

# SiVoDiM

---



## Norme di Progetto

Versione	1.0.0
Redattori	<Redattore>
Verificatori	<Verificatore>
Responsabili	<Responsabile>
Uso	<Uso>
Lista di distribuzione	<i>Stark Labs</i> Prof. Vardanega Tullio, Dr. Cardin Riccardo

Documento riguardante l'insieme di norme stabilite dal gruppo StarkLabs per la  
realizzazione di SiVoDiM.

## Indice

<b>1</b>	<b>Introduzione</b>	<b>1</b>
1.1	Scopo del documento	1
1.2	Scopo del progetto	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>3</b>
2.1	Fornitura	3
2.1.1	Attività	3
2.1.1.1	Accettazione	3
2.1.1.1.1	Discussione e scelta del capitolato	3
2.1.1.1.2	Studio di fattibilità	3
2.1.1.2	Preparazione della risposta	3
2.1.1.2.1	Definizione e preparazione della proposta	3
2.1.1.3	Pianificazione	3
2.1.1.3.1	Scelta del modello di ciclo di vita	3
2.1.1.3.2	Sviluppo e documentazione del Piano di Progetto	4
2.2	Sviluppo	4
2.2.1	Attività	4
2.2.1.1	Analisi dei requisiti	4
2.2.2	Norme	4
2.2.2.1	Classificazione requisiti	4
2.2.2.2	Classificazione casi d'uso	5
2.2.2.3	Codifica file	5
2.2.2.4	Nomi e norme di codifica	5
2.2.2.5	Ricorsione	6
2.2.3	Strumenti	6
2.2.3.1	Requirements Tool	6
2.2.3.2	PhpStorm ?	6
<b>3</b>	<b>Processi di supporto</b>	<b>7</b>
3.1	Documentazione	7
3.1.1	Attività	7
3.1.1.1	Documentazione	7
3.1.2	Procedure	7
3.1.2.1	Gestione dei documenti	7
3.1.2.2	Creazione di un nuovo documento	7
3.1.2.3	Avanzamento di un documento	7
3.1.3	Gestione del glossario	7
3.1.4	Norme	8
3.1.4.1	Progettazione e sviluppo dei documenti	8
3.1.4.2	Versionamento	8
3.1.4.3	Template	8
3.1.4.4	Struttura dei documenti	8
3.1.4.4.1	Prima pagina	8
3.1.4.4.2	Registro delle modifiche	9
3.1.4.4.3	Indici	9
3.1.4.4.4	Formattazione di una pagina	9
3.1.4.5	Suddivisione dei documenti	9
3.1.4.5.1	Norme di Progetto	9
3.1.4.5.2	Studio di Fattibilità	9

3.1.4.5.3	Analisi dei Requisiti . . . . .	9
3.1.4.5.4	Piano di Progetto . . . . .	9
3.1.4.5.5	Piano di Qualifica . . . . .	10
3.1.4.5.6	Specifica Tecnica . . . . .	10
3.1.4.5.7	Definizione di Prodotto . . . . .	10
3.1.4.5.8	Glossario . . . . .	10
3.1.4.6	Norme tipografiche . . . . .	10
3.1.4.6.1	Stile del testo . . . . .	10
3.1.4.6.2	Punteggiatura . . . . .	10
3.1.4.6.3	Composizione del testo . . . . .	10
3.1.4.6.4	Formati . . . . .	11
3.1.4.6.5	Sigle . . . . .	11
3.1.4.7	Componenti grafiche . . . . .	11
3.1.4.7.1	Immagini . . . . .	11
3.1.4.7.2	Diagrammi . . . . .	11
3.2	Processo di verifica e validazione . . . . .	11
3.2.1	Attività . . . . .	12
3.2.1.1	Analisi statica . . . . .	12
3.2.1.2	Analisi dinamica . . . . .	12
3.2.1.3	Gestione anomalie . . . . .	12
3.2.1.4	Tracciamento . . . . .	12
3.2.2	Procedure . . . . .	12
3.2.2.1	Procedure di assegnazione delle anomalie . . . . .	12
3.2.3	Norme . . . . .	12
3.2.3.1	Priorità risoluzione anomalie . . . . .	12
3.2.4	Strumenti . . . . .	12
3.2.4.1	Correzione ortografica . . . . .	12
3.2.4.2	Calcolo indice Gulpease . . . . .	12
3.2.4.3	Database online . . . . .	12
<b>4</b>	<b>Processi organizzativi</b>	<b>13</b>
4.1	Gestione dei processi . . . . .	13
4.1.1	Attività . . . . .	13
4.1.1.1	Gestione delle comunicazioni . . . . .	13
4.1.1.1.1	Comunicazione interna . . . . .	13
4.1.1.1.2	Comunicazione esterna . . . . .	13
4.1.1.1.3	Composizione email . . . . .	13
4.1.1.2	Gestione delle riunioni . . . . .	13
4.1.1.2.1	Riunioni interne . . . . .	14
4.1.1.2.2	Riunioni esterne . . . . .	14
4.1.1.3	Gestione del sistema dei task . . . . .	14
4.1.1.4	Gestione delle <i>milestone</i> . . . . .	15
4.1.1.5	Gestione dei <i>task</i> . . . . .	15
4.1.1.6	Gestione dello svolgimento dei task . . . . .	15
4.1.2	Procedure . . . . .	15
4.1.2.1	Generazione di una milestone . . . . .	15
4.1.2.2	Assegnazione di un task . . . . .	16
4.1.2.3	Svolgimento di un task . . . . .	16
4.1.2.4	Rilevamento dei rischi . . . . .	16
4.1.2.5	Ruoli di Progetto . . . . .	16
4.1.2.5.1	Responsabile di Progetto . . . . .	18
4.1.2.5.2	Amministratore di Progetto . . . . .	18
4.1.2.5.3	Analista . . . . .	18
4.1.3	Norme . . . . .	18
4.1.3.0.4	Progettista . . . . .	18
4.1.3.0.5	Programmatore . . . . .	18

	4.1.3.0.6	Verificatore . . . . .	18
4.1.4	Strumenti . . . . .		19
	4.1.4.1	Teamwork . . . . .	19
	4.1.4.2	Astah . . . . .	19
	4.1.4.3	Gantt . . . . .	19
	4.1.4.4	Telegram . . . . .	19
	4.1.4.5	Microsoft Office Power Point . . . . .	19
4.2	Gestione delle infrastrutture . . . . .		19
	4.2.0.6	Attività . . . . .	19
		4.2.0.6.1 Gestione del repository . . . . .	19
		4.2.0.6.2 Gestione del messaggio di commit . . . . .	19
	4.2.0.7	Procedure . . . . .	19
		4.2.0.7.1 Installazione di Git . . . . .	19
		4.2.0.7.2 Creazione di una cartella locale di repository . . . . .	20
		4.2.0.7.3 Creazione del branch personale . . . . .	20
4.2.1	Norme . . . . .		20
	4.2.1.1	Repository . . . . .	20
		4.2.1.1.1 Nomi dei file in SiVoDim . . . . .	20
		4.2.1.1.2 Struttura di SiVoDim . . . . .	20
		4.2.1.1.3 Messaggio di commit . . . . .	20
4.2.2	Strumenti . . . . .		21
	4.2.2.1	Git . . . . .	21
	4.2.2.2	GitHub . . . . .	21
	4.2.2.3	Dropbox . . . . .	21
	4.2.2.4	Sistema Operativo . . . . .	21
4.3	Formazione dei membri del gruppo . . . . .		21

## Elenco delle figure

- 1     Diagramma di attività - Procedura di assegnazione di un task . . . . . 17

# 1 Introduzione

## 1.1 Scopo del documento

Questo documento definisce le norme che i membri del gruppo *Stark Labs* dovranno rispettare nello svolgimento del progetto SiVoDiM: Sintesi Vocale per Dispositivi Mobili. Ogni componente del gruppo è tenuta a leggere tale documento e seguire le norme per raggiungere il miglior punto di incontro tra efficienza ed efficacia nello svolgimento delle attività. In questo modo viene garantita l'uniformità del materiale prodotto e vengono facilitate le operazioni di verifica. In particolare verranno specificate norme riguardanti:

- Interazioni tra i membri del gruppo;
- Comunicazione verso l'esterno;
- Stesura dei documenti e convenzioni tipografiche;
- Organizzazione dell'ambiente di lavoro;
- Modalità di lavoro durante le fasi del progetto;
- Stesura del codice.

## 1.2 Scopo del progetto

Lo scopo del progetto risiede nello sviluppo di un applicativo utile a dimostrare efficacemente le potenzialità del motore di sintesi vocale FA-TTS<sub>G</sub> sviluppato dall'azienda MIVOQ s.r.l. e messo a disposizione del gruppo di lavoro. L'applicazione deve implementare il motore all'interno di uno o più sistemi operativi *mobile* e metterne in risalto gli aspetti positivi attraverso un caso d'uso di facile comprensione.

## 1.3 Glossario

Al fine di aumentare la comprensione del testo ed evitare eventuali ambiguità, viene fornito un glossario (*Glossario v1.0.0*) contenente le definizioni degli acronimi e dei termini tecnici utilizzati nei documenti. Ogni vocabolo che ha un riferimento contenuto nel glossario è contrassegnato dal pedice "<sub>G</sub>".

## 1.4 Riferimenti

### 1.4.1 Normativi

- *Glossario v1.0.0*.

### 1.4.2 Informativi

- Capitolato C1 – Actorbase: a NoSQL DB based on the Actor model  
<http://www.math.unipd.it/~tullio/IS-1/2015/Progetto/C1.pdf>;
- Capitolato C2 – CLIPS: Communication & Localization with Indoor Positioning Systems  
<http://www.math.unipd.it/~tullio/IS-1/2015/Progetto/C2.pdf>;
- Capitolato C3 – UMAP: un motore per l'analisi predittiva in ambiente Internet of Things  
<http://www.math.unipd.it/~tullio/IS-1/2015/Progetto/C3.pdf>;
- Capitolato C4 – MaaS: MongoDB as an admin Service  
<http://www.math.unipd.it/~tullio/IS-1/2015/Progetto/C4.pdf>;
- Capitolato C5 – Quizzipedia: software per la gestione di questionari  
<http://www.math.unipd.it/~tullio/IS-1/2015/Progetto/C5.pdf>;

- Capitolato C6 – SiVoDiM: Sintesi Vocale per Dispositivi Mobili  
<http://www.math.unipd.it/~tullio/IS-1/2015/Progetto/C6.pdf>;
- AS/NZS ISO/IEC 12207:1997  
[http://www.math.unipd.it/~tullio/IS-1/2009/Approfondimenti/ISO\\_12207-1995.pdf](http://www.math.unipd.it/~tullio/IS-1/2009/Approfondimenti/ISO_12207-1995.pdf);
- SI/ISO 31-0 [https://en.wikipedia.org/wiki/ISO\\_31-0](https://en.wikipedia.org/wiki/ISO_31-0);
- Guida all'utilizzo di Teamwork<sub>G</sub>  
<http://support.teamwork.com/projects/start/getting-started>;
- Guida all'utilizzo di GitHub<sub>G</sub>  
<https://guides.github.com>;
- Guida all'utilizzo di Astah<sub>G</sub>  
<http://astah.net/tutorials>;
- Guida all'utilizzo di Microsoft Project 2016<sub>G</sub>  
[https://youtu.be/\\_eD2u8bxecs](https://youtu.be/_eD2u8bxecs)  
<https://support.office.com/it-it/article/Formattazione-del-diagramma-di-una-visualizzazione-di-Gantt-7473acdc-4abe-4b2f-8361-546efa9dce06#top>.

## 2 Processi primari

In questa sezione vengono descritti i processi di fornitura e di sviluppo attuati dal gruppo *Stark Labs*. Il processo di acquisizione spetta a *Proponente<sub>G</sub>* e *Committente<sub>G</sub>* del capitolato scelto, mentre il processo di manutenzione non può essere eseguito per vincoli dati dal tempo disponibile. Il prodotto *software* e la documentazione fornita con esso devono garantire la possibilità futura di essere sottoposti alle attività del processo di manutenzione.

### 2.1 Fornitura

#### 2.1.1 Attività

##### 2.1.1.1 Accettazione

**2.1.1.1.1 Discussione e scelta del capitolato** Il *Responsabile di Progetto* ha il compito di organizzare gli incontri per permettere ai componenti del gruppo di discutere sui capitolati disponibili. Le valutazioni che hanno portato a prendere questa decisione vanno documentate nello *Studio di Fattibilità*.

**2.1.1.1.2 Studio di fattibilità** Per realizzare il documento devono essere presi in considerazione i seguenti punti, adattati ai capitolati disponibili:

- Valutazione generale del capitolato;
- Valutazione dei fattori di rischio.

Per il capitolato scelto devono essere analizzati anche i seguenti punti:

- Studio del dominio applicativo;
- Studio del dominio tecnologico;
- Analisi di mercato;
- Analisi delle potenziali criticità.

##### 2.1.1.2 Preparazione della risposta

**2.1.1.2.1 Definizione e preparazione della proposta** I membri del gruppo *Stark Labs* devono redigere i seguenti documenti:

- *Norme di Progetto v1.0.0*;
- *Studio di Fattibilità v1.0.0*;
- *Analisi dei Requisiti v1.0.0*;
- *Piano di Progetto v1.0.0*;
- *Piano di Qualifica v1.0.0*.

In allegato viene consegnata anche la *Lettera di Presentazione*. Si veda la sezione 3.1.4.2 Versionamento per maggiori dettagli sul numero di versione indicato nei documenti.

##### 2.1.1.3 Pianificazione

**2.1.1.3.1 Scelta del modello di ciclo di vita** Il *Responsabile di Progetto* ha il compito di scegliere il modello di ciclo di vita<sub>G</sub> adatto per lo sviluppo del prodotto richiesto, a meno che non venga fornita un'indicazione dal *Proponente<sub>G</sub>*.



**2.1.1.3.2 Sviluppo e documentazione del Piano di Progetto** Il *Responsabile di Progetto* ha il compito di delineare i lavori che i membri del gruppo devono eseguire. Inoltre deve calcolare costi e tempistiche per le attività da svolgere. Tali pianificazioni sono trascritte nel documento *Piano di Progetto v1.0.0*.

## 2.2 Sviluppo

Il processo di sviluppo contiene le attività che definiscono la realizzazione del prodotto *software* da parte del gruppo *Stark Labs*.

### 2.2.1 Attività

**2.2.1.1 Analisi dei requisiti** Dopo aver redatto lo *Studio di Fattibilità*, gli *Analisti* devono produrre il documento *Analisi dei Requisiti v1.0.0*. L'obiettivo è produrre dei requisiti a partire dalle informazioni acquisite dal gruppo. Le risorse utilizzabili per questo scopo provengono dal capitolato d'appalto e dagli incontri con il Proponente<sub>G</sub> e con il Committente<sub>G</sub>.

### 2.2.2 Norme

**2.2.2.1 Classificazione requisiti** I requisiti prodotti devono essere classificati a seconda del tipo e dell'importanza, rispettando la seguente notazione:

R[Importanza][Tipo][Codice]

- **Importanza:** i valori che può assumere sono:
  - 0: requisito obbligatorio;
  - 1: requisito desiderabile;
  - 2: requisito opzionale.
- **Tipo:** i valori che può assumere sono:
  - F: requisito funzionale;
  - P: requisito prestazionale;
  - Q: requisito di qualità;
  - V: requisito di vincolo.
- **Codice** : è il codice gerarchico e univoco del vincolo espresso nella forma X.X.X dove X è un valore numerico.

Ogni requisito inoltre deve contenere le seguenti informazioni:

- **Descrizione:** descrizione del requisito con la minore ambiguità possibile;
- **Fonte:** la scelta può ricadere tra:
  - **Capitolato:** requisito ottenuto dalle specifiche del capitolato;
  - **Interno:** requisito elaborato dagli *Analisti* nel corso di un'analisi più approfondita del problema;
  - **Caso d'uso:** requisito ottenuto da uno o più casi d'uso. Deve essere quindi specificato il codice del caso d'uso a cui ci si riferisce;
  - **Verbale:** requisito ottenuto da un'incontro con il Proponente<sub>G</sub> o da riunioni interne tra i membri del gruppo.

**2.2.2.2 Classificazione casi d'uso** I casi d'uso devono essere suddivisi in ordine gerarchico secondo il seguente schema:

UC[Codice]

- **Codice:** è il codice gerarchico e univoco che serve a identificare ogni caso d'uso.

Per ogni caso d'uso devono essere presenti anche le seguenti informazioni:

- **Titolo:** è necessario fornire un titolo riassuntivo dell'operazione che il caso d'uso intende modellare;
- **Descrizione:** è necessario fornire una breve descrizione con la minore ambiguità possibile;
- **Attori principali:** elenco degli attori principali coinvolti nel caso d'uso;
- **Attori secondari:** elenco degli attori secondari coinvolti nel caso d'uso;
- **Scenari principali:** descrizione dei possibili scenari principali;
- **Scenari alternativi:** descrizione dei possibili scenari secondari;
- **Pre-condizioni:** una pre-condizione è una condizione sempre vera all'inizio del caso d'uso;
- **Flusso degli eventi:** ordine di esecuzione dei casi d'uso figli;
- **Inclusioni:** spiegazione di tutte le inclusioni se presenti;
- **Estensioni:** spiegazione di tutte le estensioni se presenti;
- **Generalizzazioni:** spiegazione di tutte le generalizzazioni se presenti;
- **Post-condizioni:** una post-condizione è una condizione sempre vera alla fine dell'esecuzione del caso d'uso.

Alcune fra le precedenti informazioni potrebbero essere assenti nel caso non fossero utilizzate.

**2.2.2.3 Codifica file** Tutti i *file* creati dal gruppo contenenti sia codice che documentazione devono essere codificati tramite UTF-8<sub>G</sub> senza BOM<sub>G</sub>.

Eventuali cambiamenti di codifica devono essere approvati dal *Responsabile di Progetto*.

**2.2.2.4 Nomi e norme di codifica** Questa sezione verrà redatta nel dettaglio durante le fasi successive. Vengono introdotte alcune norme che andranno applicate per lo sviluppo del codice, indipendentemente dal linguaggio di programmazione che verrà adottato.

Le norme introdotte sono le seguenti:

- In ogni *file* deve essere presente un'intestazione contenente le seguenti informazioni:
  - Percorso e nome del *file*;
  - Cognome e nome dell'autore;
  - Data di creazione;
  - Indirizzo email dell'autore;
  - Per ogni modifica effettuata devono essere specificati: la versione successiva generata dall'avanzamento, l'autore, la data e una breve descrizione.
- I nomi delle variabili devono essere chiari, descrittivi e in inglese;
- I commenti vanno scritti in italiano.

**2.2.2.5 Ricorsione** Quando possibile, la ricorsione va evitata. Per ogni funzione ricorsiva è d'obbligo fornire una prova di terminazione. Risulta inoltre necessario valutare il costo in termini di occupazione della memoria. Nel caso in cui l'utilizzo di memoria risulti eccessivo, la ricorsione deve essere rimossa.

### **2.2.3 Strumenti**

**2.2.3.1 Requirements Tool** COMPLETARE CON SPIEGAZIONE DELL'APPLICAZIONE PER MEMORIZZARE I REQUISITI

**2.2.3.2 PhpStorm ?** COMPLETARE DESCRIVENDO L'IDE UTILIZZATO PER REALIZZARE L'APPLICAZIONE PRECEDENTE.

## 3 Processi di supporto

### 3.1 Documentazione

#### 3.1.1 Attività

**3.1.1.1 Documentazione** Il gruppo di lavoro *Stark Labs* si impegna a registrare tutte le informazioni acquisite nel corso del ciclo di vita<sub>G</sub> del *software* all'interno di una serie definita di documenti che verranno descritti in questa sezione. Nello specifico, vengono spiegati gli standard da rispettare per la stesura dei documenti e i passaggi necessari per renderli formalmente corretti.

#### 3.1.2 Procedure

**3.1.2.1 Gestione dei documenti** La stesura di un nuovo documento viene decisa dal *Responsabile di Progetto*. Tutta la documentazione deve essere creata attraverso l'uso di un *template<sub>G</sub>*  $\text{\LaTeX}$  disponibile nel *repository<sub>G</sub>* *GitHub<sub>G</sub>* del gruppo, al fine di mantenerne uniforme la struttura e lo stile.

**3.1.2.2 Creazione di un nuovo documento** Nel *repository<sub>G</sub>* è presente un *file* generico denominato *new\_doc.tex* contenente un *template* adattabile ad ogni nuovo documento. Il *file* deve essere incluso nella cartella contenente il *template*  $\text{\LaTeX}$  che è messo a disposizione a ogni membro del gruppo. Ogni redattore deve possedere tutti i *file* previsti dal *template* al fine di creare correttamente un nuovo documento.

**3.1.2.3 Avanzamento di un documento** Le procedure adottate per sviluppare un documento sono le seguenti:

- Il *Responsabile di Progetto* si preoccupa di assegnare la stesura del documento a uno o più redattori, a seconda delle complessità dello stesso. L'assegnazione è gestita attraverso *task<sub>G</sub>* organizzati sul servizio *online* *Teamwork<sub>G</sub>*;
- A stesura completata, ogni assegnatario ha il compito di segnalare il *task* come completato;
- Il *Verificatore* riceve in automatico un'email che segnala il completamento del documento;
- Nel caso si riscontrassero errori durante la fase di verifica, il *Verificatore* deve occuparsi di creare un nuovo *task* da assegnare al redattore del documento;
- Nel caso di documento corretto, il *Verificatore* deve completare il *task* assegnatogli. Un'email viene quindi automaticamente recapitata al *Responsabile di Progetto*.
- Il *Responsabile di Progetto* deve infine approvare il documento. In caso di mancata approvazione, il *Verificatore* dovrà creare un nuovo *task* indirizzato al redattore, che si occuperà di correggere gli errori in base alle segnalazioni emesse.

#### 3.1.3 Gestione del glossario

Il popolamento del *Glossario v1.0.0* è un'attività che coinvolge redattori e *Verificatori*.

- **Individuazione di un nuovo termine:** se durante la stesura del documento il redattore identifica un nuovo termine che ritiene debba essere inserito nel *Glossario v1.0.0*, è tenuto a trascriverlo all'interno di un apposito documento (*Glossario Provvisorio*) presente nella sezione *Notebooks* di *Teamwork<sub>G</sub>*.
- **Inserimento del termine nei documenti:** un *Verificatore* deve occuparsi dell'approvazione di un dato termine presente in *Glossario Provvisorio* di *Teamwork* e dell'inserimento dello stesso all'interno del *Glossario v1.0.0*. Una volta inserito, il termine deve essere rimosso dal documento di *Teamwork*.

### 3.1.4 Norme

**3.1.4.1 Progettazione e sviluppo dei documenti** Ogni documento deve rispettare rigidamente la seguente serie di norme.

**3.1.4.2 Versionamento** Ogni documento prodotto deve essere corredato dal numero di versione. Il formato adottato è il seguente:

vX.Y.Z

tale che:

- **X:** indice di versione principale. Tale valore viene incrementato ad ogni approvazione del documento e ne indica la versione di rilascio;
- **Y:** indice di modifica parziale. Tale valore viene incrementato ad ogni verifica del documento;
- **Z:** indice di modifica minore. Tale valore viene incrementato ad ogni cambiamento che avviene del documento.

**3.1.4.3 Template** La creazione dei documenti avviene attraverso l'utilizzo di un *template<sub>G</sub>* sviluppato con  $\text{\LaTeX}$  e la cui struttura **non** deve essere modificata a meno di direttive imposte dal *Responsabile di Progetto*. Il *template* funge da supporto per la stesura organizzata e sistematica dei documenti e, grazie ad esso, ogni componente del documento ha una precisa impostazione che non può essere modificata o equivocata dai redattori.

**3.1.4.4 Struttura dei documenti** L'organizzazione dei documenti è la seguente:

- Una cartella generale in cui sono contenuti il *template<sub>G</sub>*  $\text{\LaTeX}$  e le varie cartelle specifiche di ogni documento;
- Nella cartella *Template* sono contenuti i *file* di configurazione e strutturazione del *template*. Il contenuto della cartella non deve essere modificato, previa autorizzazione da parte del *Responsabile di Progetto*;
- Nelle cartella specifica di ogni documento è presente un *file* di tipo nome\_documento.tex tramite cui si determina la struttura dello specifico documento accompagnato da una sotto cartella "sezioni" che contiene le varie sezioni nelle quali sono scritti i contenuti veri e propri.

**3.1.4.4.1 Prima pagina** Nella prima pagina sono presenti tutte le informazioni generali relative al documento:

- Nome del progetto;
- Logo del gruppo;
- Nome del documento;
- Versione del documento;
- Membri del gruppo che hanno lavorato come redattori, *Verificatori* e *Responsabili* per la stesura del documento. I nomi vanno scritti nel formato Nome Cognome;
- Specifica dell'uso del documento (interno o esterno);
- Lista di distribuzione del documento. I nomi vanno scritti nel formato Nome Cognome;
- Breve descrizione che identifica lo scopo del documento.

**3.1.4.4.2 Registro delle modifiche** Nella seconda pagina si trova una tabella con il registro delle modifiche effettuate al documento. Esso è indispensabile per un corretto tracciamento delle varie fasi che si sono percorse lungo la sua stesura. Ogni riga corrisponde a una modifica dove vengono segnalati:

- Descrizione dell'azione compiuta sul documento;
- Nome e cognome dell'autore della modifica;
- Data della modifica;
- Versione del documento a seguito della modifica.

**3.1.4.4.3 Indici** In terza pagina è presente l'indice che tiene traccia delle varie sezioni in cui è stato suddiviso il documento. La profondità dell'indice arriva a cinque livelli: gli argomenti trattati sono suddivisi in sezioni, sottosezioni, sotto-sottosezioni, paragrafi e sottoparagrafi. L'unico documento in cui non è presente l'indice è il *Glossario v1.0.0*. Se presenti sono contenute immagini e/o tabelle, è possibile trovare automaticamente il relativo indice associato.

**3.1.4.4.4 Formattazione di una pagina** Tutte le pagine dei documenti seguono una precisa formattazione imposta dal *template<sub>G</sub>*:

- **Intestazione:** contiene sulla sinistra il logo del gruppo e sulla destra il nome della sezione in cui è contenuta la pagina;
- **Contenuto:** contiene il contenuto effettivo della pagina le cui norme tipografiche sono descritte in 3.1.4.5.6 Norme Tipografiche;
- **Piè di pagina:** contiene sulla sinistra il nome del documento accompagnato dal numero della versione e sulla destra il numero della pagina, scritto nel formato "X di Y", con X numero di pagina corrente e Y numero delle pagine totali.

La suddetta struttura si ripete per ogni pagina ad eccezione della prima, il cui formato è descritto in 3.1.4.4.1 Prima Pagina.

### 3.1.4.5 Suddivisione dei documenti

**3.1.4.5.1 Norme di Progetto** Il documento ha lo scopo di definire le linee guida per le varie attività di sviluppo. Al suo interno sono raccolte le norme, le procedure e gli strumenti che il gruppo adotterà nel corso della realizzazione del progetto. Il documento è destinato a uso interno.

**3.1.4.5.2 Studio di Fattibilità** Il documento ha lo scopo di descrivere le considerazioni elaborate dal gruppo per l'accettazione del progetto che si è deciso di prendere in carico, con valutazione di rischi, costi e benefici calcolati sulla base di una prima analisi del capitolato. Al suo interno vengono motivate le scelte che hanno spinto il gruppo all'esclusione degli altri progetti. Il documento è destinato a uso interno.

**3.1.4.5.3 Analisi dei Requisiti** Il documento ha lo scopo di identificare e descrivere i requisiti, i vincoli e gli obiettivi necessari allo sviluppo del progetto. Al suo interno sono contenuti i casi d'uso e i requisiti utili alla realizzazione del progetto, accompagnati da diagrammi e grafici di interazione fra utenti e sistema. Il documento è destinato a uso esterno.

**3.1.4.5.4 Piano di Progetto** Il documento ha lo scopo di pianificare lo svolgimento del progetto. Al suo interno sono fissate le risorse disponibili, la suddivisione e il calendario delle attività, e gli obiettivi necessari per valutare in modo corretto il grado di avanzamento dello sviluppo. Il documento è destinato a uso esterno.

**3.1.4.5.5 Piano di Qualifica** Il documento ha lo scopo di spiegare le strategie applicate al progetto per ottenere gli obiettivi di qualità. Al suo interno sono presenti le attività di verifica e pianificazione con i relativi test da sviluppare. Il documento è destinato a uso esterno.

**3.1.4.5.6 Specifica Tecnica** Il documento ha lo scopo di descrivere l'architettura logica del progetto, senza fissare i dettagli implementativi, ma definendo linee e strategie di realizzazione, al fine di stabilire cause ed effetti e avere una visione complessiva della soluzione. Al suo interno è contenuta una prima progettazione ad alto livello del sistema da sviluppare. In esso vengono specificati i *design pattern*<sub>G</sub> utilizzati. Il documento è destinato a uso esterno.

**3.1.4.5.7 Definizione di Prodotto** Il documento ha lo scopo di descrivere nel dettaglio l'architettura del prodotto da sviluppare. Il suo contenuto viene utilizzato dai *Programmatore* per sviluppare il *software*. Il documento è destinato a uso esterno.

**3.1.4.5.8 Glossario** Il documento ha lo scopo di raccogliere, in ordine alfabetico, tutti i termini ambigui presenti nei documenti accompagnati da una loro definizione. Il documento è destinato a uso esterno.

**3.1.4.6 Norme tipografiche** Al fine di rendere omogenea e coesa la stesura dei documenti, il contenuto deve rispettare le seguenti norme tipografiche.

#### 3.1.4.6.1 Stile del testo

- **Corsivo:** va utilizzato tassativamente per indicare termini in lingua inglese, citazioni, nomi di documenti interni, ruoli dei membri del gruppo, nome del *team* ed eventualmente (ma con moderazione) per parole che si ritiene debbano essere messe in risalto rispetto al resto del testo;
- **Grassetto:** va usato per evidenziare parole significative di estrema importanza. È importante che non se ne abusì. Viene applicato automaticamente a titoli di sezioni, sottosezioni e paragrafi;
- **Maiuscolo:** viene utilizzato unicamente per scrivere acronimi e macro  $\LaTeX$ ;
- **Monospace<sub>G</sub>:** serve per formattare il testo contenente porzioni di codice, percorsi dei *file*, comandi e indirizzi *web*;
- $\LaTeX$ : viene usato il comando `\LaTeX` per ogni occorrenza del termine  $\LaTeX$ .
- **Font:** nel *template* è impostato Gillius, un *font* professionale di tipo *sans-serif* il cui scopo è garantire maggiore leggibilità dei documenti su schermo.

#### 3.1.4.6.2 Punteggiatura

- **Spaziatura:** lo spazio non può mai precedere un carattere di punteggiatura;
- **Maiuscolo:** vanno utilizzate lettere maiuscole per riferirsi ai ruoli di progetto, alle fasi di lavoro e ai seguenti nomi: del *team*, del progetto e dei documenti.
- **Numerazione:** viene utilizzato lo standard internazionale SI/ISO 31-0 per indicare quantità e unità di misura.

#### 3.1.4.6.3 Composizione del testo

- **Elenchi puntati:** l'ultima voce deve terminare con un punto, mentre le altre con un punto e virgola. La prima lettera di ogni punto va scritta in maiuscolo e la prima parola va in grassetto se seguita da una descrizione della stessa. elenchi numerati... non numerati...?
- **Pedice G:** il pedice <sub>G</sub> è utilizzato al solo scopo di indicare termini potenzialmente ambigui contenuti nel documento *Glossario v1.0.0*.

#### 3.1.4.6.4 Formati Elenco dei formati rispettati dai documenti.

- **Data:** il formato utilizzato è dd/mm/yyyy. Per esempio, 04/06/1996 indica il 4 giugno 1996;
- **Ora:** si utilizza il formato internazionale previsto dalla norma ISO 8601 del tipo [hh]:[mm]:[ss] ove [hh] indica l'ora, [mm] i minuti, [ss] i secondi, espressi con due cifre. Ad esempio, 18:35:26 indica le ore 18, 35 minuti e 26 secondi; 04:09:01 indica le ore 4, 9 minuti e 1 secondo;
- **Percorsi:** viene utilizzato il separatore *slash* (/) per indicare il percorso di un file. Per esempio, cartella1/cartella2/file\_esempio.txt;
- **Nomi di persona:** espressi nel formato Nome Cognome.

#### 3.1.4.6.5 Sigle Elenco delle sigle che possono apparire nel corso dei documenti.

- **AdR:** Analisi dei Requisiti;
- **GL:** Glossario;
- **NdP:** Norme di Progetto;
- **PdP:** Piano di Progetto;
- **PdQ:** Piano di Qualifica;
- **SdF:** Studio di Fattibilità;
- **ST:** Specifica Tecnica;
- **RA:** Revisione di Accettazione;
- **RP:** Revisione di Progettazione;
- **RQ:** Revisione di Qualifica;
- **RR:** Revisione dei Requisiti.

#### 3.1.4.7 Componenti grafiche

**3.1.4.7.1 Immagini** Tutte le immagini incluse nei documenti devono essere salvate in formato *Portable Network Graphics* (PNG<sub>G</sub>).

#### 3.1.4.7.2 Diagrammi

### 3.2 Processo di verifica e validazione

Il processo di verifica consiste nel controllare che il materiale prodotto al raggiungimento delle *milestone<sub>G</sub>* sia conforme agli obiettivi prefissati. Pertanto, è necessario verificare che non siano stati prodotti errori. Il prodotto è validato se il risultato ottenuto è consistente e conforme alle attese. Una corretta applicazione del processo di verifica genera un aumento del rapporto fra efficienza ed efficacia, riducendo il tempo impiegato nel percorso di analisi.



### 3.2.1 Attività

**3.2.1.1 Analisi statica** L'attività di analisi statica è una tecnica di verifica applicabile sia a documenti che a codice sorgente che va ad analizzare il solo testo del *file* senza mandarlo in esecuzione. Tale tecnica viene utilizzata durante l'intero sviluppo del progetto e si pone come obiettivo il ritrovamento di eventuali anomalie. I metodi di controllo sono i seguenti:

- **Walkthrough:** si ricercano all'interno di testo o codice tutte le possibili anomalie; l'attività è eseguita da un umano. L'analisi si basa sulla lettura di tutto il contenuto del *file*. In seguito al ritrovamento di anomalie, le si analizza con il redattore del documento (o *Programmatore* nel caso di codice) e si indaga per raggiungere una soluzione del problema. Tale tecnica risulta utile durante le prime fasi di sviluppo del progetto, in quanto manca, ai componenti, una visione complessiva del documento o del codice che si sta scrivendo. Permette così ai *Verificatori*, dopo aver svolto le prime correzioni, di preparare una *lista di controllo* con gli errori più frequenti in modo da migliorare l'efficienza delle verifiche future. A questo punto è possibile implementare un'analisi mirata e più efficiente attraverso il metodo dell'*Inspection*.
- **Inspection:** si ricercano all'interno di un testo errori specifici; l'attività può essere eseguita sia da un umano che, nel caso di anomalie nella sintassi, da uno script. Il metodo focalizza la ricerca su errori presupposti identificati dalla *lista di controllo*.

**3.2.1.2 Analisi dinamica** L'attività di analisi dinamica è una tecnica di verifica applicabile solamente al *software*. Tale tecnica può essere utilizzata per analizzare l'intero *software* o una porzione limitata dello stesso. L'attività consiste nell'esecuzione di *test* automatici realizzati dal *team*. Le verifiche devono essere effettuate su un insieme finito di casi, con valori di ingresso, uno stato iniziale e un esito decidibile. Tutti i *test* producono risultati automatici che inviano notifiche sulla tipologia di problema individuato. Ogni *test* è ripetibile, ossia applicabile durante l'intero ciclo di vita<sub>G</sub> del *software*. Le caratteristiche da rispettare sono le seguenti:

- **Ambiente:** è necessario riportare l'ambiente sia *software* che *hardware* in cui il sistema esegue il *test*. Deve essere specificato lo stato iniziale del sistema;
- **Specificità:** è necessario riportare i dati in ingresso e in uscita per verificare l'esito del *test*, ossia se il codice analizzato è conforme alle aspettative;
- **Procedure:** è possibile specificare ulteriori istruzioni per l'esecuzione dei *test*. Inoltre possono essere riportate istruzioni sulle corretta lettura dei risultati.

**3.2.1.3 Gestione anomalie** Se si dovessero riscontrare anomalie o discordanze normative durante le attività di verifica, il *Verificatore* ha il dovere di notificarle all'assegnatario del *task<sub>G</sub>*

**3.2.1.4 Tracciamento** L'attività di tracciamento svolta dai *Verificatori* consiste nella catalogazione di tutti i casi d'uso e i requisiti che ne derivano ed evidenziare la corrispondenza fra di essi in modo da avere ben chiaro i processi di derivazione.

### 3.2.2 Procedure

#### 3.2.2.1 Procedure di assegnazione delle anomalie

#### 3.2.3 Norme

##### 3.2.3.1 Priorità risoluzione anomalie

#### 3.2.4 Strumenti

##### 3.2.4.1 Correzione ortografica

##### 3.2.4.2 Calcolo indice Gulpease

### 3.2.4.3 Database online

## 4 Processi organizzativi

### 4.1 Gestione dei processi

#### 4.1.1 Attività

##### 4.1.1.1 Gestione delle comunicazioni

**4.1.1.1.1 Comunicazione interna** Viene utilizzato un gruppo Telegram<sub>G</sub> per comunicare informalmente fra i membri del *team*. Il servizio fornisce il vantaggio di essere un'applicazione multi-piattaforma<sub>G</sub> e disponibile in versione sia *desktop* che *web*. Inoltre è possibile catalogare e tenere traccia degli argomenti tramite l'uso di *hashtag* (per esempio: #analisiRequisiti), e chiamare all'attenzione un dato membro attraverso l'utilizzo del comando chiocciola (per esempio: @RiccardoRizzo).

**4.1.1.1.2 Comunicazione esterna** Il *Responsabile di Progetto* è la persona preposta a mantenere i contatti con individui esterni al gruppo. Per tali comunicazioni è stato creato il seguente indirizzo di posta elettronica:

starklabs.swe@gmail.com

Tutti i componenti del gruppo possono accedere all'indirizzo di posta, tuttavia solo il *Responsabile di Progetto* può inviare comunicazioni con tale indirizzo email. Tutte le email ricevute verranno automaticamente inoltrate agli indirizzi personali dei membri del *team*.

##### 4.1.1.1.3 Composizione email

- **Destinatario:**

- **Esterno:** i destinatari possono essere il Proponente (Giulio Paci e l'azienda MIVOQ s.r.l.), il Prof. Tullio Vardanega e il Prof. Riccardo Cardin.

- **Mittente:**

- **Esterno:** l'unico indirizzo utilizzabile è starklabs.swe@gmail.com e può essere usato solamente dal *Responsabile di Progetto*.

- **Oggetto:** l'oggetto deve contenere la dicitura [UNIPD-TTS] così come è stato specificato nel capitolato dell'azienda Proponente. Nel caso il messaggio sia una risposta è consigliabile aggiungere la particella "Re:" all'inizio dell'oggetto per rendere chiara la distinzione del livello di risposta; se si dovesse trattare di un inoltro si deve usare la particella "I:". L'oggetto non va mai cambiato.

- **Corpo:** in caso di risposta da parte dell'azienda MIVOQ o del fornitore, risulta utile la citazione della frase a cui si intende rispondere. Il modello per citare correttamente una porzione del messaggio deve seguire le seguenti regole: devono essere presenti data e ora della mail a cui si risponde, il nome del mittente e il suo indirizzo email tra parentesi angolari. Per esempio: <starklabs.swe@gmail.com>, la dicitura "ha scritto:" e infine il testo con all'inizio una parentesi angolare chiusa (">testo di prova"). Se dovessero essere presenti alcune parti con uno o più destinatari specifici, il nome dovrà essere indicato all'inizio del paragrafo attraverso la dicitura: @destinatario.

- **Allegati:** qualora vi fosse necessità, è possibile allegare alcuni *file* al messaggio email. Possono per esempio essere allegati i verbali di eventuali incontri con Proponente o Committente, oppure *file* facenti parte della documentazione spiegata in sezione 3.1.4.5 Suddivisione dei documenti.

##### 4.1.1.2 Gestione delle riunioni

#### 4.1.1.2.1 Riunioni interne

- **Frequenza:** le riunioni del gruppo di lavoro avranno cadenza settimanale;
- **Convocazione:** Il *Responsabile di Progetto* ha il compito di convocare le riunioni generali, a cui dovranno partecipare tutti i membri del gruppo. Su decisione del *Responsabile di Progetto* le riunioni possono coinvolgere anche solo specifici componenti del gruppo, a seconda del ruolo che si ritiene più utile in una data fase del progetto. Al termine di ogni riunione viene redatto un verbale. Il *Responsabile* deve convocare l'assemblea con almeno un giorno di preavviso attraverso l'invio di una comunicazione ufficiale nel gruppo Telegram<sub>G</sub>, messa in rilievo tramite l'uso dell'*hashtag* #riunioneDATA, con DATA espressa nel formato d/mmmm/yyyy. Per esempio: #riunione1marzo2016 raccoglie i messaggi inerenti alla riunione tenutasi in data 1 marzo 2016. Nel corpo del messaggio deve essere specificato:
  - **Data:** data e ora prevista;
  - **Luogo:** luogo previsto;
  - **Tipo:** ordinaria o straordinaria;
  - **Ordine del giorno:** elenco ordinato delle voci da esaminare.

Ogni componente del gruppo deve rispondere al messaggio nel più breve tempo possibile, confermando la presenza o giustificando l'eventuale assenza. In assenza di una risposta di uno o più membri entro 24 ore, il *Responsabile di Progetto* ha il compito di contattare telefonicamente gli interessati. Una volta ricevute le risposte e verificata l'assenza o presenza dei membri convocati, il *Responsabile di Progetto* ha la possibilità di decidere se confermare o posticipare la riunione, in modo da permettere la presenza di tutti i membri chiamati; tutte le eventuali modifiche dovranno essere notificate tramite lo stesso *hashtag* utilizzato per organizzare la riunione.

- **Verbale:** il verbale di riunione interna si presenta in forma di documento informale, utile al solo scopo di fissare in modo ordinato i punti principali trattati e le relative soluzioni proposte. Verrà redatto come documento testuale utilizzando la funzione Notebooks<sub>G</sub> di TeamWork<sub>G</sub> permettendo così la condivisione, tra tutti i membri del gruppo, di un documento aggiornato all'ultima modifica dal segretario della riunione, ruolo scelto a rotazione tra i membri convocati. È inoltre compito del segretario annotare ogni argomento trattato e controllare che venga rispettato l'ordine del giorno.

#### 4.1.1.2.2 Riunioni esterne

- **Convocazione:** in questo caso viene utilizzata l'email starklabs.swe@gmail.com attraverso la quale il *Responsabile di Progetto* si occupa di contattare l'azienda Proponente e di mettere in cc<sub>G</sub> i membri del gruppo. Per quanto sia auspicabile una riunione plenaria, eventuali assenze dei componenti del *team* non causeranno posticipazioni o spostamenti delle date di incontro, dovendo ovviamente considerare gli impegni dell'azienda stessa.
- **Verbale:** in caso di riunione con il Committente o il Proponente, il verbale è un documento che assume carattere ufficiale e che pertanto viene redatto secondo uno schema specifico. Per agevolare la scrittura di tale documento viene utilizzato un *template*<sub>G</sub>  $\LaTeX$  per definire la struttura e organizzare i contenuti. Tale documento dovrà essere redatto e inviato come allegato in risposta all'email di convocazione dell'assemblea al Proponente Giulio Paci direttamente dal segretario scelto tra i membri presenti.

**4.1.1.3 Gestione del sistema dei task** Il sistema selezionato per la gestione dei *task*<sub>G</sub> è Teamwork<sub>G</sub>, un servizio *web* di *project management*<sub>G</sub>. Le viste presenti sono:

- **Dashboard:** dove vengono visualizzati i progetti attivi e le ultime notizie relative ad essi;
- **Everything:** che consente di visualizzare i *task*, le *milestone*<sub>G</sub> e i *file* con possibilità di filtrarli per data;

- **Project:** permette di visualizzare la lista di tutti i progetti suddivisi per categoria e ne permette l'accesso;
- **Calendar:** mostra un calendario per la gestione degli impegni e delle scadenze;
- **Statuses:** consente di verificare gli stati dei collaboratori del progetto;
- **People:** permette di visualizzare l'elenco dei singoli elementi del gruppo di lavoro e di accedere al loro profilo.

Le funzionalità principali si hanno una volta avuto accesso al progetto desiderato. Esse si suddividono nei seguenti punti:

- Aggiunta di nuovi *task*, ed eventualmente di sotto *task*, da associare ad uno o più membri del *team*;
- Assegnazione a ciascun *task* di una data d'inizio e di completamento;
- Aggiunta di nuove *milestone* con relativi dettagli come responsabile, descrizione e data di scadenza;
- *Upload<sub>G</sub>* di file potenzialmente utili al gruppo di lavoro;
- Utilizzo di un blocco note.

**4.1.1.4 Gestione delle *milestone*** Il *Responsabile di Progetto* ha il compito pianificare i punti di controllo che il *team* deve raggiungere, assicurandosi che ogni *task<sub>G</sub>* necessario al suo soddisfacimento venga terminato entro la data prestabilita.

**4.1.1.5 Gestione dei *task*** È compito del *Responsabile di Progetto* individuare ogni singolo *task<sub>G</sub>* e, al seguito di un'accurata valutazione, assegnarlo al membro del gruppo più adatto. Deve inoltre associare una data di inizio e di scadenza di fattibilità. Tutte queste attività possono essere facilmente accedute attraverso l'interfaccia grafica di *Teamwork<sub>G</sub>*.

**4.1.1.6 Gestione dello svolgimento dei *task*** Ogni membro del gruppo di lavoro è tenuto ad accettare il *task<sub>G</sub>* assegnatigli dal *Responsabile di Progetto* e fare quanto possibile per portarlo a termine entro la data di scadenza. Nel caso in cui l'assegnatario non fosse in grado di adempiere al suo compito è tenuto a renderlo noto al *Responsabile di Progetto* entro 24 ore dall'assegnazione del *task*, altrimenti quest'ultimo verrà considerato come accettato; solo dopo un'accurata valutazione delle motivazioni riportate, il *Responsabile di Progetto* può provvedere a trovare un nuovo membro da incaricare allo svolgimento del *task*.

## 4.1.2 Procedure

**4.1.2.1 Generazione di una *milestone*** Il *Responsabile di Progetto* dovrà eseguire i seguenti passi per generare una *milestone<sub>G</sub>* dopo aver avuto accesso al progetto a *Teamwork<sub>G</sub>*:

- Cliccare sul pulsante "*Add a milestone*";
- Definire il titolo della *milestone<sub>G</sub>*;
- Definire la data di scadenza;
- Assegnare un responsabile.

**4.1.2.2 Assegnazione di un task** Dopo essere acceduto al progetto di Teamwork<sub>G</sub>, il *Responsabile di Progetto* deve eseguire i seguenti passi, riassunti in Figura 1, per generare un *task<sub>G</sub>*

- Cliccare sul pulsante "Add a task";
- Definire il titolo del *task*;
- Definire la data di inizio;
- Definire la data di scadenza;
- Assegnarlo ad uno o più membri del gruppo.

**4.1.2.3 Svolgimento di un task** Il membro assegnatario del *task<sub>G</sub>*, ricevuta la notifica e non avendo alcun impedimento, deve procedere secondo le seguenti direttive:

- Se il *task* ricevuto ha una scadenza più immediata rispetto a quello su cui sta lavorando, deve sospendere lo svolgimento di quest'ultimo, metterlo in coda e dedicarsi al *task* appena notificato;
- Se, dopo aver iniziato lo svolgimento del *task*, si riceve la notifica di uno nuovo con scadenza più immediata si procede come riportato nel punto precedente;
- Se si dovesse superare la data di scadenza prevista, è necessario impostare il *tag<sub>G</sub>* "Delay" dal sistema offerto su Teamwork<sub>G</sub>.

Questa situazione si può verificare se:

- Il tempo assegnato dal *Responsabile di progetto* non è sufficiente al completamento del *task*;
- Il *task* in ritardo sta alle dipendenze di un altro *task* non ancora completato;
- L'assegnatario è rallentato da cause esterne non rese note al *Responsabile di Progetto*;
- Il membro incaricato non ha a disposizione tutte le conoscenze necessarie per un corretto svolgimento del *task*.

Spetta al *Responsabile di Progetto* fare in modo che i primi due casi non si verifichino.

- Al completamento del lavoro l'assegnatario deve spuntare il *task* dalla lista presente su Teamwork<sub>G</sub>;
- A questo punto può proseguire con lo svolgimento dei *task* rimanenti riprendendo la procedura dall'inizio.

**4.1.2.4 Rilevamento dei rischi** È compito del *Responsabile di Progetto* individuare i rischi trovati nel *Piano di Progetto v1.0.0*. Questa attività necessita di un continuo monitoraggio, in quanto è plausibile che insorgano nuovi rischi in seguito a quelli rilevati nella fase preliminare. In tal caso il *Responsabile di Progetto* deve agire come segue:

- Registrare il resoconto effettivo dei rischi nel *Piano di Progetto v1.0.0*;
- Pianificare per gestire i nuovi rischi;
- Aggiornare le metodologie per far fronte alla nuova pianificazione;
- Monitorare i nuovi rischi riscontrati durante lo sviluppo del progetto.

**4.1.2.5 Ruoli di Progetto** Ogni componente del gruppo *Stark Labs* deve ricoprire almeno una volta ciascuno dei ruoli necessari allo sviluppo del progetto. Di seguito vengono presentati i diversi incarichi, delineando per ciascuno mansioni e responsabilità.

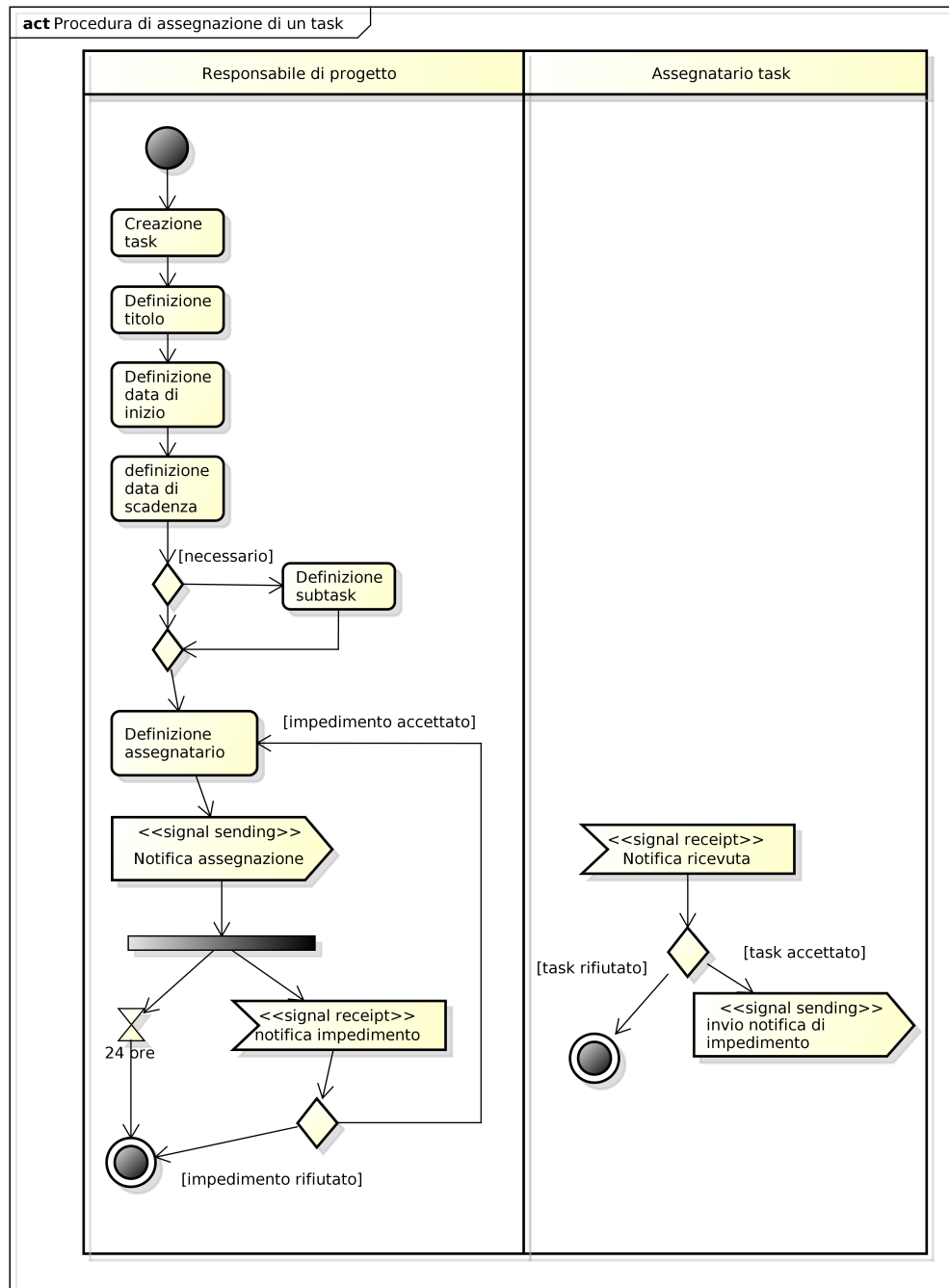


Figura 1: Diagramma di attività - Procedura di assegnazione di un task

**4.1.2.5.1 Responsabile di Progetto** Il *Responsabile di Progetto* rappresenta il  $team_G$  e il progetto nei confronti di Committente e Proponente; accentra le responsabilità di scelta e approvazione. Detiene inoltre le seguenti responsabilità:

- Pianificazione e coordinamento delle attività;
- Gestione e controllo delle risorse;
- Analisi e gestione dei rischi;
- Approvazione dei documenti;
- Assicurarsi che tutte le attività svolte siano conformi alle *Norme di Progetto v1.0.0* e rispettino la pianificazione effettuata nel *Piano di Progetto v1.0.0*.

**4.1.2.5.2 Amministratore di Progetto** L' *Amministratore di Progetto* deve svolgere i seguenti compiti:

- Assicurarsi che tutte le risorse siano presenti e operanti;
- Garantire un'infrastruttura funzionale;
- Fornire procedure che servono a garantire la qualità del prodotto uscente da un determinato compito.

**4.1.2.5.3 Analista** L' *Analista* deve svolgere i seguenti compiti:

- Tradurre il bisogno del cliente in una specifica utile per trovare una soluzione;
- Comprendere la complessità del problema;
- Capire il dominio nel quale lavora il cliente;
- Analizzare il dominio applicativo e le specifiche per poi produrre i documenti di analisi.

### 4.1.3 Norme

**4.1.3.0.4 Progettista** Il *Progettista* deve svolgere i seguenti compiti:

- Individuare la tecnologia più idonea per risolvere il problema indicato dall'*Analista*;
- Descrivere il funzionamento interno del sistema a diversi livelli di dettaglio;
- Produrre una soluzione comprensibile e attuabile.

**4.1.3.0.5 Programmatore** Il *Programmatore* ha responsabilità sulle attività di codifica e pertanto deve svolgere i seguenti compiti:

- Scrivere codice documentato, versionato e manutenibile;
- Implementare le soluzioni descritte dal *Progettista*;
- Implementare i test sul codice prodotto.

**4.1.3.0.6 Verificatore** Il *Verificatore* è il responsabile delle attività di verifica e pertanto deve svolgere i seguenti compiti:

- Controllare che vengano rispettate le norme di progetto;
- Assicurarsi la conformità di ogni stadio del ciclo di vita $_G$  del prodotto.

#### 4.1.4 Strumenti

**4.1.4.1 Teamwork** Teamwork<sub>G</sub> è l'applicazione web scelta per la gestione dei *task<sub>G</sub>*; permette anche di gestire un calendario dove inserire note o fissare appuntamenti e/o traguardi importanti.

**4.1.4.2 Astah** Astah<sub>G</sub> è l'applicativo scelto per la creazione di grafici UML<sub>G</sub>. La versione adottata è quella *Professional*, resa disponibile gratuitamente per un utilizzo da parte di studenti.

#### 4.1.4.3 Gantt

**4.1.4.4 Telegram** Si utilizza Telegram<sub>G</sub> per una comunicazione informale all'interno del gruppo. Inoltre Telegram fornisce il vantaggio di essere un'applicazione multi-piattaforma<sub>G</sub> disponibile nelle seguenti versioni: *desktop*, *web* e *mobile*.

**4.1.4.5 Microsoft Office Power Point** Power Point<sub>G</sub> è il software utilizzato per creare presentazioni.

### 4.2 Gestione delle infrastrutture

#### 4.2.0.6 Attività

**4.2.0.6.1 Gestione del repository** Il gruppo ha deciso di utilizzare un repository<sub>G</sub> utile a svolgere funzioni diverse, ma necessarie, allo sviluppo del sistema finale. Una volta iscritto, ciascun membro ha la possibilità di creare il suo *branch<sub>G</sub>* personale contenente una copia dei file originali del *branch master* in modo da poter lavorare su delle copie in locale.

**4.2.0.6.2 Gestione del messaggio di commit** Per mantenere l'ambiente di lavoro il meno ambiguo possibile, è stato deciso di adottare un formato standard per andare a scrivere il messaggio della *commit<sub>G</sub>*.

#### 4.2.0.7 Procedure

**4.2.0.7.1 Installazione di Git** La procedura di installazione varia a seconda del sistema operativo utilizzato. Per i sistemi Linux<sub>G</sub> occorre rispettare la seguente procedura:

- Aprire il terminale;
- Immettere il comando *sudo apt-get update*;
- Immettere il comando *apt-get install git*;

Per sistemi OS X<sub>G</sub>:

- Recarsi nella sezione dedicata ai *download* <https://git-scm.com/download/mac>;
- Scaricare il *file* in formato DMG<sub>G</sub>;
- Aprire il file appena scaricato;
- Lanciare l'installazione cliccando su *git.pkg*.

Infine per i sistemi Windows<sub>G</sub>, è necessario fare quanto segue:

- Accedere al sito ufficiale <https://git-for-windows.github.io>;
- Scaricare l'eseguibile;
- Lanciare l'eseguibile;
- Seguire la procedura riportata dalla finestra di dialogo.



#### 4.2.0.7.2 Creazione di una cartella locale di repository Seguire la seguente procedura:

- Creare una nuova cartella;
- Aprire il terminale;
- Collocarsi all'interno della cartella appena creata;
- Eseguire il comando `git init`.
- Immettere il comando `git clone <indirizzo>`, sostituendo <indirizzo> con l'URL del progetto su GitHub<sub>G</sub>.

#### 4.2.0.7.3 Creazione del branch personale Per creare il *branch<sub>G</sub>* personale occorre seguire i seguenti passi:

- Muoversi nella cartella di *repository<sub>G</sub>*.
- Accedere alla cartella di progetto;
- Eseguire il comando `git branch <nome>`, sostituendo <nome> con il nome del nuovo *branch* da creare.

### 4.2.1 Norme

#### 4.2.1.1 Repository

**4.2.1.1.1 Nomi dei file in SiVoDim** I file e le cartelle presenti nel *repository<sub>G</sub>* devono essere conformi al seguente formalismo tratto dallo Standard ISO<sub>G</sub> 9660:1999 (Level 2):

- I caratteri usati sono solo quelli minuscoli a-z, 0-9, l'underscore e il punto, ad esempio: `nome_del_documento.tex`;
- Non sono ammessi caratteri accentati;
- I nomi non possono includere spazi o finire con un punto (.);
- I nomi non devono contenere più di un punto (.) ad eccezione di quelli che fanno riferimento ad una specifica versione (esempio: `studio_di_fattibilita_v1.0.0.pdf`);
- I nomi non devono essere più lunghi di 21 caratteri esclusi i 3 destinati all'estensione.

#### 4.2.1.1.2 Struttura di SiVoDim

**4.2.1.1.3 Messaggio di commit** Il messaggio di `commitG` dovrà essere conforme alla seguente notazione:

Desc:  
Data:  
Note:

dove:

- **Desc:** fornisce una descrizione esaustiva dell'attività svolta;
- **Data:** fornisce la data in cui si è apportata la modifica;
- **Note:** aiuta a specificare lo stato del lavoro, nello specifico si adotteranno le seguenti notazioni:
  - C se il lavoro è stato completato;
  - NC se il lavoro non è stato completato;
  - V se il lavoro necessita di verifica;

#### 4.2.2 Strumenti

**4.2.2.1 Git** Git<sub>G</sub> è il sistema di controllo di versione utilizzato per il *repository<sub>G</sub>* del *team*.

**4.2.2.2 GitHub** GitHub<sub>G</sub> è il servizio *web* di *hosting<sub>G</sub>* adottato per tenere una copia del *repository<sub>G</sub>* del progetto.

**4.2.2.3 Dropbox** Dropbox<sub>G</sub> è lo strumento di *cloud<sub>G</sub>* che si è scelto per gestire file che non necessitano di essere sottoposti a controllo di versione.

**4.2.2.4 Sistema Operativo** I membri del gruppo operano su tre diversi sistemi operativi:

- Ubuntu<sub>G</sub>;
- Windows10<sub>G</sub>;
- OS X<sub>G</sub>.

### 4.3 Formazione dei membri del gruppo

I membri del gruppo che non hanno conoscenze sufficienti per far fronte ai problemi assegnati dal *Responsabile di Progetto*, dovranno documentarsi e colmare eventuali lacune durante ore esterne a quelle di lavoro, non imputabili perciò ai costi del Proponente<sub>G</sub>.