1 Configurazione

1.1 Controllo di versione

Il controllo di versione di documenti e file sorgente viene fatto utilizzando il software Git. Nonostante siano presenti varie alternative, come Mercurial, Subversion e Bazaar, è stato scelto Git in quanto soddisfa appieno le necessita di versionamento dei file per questo progetto e, inoltre, permette di versionare i file localmente, senza bisogno di una connessione internet attiva. Come servizio di hosting per la repository è stato scelto Github. Entrambe le scelte sono state fatte anche perché più membri del gruppo avevano già avuto la possibilità di lavorare con tali servizi.
Ogni qualvolta viene eseguita una modifica sostanziale ad un documento o ad un file sorgente deve essere assegnato un nuovo numero di versione a questo, per tener traccia delle modifiche fatte.
Per quanto riguarda le norme di versionamento dei file inerenti alla documentazione si rimanda alla sezione Versionamento dei documenti del presente documento.

1.2 Richieste di modifica

Ogni componente del gruppo può avanzare una richiesta di modifica al *Responsabile di progetto* , qualora lo ritenga necessario. Il *Responsabile di progetto* ha il compito di analizzare tale richiesta e decidere se approvarla o meno. In caso affermativo, deve assegnare il compito di realizzare tale modifica ad un membro del gruppo, in base al ruolo da lui ricoperto in quel momento. Una volta effettuata la modifica questa deve essere sottoposta a verifica e dev'essere fatta mantenendo traccia dello stato precedente. In ogni caso, sia di accettazione della richiesta di modifica, sia in caso di rifiuto, il *Responsabile di progetto* è tenuto a motivare la scelta.

1.2.1 Procedura di richiesta di modifica

Il membro del gruppo che vuole effettuare una modifica deve presentare una richiesta formale al *Responsabile di progetto* . La richiesta deve avere i seguenti campi:

1. Autore: contiene nome, cognome e ruolo del richiedente della modifica;
2. Documento: contiene il nome del documento di cui si ritiene in cui andrebbe fatta tale modifica;

3. Urgenza: indica una misura di quanto il richiedente ritiene necessario fare una modifica, può assumere solamente tali valori:

   (a) "Alta": in questo caso si ritiene che la modifica da fare sia importante, con pesanti conseguenza nell'organizzazione dello svolgimento delle attività future o che possa in qualche modo compromettere l'organizzazione delle attività future;

   (b) "Media": in questo caso il richiedente considera la modifica importante, ma non comporta forti conseguenza nell'organizzazione generale delle attività, ma può influenzare lo svolgimento di alcune di queste;

   (c) "Bassa": in questo caso la modifica proposta di modifica è di importanza secondaria, un suggerimento nell'effettuare alcune attività oppure che non influisce, o il suo peso è poco rilevante, nello svolgimento delle attività.

4. Decisione del *Responsabile di progetto* . Questo campo va aggiunto successivamente alla decisione del *Responsabile di progetto* . Questo campo può assumere due valori:

   (a) "Approvata";

   (b) "Respinta".

   Il *Responsabile di progetto* può aggiungere a questo campo la motivazione per cui ha approvato o meno la modifica, in caso lo ritenga necessario.

1.3 Struttura del repository

I file all'interno del repository verranno organizzati secondo questa struttura:

- /Documents
  - NormeDiProgetto;
  - StudioDiFattibilità;
  - AnalisiDeiRequisiti;
  - PianoDiProgetto;
  - PianoDiQualifica;
  - Verbali;
  - Glossario.

- /Source

La struttura di /Source verrà decisa all'inizio della progettazione architetturale.

1.4 Nomi dei file

I nomi dei file interni al repository devono sottostare alle seguenti norme:

- Devono contenere solo lettere, numeri, il carattere "underscore", il segno meno ed il punto;
- Devono avere lunghezza minima di tre caratteri;
- Devono identificare in modo non ambiguo i file;
- Devono riportare le informazioni dal generale al particolare;
- Devono, nel caso contengano date, rispettare il formato YYYYMMDD.

È consigliato invece:

- Utilizzare la notazione CamelCase, invece di caratteri "underscore" e segni meno;
- Utilizzare nomi né troppo lunghi, né troppo corti, indicativamente tra i 10 e i 25 caratteri, estensione compresa;
- Specificare, quando possibile, l'estensione del file.

1.5 Commit

L'esecuzione di ogni comando commit deve sottostare alle seguenti norme:

- Ad ogni commit è necessario specificare un messaggio nel quale si deve dare una descrizione sintetica e più possibile precisa delle modifiche effettuate;
- Le modifiche apportate devono essere complete e testate con successo;
- È vietato l'utilizzo del comando "git add *" prima di un comando commit per evitare di includere nel repository file nascosti, temporanei o non voluti.

1.6 Codifica dei file

I file contenenti codice oppure documentazione dovranno utilizzare la codifica UTF-8 senza BOM.

1.7 Aggiornamento del repository

Per l'aggiornamento del repository è prevista la seguente procedura:

- Dare il comando "git pull". Nel caso in cui si verifichino dei conflitti:
  - Dare il comando "git stash" per accantonare momentaneamente le modifiche apportate;
  - Dare il comando "git pull";
  - Dare il comando "git stach apply" per ripristinare le modifiche.

  In questo modo il repository risulta aggiornato rispetto il repository remoto;
- Dare il comando "git add NomeDelFile", dove al posto di "NomeDelFile" si deve mettere il nome del file, o lista di nomi dei file, sul quale sono state effettuate delle modifiche;
- Dare il comando "git commit -m "Riassunto delle modifiche" ", dove al posto di "Riassunto delle modifiche" deve essere specificato, tra virgolette, il riassunto delle modifiche effettuate;
- Dare il comando "git push".

1.8 Visibilità del repository

Per la condivisione e il versionamento dei configuration item è stato creato un repository privato su GitHub, raggiungibile all'indirizzo https://github.com/mzanella/Leaf. L'accesso è consentito solo ai membri del gruppo.