



## *Norme di progetto* Progetto Trustify

*pentasoftswe@gmail.com*

### Informazioni sul documento

Responsabile	Luca Marcato
Redattori	Nicola Lazzarin Marco Brugin Stefano Meneguzzo
Verificatori	Marco Rosin Pietro Lauriola Stefano Meneguzzo
Uso	Interno
Destinatari	Prof. Tullio Vardanega Prof. Riccardo Cardin
Versione	<i>v2.0.0</i>

### Sommario

Questo documento descrive le *norme di progetto* adottate dal gruppo *PentaSoft* nella realizzazione del progetto *Trustify*.

## Registro delle Modifiche

Versione	Data	Autore	Verificatore	Descrizione
2.0.0	2023/05/12	Luca Marcato (Responsabile)	Stefano Meneguzzo	Approvazione per il rilascio
1.0.2	2023/05/10	Stefano Meneguzzo (Amministratore)	Marco Brugin	Stesura § Processi di Codifica
1.0.1	2023/03/12	Stefano Meneguzzo (Amministratore)	Nicola Lazzarin	Aggiornamento § Processi di Supporto

Dalla presente modifica vengono messe in atto le correzioni agli errori illustrati nell'esito dell' $RTB_G$  del 2023-03-07.

Versione	Data	Autore	Ruolo	Descrizione
1.0.0	2023/02/05	Lazzarin Nicola	Responsabile	Approvazione per il rilascio
0.4.0	2023/01/19	Pietro Lauriola	Verificatore	Verifica complessiva del documento
0.3.0	2023/01/11	Nicola Lazzarin	Verificatore	Verifica aggiornamento del documento
0.2.2	2023/01/11	Luca Marcato	Analista	Aggiornamento del documento
0.2.1	2022/12/16	Marco Rosin	Analista	Stesura § Analisi dei requisiti
0.2.0	2022/11/24	Pietro Lauriola	Verificatore	Verifica § Processi primari
0.1.1	2022/11/22	Marco Brugin	Analista	Aggiornamento § Processi di supporto in seguito a VI.2022.11.19
0.1.0	2022/11/22	Marco Rosin	Verificatore	Verifica § Processi organizzativi
0.0.7	2022/11/22	Luca Marcato	Analista	Stesura § Processi di Sviluppo
0.0.6	2022/11/20	Luca Marcato	Analista	Stesura § Processi di Fornitura
0.0.5	2022/11/14	Nicola Lazzarin	Analista	Stesura § Processi Organizzativi
0.0.4	2022/11/14	Marco Brugin	Analista	Stesura § Processi di Supporto
0.0.3	2022/11/11	Marco Rosin	Analista	Stesura § Glossario § Riferimenti
0.0.2	2022/11/09	Nicola Lazzarin	Analista	Stesura § Scopo del documento

... Continua nella prossima pagina

· *Continua dalla pagina precedente*

Versione	Data	Autore	Ruolo	Descrizione
0.0.1	2022/11/09	Nicola Lazzarin	Analista	Creazione struttura documento

# Indice

<b>1</b>	<b>Introduzione</b>	<b>1</b>
1.1	Scopo del Documento . . . . .	1
1.2	Scopo del Capitolato . . . . .	1
1.3	Glossario . . . . .	1
1.4	Riferimenti . . . . .	1
1.4.1	Normativi . . . . .	1
1.4.2	Informativi . . . . .	1
<b>2</b>	<b>Processi Primari</b>	<b>2</b>
2.1	Introduzione . . . . .	2
2.2	Processo di fornitura . . . . .	2
2.2.1	Scopo . . . . .	2
2.2.2	Aspettative . . . . .	2
2.2.3	Descrizione . . . . .	2
2.2.4	Inizializzazione . . . . .	2
2.2.5	Preparazione della risposta . . . . .	2
2.2.6	Contratto . . . . .	3
2.2.7	Pianificazione . . . . .	3
2.2.8	Esecuzione e controllo . . . . .	4
2.2.9	Revisione e valutazione . . . . .	4
2.2.10	Consegna e completamento . . . . .	4
2.3	Processo di sviluppo . . . . .	5
2.3.1	Scopo . . . . .	5
2.3.2	Aspettative . . . . .	5
2.3.3	Descrizione . . . . .	5
2.3.4	Processi di implementazione . . . . .	5
2.3.5	Analisi dei requisiti . . . . .	5
2.3.5.1	Casi d'uso . . . . .	5
2.3.5.2	Requisiti . . . . .	6
2.3.5.3	Funzionalità e capacità del sistema . . . . .	6
2.3.6	Processi di codifica . . . . .	7
2.3.6.1	Scopo . . . . .	7
2.3.6.2	Aspettative . . . . .	7
2.3.6.3	Stile della codifica . . . . .	7
<b>3</b>	<b>Processi di Supporto</b>	<b>8</b>
3.1	Documentazione . . . . .	8
3.1.1	Ciclo di Vita dei Documenti . . . . .	8
3.1.2	Versionamento dei documenti . . . . .	8
3.1.3	Modifica documenti . . . . .	8
3.1.4	Suddivisione dei Documenti . . . . .	8
3.1.5	Template per tutti i documenti e presentazioni . . . . .	9
3.1.5.1	Nomi delle immagini . . . . .	9
3.1.5.2	Formati immagini nei documenti . . . . .	9
3.1.6	Verbali . . . . .	10
3.1.7	Norme di progetto . . . . .	10
3.1.8	Candidatura . . . . .	10
3.1.9	Glossario . . . . .	10
3.1.9.1	Aggiornamento del glossario . . . . .	10

3.1.10	Studio dei capitolati . . . . .	10
3.1.11	Piano di progetto . . . . .	10
3.1.12	Piano di qualifica . . . . .	11
3.1.13	Analisi dei requisiti . . . . .	11
3.1.13.1	Casi d'uso . . . . .	11
3.1.13.2	Requisiti . . . . .	11
3.1.14	Nomi dei Documenti . . . . .	12
3.1.15	Tipografia utilizzata . . . . .	12
3.1.16	Scrittura della data . . . . .	12
3.1.17	Sigle dei documenti: . . . . .	13
3.2	Revisione . . . . .	13
3.3	Risoluzione dei problemi . . . . .	13
3.4	Gestione della Qualità . . . . .	13
3.5	Verifica . . . . .	14
3.5.1	Svolgimento . . . . .	14
3.5.2	Come eseguirla . . . . .	14
3.5.3	Tecniche di Verifica . . . . .	15
3.5.4	Verifica della Documentazione . . . . .	15
3.5.5	Verifica del codice . . . . .	15
3.5.6	Test . . . . .	15
3.5.7	Strumenti . . . . .	16
3.6	Validazione . . . . .	16
3.7	Gestione dei rischi . . . . .	16
3.7.1	Codifica dei rischi . . . . .	16

<b>4</b>	<b>Processi Organizzativi</b>	<b>18</b>
4.1	Gestione organizzativa . . . . .	18
4.1.1	Scopo . . . . .	18
4.1.2	Aspettative . . . . .	18
4.1.3	Descrizione . . . . .	18
4.1.4	Ruoli di progetto . . . . .	18
4.1.5	Rotazione dei ruoli di progetto . . . . .	19
4.1.6	Gestione delle riunioni . . . . .	20
4.1.7	Orario di reperibilità . . . . .	20
4.2	Gestione delle infrastrutture interne . . . . .	20
4.2.1	Comunicazioni . . . . .	20
4.2.1.1	Comunicazioni interne . . . . .	20
4.2.1.2	Comunicazioni esterne . . . . .	21
4.2.2	Riunioni . . . . .	21
4.2.2.1	Riunioni interne . . . . .	21
4.2.2.2	Riunioni esterne . . . . .	21
4.2.3	Gestione degli strumenti di coordinamento . . . . .	21
4.2.3.1	Ticketing . . . . .	21
4.2.4	Gestione degli strumenti di versionamento . . . . .	22
4.3	Formazione . . . . .	22
4.3.1	Scopo . . . . .	22
4.3.2	Aspettative . . . . .	22
4.3.3	Modalità di formazione . . . . .	22
4.4	Pianificazione . . . . .	23
4.4.1	Scopo . . . . .	23
4.4.2	Aspettative . . . . .	23

4.4.3	Descrizione . . . . .	23
-------	-----------------------	----

# 1 Introduzione

## 1.1 Scopo del Documento

Lo scopo di questo documento è definire tutte le *best practices*<sub>G</sub> e il *way of working*<sub>G</sub> che, al fine di migliorare la qualità del lavoro svolto e l'organizzazione del gruppo, ogni componente è tenuto a rispettare. Il documento nel suo stato attuale non è da considerarsi definitivo, in quanto verrà integrato in modalità incrementale da modifiche e/o aggiornamenti derivanti dalle decisioni prese dall'intero gruppo. È compito del *responsabile* notificare il resto del gruppo in caso di modifiche al documento.

## 1.2 Scopo del Capitolato

## 1.3 Glossario

Alcuni dei termini utilizzati in questo documento potrebbero generare dei dubbi riguardo al loro significato, al fine di evitare tali ambiguità è necessario dar loro una definizione. Tali termini vengono contassegnati da una G maiuscola finale, se questa non compare in un titolo di sezione, a pedice della parola ed essa non verrà ripetuta più di una volta per paragrafo/sottosezione/sezione onde evitare fastidiose ripetizioni. La loro spiegazione è riportata nel *Glossario v2.0.0*.

## 1.4 Riferimenti

### 1.4.1 Normativi

- **Regolamento del progetto didattico:**

<https://www.math.unipd.it/tullio/IS-1/2022/Dispense/PD02.pdf>

- **Presentazione Capitolato C7 - Trustify:**

<https://www.math.unipd.it/tullio/IS-1/2022/Progetto/C7.pdf>

### 1.4.2 Informativi

- **ISO/IEC 12207:**

[https://en.wikipedia.org/wiki/ISO/IEC\\_12207](https://en.wikipedia.org/wiki/ISO/IEC_12207)

- **ISO/IEC 9126:**

[https://en.wikipedia.org/wiki/ISO/IEC\\_9126](https://en.wikipedia.org/wiki/ISO/IEC_9126)

## 2 Processi Primari

### 2.1 Introduzione

In questa sezione vengono elencate e descritte le attività e i compiti svolti per i processi di **fornitura** e **sviluppo** del progetto.

### 2.2 Processo di fornitura

#### 2.2.1 Scopo

Il processo di fornitura contiene e descrive le attività di **analisi della proposta**, lo **sviluppo di un piano di progetto** e le attività di **controllo** e **revisione** del prodotto software.

#### 2.2.2 Aspettative

Si prevede che al termine di questo processo siano stati raggiunti i seguenti obiettivi:

- Definizione delle procedure e delle risorse necessarie per gestire e garantire il progetto
- Creazione ed esecuzione dei piani progettuali attraverso la consegna del prodotto software

#### 2.2.3 Descrizione

Questo processo consiste nello svolgimento delle seguenti attività:

- Inizializzazione
- Preparazione della risposta
- Contratto
- Pianificazione
- Esecuzione e controllo
- Revisione e valutazione
- Consegna e completamento

#### 2.2.4 Inizializzazione

La prima attività da svolgere prevede di condurre un'analisi dei requisiti dei progetti esposti dalle aziende proponenti. L'attività riguarda la revisione dei documenti dei capitolati e come tale viene riportata nel documento di *Studio dei capitolati*.

#### 2.2.5 Preparazione della risposta

Prima di potersi candidare ad uno dei capitolati presentando la *Lettera di candidatura* il gruppo ha effettuato uno studio di fattibilità. Tale studio prevede:

1. la stima dei costi orari di ogni ruolo
2. la definizione dell'impegno orario per singolo membro, ottenuta dividendo il monte ore totale per il numero dei membri del gruppo
3. la stima del costo complessivo del progetto, ottenuta sommando il costo di ogni ruolo per il numero di ore di lavoro previste



4. la definizione delle *milestones<sub>G</sub>*
5. la previsione della data di consegna del prodotto, ottenuta sulla base del *calendario<sub>G</sub>* di impegno personale del gruppo

### 2.2.6 Contratto

La proposta di candidatura presentata viene revisionata e approvata. In caso contrario questa verrà momentaneamente sospesa per permettere al gruppo di apportare le modifiche necessarie alle stime precedentemente prodotte.

### 2.2.7 Pianificazione

Successivamente all'aggiudicazione del capitolato il gruppo, effettuando dei *brainstorming<sub>G</sub>*, ha stabilito le caratteristiche dei **§processi organizzativi** e dei **§processi di supporto**. Basandosi su tali requisiti, il gruppo ha continuato a sviluppare e documentare il *Piano di progetto* prevedendo che lo sviluppo del prodotto software andrà ad utilizzare le risorse interne disponibili.

Gli elementi considerati nel *Piano di progetto* vengono riportati di seguito.

#### Impegni e Scadenze

Viene definita una tabella di ripartizione oraria per ogni ruolo e il metodo di rotazione dei ruoli tra i membri del gruppo. Sono state inoltre definite le date delle tre revisioni: *RTB<sub>G</sub>*, *PB<sub>G</sub>* e *CA<sub>G</sub>*.

#### Analisi dei rischi

Sono state identificate e classificate le zone del progetto che involgono un potenziale rischio tecnico, di risorse o organizzativo. Ogni rischio è stato identificato, classificato in base alla probabilità che si verifichi e associato ad una *modalità di controllo* e ad un *piano di contingenza*.

#### Pianificazione dei periodi

Vengono stabiliti dei periodi di lavoro della durata massima di 20 giorni durante i quali vengono definite e schedate le varie *milestones<sub>G</sub>* da completare per arrivare ad ottenere una *baseline<sub>G</sub>* alla fine di ogni periodo. Affiancato alla pianificazione testuale vengono riportati i diagrammi di Gantt per la rappresentazione dell'arco temporale del progetto, suddiviso nei vari periodi.

#### Preventivo orario ed economico

Viene effettuata una stima delle ore personali di lavoro durante uno specifico periodo tenendo conto dei ruoli assunti da ogni membro. Infine utilizzando le ore totali di lavoro svolte per ogni ruolo in un determinato periodo, è possibile calcolare un preventivo economico basato sul costo orario di ogni ruolo.

#### Bilancio a consuntivo orario ed economico

Viene effettuato un calcolo delle ore personali di lavoro effettivamente utilizzate durante uno specifico periodo tenendo conto dei ruoli assunti da ogni membro. Infine utilizzando le ore totali di lavoro svolte per ogni ruolo in un determinato periodo, è possibile calcolare un bilancio economico basato sul costo orario di ogni ruolo.

#### Consuntivo e preventivo a finire

Considerando le ore e il budget effettivamente consumato in determinato periodo, o insieme di periodi, si va a calcolare il monte ore a disposizione ancora di ogni membro e il budget rimanente.

La pianificazione prevede anche lo svolgimento delle attività di:

- Apprendimento
- Verifica e il modo di interfacciarsi a tale attività
- Definizione di un documento di *controllo della qualità*

### 2.2.8 Esecuzione e controllo

Durante i vari stati del ciclo di vita del progetto il gruppo ha adottato e adotterà i seguenti metodi per monitorare lo stato di avanzamento del progetto:

- Implementare ed eseguire in maniera disciplinata e rigorosa quanto riportato dal *Piano di progetto* e dalle *Norme di progetto*
- Utilizzare in maniera disciplinata e rigorosa gli strumenti di gestione di progetto
- Eseguire in maniera disciplinata e rigorosa le attività di revisione, validazione e test seguendo le specifiche del *Piano di progetto*
- Effettuare delle riunioni interne a cadenza settimanale
- Seguire le scadenze definite dalle *milestones<sub>G</sub>*
- Effettuare delle riunioni esterne con l'azienda Sync Lab S.r.l. per assicurarsi che il progetto stia seguendo i *requisiti contrattuali<sub>G</sub>*
- Interfacciarsi a figure professionali e con maggiore esperienza come il Prof. Tullio Vardanega e il Prof. Riccardo Cardin

### 2.2.9 Revisione e valutazione

Il gruppo assieme all'azienda Sync Lab S.r.l. ha concordato un canale dedicato per poter coordinare le attività di revisione, interfacciamento e comunicazione interne. Il gruppo si prende l'onere di:

- Produrre il materiale informativo necessario per poter comunicare con l'azienda
- Effettuare revisioni di accettazione
  - Identificare la motivazione della revisione
  - Accordarsi con il proponente
  - Apportare e approvare la revisione
- Effettuare testing di accettazione prima del rilascio del prodotto
- Effettuare un controllo finale con l'azienda proponente
- Rendere disponibili all'azienda i *report<sub>G</sub>* di valutazione, controllo e test del software prodotto come specificato dal *Piano di progetto*
- Lavorare con la garanzia di rispettare i principi di qualità di sviluppo.

### 2.2.10 Consegna e completamento

Il gruppo prevede di rilasciare la prima versione del prodotto software sotto forma di *MVP<sub>G</sub>* alla revisione *PB<sub>G</sub>*.

## 2.3 Processo di sviluppo

### 2.3.1 Scopo

Il processo di sviluppo contiene e descrive le attività di **analisi dei requisiti**, **design**, **codifica** e **testing** del prodotto software.

### 2.3.2 Aspettative

Si prevede che alla termine di questo processo siano stati raggiunti i seguenti obiettivi:

- Produzione di un documento di *analisi dei requisiti*
- Definizione del *design* del prodotto software
- *Codifica* completa del prodotto software
- Copertura di tutti i requisiti del prodotto software tramite *testing*

### 2.3.3 Descrizione

Questo processo consiste delle seguenti attività:

- Processi di implementazione
- Analisi dei requisiti
- Progettazione logica
- Progettazione di dettaglio
- Codifica e testing
- Integrazione del software
- Test di qualità del software

### 2.3.4 Processi di implementazione

Durante il processo di sviluppo software il gruppo dovrà produrre la documentazione dei risultati ottenuti, effettuare la risoluzione dei problemi, risolvere le non conformità riscontrate ed attuare i cambiamenti di controllo in conformità con quanto stabilito dai **§processi di supporto**.

Tali attività e compiti potrebbero sovrapporsi e interagire tra di loro e potrebbero essere eseguite più di una volta durante l'avanzamento dello stato di sviluppo.

Durante il processo di sviluppo software il gruppo farà uso di metodi, strumenti, standard e linguaggi di programmazione appropriati e dotati di documentazione.

### 2.3.5 Analisi dei requisiti

Il preciso scopo di utilizzo del sistema da sviluppare deve essere specificato da parte del gruppo nel documento di *Analisi dei requisiti* che conterrà i seguenti elementi.

#### 2.3.5.1 Casi d'uso

Vengono individuati gli *attori*<sub>G</sub> del sistema e gli obiettivi da raggiungere per ciascun *attore*. La realizzazione degli *use case*<sub>G</sub> viene svolta fino al livello di dettaglio, per poter individuare direttamente i requisiti di sistema che lo *use case*<sub>G</sub> modella.

### 2.3.5.2 Requisiti

I requisiti di progetto vengono identificati dalle seguenti fonti:

- Analisi approfondita del *Capitolato<sub>G</sub>*;
- Discussione critica con la *Proponente<sub>G</sub>*;
- Analisi dei *casi d'uso<sub>G</sub>* precedentemente individuati;
- Discussione critica tra i membri del gruppo di progetto.

### 2.3.5.3 Funzionalità e capacità del sistema

#### Requisiti utente, organizzativi e di business

I *requisiti utente* sono i requisiti specificati dall'utente finale del sistema che descrivono come l'utente vorrebbe che il sistema funzioni e quali funzionalità deve fornire.

I *requisiti organizzativi* sono i requisiti che garantiscono che il sistema rispetti gli standard e le norme del gruppo.

I *requisiti di business* sono i requisiti che descrivono le esigenze del progetto per quanto riguarda la capacità del gruppo di lavorare in maniera efficace e efficiente.

Tutti questi requisiti devono essere raccolti e documentati prima che il progetto possa iniziare il periodo di sviluppo, poiché essi descrivono le funzionalità e gli obiettivi del sistema. Tali requisiti vengono utilizzati come base per la *progettazione* e lo *sviluppo* del prodotto software stesso.

#### Requisiti di funzionamento

I requisiti di funzionamento descrivono le specifiche *hardware* e *software* che devono essere soddisfatte per garantire il corretto funzionamento del sistema.

Tali requisiti includono:

- Sistemi operativi e browser supportati
- Specifiche hardware
- Dipendenze software

Questi requisiti devono essere considerati durante la fase di progettazione e di sviluppo del prodotto per garantire che siano soddisfatti.

#### Vincoli di progettazione e requisiti di qualifica

I *vincoli di progettazione* sono restrizioni imposte al progetto che devono essere rispettate durante la progettazione e lo sviluppo del sistema. Questi possono includere limitazione tecniche, come la disponibilità di una determinata tecnologia o la necessità di utilizzare un determinato linguaggio di programmazione.

I *requisiti di qualifica* sono criteri che devono essere soddisfatti per dimostrare che il sistema soddisfa i requisiti specificati e che funzioni correttamente. Questi possono includere test di accettazione, verifiche e validazioni, che sono eseguiti per verificare che il sistema soddisfi i requisiti utente, organizzativi e di business.

I vincoli di progettazione e i requisiti di qualifica sono entrambi importanti perché garantiscono che il sistema soddisfi i requisiti specificati e funzioni correttamente.

### 2.3.6 Processi di codifica

**2.3.6.1 Scopo** L'attività di codifica segue la fase di progettazione e verte nel cercare di ottenere i risultati attesi durante la programmazione del software.

**2.3.6.2 Aspettative** Questa attività ha l'obiettivo di generare un prodotto software conforme a ciò che è stato concordato con la proponente, in termini di requisiti e funzionalità. È buona prassi che il gruppo si accordi nel cercare di fissare alcune norme di codifica affinché lo stile di scrittura del codice possa essere leggibile e facilmente comprensibile in modo da garantire la manutenibilità dello stesso per future eventuali modifiche.

#### 2.3.6.3 Stile della codifica

- **Indentazione:** i blocchi di codice che si andranno a realizzare dovranno avere un'indentazione di quattro spazi;
- **Parentesi:** la parentesi aperta dovrà essere inserita nella stessa riga di dichiarazione del costrutto, separata da uno spazio, mentre la parentesi chiusa dovrà essere inserita con la giusta indentazione alla riga immediatamente successiva all'ultima riga di codice del costrutto;
- **Metodi:** il nome dei metodi dovrà iniziare con lettera minuscola e, se composto da più parole, le successive dovranno iniziare con lettera maiuscola. È preferibile mantenere metodi brevi, con poche righe di codice;
- **Classi:** il nome delle classi dovrà sempre iniziare con la lettera maiuscola e, come per i metodi, se composto da più parole, le successive dovranno iniziare con la lettera maiuscola;
- **Variabili:** il nome delle variabili deve sempre essere scritto in minuscolo e in inglese. Se il nome è composto da più parole, la seconda dovrà iniziare con la lettera maiuscola;
- **Costanti:** il nome deve essere sempre scritto in maiuscolo e in inglese. Se il nome è composto da più parole, queste dovranno essere separate dal carattere ' ';
- **Univocità dei nomi:** tutti i costrutti dovranno avere nomi univoci e significativi;
- **Commenti:** i commenti dovranno essere inseriti prima dell'inizio del costrutto e presentati in lingua italiana;
- **File:** dovranno avere un nome che inizia per lettera maiuscola che ne specifichi il contenuto.

## 3 Processi di Supporto

### 3.1 Documentazione

#### 3.1.1 Ciclo di Vita dei Documenti

Ogni documento nasce dalla necessità di fissare per iscritto un fatto in modo univoco a tutti i membri del gruppo di progetto.

Agli arbori di esso è presente la pianificazione: si va a decidere la struttura e il contenuto che esso debba avere.

In seguito i *redattori*<sub>G</sub> con la collaborazione degli *amministratori di sistema*<sub>G</sub> scrivono il documento, secondo quanto pianificato. Quest'ultimo, alla conclusione della stesura, passa al *verificatore*<sub>G</sub>, i quali controlleranno il documento nella sua interezza. Tale attività potrà terminare al più con uno dei seguenti due risultati: se la verifica termina senza errori o con errori lievi (errori di battitura, sintassi errata...) il documento viene corretto e passato al *responsabile di progetto*<sub>G</sub> per l'approvazione, mentre se presenta errori logici-strutturali non ammissibili il documento viene rinviato ai redattori per la correzione. Tale ciclo terminerà solo in caso di superamento della verifica e approvazione del documento.

#### 3.1.2 Versionamento dei documenti

Come è facilmente intuibile i documenti verranno creati con un metodo incrementale: da una versione iniziale base fino a raggiungere una versione finale attraverso raffinamenti successivi. Il metodo di versionamento scelto è il seguente: una versione **a.v.r** è caratterizzata da:

- **a** rappresenta la versione approvata dal *responsabile di progetto*<sub>G</sub>. È il valore più importante e permette la pubblicazione e invio del documento al *committente*<sub>G</sub>
- **v** rappresenta il livello di verifica del documento. Avanza soltanto quando il *verificatore*<sub>G</sub> ha approvato tale documento
- **r** rappresenta lo stato di scrittura del documento. Avanza soltanto se l'attività di scrittura dei *redattori*<sub>G</sub> e *amministratore di sistema*<sub>G</sub> è terminata. La modifica di una sezione/sottosezione deve essere accompagnata dalla verifica della medesima. La corrispondente riga del changelog dovrà quindi indicare sia il redattore che il verificatore.

Esempio esemplificativo: il documento nel momento in cui viene pianificato presenta la versione 0.0.0, mentre dopo la prima scrittura passa alla versione 0.0.1 e così via.

#### 3.1.3 Modifica documenti

In caso di modifica di un particolare denotato in un documento, questo dopo essere stato discusso durante una riunione interna e documentato con un apposito "Verbale di Modifica" dovrà essere riportato nel relativo documento, accopagnato dal relativo aggiornamento di versione.

#### 3.1.4 Suddivisione dei Documenti

I vari documenti vengono suddivisi in base al loro scopo e destinatario. Chiamiamo **Documenti Interni** tutta la documentazione destinata all'uso esclusivo dei membri di progetto e **Documenti Esterni** tutti quei documenti destinati anche all'azienda proponente e al *committente*<sub>G</sub>. Elenco dei documenti:

- **INTERNI**

- **Verbali** delle riunioni periodiche con i componenti del team
- **Norme di progetto** disponibili a tutto il team

- **ESTERNI**

- **Verbali** di riunioni con il proponente
- **Glossario** dei termini tecnici del progetto da sviluppare, disponibile e integrabile da tutto il team
- **Studio dei capitolati**
- **Lettera di candidatura** al progetto indicato come scelta preferita dal gruppo
- **Piano di progetto**
- **Piano di qualifica**
- **Analisi dei requisiti**

### 3.1.5 Template per tutti i documenti e presentazioni

Tutti i documenti e presentazioni vengono realizzati con template comune. Viene fatta tale scelta per una migliore standardizzazione e per facilitarne la modifica. Inoltre tutti i documenti saranno suddivisi in sezioni per garantirne l'aggiornamento. Struttura dei documenti:

- **Template** è il file che va a definire la struttura, i margini e tutti i particolari tecnici del documento.
- **Intestazione** del documento, rappresenta la prima pagina del documento e va a definire una serie di caratteristiche proprie di quest'ultimo:
  - Uso
  - Versione
  - *Redattori<sub>G</sub>*
  - *Verificatori<sub>G</sub>*
  - *Approvatore<sub>G</sub>*
  - Destinatari
  - Sommario del contenuto
- **Contenuto** componente che permette di rendere scalabile la modifica del documento
- **Registro delle Modifiche** permette di tracciare le integrazioni e modifiche la documento
- **Sezioni** del contenuto del documento

**3.1.5.1 Nomi delle immagini** Tutte le immagini che verranno generate per essere poi inserite in un documento attinente al gruppo di progetto saranno salvate sul *Repository<sub>G</sub>* del gruppo e dovranno riportare un identificativo attinente all'argomento trattato.

**3.1.5.2 Formati immagini nei documenti** Tutte le immagini ed i diagrammi che necessitano di essere nei documenti di progetto, devono essere forniti in formato .png così da permettere una più dettagliata visione, chiarezza espositiva e praticità nella compilazione.

### 3.1.6 Verball

Tutti i verball vengono redatti attraverso un template comune (citato sopra). Presentano al loro interno oltre alle componenti appena citate comuni a tutti i documenti le seguenti sezioni:

- **Generale** descrizione delle informazioni e obiettivi della riunione in oggetto
- **Svolgimento**
- **Decisioni prese**

I verball sono dei documenti particolari: a differenza degli altri non sono caratterizzati da revisioni e incrementi successivi.

### 3.1.7 Norme di progetto

È il documento contenente tutte le regole e buone pratiche che i componenti del team sono tenuti a seguire. Tale documento viene suddiviso in: Introduzione, Processi Primari, Processi di Supporto e Processi Organizzativi.

### 3.1.8 Candidatura

È il documento che contiene la preferenza da parte del team nei confronti dei progetti proposti e tutti i riferimenti ai documenti che motivano tale scelta.

### 3.1.9 Glossario

Il Glossario è un documento che contiene il significato di alcuni termini specifici per il dominio di interesse del progetto preso in esame. Esso viene realizzato utilizzando il classico template del resto della documentazione, ma a differenza degli altri documenti, al suo interno presenterà non delle sezioni ma un'elenco delle definizioni dei termini. Sarà possibile in tutti i documenti inserire dei richiami a quest'ultimo.

**3.1.9.1 Aggiornamento del glossario** Un requisito essenziale per svolgere un'attività di progetto è avere un glossario sempre aggiornato. Nel momento in cui un componente del team di progetto, incaricato di redigere un documento, si trova nella situazione di dover utilizzare un termine specifico di dominio è tenuto a:

- Verificare che tale termine sia già presente nel glossario, se lo fosse è tenuto ad aggiungere a tale termine la lettera G con il comando \glo scritto accanto al termine
- Aggiungere tale termine al Glossario, essendosi prima documentato sulla definizione dello stesso

### 3.1.10 Studio dei capitoli

Lo studio dei capitoli è un documento che contiene le prime considerazioni e analisi del team nei confronti dei *capitoli*<sub>G</sub> proposti dalle aziende anche dopo un primo confronto con alcune di esse.

### 3.1.11 Piano di progetto

È il documento che contiene la pianificazione, scadenze, stima dei costi e analisi dei rischi che si ipotizza possano accadere nell'eseguire un'attività di progetto nei confronti del *capitolo*<sub>G</sub> scelto.



### 3.1.12 Piano di qualifica

È il documento che descrive le modalità, perseguite dal gruppo, per verificare, validare e garantire la qualità del prodotto e processo.

### 3.1.13 Analisi dei requisiti

E' il documento che contiene l'analisi dei bisogni espressi dagli stakeholder riguardo il  $capitolato_G$  di interesse. Inoltre in tale documento sono contenuti anche i relativi casi d'uso,  $requisiti\_soluzione_G$  (funzionalità) atti a soddisfare i corrispettivi  $requisiti\_utente_G$ .

#### 3.1.13.1 Casi d'uso

Ogni caso d'uso tracciato viene identificato da un codice univoco che rispetta il seguente formalismo:

$$UC[ID]$$

Dove:

- **UC:** Sta per Use Case;
- **ID:** Valore numerico univoco per ogni use case.

Ogni use case può inoltre possedere le seguenti informazioni aggiuntive:

- **Attori primari:** rappresentano gli utenti che interagiscono con il sistema
- **Attori secondari** rappresentano le parti esterne che interagiscono con il sistema
- **Precondizione:** sono le condizioni che devono essere verificate prima che il caso d'uso possa iniziare
  - Nelle precondizioni dei *sotto-casi* e degli *scenari alternativi* viene utilizzata la terminologia "UC[ID] - Riferimento al caso d'uso" per far riferimento alla specifica situazione del caso d'uso.
- **Postcondizione:** sono le condizioni che devono essere verificate al termine del caso d'uso
- **Scenario principale:** la sequenza di passi che descrivono come il sistema deve rispondere all'attività dell'utente
- **Scenari alternativi:** descrivono come il sistema deve rispondere in situazioni diverse dal normale flusso principale

Viene inoltre fornita la rappresentazione grafica di ogni use case sotto forma di diagramma  $UML_G$  use case.

#### 3.1.13.2 Requisiti

Ogni requisito tracciato viene identificato da un codice univoco che rispetta il seguente formalismo:

$$R[Tipologia][Importanza][ID]$$

Dove:

- **R:** Sta per *Requisito*;
- **Tipologia:** Rappresenta il tipo del requisito. Può essere al più uno tra:
  - **F:** requisito funzionale
  - **P:** requisito prestazionale
  - **Q:** requisito qualitativo
  - **V:** requisito di vincolo
- **Importanza:** Rappresenta il grado di necessità del requisito. Può essere al più uno tra:
  - **O:** requisito obbligatorio;
  - **F:** requisito facoltativo;
  - **D:** requisito desiderabile.
- **ID:** Valore numerico univoco per ogni requisito.

In seguito all'associazione ad un requisito il codice identificativo non potrà essere modificato. Ogni requisito deve inoltre possedere le seguenti informazioni aggiuntive:

- **Descrizione:** Breve descrizione del requisito;
- **Classificazione:** Necessità del requisito;
- **Fonte:** Origine del requisito.

Si è scelto di duplicare la rappresentazione del grado di necessità dei requisiti per semplificare l'interpretazione del codice identificativo e per enfatizzarne l'importanza.

#### 3.1.14 Nomi dei Documenti

I documenti inizieranno con la lettera "d" seguita dal nome proprio del documento. Per quanto riguarda i verbali si dovrà riportare anche la data dell'incontro a cui si fa riferimento.

#### 3.1.15 Tipografia utilizzata

- **Grandezza font** di 10pt per tutto il documento
- **Grassetto** per termini importanti o titoli di sezioni (la grandezza è automaticamente determinata da Latex)
- **Elenchi non annidati:** il termine chiave viene scritto in Grassetto

#### 3.1.16 Scrittura della data

La data viene fornita in formata americano yyyy-mm-dd.

### 3.1.17 Sigle dei documenti:

- VI: Verbale interno
- VE: Verbale esterno
- NdP: Norme di progetto
- PdP: Piano di progetto
- AdR: Analisi dei requisiti
- PdQ: Piano di qualifica

## 3.2 Revisione

Con il termine revisione si intende una valutazione del periodo appena trascorso che il team di progetto intende fare per migliorare il proprio *way of working*<sub>G</sub>, aumentando la produttività. Tale attività viene svolta a cadenze periodiche in determinate riunioni interne nelle quali si vanno a illustrare valori negativi e valori positivi della metodologia utilizzata. Al termine di tali riunioni si designa un componente del team di progetto incaricato di apportare le rispettive modifiche al *way of working*

## 3.3 Risoluzione dei problemi

Con il termine segnalazione di problemi si va ad intendere qualsiasi malfunzionamento o imprevisto che si possa andare a verifica nel periodo dell'attività di progetto. La procedura di segnalazione dei problemi sarà la seguente:

1. Il componente si occorre della presenza di un problema
2. Costui ha la responsabilità di approfondire il problema e di segnalarlo agli altri componenti attraverso l'apertura di un *issue*<sub>G</sub> su git hub esplicitando: Tipo di problematica e descrizione della problematica se quest'ultima non è risolvibile in modo autonomo
3. In seguito alla pubblicazione della *issue* ogni componente del gruppo attraverso la propria conoscenza o con riferimenti a soluzioni può contribuire alla risoluzione
4. La problematica si detiene risolta solo alla chiusura della *issue* e viene archiviata come *storico*<sub>G</sub> per eventuali ricomparsa di quest'ultima

## 3.4 Gestione della Qualità

La gestione della qualità del software è un insieme di attività e di pratiche che hanno l'obiettivo di garantire che un prodotto software soddisfi i *requisiti* del cliente. La gestione della qualità del software viene definita nel documento di *Piano di qualifica* che include diverse attività come le metodologie di controllo degli obiettivi di qualità, la pianificazione dei test, la progettazione dei test, l'esecuzione dei test, la valutazione dei risultati dei test e la risoluzione dei problemi.

Ogni metrica di qualità viene identificata da un codice univoco che rispetta il seguente formalismo:

**MPC[ID]**

Dove:

- **MPC:** Sta per *Metrica ProCesso*
- **ID:** Valore numerico univoco per ogni metrica

**MPD[ID]**

Dove:

- **MPD:** Sta per *Metrica ProDotto*
- **ID:** Valore numerico univoco per ogni metrica

Ogni metrica deve inoltre possedere le seguenti informazioni aggiuntive:

- **Nome:** Breve descrizione della metrica
- **Valore Accettabile:** Valore da rispettare
- **Valore Ottimale:** Valore assunto dalla metrica in condizioni ottimali

Per una descrizione più dettagliata delle metriche di qualità si faccia riferimento al documento *Piano di qualifica v2.0.0*.

### 3.5 Verifica

Il processo di *verifica<sub>G</sub>* si occupa di controllare che il prodotto preso in esame sia coretto e completo. Deve essere svolta ad intervalli regolari, secondo quando deciso a maggioranza al momento dell'assegnazione di ogni compito ai componenti del team di progetto in sessione pubblica durante una riunione con la partecipazione di tutti i membri (a parte giustificazioni valide).

#### 3.5.1 Svolgimento

Nella *verifica<sub>G</sub>* in particolare si va a verificare che determinate azioni diano effetti previsti. L'attività di *verifica* va svolta ogni qual volta che si:

- Esegue un cambiamento consistente allo svolgimento di un processo
- E' determinato a priori che tale livello di maturità necessiti di *verifica*

#### 3.5.2 Come eseguirla

L'attività di *Verifica<sub>G</sub>* dovrà attenersi a considerare e verificare i parametri indicati nelle rispettive procedure di ogni processo, controllando che ogni azione dia l'effetto previsto. Conclusa la *Verifica* si prosegue con il processo di *validazione<sub>G</sub>*.

### 3.5.3 Tecniche di Verifica

Il processo di *verifica*, secondo come viene svolto, può essere suddiviso in 2 tecniche:

- Walkthrough: tecnica attraverso cui il *Verificatore<sub>G</sub>* esegue una lettura e controllo ad ampio spettro del documento, non tralasciando nulla
- Inspection: tecnica attraverso la quale il *Verificatore* va a controllare soltanto i punti chiave o parti più importanti del documento essendo valori di interesse strategico

### 3.5.4 Verifica della Documentazione

La *verifica<sub>G</sub>* della documentazione si attiene ai seguenti passi:

- Controllo sintattico e ortografico (eventuali errori di battitura)
- Controllo della coerenza del dizionario utilizzato nei confronti del dominio di interesse
- Controllo della aderenza dei contenuti del documento secondo quanto assegnato

### 3.5.5 Verifica del codice

Il processo di *Verifica<sub>G</sub>* del codice si servirà di test automatici scritti apposta con lo scopo di rintracciare eventuali bug all'interno del codice. Durante questa fase si farà uso di due particolari tipologie di analisi:

- **Analisi Statica:** In questa fase è premura di programmatori e verificatori accertarsi che durante la fase di scrittura del codice siano state rispettate le norme decise dal gruppo di lavoro
- **Analisi Dinamica:** Programmatori e verificatori controlleranno la presenza di eventuali bug tramite l'esecuzione del software e procederanno nel risolverli.

### 3.5.6 Test

L'attività di testing prevede la scrittura di parti di codice con lo scopo di trovare eventuali bug nel software a run time. È importante che i test siano quanto più possibile automatizzati in modo tale da renderli ripetibili.

#### Classificazione dei test

I test vengono contrassegnati da un codice con la seguente struttura:

**T[TipologiaTest][ImportazioneRequisito]\*[TipologiaRequisito]\*[IDNumerico]**

dove **TipologiaTest** può assumere i seguenti valori:

- **A**(Test di accettazione): Si eseguono prima del rilascio del prodotto
- **S**(Test di sistema): Vengono eseguiti su di un sistema completo con lo scopo di verificare che rispetti i requisiti desiderati.
- **U**(Test di unità): Test eseguiti su una piccola porzione di codice, isolata dal tutto il resto

I test di unità vengono a loro volta identificati con i codici:

- **TC:** *Solidity<sub>G</sub>* - *TCoin Token<sub>G</sub>*
- **TR:** *Solidity* - *Trustify<sub>G</sub>*
- **AP:** *Java<sub>G</sub>* - API-REST

**ImportanzaRequisito** e **TipologiaRequisito** vengono usati solo per i test di sistema e accettazione, servono a indicare quale requisito si vuole andare a verificare con quel particolare test.

**IDNumerico** viene usato come indice crescente che parte da 1 e viene incrementato per contraddistinguere ogni test all'interno di uno stesso gruppo.

### 3.5.7 Strumenti

- **Truffle:** *Framework<sub>G</sub>* usato per eseguire i test sullo *Smart Contract<sub>G</sub>*.
- **Karma:** Strumento di testing che si basa sull'utilizzo di Nodejs per funzionare, utilizzato per testare codice *Javascript<sub>G</sub>*/*Typescript<sub>G</sub>* su diversi tipi di browser. In particolare il gruppo ne ha fatto uso per eseguire i test sulla *Webapp<sub>G</sub>*.
- **Jasmine:** Framework usato per eseguire i test su codice Javascript/Typescript, impiegato per testare la webapp.
- **Junit:** Framework usato per eseguire i test su codice *Java<sub>G</sub>*, impiegato per testare l'API-Rest.

## 3.6 Validazione

Con il termine *validazione<sub>G</sub>* si va ad intendere la verifica del soddisfacimento di tutti i *requisiti utente<sub>G</sub>* esposti dal proponente nel *capitolato<sub>G</sub>*. Per verificare tutto ciò verranno eseguiti dei test che sappiano dimostrare che il prodotto fornisce le funzionalità richieste. Naturalmente tali test dovranno essere già verificati in precedenza, avendo così un risultato quasi predicibile dei test di validazione.

## 3.7 Gestione dei rischi

È compito del *Responsabile* di progetto mantenere aggiornata l'analisi dei rischi contenuta nel *Piano di progetto v4.0.0* e riconoscere il sorgere dei rischi documentati durante lo svolgimento del progetto.

La norma che il *Responsabile* dovrà seguire è la seguente:

1. Monitoraggio del sorgere di rischi già tracciati e individuazione di quelli non tracciati;
2. Aggiungere eventuali nuovi rischi individuati al *Piano di progetto v4.0.0*;
3. Se necessario, ridefinire le strategie da implementare per mitigare i rischi.

### 3.7.1 Codifica dei rischi

Ogni rischio tracciato nel *Piano di progetto v4.0.0* viene identificato tramite codice univoco che rispetti la seguente codifica:

**R[Tipologia][ID]**

Dove:

- **R:** Sta per "rischio";
- **Tipo:** Rappresenta il tipo di rischio. Le tipologie attualmente tracciate sono:

- **T**: Tecnologico;
- **O**: Organizzativo;
- **I**: Interpersonale;
- **R**: Requisito.

- **ID**: Contatore iniziante da 1 che viene incrementato di 1 per ogni rischio tracciato.

Per una trattazione più dettagliata dei rischi tracciati si veda il *Piano di progetto v4.0.0*.

## 4 Processi Organizzativi

I processi organizzativi sono tutti quei processi che determinano l'organizzazione del gruppo *PentaSoft*. Vengono identificati in:

- **Processo di gestione organizzativa**
- **Processo di gestione dell'infrastruttura**
- **Processo di formazione**
- **Pianificazione**

### 4.1 Gestione organizzativa

#### 4.1.1 Scopo

Lo scopo principale di questo processo è la stesura del documento *Piano di progetto*, la cui funzione è normare spese, scadenze e rischi del progetto Trustify. Lo scopo secondario è normare tutte le attività inerenti all'organizzazione del lavoro.

#### 4.1.2 Aspettative

Si prevede che al termine di questo processo siano stati raggiunti i seguenti obiettivi:

- produzione del documento *Piano di progetto*;
- definizione dei ruoli di progetto e relativa rotazione;
- definizione di norme e metodologie per disciplinare il lavoro.

#### 4.1.3 Descrizione

Il processo di *Gestione Organizzativa* copre i seguenti argomenti:

- Ruoli di progetto;
- Rotazione dei ruoli di progetto;
- Gestione delle riunioni;
- Orari di reperibilità;
- Gestione dei rischi.

#### 4.1.4 Ruoli di progetto

- ***Responsabile di progetto*[RE]**

Colui che guida il progetto e gestisce lo svolgimento dei vari processi, come ad esempio:

- Gestire la pianificazione delle attività assegnandole ai membri del gruppo;
- Gestire le comunicazioni esterne;
- Organizzare il lavoro in modo da minimizzare la probabilità che si verifichino problemi;
- Approvare il rilascio della documentazione al termine del processo di verifica.



- **Amministratore[AM]**

É il responsabile degli strumenti necessari per garantire efficacia ed efficienza dei processi.  
Si occupa di:

- Gestire il sistema di archiviazione e versionamento;
- Gestire l'infrastruttura e gli strumenti utilizzati;
- Individuare punti di miglioramento dei processi;
- Redigere le *Norme di progetto*.

- **Analista[AN]**

Colui che comprende a fondo le necessità del proponente e le formalizza sotto forma di requisiti.  
Le sue responsabilità sono:

- Individuazione dei requisiti del progetto e delle varie funzionalità;
- Redigere la *Analisi dei requisiti*.

- **Progettista[PG]**

Ha il compito di definire la struttura architeturale del sistema, basandosi sul lavoro svolto dall'*analista*. Nello specifico si occupa di:

- Studiare l'architettura più adatta al prodotto da realizzare;
- Produrre una architettura ad alto livello di manutenibilità che soddisfi i requisiti;
- Garantire la qualità del prodotto.

- **Programmatore[PR]**

Ha il compito di implementare l'architettura delineata dal *progettista*, in particolare:

- Scrivere codice che soddisfi le specifiche architettureali del progetto;
- Scrivere codice documentato, mantenibile e versionato;
- Scrivere i test relativi al codice prodotto.

- **Verificatore[VE]**

Il suo compito consiste nella verifica di contenuti, caratteristiche e qualità di ogni attività di progetto.

Si occupa di:

- Controllare che il lavoro svolto dai membri del gruppo sia conforme alle norme delineate nell'omonimo documento *Norme di progetto*;
- Segnalare tempestivamente eventuali in conformità rilevate durante la verifica affinché vengano corrette.

Il *verificatore*, per ovvie ragioni, non può effettuare il controllo di una attività svolta da lui stesso.

#### 4.1.5 Rotazione dei ruoli di progetto

L'assegnazione del ruolo di *Responsabile* viene dettata da una schedulazione *Round Robin<sub>G</sub>* a quanto di tempo mensile. In questo modo viene garantita un'equa distribuzione delle responsabilità. Gli altri ruoli di progetto verranno assegnati dal *Responsabile* in base alle attività da svolgere. L'ordine di rotazione previsto è il seguente:

Data inizio	Data fine	Responsabile designato
03/11/2022	03/12/2022	Luca Marcato
03/12/2022	03/01/2023	Marco Brugin
03/01/2023	03/02/2023	Marco Rosin
03/02/2023	03/03/2023	Nicola Lazzarin
03/03/2023	03/04/2023	Pietro Lauriola
03/04/2022	03/05/2022	Stefano Meneguzzo

#### 4.1.6 Gestione delle riunioni

L'organizzazione delle riunioni interne ed esterne spetta al *Responsabile*, il quale:

1. (In caso di riunione esterna) Contatta la Proponente per schedulare la riunione;
2. Propone ai membri del gruppo la data prevista per la riunione;
3. Solo se almeno 2/3 del gruppo garantiscono la partecipazione conferma la riunione;
4. Aggiunge la riunione al calendario di progetto;
5. Notifica i membri del gruppo;
6. Assegna ad un membro del gruppo la redazione del verbale della riunione, che dovrà essere conforme alle regole di stesura delineate nelle *Norme di progetto*.

Gli strumenti utilizzati per la comunicazione sono elencati nella sottosezione §Comunicazioni.

#### 4.1.7 Orario di reperibilità

Ogni membro del gruppo *PentaSoft* si impegna ad essere reperibile nei seguenti orari dei giorni feriali:

- dalle 09:00 alle 13:00;
- dalle 14:00 alle 18:00.

È da notare che queste fasce orarie non rappresentano le effettive ore di lavoro contabilizzate ma solo i periodi in cui ogni membro sarà disponibile per collaborare in modalità sincrona/asincrona.

### 4.2 Gestione delle infrastrutture interne

#### 4.2.1 Comunicazioni

##### 4.2.1.1 Comunicazioni interne

Le comunicazioni interne del gruppo *PentaSoft* avvengono tramite i seguenti canali:

- **Slack<sub>G</sub>**: Utilizzato per comunicazioni formali strettamente inerenti al progetto;
- **Telegram<sub>G</sub>**: Utilizzato per comunicazioni informali di carattere generale;
- **Discord<sub>G</sub>**: Utilizzato per riunioni vocali formali di tipo organizzativo.

#### 4.2.1.2 Comunicazioni esterne

Comunicazioni che avvengono principalmente con committente e proponente. Essendo di notevole importanza vanno adeguatamente preparate.

Gli strumenti principalmente utilizzati sono:

- **Google meet<sub>G</sub>**: Utilizzato per riunioni formali di tipo vocale;
- **Gmail<sub>G</sub>**: Utilizzato per comunicazioni formali di tipo testuale.

#### 4.2.2 Riunioni

##### 4.2.2.1 Riunioni interne

Le riunioni interne fungono da *milestone<sub>G</sub>* per l'avanzamento del progetto e hanno cadenza al più settimanale. Vengono organizzate dal *Responsabile* rispettando le norme descritte alla sezione 4.1.6.

Le svolgimento delle riunioni dovrà aderire obbligatoriamente alle seguenti norme:

- Ogni membro del gruppo è tenuto ad annotarsi i punti chiave della riunione e a comunicarli al *verbalista* designato al termine della stessa;
- Ciascun membro del gruppo si impegna a possedere adeguata conoscenza degli argomenti oggetto della riunione e/o che intende esporre al resto del gruppo;
- Ciascun membro del gruppo si impegna a partecipare attivamente alla riunione.

##### 4.2.2.2 Riunioni esterne

Le riunioni esterne coinvolgono attivamente *Committente<sub>G</sub>* e *Proponente<sub>G</sub>*. Ogni riunione esterna deve essere adeguatamente e preparata in anticipo, in modo da avere una struttura ben delineata e una durata contenuta.

Vengono organizzate dal *Responsabile* rispettando le norme descritte alla sezione §4.1.6.

#### 4.2.3 Gestione degli strumenti di coordinamento

##### 4.2.3.1 Ticketing

Per la gestione delle *issue<sub>G</sub>* viene utilizzato *Trello<sub>G</sub>*. È compito del *Responsabile<sub>G</sub>* creare e assegnare le varie issue ai membri del gruppo, cercando di partizionare equamente il carico di lavoro. Ogni issue contiene una descrizione del compito da svolgere ed eventuali commenti relativi ad esso.

Il ciclo di vita di ogni issue è diviso nei seguenti stati, rappresentati in:

- **Programmato:** rappresenta il *backlog<sub>G</sub>* di cose da fare;
- **Da fare:** *issue* trasferita spostata dal *backlog* e assegnata ad un membro del gruppo;
- **In esecuzione:** *issue* in svolgimento;
- **Sotto verifica:** *issue* svolta ma non ancora verificata;
- **Completata:** *issue* svolta e verificata, considerata formalmente *terminata*.

La gestione delle issue dovrà rispettare le seguenti norme:

- Solamente il *Responsabile* e l'assegnatario di una issue ne possono far avanzare lo stato;
- Lo stato di una issue non può mai regredire, eccezion fatta per La transizione tra "in esecuzione" e il "sotto verifica";

#### 4.2.4 Gestione degli strumenti di versionamento

Per il versionamento e il salvataggio dei file è previsto l'utilizzo della piattaforma *GitHub<sub>G</sub>*.

È compito dell'*Amministratore* di progetto creare i *repository<sub>G</sub>* necessari e aggiungere i rimanenti membri del gruppo come collaboratori.

Attualmente è previsto l'uso di due *repository<sub>G</sub>*:

- **Docs-privati:** *repository* documentale contenente i file sorgente (*.tex*) e le immagini (*.png*);
- **Docs:** *repository* documentale contenente i documenti in formato *.pdf* prodotti tramite compilazione automatica dei file sorgente.

Entrambi i *repository* contengono un apposito file *.gitignore* per filtrare automaticamente i contenuti e garantire l'assenza di file indesiderati.

### 4.3 Formazione

#### 4.3.1 Scopo

Lo scopo di questa sezione è definire le norme riguardanti il processo di formazione e apprendimento che i membri del gruppo *PentaSoft* sono tenuti a rispettare per ottenere una buona conoscenza degli strumenti e linguaggi necessari per sviluppare il prodotto e i documenti richiesti.

#### 4.3.2 Aspettative

Si prevede che al termine del processo di formazione ogni membro abbia ottenuto:

- Una buona conoscenza del linguaggio *L<sup>A</sup>T<sub>E</sub>X*;
- Una buona comprensione dell'ambito di sviluppo del capitolato;
- Una buona familiarità con i linguaggi di programmazione, le librerie e gli strumenti utilizzati per realizzare il prodotto software;
- Una buona comprensione degli strumenti organizzativi usati per la buona riuscita di un progetto software.

#### 4.3.3 Modalità di formazione

La principale modalità di formazione di ogni componente del gruppo *PentaSoft* è l'apprendimento in autonomia tramite documentazione reperita sul web e materiale fornito dalla proponente e dal committente.

In caso di difficoltà i rimanenti membri del gruppo si impegnano a condividere la propria conoscenza e ad aiutare per quanto possibile il componente in difficoltà al fine di velocizzare il processo di formazione.

Vengono riportati i principali strumenti usati:

- **L<sup>A</sup>T<sub>E</sub>X<sub>G</sub>:** Per la stesura di tutti i documenti versionabili relativi al progetto;

<https://www.overleaf.com/learn>

- **GitHub<sub>G</sub>:** Per il versionamento e archiviazione di documenti e codice sorgente;

<https://docs.github.com>

- **Strumenti di comunicazione:** Per riunioni interne ed esterne:

- **Discord<sub>G</sub>**
- **Zoom<sub>G</sub>**
- **Google Meet<sub>G</sub>**

Per comunicazioni veloci tra i membri del gruppo:

- **Telegram<sub>G</sub>**
- **Slack<sub>G</sub>**
- **Google Drive<sub>G</sub>**: Per archiviare presentazioni ed altri documenti non versionabili;
- **Strumenti di pianificazione**:
  - **Jira<sub>G</sub>**
  - **Trello<sub>G</sub>**
- **PlantUML<sub>G</sub>**: Per la realizzazione di diagrammi *UML<sub>G</sub>*.

## 4.4 Pianificazione

### 4.4.1 Scopo

Il *responsabile del progetto<sub>G</sub>* deve preparare un piano per l'esecuzione dei vari processi durante il ciclo di vita del progetto. Il piano associato all'esecuzione dei processi in un determinato periodo deve contenere una descrizione delle attività e dei compiti.

### 4.4.2 Aspettative

Si prevede che al termine di questo processo siano stati raggiunti i seguenti obiettivi:

### 4.4.3 Descrizione

Il processo di *pianificazione* copre i seguenti argomenti:

- Definizione dei piani per il completamento delle attività
- Assegnazione adeguata delle risorse necessarie per eseguire le attività
- Assegnazione delle attività
- Assegnazione delle responsabilità
- Quantificazione dei rischi associati alle attività o al processo stesso