NORME DI PROGETTO

SWEg Group

1 Registro Modifiche

Modifica	**Nome**	**Data**	**Ver.**
Creazione Documento	Piergiorgio Danieli	12/12/2016	0.0.1
Aggiunti Ruoli	Sebastiano Marchesini	12/12/2016	0.0.2
Aggiunte Procedure di Sviluppo	Piergiorgio Danieli	13/12/2016	0.0.3
Aggiunti Analisi dei Requisiti	Sebastiano Marchesini	14/12/2016	0.0.4
Aggiunto Ambiente di Lavoro	Piergiorgio Danieli	15/12/2016	0.0.5
Aggiunto Versionamento	Sebastiano Marchesini	15/12/2016	0.0.6
Stesura Iniziale	Sebastiano Marchesini	30/12/2016	0.1.0

Indice

2 Introduzione

2.1 Scopo del Documento

Questo documento ha l'obiettivo di definire le regole che i membri del gruppo *SWEG* seguiranno nello sviluppo del progetto.
Ogni componente del team e' tenuto a leggere e seguire le norme qui contenute per ottimizzare il lavoro, uniformare tutti i documenti e minimizzare gli errori.
In particolare verranno specificate le norme per:

- interazioni tra i membri del gruppo e componenti esterni
- stesura dei documenti e relative convenzioni
- ambiente di lavoro
- stesura del codice.

2.2 Glossario

Al fine di evitare ambiguità e ottimizzare la comprensione dei documenti, viene incluso un Glossario, nel quale saranno inseriti i termini tecnici, acronimi e parole che necessitano di essere chiarite.
Un glossario è una raccolta di termini di un ambito specifico e circoscritto. In questo caso per raccogliere termini desueti e specialistici inerenti al progetto.

2.3 Riferimenti

3 Comunicazioni e Riunioni

3.1 Comunicazioni Interne ed Esterne

3.1.1 Interne

Per le comunicazioni interne è stato creato un gruppo su *Telegram*, un'app di messaggistica istantanea.

3.1.2 Esterne

Per le comunicazioni esterne è stata creata la casella di posta *sweg.group@gmail.com*. Questo può essere l'unico canale utilizzabile. Solo il *Project Manager* può avere delle interazioni con l'esterno, sarà poi compito suo riferire il contenuto delle discussioni agli altri componenti del gruppo.

3.2 Riunioni

3.2.1 Interne

Le convocazioni alle riunioni sono notificate dal *Project Manager* e confermate informalmente dai componenti del gruppo. Ogni membro è tenuto a partecipare a meno di una giustificazione valida. Una riunione e' fissata per discutere di argomenti di interesse generale di tutti i componenti. E' pero' possibile che ci sia la necessità di effettuare delle riunioni alle quali non e' richiesta la presenza di tutti i membri ma solo di alcuni di essi; in questo caso il *Project Manager* autorizza l'incontro e poi vi prenderanno parte solo le persone interessate, le quali dovranno poi informare gli altri componenti del gruppo delle decisioni prese.

3.2.2 Esterne

E' compito del *Project Manager* fissare degli incontri con il proponete ed il committente.
Tutti i membri del gruppo ne devono essere informati. Ad ogni incontro verrà scelto un segretario il quale avrà il compito di redigere un verbale che verrà poi inviato a tutti i componenti.
Ogni verbale redatto dovrà avere la seguente composizione:

1. Data e ora
2. Luogo
3. Partecipanti interni
4. Partecipanti esterni
5. Argomenti trattati
6. Domande e risposte

4 Documenti

4.1 Template

Per facilitare ed unificare la redazione della documentazione e' stato creato un template apposito in LaTeX.

4.1.1 Stile del testo

- **Grassetto**: Deve essere applicato alle parole significative che devono essere messe in risalto;
- **Corsivo**: va utilizzato per indicare termini in lingua inglese e nomi dei file;
- **Maiuscolo**: va usato per scrivere gli acronimi;
- **Latex**: ogni riferimento a LaTeXva scritto utilizzando il comando (slash)LaTeX .

4.2 Convenzioni tipografiche

I principali comandi utilizzati in LaTeXsono indicati sotto per argomento.
Per le liste:

- `\begin{itemize} \end{itemize}` dove tra i due comandi posso mettere i vari punti di una lista;
- gli oggetti della lista si indicano col comando `\item`;
- vi possono essere vari tipi di liste : `itemize` puntata, `enumerate` numerata e `trivlist` vuota.

Per le tabelle:

- `\renewcommand\arraystretch{1.2}` allarga ogni riga dello spazio indicato. La tabella va racchiusa tra parentesi graffe;
- `\begin{tabular}{|c|c|c|} \end{tabular}` indica una tabella con 3 colonne e pos. testo centrale. La barra verticale (|) indica che vi e' una linea divisoria verticale tra le celle;
- `\hline` indica una linea separatrice orizzontale;
- `&` è il simbolo separatore tra le celle orizzontalmente;
- `\\` passa alla riga successiva della tabella.

Per le immagini:

- `\begin{figure} \end{figure}` inizio e fine di una sezione con figure;
- `\centering` imposta centrale la tabella;

4.3 Formati

- **Date**: saranno espresse nel formato italiano gg/mm/aaaa dove:
  - gg : indica il giorno, va scritto sempre con 2 cifre;
  - mm : rappresenta il mese, va scritto sempre con 2 cifre;
  - aaaa : rappresenta l'anno, va scritto sempre con 4 cifre.
- **Anni accademici**: si userà il formato aaaa1-aaaa2 dove:
  - aaaa1: indica l'anno solare di inizio, va scritto sempre con 4 cifre;
  - aaaa2: indica l'anno solare di fine, va scritto sempre con 4 cifre.
- **Orari**: verranno espressi secondo lo standard *ISO 8601* HH:MM dove:
  - HH: indica le ore, va scritto sempre con 2 cifre;
  - MM: indica i minuti, va scritto sempre con 2 cifre.
- **URL**: collegamento ad un indirizzo web deve essere scritto con un font di colore blu;
- **Sigle**: i nomi dei documenti potranno essere sostituiti dalle rispettive sigle:
  - AdR ad indicare il documento ?Analisi dei Requisiti?;
  - PdP ad indicare il documento ?Piano di Progetto?;
  - PdQ ad indicare il documento ?Piano di Qualifica?;
  - NdP ad indicare il documento ?Norme di Progetto?;
  - Gl ad indicare il documento ?Glossario?;
  - ST ad indicare il documento ?Specifica Tecnica?;
  - SdF ad indicare il documento ?Studio di Fattibilità?.

4.4 Struttura del documento

4.4.1 Frontespizio

Il frontespizio di ogni documento dovrà essere così strutturato:

- Nome e logo del gruppo;
- Nome del progetto;
- Titolo del documento;
- Versione del documento;
- Nome e cognome dei redattori del documento;
- Nome e cognome dei revisori del documento;
- Uso del documento;
- Destinatari del documento.

4.4.2 Diario delle modifiche

La seconda pagina deve essere una tabella contenente i cambiamenti che sono stati effettuati nel documento. Deve essere così strutturata:

- Modifica: descrizione delle modifiche effettuate;
- Nome: nome e cognome dell'autore delle modifiche;
- Data: il giorno nel quale sono state apportate le modifiche;
- Versione: la versione del documento dopo la modifica;

La tabella è ordinata per data in ordine decrescente, in modo che la prima riga sia l'ultima modifica eseguita, e quindi corrisponda alla versione attuale del documento.

4.4.3 Indice

La pagine successiva al diario delle modifiche deve essere l'indice del documento. Ogni documento tranne i verbali deve contenere l'indice che serve a dare una visione globale degli argomenti trattati.
L'indice è autogenerato con un comando di LaTeX.

4.4.4 Formattazione generale

Tutte le altre pagine del documento dovranno rispettare la struttura di base che segue:

- Logo del gruppo: sarà posizionato sempre in alto a sinistra;
- Sezione corrente: nell'intestazione deve comparire il numero ed il nome della sezione in cui ci si trova. Questa informazione sarà in alto a destra;
- Numero di pagina: compare a piè di pagina, a destra.

Ogni pagina deve rispettare i seguenti margini:

- Superiore: 2cm;
- Inferiore: 2cm;
- Destro: 1cm;
- Sinistro: 1cm;

4.5 Classificazione documenti

4.5.1 Documenti formali

Tutti i documenti che sono stati approvati dal *Project Manager* sono da ritenersi formali.
Se un documento formale viene modificato, deve essere successivamente approvato nuovamente dal *Project Manager*.

4.5.2 Documenti informali

Sono quei documenti che non sono stati approvati del Project Manager, poiché non lo necessitano e quindi sono ad esclusivo uso interno, o perché sono ancora in fase di sviluppo e verranno approvati successivamente.

4.6 Componenti grafiche

4.6.1 Tabelle

Tutte le tabelle presenti nella documentazione necessitano di una didascalia che le descrive brevemente e che porta un numero progressivo in modo da identificarla univocamente.

4.6.2 Immagini

Le immagini come le tabelle dovranno essere accompagnate da una didascalia ed un numero progressivo che le identifica.

4.7 Intestazione file di documentazione

Ogni file di documentazione dovrà iniziare con la seguente intestazione:

- %FILE: nome del file;
- %PERCORSO: /PercorsoDelFile/NomeDelDocumento/;
- %DATA CREAZIONE: gg/mm/aa;
- %AUTORE: SWEg;
- %EMAIL: sweg.group@gmail.com;

5 Ruoli

5.1 Ruoli all'interno del progetto

Ad ogni componente è assegnato un ruolo all'interno del progetto che varia a seconda di regole che sono riportate in seguito. Importante è la nessun conflitto di interesse infatti non è possibile validare il processo progettato o realizzato. Può però capitare che si svolgano più compiti nello stesso periodo causa carenza di personale. Importante è la flessibilità all'interno del team per diminuire i tempi di scambio mansione.
Ogni membro a qualunque ruolo ha il diritto di rafforzare la sua funzione con corsi e il dovere di specializzarsi con ricerca ed esperienze personali.

5.1.1 Project Manager (PM)

Titolare di responsabilità decisionale. Rappresenta il progetto di fronte al fornitore e al committente. Tale figura è garante unico dell'avvio, pianificazione, svolgimento, controllo e chiusura di un progetto. In particolare :

- Pianifica riunioni e compiti;
- Gestisce i ruoli;
- Ruolo consultivo e preventivo;
- Capacità tecniche e conosce bene gli strumenti utili;
- Controllo, coordinamento con relazioni esterne;
- Approvazione documenti.

5.1.2 Amministratore (AM))

Responsabilità di cura dei processi. In dettaglio :

- Gestione delle risorse e delle infrastrutture
- Ispettore di versioni e configurazioni
- Risolutore di difficoltà di processi

5.1.3 Analista (AN)

Il ruolo principale dell'analista è l'attuazione dei requisiti. Quindi comprendere il problema in ogni sfaccettatura. Senza l'obbligo di trovare una soluzione. Tra le sue competenze :

- Cogliere il problema
- Conoscere il dominio del quesito
- Vasta esperienza professionale

5.1.4 Progettista (PL)

Il compito del progettista è quello di realizzare e seguire la manutenzione del prodotto. E' di fatto un attuatore :

- Risolve il problema
- Definisce cosa viene costruito e come
- Ampia conoscenza nei linguaggi e delle risorse utilizzate
- Esegue una soluzione attuabile e comprensibile

5.1.5 Programmatore (PR)

Il programmatore è la mano che muove crea il progetto. Il suo compito principale è infatti la realizzazione e la manutenzione del prodotto. Come il progettista anche nel suo ambito è un attuatole :

- Segue i piani progettati e attua il disegno previsto
- Implementa test di verifica automatica per il suo codice
- Il codice che scrivere deve essere versionato e documentato comprensibile

5.1.6 Verificatore (VR)

Al termine del lavoro eseguito dagli altri membri del team il verificatore esegue un controllo tramite analisi statica del ticket . Questa può essere attuata tramite *Walkthrough* o *Ispection*. Nel secondo caso la check list è redata concordata insieme al *Project Manager*.

- Controlla siano rispettati gli standard aderiti nel progetto
- Assicura la conformità del prodotto in ogni fase del ciclo di vita

5.1.7 Responsabile Qualità

Non è richiesto tale ruolo all'interno del team e non sarà interpretato nella rotazione. Molto spesso è incaricato un ente fuori dall'azienda. In questo caso è compito del committente.

5.2 Sistema rotativo dei ruoli

Il progetto a scopo didattico ha l'obbligo di far interpretare ogni ruolo a tutti i partecipanti del team. Per il momento assumiamo un modello agile di tipo *scrum* per eseguire i compiti (o *ticket*). Quindi dopo aver costruito una prima check list in gruppo ogni obbiettivo viene risolto da un sottogruppo in cui il primo che preferisce eseguirlo assume il ruolo di *Project Manager* e vaglia oggettivamente ogni sfaccettatura per risolvere il *backlog* richiedendo le risorse di cui ha bisogno.

5.3 Costi Ruoli

Ciascun ruolo e mansione ha un peso diverso nel progetto e quindi un costo. Che deve rientrare nel *budget* pattuito dal committente $9300,00$[1] . Non specificato nel

[1] Il calcolo del budjet per noi gruppo composto da 5 persone è stato calcolato proporzionalmente al costo di 7 persone arrotondato per eccesso.

nostro caso.

5.3.1 Tabella Costi

Responsabile/Project Manager	€30,00
Amministratore	€20,00
Analista	€25,00
Progettista	€22,00
Programmatore	€15,00
Verificatore	€15,00

6 Analisi dei Requisiti

6.1 Studio di Fattibilità e Analisi dei Rischi

Dopo aver indetto le riunioni e stabilito i vari ruoli gli *Analisti* valuteranno capacità e preferenze del progetto che gioveranno alla completa realizzazione degli obbiettivi. E? quindi compito di questi ultimi, sulla base di quanto deciso, redigere uno *Studio di Fattibilità* con:

- Contributi agli obbiettivi dell'organizzazione
- Ingegnerizzare con la tecnologia corrente entro il *budget*
- Integrazione con altri sistemi usati

6.2 Documento Analisi dei Requisiti

Sempre gli *Analisti* hanno compito di redigere l' *Analisi dei Requisiti*. Comunicheranno poi al *Project Manager* il bisogno di negoziare e chiarire con il committente punti meno chiari del capitolato.

6.2.1 Classificazione dei requisiti

Ogni requisito può essere provvisto di più sotto-requisiti. I requisiti vengono organizzati in forma gerarchica.
Al compimento di tutti i moduli che compongono il requisito padre questo viene soddisfatto.
La notazione da utilizzare e? quella che segue:

<0|1|2><F|P|Q|V>-X<.Y<.Z»

Dove le seguenti sigle indicano:

0: obbligatorio

1: desiderabile

2: opzionale

Hanno diverso tipo:

F: requisito funzionale

P: requisito prestazionale

Q: requisito di qualità

V: requisito dichiarativi(vincoli)

Tale parte indica il codice identificativo di ogni requisito.
E' espresso in modo gerarchico ed univoco.

X: requisito di primo livello

Y: sotto-requisito

Z: sotto-requisito di un sotto-requisito

I campi Y e Z possono essere assenti.

Si ricorda che per ogni requisito c'è bisogno di una descrizione e riportare le dipendenze che ha verso altri. I requisiti non devono comunque essere in conflitto tra loro.

6.2.2 Allocazione e modellazione concettuale del sistema

Il team *Analisti*, dopo aver preso visione nello specifico del capitolato e aver fatto emergere i requisiti , procede con la costruzione dei diagrammi dei casi d'uso (*UC*). Nello specifico é richiesto:

- **Titolo** del caso d'uso sintetico
- **Attori Principali**
- **Attori Secondari**, nel caso vi fossero.

Descrizione del caso con:

- **Scenario principale**
- **Scenari/o alternativo**, se presenti
- **Precondizione**
- **Postcondizione**
- **Requisiti del caso d'uso ricavati**

Il caso d'uso sarà sempre associato da un grafico specializzato. Si è scelto uno standard ibrido tra *UML 2.x e 1.x* con le funzioni principali che racchiudono le due versioni di linguaggio.

6.3 Tracciamento

Il tracciamento dimostra completezza ed economicità della soluzione. Se tracciati tutti i requisiti conseguentemente:

- Possiamo soddisfarli
- Nessuna funzionalità sarà superflua
- Nessun comportamento è ingiustificato

E' nostro compito automatizzare il processo con software adeguato.

7 Procedure di sviluppo

In questa sezione verranno descritte le procedure che tutti i componenti del gruppo sono tenuti a seguire per ottenere un risultato soddisfacente degli obiettivi stabiliti dal Piano di qualifica v1.0.0 .

7.1 Gestione di progetto

Il *Project Manager* ha la responsabilità di gestire l'intero progetto. Dovrà utilizzare gli strumenti in suo possesso per:

- Coordinare e pianificare le attività;
- Coordinare e gestire le risorse umane;
- Analizzare i rischi;
- Elaborare e monitorare i dati d'avanzamento;

7.1.1 Pianificazione delle attività

Per ogni periodo di lavoro il *Project Manager* deve stabilire le seguenti attività:

- Creare un calendario lavorativo;
- Inserire le attività da svolgere;
- Inserire eventuali dipendenze delle attività;
- Inserire le *milestone* indicante il termine previsto delle attività;
- Assegnare ad ogni attività le risorse necessarie;

7.1.2 Coordinamento delle attività

Una volta pianificate le attività, il *Project Manager* le assegna ad ognuno dei membri attraverso l'ausilio offerto da *Asana*. Ogni componente può quindi verificare quali attività gli sono state assegnate e modificarne lo stato, facendo in modo che il *Project Manager* possa seguire con facilità i progressi effettuati.

7.1.3 Analisi dei rischi

Durante tutto l'avanzamento del progetto il *Project Manager* ha il compito di valutare eventuali rischi che possono venirsi a creare e che sono descritti nel *Piano di Progetto* v1.0.0 attuando le adeguate contromisure descritte.

7.1.4 Elaborazione dei dati

Il *Project Manager* dovrà creare eventuali grafici esplicativi dell'andamento del progetto, e riportarli nel *Piano di Progetto* v1.0.0.

7.2 Verifica

In questa sezione verranno descritte le tecniche e le strategie che dovranno utilizzare i *Verificatori* durante il loro lavoro, in modo da uniformare le attività ed ottenre il miglior risultato possibile.

7.2.1 Tecniche di analisi

I Verificatori dovranno utilizzare le seguenti tecniche per svolgere al meglio il proprio compito:

- **Walkthrough**: consiste in una lettura del documento/codice cercando errori ed anomalie a largo spettro senza un'idea precisa del tipo di errori che si potranno trovare.
  Ogni errore verrà discusso con gli autori per evitare incomprensioni e per valutare insieme le modifiche da apportare. Il *walkthrough* è necessario nelle prime fasi dello sviluppo, quando ancora non è del tutto chiaro quali siano gli errori possibili. Utilizzando questa tecnica più volte è possibile scrivere una lista degli errori più comuni.

- **Inspection**: l'*inspection* si basa sulla lettura mirata del documento/codice. Questa fase è facilitata dall'utilizzo della lista stilata durante la fase di *walkthrough*. In questo modo si cercano gli errori maggiormente fatti. Con l'acquisizione di esperienza la lista di controllo verrà estesa, rendendo questa fase sempre più efficace.

- **Analisi dinamica**: questa analisi si applica solo al prodotto software e viene svolta durante l'esecuzione del codice. Per questo tipo di analisi vengono utilizzati degli appositi test per verificare il funzionamento e rilevare possibili errori d'implementazione.
  Questi *test* devono essere utili e ripetibili. Sono definiti ripetibili quando uno stesso test, con gli stessi dati in input, sullo stesso hardware e con lo stesso software, produce lo stesso risultato.

  Test di unità: questo *test* verifica che ogni singola unità di software funzioni correttamente. Con questo test si verifica la correttezza di tutti i moduli base che compongono il software, limitando così gli errori d'implementazione.

  Test di integrazione: questo test verifica che due moduli testati durante il test precedente, funzioni anche quando vengono assemblati insieme. E' possibile in questo modo scovare degli errori che precedentemente erano sfuggiti.

  Test di sistema: consiste nella validazione dei prodotti software. Viene eseguito soltanto quando si raggiunge una versione che può essere definitiva. Viene quindi verificata la copertura completa dei requisiti da parte del prodotto.

  Test di regressione: viene eseguito immediatamente dopo che una componente è stata modificata. Di fatto consiste nel eseguire nuovamente tutti i test precedenti per verificare che le modifiche permettano ancora ai moduli di funzionare nel modo previsto.

  Test di accettazione: coincide con il collaudo del software in presenza del proponente. Se questo test ha esito positivo, il prodotto sarà considerato abbastanza maturo da permettere il rilascio.

7.2.2 Lista di controllo

Durante la fase di *walkthrough* sono stati rilevati questi come errori più frequenti:

- Norme Stilistiche:

  - Un elemento dell'elenco inizia con lettera minuscola;
  -
  -

- Italiano:
  - Scorretto utilizzo di accenti acuti e gravi;
  - Punteggiatura nelle liste;
  -

- LaTeX:
  -

- UML:
  - Direzione delle frecce del diagramma errate;
  -

7.2.3 Verifica documenti

Il *Verificatore* nei documenti dovrà verificare:

- Controllare nel diario delle modifiche i cambiamenti avvenuti dall'ultima volta;
- Cercare eventuali errori nelle parti che sono state modificate del documento;
- Fare un resoconto degli errori riscontarti, in modo che il *Project Manager* possa creare un *ticket* per la correzione;
- Aggiornare il registro delle modifiche e la versione del documento.

7.2.4 Verifica requisiti

Il verificatore, nel controllare i requisiti dovrà porre la propria attenzione su:

- Correttezza ortografica e lessicale del testo;
- Atomicità dei requisiti foglia.

7.2.5 Verifica diagrammi UML

Nel controllare i diagrammi *UML*, il *Verificatore* dovrà prestare attenzione a questi elementi:

- Correttezza ortografica e lessicale del testo;
- Correttezza del formalismo grafico utilizzato;
- Correttezza logica del programma.

8 Ambiente di lavoro

8.1 Sistema operativo

Il progetto e l?insieme delle applicazioni di supporto non sono progettate su un particolare sistema operativo. I componenti del team avranno quindi la possibilità di installare i propri *client* dei mezzi in qualunque macchina: *Windows* , *Mac OS X*, *Linux*.

8.2 Applicativo Guida

Si è stato scelto per il coordinamento :

- Applicativo web Asana;
- Repository Git;
- Comunicazione tramite Telegram.

8.3 Ticket-ing

Il web applicativo Asana è stato scelto per la comunicazione e il ticketing.
E' prediletto questa piattaforma per la sua versione sia web sia mobile.
Collegato tramite mail così da avere una notifica sui compiti seguiti.
Oltre ovviamente a essere gratuito Asana offre la possibilità di collegare direttamente file e link su ogni singolo ticket.
Cosa infine forse più importante è l'associazione al calendario così da avere sotto controllo migliore gli eventi datati.

8.4 Documentazione

Per la documentazione inizialmente si redige una prima stesura su un documento semplice in qualsiasi text editor per poi elaborarlo in linguaggio LaTeX.

8.4.1 Latex

E' stato scelto di scrivere i documenti in linguaggio LaTeXperché:

- nerente al nostro percorso di studi essendo un linguaggio compilabile. Si usa di più la mente logica che serve a costruire un programma informatico;
- I documenti scritti con il mark-up di Latex posso venire composti direttamente in pdf, oltre ad altri formati;
- Sono più precisi gli automatismi di creazione di indice e l'impaginazione;
- E' distribuito con una licenza di software libero;
- Permette di definire template;
- Consente di suddividere il documento in file separati.

8.4.2 Edior

L'editor di testo utilizzato multipiattaforma è *TexStudio*. Programma efficiente con compilazione rapida. Offre una insieme di mezzi per velocizzare e facilitare la creazione della documentazione.

8.4.3 Diagrammi UML

Per la creazione dei diagrammi *UML* si è preso riferimento l'applicazione usata dal professore nelle ore di lezione. Inserendo la mail concessaci dell'università è possibile avere una versione gratuita del software *Astah*.
Progettato per il design dei principali grafici a uso informatico.

8.4.4 Verifica

Tutti i documenti creati con LaTeXsono automaticamente controllati con uno script per il controllo ortografico.

8.5 Pianificazione

8.5.1 Diagrammi

Per la pianificazione delle attività è stato scelto il software

8.5.2 Attività

Asana è un'applicazione web e mobile che permette la suddivisione delle attività in modo strutturato. Offre un'organizzazione ad albero suddivisa in questo modo:

- Creare più **Workspace**;
- Creare *Project* all'interno di *Workspace*;
- Creare *Task* all'interno di un *Project*;
- Creare *Sub-Task* all'interno di un *Task*.

Alcune delle funzionalità offerte sono:

- Assegnare un *Task* ad uno o più individui;
- Assegnare una data di scadenza;
- Vedere le attività in scadenza sul calendario;
- Seguire lo stato di avanzamento dei *Task*;
- Avere un riepilogo dei *Task*;
- Riceve e-mail automatiche con l'aggiornameto dei Task che si seguono.

I nome delle etichette relativi ai Task dovranno seguire la seguente forma [elemento di interesse][messaggio]:

- **Elemento di interesse**: si riferisce al documento o all'ambito al quale appartiene il Task. Nel caso un cui si tratti di un documento, dovranno essere messe le iniziali del suo nome.
  Ad esempio NdP, per indicare *Norme di Progetto*. Nel secondo caso si userà il nome completo dell'ambito che si sta trattando. In entrambi i casi viene utilizzata la lettera maiuscola.

- **Messaggio**: breve messaggio quanto più esaustivo possibile per indicare la problematica da risolvere o il compito da eseguire.

8.6 Versionamento

Il software di versionamento scelto è *Git*. Le motivazioni a base della scelta è perché con *Git*:

- Strutturato come un File System;
- Salva uno snapshot del progetto a ogni commit;
- Veloce e Distribuito avendo in locale la copia della repository per ogni utente;
- Sfrutta facilmente le operazioni di merge e branch;
- Oltre al fatto che è stato già usato da alcuni membri del team.

All'interno della *repository* vi sono due macro sezioni . La sezione documentazione e la sezione progetto.

8.7 Repository

8.7.1 Repository documentazione

La *repository* della documentazione è composta dalle varie cartelle :

- AnalisiDeiRequisiti
- Glossario
- NormeDiProgetto
- PianoDiProgetto
- PianoDiQualifica
- StudioDiFattibilità
- Verbali

I file all'interno possono essere di qualsiasi genere. Anche appunti. Nella capitolo della documentazione è stato trattata il modo in cui crearli ed elaborarli.

8.7.2 Repository progetto