le comunicazioni interne al gruppo, come spiegato al punto 4.2.1. Permette la creazione gratuita di un server nel quale comunicare con gli altri membri mediante canali testuali e vocali. Permette inoltre la creazione di un numero illimitato di *Webhooks<sub>g</sub>*, utili per integrare molti altri strumenti, come *Github<sub>g</sub>*.

### 4.4.2 Github

Github è il servizio di hosting scelto per la repository. Funziona utilizzando *git<sub>g</sub>*, un software di *versionamento<sub>g</sub>* descritto al punto 4.4.3 Oltre all'hosting del codice sorgente, offre altre caratteristiche utili, tra le quali:

- Un *Issue tracking System<sub>g</sub>*, con etichette e *milestone<sub>g</sub>*;
- Una lista dei *commit<sub>g</sub>* passati e delle relative modifiche ai file semplice da consultare;

- *Account Educational<sub>g</sub>* gratuito per gli studenti, il quale permette la creazione di repository private, normalmente creabili solo con un account a pagamento.

#### 4.4.3 Git

Git è un software open-source di controllo *versione<sub>g</sub>* distribuito. Viene usato con Github, in quanto è un servizio già conosciuto dai membri del team e molto comodo da utilizzare.

#### 4.4.4 Client Git

A seconda delle preferenze, ogni membro del team deciderà se usare come client *GitKraken<sub>g</sub>* o *GitHub Desktop<sub>g</sub>*. Entrambi sono client desktop che facilitano l'utilizzo di git, rendendo molto più semplice inviare le modifiche locali al repository remoto, o viceversa.

### 4.5 Norme di versionamento

Come già detto, il servizio di hosting scelto per la repository è GitHub. La repository è privata e sarà resa pubblica a fine progetto. Seguono le norme che ne riguardano gli aspetti principali.

#### 4.5.1 Norme sui file

I file devono rispettare le norme di nomenclatura già specificate in 3.1.6 Nomenclatura dei documenti. Tramite il file *.gitignore<sub>g</sub>* vengono esclusi i file intermedi di compilazione dei file latex. Gli unici file ammessi nella repository sono .tex .jpg .png .pdf.

#### 4.5.2 Norme sui commit

Ogni modifica sostanziale effettuata ai file dev'essere seguita da un commit. Il commento associato ad ogni commit deve essere nella forma: [nome file] descrizione. La descrizione deve essere concisa e quanto più esaustiva possibile. Ad ogni commit deve corrispondere un aggiornamento del diario delle modifiche.

#### 4.5.3 Norme su branching e merging

Le norme usate per *branching<sub>g</sub>* e *merging<sub>g</sub>* si rifanno al *Git Feature Branch Workflow<sub>g</sub>*: master è il branch in cui risiederà documentazione e codice verificati e approvati. Le modifiche o le nuove feature vanno implementate su un feature branch che saranno verificati prima di effettuare una pull request verso master. La pull request permette di non modificare master fintanto che non viene verificato il branch. Ogni nuovo feature branch deve:

- essere creato a partire da master;
- avere un nome che permetta di capire a quale feature è associato.

Queste norme saranno rivisitate all'inizio del processo di sviluppo per accertarsi che il feature branch workflow sia il più opportuno da usare. Una possibile alternativa con cui verrà confrontato è *GitFlow<sub>g</sub>*.



## 4.6 Formazione

La formazione ai fini del progetto è responsabilità di ogni singolo membro del gruppo. Qualora ai fini del progetto un membro del gruppo dovesse formarsi autonomamente su un argomento sconosciuto agli altri, sarà sua responsabilità informare il gruppo e se ritenuto opportuno formarlo secondo una delle seguenti modalità:

- Redigere un documento a scopo introduttivo che illustri le tematiche principali ed elenchi le risorse usate per la formazione. Tale modalità è da preferirsi alla successiva che verrà adottata se le circostanze dovessero ostacolare la redazione del documento;
- Formare il gruppo tramite un incontro a scopo formativo.