

# 1 Processi di supporto

## 1.1 Documentazione

In questa sezione sono indicati gli standard riguardanti la struttura e la stesura dei documenti prodotti.

### 1.1.1 Struttura dei documenti

Ogni documento è realizzato a partire da una struttura prestabilita che dovrà essere uguale per tutti i documenti ufficiali ad eccezione dei verbali:

1. frontespizio;
2. informazione sul documento;
3. diario delle modifiche;
4. indice delle sezioni;
5. indice delle tabelle;
6. indice delle figure;
7. introduzione;
8. contenuto.

L'ordine di ognuna delle sezioni è fissato. La numerazione delle prime pagine sarà quella romana, mentre dall'introduzione fino alla fine del documento quella araba.

**1.1.1.1 Frontespizio** Questa sezione deve trovarsi nella prima pagina di ogni documento e contiene:

1. informazioni sul gruppo:
  - a. nome;
  - b. logo;
  - c. email.
2. informazioni sul progetto:
  - a. nome progetto;
  - b. nome azienda proponente.

3. informazioni sul documento:

- a. nome;
- b. versione.

**1.1.1.2 Informazioni sul documento** In questa sezione vengono indicate le principali informazioni riguardanti il documento quali:

- 1. versione;
- 2. data di redazione;
- 3. cognome e nome di coloro che hanno redatto il documento (in ordine alfabetico);
- 4. cognome e nome di coloro che hanno verificato il documento (in ordine alfabetico);
- 5. ambito d'uso del documento (interno oppure esterno);
- 6. cognome e nome di coloro ai quali è destinato il documento (in ordine alfabetico).

**1.1.1.3 Diario delle modifiche** Questa sezione descrive, attraverso l'utilizzo di una tabella, le modifiche che sono state apportate al documento. Ogni riga della tabella corrisponde ad una modifica effettuata al documento. La struttura della riga della tabella è la seguente:

- 1. versione del documento;
- 2. data della modifica;
- 3. cognome e nome dell'autore della modifica;
- 4. ruolo dell'autore della modifica nel momento in cui essa è avvenuta;
- 5. descrizione delle modifiche apportate.

Le righe della tabella sono ordinate a partire dalla data dell'ultima modifica effettuata, in ordine cronologico inverso.

**1.1.1.4 Indice delle sezioni** L'indice delle sezioni contiene l'indice di tutti gli argomenti trattati all'interno del documento. La sua struttura è la seguente:

1. titolo dell'argomento trattato;
2. numero di pagina.

**1.1.1.5 Indice delle tabelle** Questa sezione contiene l'indice delle tabelle. Per ogni tabella deve essere specificato:

1. titolo della tabella;
2. numero di pagina di riferimento.

Nel caso in cui non siano presenti tabelle all'interno del documento, è possibile omettere questa sezione.

**1.1.1.6 Indice delle figure** In questa sezione sono riportate tutte le figure presenti all'interno del documento. Per ogni figura deve essere specificato:

- nome figura;
- pagina di riferimento della figura.

Nel caso in cui non siano presenti figure all'interno del documento, è possibile omettere questa sezione.

**1.1.1.7 Introduzione** Questa sezione deve riportare le seguenti informazioni:

1. scopo del documento;
2. glossario;
3. riferimenti utili:
  - a. riferimenti normativi;
  - b. riferimenti informativi.

**1.1.1.8 Contenuto** Questa sezione contiene il contenuto del documento. Anch'esso deve essere propriamente diviso in sezioni e sottosezioni.

## 1.1.2 Norme tipografiche

### 1.1.2.1 Formattazione generale

**1.1.2.1.1 Testatine** Ogni pagina di un documento, fatta eccezione per il frontespizio, deve contenere la testina. Essa è composta da:

- logo del gruppo, posizionato in alto a sinistra;
- nome del documento, posizionato in alto a destra.

**1.1.2.1.2 Piè pagina** Ogni pagina di un documento, deve contenere il piè pagina. Esso contiene:

- numero della pagina, posizionato al centro.

**1.1.2.1.3 Orfani e vedove** Si considerano vedova, la riga di un paragrafo che inizia alla fine di una pagina, mentre si considera orfana, la riga di un paragrafo che finisce all'inizio di una pagina. I documenti dovranno essere redatti cercando di evitare il più possibile queste due tipologie di righe poiché risultano poco gradevoli.

### 1.1.2.2 Caratteri

#### 1.1.2.2.1 Virgolette

- **Virgolette alte singole ‘ ’** : devono essere utilizzate per racchiudere un singolo carattere;
- **Virgolette alte doppie “ ”** : devono essere utilizzate per racchiudere:
  - nomi di file;
  - comandi;
  - collegamenti a sezioni interne dello stesso documento;
  - parole a cui è stato dato un significato particolare;
  - parole a cui è stato dato un senso diverso da quello originale.
- **Virgolette basse « »** : devono essere utilizzate per racchiudere citazioni.

Non sono ammessi ulteriori casi d'uso per le virgolette.

#### 1.1.2.2.2 Parentesi

- **Tonde:** possono essere utilizzate per descrivere esempi, per fornire dei sinonimi oppure per dare delle precisazioni. Sono le uniche parentesi ammesse all'interno di una frase.
- **Quadre:** possono rappresentare uno standard ISO oppure uno stato relativo ad un ticket<sub>g</sub>.

**1.1.2.2.3 Punteggiatura** La punteggiatura deve essere sempre utilizzata attentamente per cercare di rendere il discorso il più chiaro e coeso possibile. Non sono ammesse spaziature prima dell'utilizzo di un carattere di punteggiatura. L'utilizzo del punto è necessario per indicare la fine di un concetto e poter iniziare un altro.

**1.1.2.2.4 Numeri** I numeri all'interno dei documenti devono essere formattati seguendo lo standard [SI/ISO 31-0]. Esso prevede che la parte frazionaria sia separata da quella decimale utilizzando la virgola. I numeri la cui parte intera supera le tre cifre, devono essere scritti raggruppando in gruppi di tre le cifre di cui è composta la parte intera, partendo dalla cifra meno significativa e separandoli con uno spazio unificatore.

#### 1.1.2.3 Stile del testo

**1.1.2.3.1 Corsivo** Il corsivo va utilizzato per riportare le seguenti informazioni:

- nome di un documento;
- nome di un ruolo;
- percorsi di cartelle.

**1.1.2.3.2 Grassetto** Il grassetto va utilizzato per riportare le seguenti informazioni:

- titoli;
- parole su cui è utile focalizzare l'attenzione del lettore all'interno di un argomento;
- parole chiave all'interno di elenchi.

**1.1.2.3.3 Sottolineato** La sottolineatura è indicata qualora si voglia evidenziare l'importanza di una parola all'interno di una frase.

**1.1.2.3.4 Monospace** Lo stile Monospace<sub>g</sub> va applicato nel caso in cui si vogliano riportare all'interno di un documento comandi oppure parti di codice.

**1.1.2.3.5 Glossario** Questo stile va applicato per tutte le parole che hanno una corrispondenza all'interno del glossario. Ogni parola presente nel glossario deve essere seguita da un pedice contenente il carattere 'g' scritto in corsivo. Non si applica questa regola nei casi in cui la parola compaia all'interno di titoli, percorsi, nomi di cartelle, comandi o parti di codice.

#### **1.1.2.4 Composizione del testo**

**1.1.2.4.1 Elenchi** Le norme che regolano un elenco sono le seguenti:

- la prima parola di un elenco deve essere maiuscola, fatta eccezione nel caso in cui l'elenco inizi con il carattere ':';
- ogni elemento dell'elenco, tranne l'ultimo, deve terminare con il carattere ','. È fatta eccezione nel caso in cui l'elemento sia composto da molte frasi, allora è permesso il '.';
- l'ultimo elemento di un elenco deve sempre terminare con il carattere '.'.

È necessario usare elenchi numerati quando l'ordine degli elementi è rilevante. Per gli elenchi numerati valgono le seguenti regole:

- nel primo livello si usano numeri interni a partire da uno;
- nel secondo livello si usano lettere dell'alfabeto a partire dalla 'a'.

Gli elenchi puntati servono per descrivere elementi di cui non è importante l'ordine espositivo. Essi seguono le seguenti regole:

- nel primo livello bisogna utilizzare cerchi neri pieni;
- nel secondo livello trattini neri.

**1.1.2.4.2 Descrizioni** Nel caso in cui si voglia strutturare la descrizione di qualcosa nella forma di elenco è necessario utilizzare il costrutto `LATEX \begin{description} \item[elemento] \end{description}`. All'interno delle parentesi quadre viene inserito il nome dell'elemento che si vuole descrivere mentre, dopo di esse, la sua descrizione.

**1.1.2.4.3 Note a piè pagina** Le note a piè pagina seguono le seguenti regole:

- la loro numerazione è progressiva all'interno di tutto il documento;
- devono essere scritte una volta sola;
- il primo carattere di ogni nota deve essere maiuscolo. Fanno eccezione i casi in cui la parola sia un acronimo. In questo caso bisogna seguire le regole di formattazione di tale acronimo.

### 1.1.2.5 Formati

**1.1.2.5.1 Date** La formattazione delle date segue lo standard [ISO 8601]. Tale standard prevede che una data sia scritta secondo il seguente formalismo: AAAA-MM-GG. Questa rappresentazione va letta nel seguente modo:

- AAAA: numero a quattro cifre che rappresenta l'anno;
- MM: numero a due cifre che rappresenta il mese;
- GG: numero a due cifre che rappresenta il giorno.

Nei casi in cui risulti possibile esprimere i mesi o giorni omettendo una cifra, è necessario anteporre uno zero davanti a tale cifra. Per velocizzare la scrittura delle date è stato creato il comando `LATEX \frmdate{giorno}{mese}{anno}`.

**1.1.2.5.2 Orari** La formattazione degli orari segue lo standard [ISO 8601]. Tale standard prevede che gli orari siano scritti secondo il seguente formalismo: hh:mm. Questa rappresentazione va letta nel seguente modo:

- hh: numero a due cifre che rappresenta il numero di ore trascorse dalla mezzanotte;
- mm: numero a due cifre che rappresenta i minuti.

Nei casi in cui risulti possibile esprimere le ore o i minuti omettendo una cifra, è necessario anteporre uno zero davanti a tale cifra. Per velocizzare la scrittura degli orari è stato creato il comando `LATEX \frmtime{ora}{minuti}`.

**1.1.2.5.3 URI** La stile utilizzato per rappresentare un URI è il corsivo ed il testo deve essere di colore blu. Per velocizzare la scrittura degli URI è stato creato il comando `LATEX \frmURI{URI}`.

**1.1.2.5.4 Sigle** È possibile fare riferimento a ruoli, documenti e revisioni pianificate utilizzando le seguenti sigle:

- Rp (*Responsabile di progetto*);
- Am (*Amministratore*);
- An (*Analista*);
- Pt (*Progettista*);
- Pm (*Programmatore*);
- Ve (*Verificatore*);
- AR (*Analisi dei requisiti*);
- GL (*Glossario*);
- NP (*Norme di progetto*);
- PP (*Piano di progetto*);
- PQ (*Piano di qualifica*);
- SF (*Studio di fattibilità*);
- ST (*Specifica tecnica*);
- RR (**Revisione dei requisiti**);
- RA (**Revisione di accettazione**);
- RP (**Revisione di progettazione**);
- RQ (**Revisione di qualifica**);

L'utilizzo di tale sigle è permesso solo all'interno di:

- Tabelle;
- Diagrammi (immagini);
- Didascalie di tabelle e immagini;
- Note a piè di pagina.



**1.1.2.5.5 Ruoli di progetto** Quando si fa riferimento ad un ruolo di progetto bisogna adottare lo stile corsivo e la prima lettera deve essere maiuscola. Per velocizzare la scrittura di un ruolo è stato creato il comando `LATEX \frmrole{Ruolo}`.

**1.1.2.5.6 Fasi del progetto** Quando si fa riferimento ad una fase<sub>g</sub> del progetto bisogna adottare lo stile grassetto e la prima lettera deve essere maiuscola. Per velocizzare la scrittura di una fase<sub>g</sub> è stato creato il comando `LATEX \frmphase{Fase}`.

**1.1.2.5.7 Revisioni** Quando si fa riferimento ad una revisione bisogna adottare lo stile grassetto e la prima lettera deve essere maiuscola. Per velocizzare la scrittura di una revisione è stato creato il comando `LATEX \frmrev{Revisione}`.

#### **1.1.2.5.8 Nomi**

- **Nome del gruppo:** quando si fa riferimento al nome del gruppo bisogna adottare lo stile corsivo e la prima lettera deve essere maiuscola. Il nome del gruppo deve essere sempre indicato tramite il comando `LATEX \leaf`, in modo tale da avere sempre la stessa formattazione.
- **Nome del progetto:** il nome del progetto deve essere sempre indicato tramite il comando `LATEX \progetto`, in modo tale da avere sempre la stessa formattazione.
- **Nome proprio:** ogniqualvolta si intende utilizzare un nome proprio si deve scrivere prima il nome e successivamente il cognome. Quando si fa riferimento a disporre i nomi in ordine alfabetico, i nomi devono essere scritti mettendo il cognome davanti al nome e l'ordine è dettato dalle lettere del cognome, a meno di ulteriori specificazioni.
- **Nome di un file:** i nomi dei file vanno formattati utilizzando lo stile Monospace<sub>g</sub> e devono essere racchiusi dalle doppie virgolette alte. Per velocizzare la scrittura del nome di un file è stato creato il comando `LATEX \frmFile{File}`.
- **Nome di un documento:** quando si fa riferimento ad un documento bisogna adottare lo stile corsivo e la prima lettera deve essere maiuscola. I nomi dei documenti devono essere racchiusi dalle doppie virgolette alte. Per velocizzare la scrittura di un documento è stato creato il comando `LATEX \frmdoc{documento}`.

### 1.1.2.6 Componenti grafiche

**1.1.2.6.1 Immagini** L'utilizzo delle immagini all'interno di un documento è regolamentato secondo quanto segue:

- i formati ammessi per le immagini sono il PNG<sub>g</sub> e il PDF<sub>g</sub>;
- devono essere numerate in ordine crescente;
- devono essere seguite da una breve descrizione;
- deve essere presente un riferimento all'immagine all'interno dell'indice immagini.

**1.1.2.6.2 Tabelle** L'utilizzo delle tabelle all'interno di un documento è regolamentato secondo quanto segue:

- devono essere numerate in ordine crescente;
- devono essere seguite da una breve descrizione;
- devono essere presente un riferimento alla tabella all'interno dell'indice delle tabelle;
- si consigliano le linee verticali all'interno delle tabelle solo se queste ne dovessero aumentare la leggibilità.

Al fine di uniformare lo stile delle tabelle in tutti i documenti, è consigliabile utilizzare il template L<sup>A</sup>T<sub>E</sub>X che è stato creato, il quale contiene lo stile predefinito delle tabelle.

### 1.1.3 Tipologie di documenti

**1.1.3.1 Documenti formali** I documenti formali possono essere descritti secondo quanto segue:

- sono documenti approvati dal responsabile di progetto;
- eventuali modifiche ad un documento formale, lo rendono informale;
- sono gli unici documenti che possono essere distribuiti all'esterno del team<sub>g</sub> di progetto;

**1.1.3.2 Documenti informali** I documenti informali possono essere descritti secondo quanto segue:

- sono documenti non ancora approvati dal responsabile di progetto;
- possono essere distribuiti solamente all'interno del team<sub>g</sub> di progetto;
- possono essere sottoposti a revisione.

**1.1.3.3 Glossario** Il glossario nasce dall'esigenza di chiarire il significato ambiguo che possono avere certe parole all'interno di determinati contesti. Al suo interno saranno quindi presenti alcune parole, prese dai documenti, che hanno le seguenti caratteristiche:

- trattano argomenti tecnici;
- trattano argomenti poco conosciuti o che possono scatenare ambiguità;
- rappresentano delle sigle.

Il glossario deve essere strutturato secondo quanto segue:

- i termini devono seguire l'ordine lessicografico;
- ogni termine deve essere seguito da una spiegazione chiara e concisa del significato del termine stesso. Questa spiegazione non deve essere in alcun modo ambigua.

Per evitare confusione, la stesura del glossario deve avvenire in maniera parallela alla stesura dei documenti. Al fine di evitare dimenticanze, è ammesso inserire un termine all'interno del glossario senza inserirne immediatamente la spiegazione. È comunque doveroso completare la spiegazione non appena possibile.

All'interno dei documenti i termini presenti nel glossario devono essere marcati con il pedice <sub>g</sub>(Regole per la formattazione di termini nel Glossario).

**1.1.3.4 Verbali** Lo scopo di un verbale è di riassumere, cercando di essere il più possibile fedeli, ciò che è stato discusso durante un incontro. È previsto che ad ogni riunione tra i membri del gruppo e/o soggetti esterni sia redatto un verbale. Il verbale è soggetto ad un'unica stesura e non può subire modifiche.

**1.1.3.4.1 Verbalì di riunioni interne** Si definisce interna una riunione che coinvolge solamente i membri del gruppo. Il verbale per questo tipo di incontro è da considerarsi di carattere informale.

Il verbale deve essere redatto seguendo la seguente struttura:

1. frontespizio;
2. una sezione “Estremi del Verbale” contenente le seguenti informazioni:
  - a. data;
  - b. luogo, a meno di incontri telematici;
  - c. partecipanti.
3. introduzione contenente le motivazioni per le quali è stato richiesto l’incontro;
4. ordine del giorno;
5. verbale della riunione.

**1.1.3.4.2 Verbalì di riunioni esterne** Si definisce esterna una riunione che avviene tra i membri del team<sub>g</sub> e soggetti esterni. Il verbale redatto per questo tipo di incontro è da considerarsi come parte integrante della documentazione ufficiale e per questo motivo può avere un valore normativo o fornire nuovi requisiti. È previsto che ad ogni incontro venga nominata una persona che si occupi della sua stesura.

Il verbale deve essere redatto seguendo la seguente struttura:

1. frontespizio;
2. indice;
3. una sezione “Estremi del Verbale” contenente le seguenti informazioni:
  - a. data;
  - b. ora;
  - c. luogo, a meno di incontri telematici;
  - d. partecipanti.
4. introduzione contenente le motivazioni per le quali è stato richiesto l’incontro;
5. una sezione dedicata alle domande poste e alle risposte ottenute durante l’incontro.

#### 1.1.4 Procedure

**1.1.4.1 Versionamento dei documenti** Tutti i documenti, ad eccezione dei verbali, sono sottoposti a versionamento. Il versionamento prevede che la versione di un documento venga incrementata ogniqualvolta avvengano delle modifiche all'interno del documento stesso. La sintassi che indica la versione di un documento è la seguente: vX.YY. La sua interpretazione è la seguente:

- 'v' è un carattere che si riferisce alla parola versione;
- X è un numero che indica quante volte è stato formalizzato il documento;
- YY è un numero a due cifre che indica quante modifiche sono state effettuate al documento dalla sua ultima formalizzazione.

#### 1.1.4.2 Avanzamento di un documento


**1.1.4.2.1 Regole di avanzamento di versione** L'avanzamento di versione avviene secondo le seguenti regole:

- X inizia da 0 e viene incrementato di una unità nel momento in cui il *Responsabile di progetto* formalizza il documento;
- YY inizia da 00 e viene incrementato di una unità ad ogni modifica che viene effettuata al documento. Ogni volta che il documento viene formalizzato riparte da 00.

La prima versione di ogni documento è indicata dalla versione v0.01.

**1.1.4.2.2 Formalizzazione di un documento** La formalizzazione di un documento segue la seguente procedura:

1. il documento viene redatto da coloro che sono incaricati della sua stesura ed eventuale correzione di errori;
2. il *Responsabile di progetto* assegna uno o più *Verificatori* al documento i quali dovranno occuparsi di controllare la correttezza del documento stesso;
3. se i *Verificatori* riscontrano anomalie si ritorna al punto 1 altrimenti il documento viene consegnato al *Responsabile di progetto*;
4. il *Responsabile di progetto* decide se approvare il documento, e quindi formalizzarlo, oppure se rifiutarlo, e quindi ritornare al punto 1.



# section

**Figura 1:** Procedura di formalizzazione di un documento

## 1.1.5 Strumenti

**1.1.5.1 Latex** La stesura dei documenti deve essere effettuata utilizzando il linguaggio di markup,  $\text{\LaTeX}$ . È stato scelto questo strumento poiché permette una facile separazione tra formattazione e presentazione. La scelta dell'editor da utilizzare è lasciata libera ai membri del gruppo.

**1.1.5.1.1 Template** Per poter creare omogeneità tra i documenti, è stato creato un template  $\text{\LaTeX}$ , nel quale sono state definite tutte le regole tipografiche da applicare al documento. Questo permette di poter scrivere i documenti senza dover tener conto della loro formattazione.

**1.1.5.1.2 Comandi personalizzati** Sono stati definiti dei comandi  $\text{\LaTeX}_g$  personalizzati al fine di poter rendere più semplice ed immediata l'applicazione delle norme tipografiche. Questi comandi si occupano delle corretta formattazione del testo secondo le norme che sono state definite. La lista dei comandi è presente nella sezione formati.

**1.1.5.1.3 Rilevamento errori ortografici** Al fine di semplificare l'operazione di individuazione e correzione degli errori ortografici, è stato creato uno script bash “OrtographicCheck.sh”.

## 1.2 Configurazione

### 1.2.1 Controllo di versione

Il controllo di versione di documenti e file sorgente viene fatto utilizzando il software  $_g$  Git, e la piattaforma GitHub $_g$ . Ad ogni modifica sostanziale ad un documento o ad un file sorgente impone che a quest'ultimo venga assegnato nuovo numero di versione.

Per quanto riguarda le norme di versionamento dei file inerenti alla documentazione si rimanda alla sezione Versionamento dei documenti del presente documento.

**1.2.1.1 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.

### 1.2.1.2 Repository

**1.2.1.2.1 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 architettuale.

**1.2.1.2.2 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<sub>g</sub> file nascosti, temporanei o non voluti;
- È consigliato specificare il riassunto delle modifiche apportate ad ogni commit tramite il comando `git commit` rispetto al comando `git commit -m`, dove il messaggio deve essere scritto inline tra virgolette.

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

#### 1.2.1.2.4 File

**1.2.1.2.4.1 Nomi dei file** I nomi dei file interni al repository<sub>g</sub> devono sottostare alle seguenti norme:

- Devono contenere solo lettere, numeri, il carattere '`_`', 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 YYYY-MM-DD.

È consigliato invece:

- Utilizzare la notazione CamelCase<sub>g</sub>, invece di caratteri ‘\_’ e ‘-’ qualora il nome di un file fosse composto da più parole;
- 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.


**1.2.1.2.4.2 Codifica dei file** I file contenenti codice oppure documentazione dovranno utilizzare la codifica UTF-8 senza BOM<sub>g</sub>.

### 1.2.1.3 Procedure

**1.2.1.3.1 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 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. Descrizione: una descrizione dettagliata e motivata delle modifiche che si vogliono apportare;
5. 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”.

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



# section

**Figura 2:** Procedura di richiesta modifica di un documento

**1.2.1.3.2 Aggiornamento del repository** Per l'aggiornamento del repository<sub>i</sub> 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 stash apply` per ripristinare le modifiche.

In questo modo il repository<sub>i</sub> risulta aggiornato rispetto il repository<sub>i</sub> 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`, e successivamente specificare sinteticamente il riassunto delle modifiche effettuate;
- Dare il comando `git push`.



**Figura 3:** Procedura di aggiornamento del repository

**1.2.1.4 Strumenti** Lo strumento utilizzato per il versionamento di documenti e file sorgenti è `Gitg`. Nonostante siano presenti varie alternative,

come Mercurial, Subversion e Bazaar, è stato scelto Git<sub>g</sub> in quanto soddisfa appieno le necessità 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 il repository<sub>g</sub> è stato scelto, invece, GitHub<sub>g</sub>. Entrambe le scelte sono state fatte anche perché più membri del gruppo avevano già avuto la possibilità di lavorare con tali servizi.

## 1.3 Verifica

### 1.3.1 Documenti

Il *Responsabile di progetto* ha il compito di dare inizio alla fase<sub>g</sub> di verifica, assegnando i compiti ai *Verificatori*. Quest'ultimi, devono effettuare un'accurata verifica delle seguenti regole:

- Deve essere utilizzata una sintassi corretta e il più possibile semplice;
- Devono essere utilizzati periodi brevi;
- La struttura del documento deve essere semplice e intuitiva;
- Devono essere rispettate le norme tipografiche;
- Devono essere rispettate le regole riguardanti il glossario.

Durante l'intera attività di verifica i *Verificatori* dovranno utilizzare lo strumento del diario delle modifiche in modo tale da concentrare la loro attenzione sulle modifiche effettuate ad un documento e tenere traccia degli errori più comuni commessi nel redigere i documenti.

**1.3.1.1 Sintassi** I *Verificatori* hanno il compito di identificare errori sintattici nei documenti. Ciò può essere fatto con l'ausilio di strumenti automatici, ma ogni documento deve essere sempre sottoposto a walkthrough in modo tale da individuare errori che non sono stati in grado di evidenziare. Inoltre hanno il compito di inserire nel *Glossario* tutte quelle parole che possono essere fonte di ambiguità.

**1.3.1.2 Periodi** I *Verificatori* hanno il compito di calcolare l'indice Gulease<sub>g</sub> di ogni documento utilizzando lo strumento automatico preposto (vedi sezione Strumenti). Qualora non si ottenga un risultato soddisfacente il *Verificatore* deve applicare la tecnica walkthrough con l'obiettivo di individuare periodi troppo lunghi, che possono essere di difficile comprensione o leggibilità.

**1.3.1.3 Struttura del documento** I *Verificatori* devono verificare che la struttura di ogni documento rispetti i contenuti del documento stesso.

#### **1.3.1.4 Procedure**

**1.3.1.4.1 Resoconto attività di verifica sui documenti** La stesura dell'attività del resoconto dell'attività di verifica dovrà essere fatta al termine dell'attività di verifica. Il *Verificatore* dovrà riportare il numero di occorrenze per ognuna delle voci definite nella sezione norme. Qualora fossero stati trovati degli errori nelle attività di verifica precedenti, il numero di errori trovati andrà a sommarsi al numero di errori trovati nelle attività precedenti. Inoltre, il *Verificatore*, nel caso in cui dovesse trovare termini mancanti nel glossario, dovrà aggiornare la tabella definita nelle norme che regolano l'attività manuale di verifica.

**1.3.1.4.1.1 Attività manuale di verifica** Ogni attività manuale di verifica sui documenti deve essere accompagnata da un resoconto. Il resoconto deve contenere il numero di occorrenze riscontrate durante la verifica per ognuna delle seguenti voci:

- periodi troppo lunghi o complessi;
- parole non appropriate;
- incongruenze;
- errori concettuali;
- violazioni di quanto stabilito nelle norme tipografiche.

Al fine di rendere più veloce e chiara l'analisi di queste voci, è consigliabile che questi dati vengano rappresentati in forma tabellare.

Inoltre, deve essere sempre presente una tabella contenente il numero di occorrenze riscontrate durante la verifica per ognuna delle seguenti voci:

- termini candidati ad essere aggiunti al *Glossario*;
- termini aggiunti al *Glossario*.

**1.3.1.4.1.2 Attività automatica di verifica** Ogni attività automatica di verifica sui documenti deve essere accompagnata da un resoconto. Il resoconto deve contenere il numero di occorrenze riscontrate durante la verifica per ognuna delle seguenti voci:

- errori ortografici;
- utilizzo errato  $\text{\LaTeX}$ ;

Inoltre, devono essere riportati gli indici Gulpease calcolati per ognuno dei documenti sui quali è stata effettuata attività di verifica.

## **1.3.2 Diagrammi UML**

I *Verificatori* devono controllare tutti i diagrammi  $\text{UML}_g$  prodotti, sia che venga rispettato lo standard  $\text{UML}_g$ , sia che siano corretti semanticamente. I diagrammi utilizzati finora sono solamente i diagrammi dei casi d'uso.

**1.3.2.1 Diagrammi dei casi d'uso** Le verifiche che devono essere effettuate ai diagrammi dei casi d'uso, innanzitutto, devono riguardare il rispetto dello standard  $\text{UML}_g$ , soprattutto riguardanti inclusioni, generalizzazioni e estensioni. Successivamente si deve verificare che la semantica del diagramma sia corretta, ovvero che il diagramma rappresenti effettivamente ciò che si vorrebbe descrivere e modellare. Per quest'ultimo punto bisogna porre particolare attenzione sull'identificazione degli attori. Per questo motivo è prevista una procedura di verifica degli attori dei diagrammi UML in questo documento.

### **1.3.2.2 Procedure**

**1.3.2.2.1 Procedura di verifica attori dei diagrammi UML** Per identificare un attore può risultare utile seguire la seguente procedura:

1. Per prima cosa è utile chiedersi se ciò che vogliamo rappresentare come attore è una persona che interagisce col sistema. In caso affermativo è giusto rappresentarlo come attore, in caso negativo si deve andare al passo 2;
2. Se ciò che vogliamo rappresentare come attore non è una persona che interagisce col sistema bisogna chiedersi se questa "cosa" può cambiare insieme al design del sistema. In caso affermativo probabilmente ciò

che vogliamo rappresentare non è un attore ma una parte del sistema stesso. In caso negativo, invece, è con buona probabilità un attore.



**Figura 4:** Procedura di verifica di un attore

### 1.3.3 Issue tracking

L'*issue*<sub>g</sub> tracking è un'attività di supporto per i *Verificatori* ai quali permette di tenere traccia potenziali errori, segnalandoli al *Responsabile di progetto*. Lo strumento scelto per la gestione di questa attività è GitHub(vedi sezione Strumenti). Questo strumento è utile anche al *Responsabile di progetto* per assegnare i compiti di correzione degli errori.



**1.3.3.1 Sintassi di una label** Le label sono delle etichette che è possibile associare ad ogni  $issue_g$ . Ogni etichetta rappresenta lo stato nel quale si trova la issue. Le uniche etichette che sono accettate sono le seguenti:

- **Request:** rappresenta la richiesta di una  $issue_g$ ;
- **Question:** identifica una  $issue_g$  nella quale si sta discutendo per risolvere il problema;
- **ToDo:** identifica una  $issue_g$  per la quale è stata trovata la soluzione ma questa deve ancora essere applicata;
- **Working:** identifica una  $issue_g$  alla quale un membro del gruppo sta lavorando per applicare la soluzione;
- **Rejected:** identifica una  $issue_g$  che non è stata accettata poiché risulta essere futile o dannosa;
- **Done:** identifica una  $issue_g$  che è stata completata.

**1.3.3.2 Sintassi di una Issue** Ogni  $issue_g$  avrà un nome che dovrà seguire la seguente notazione:

**[D]:[S]**

dove:

- **D** rappresenta la sigla del documento di interesse;
- **S** rappresenta la sintesi della descrizione del problema.

Segue, poi, la descrizione della  $issue_g$ . Questa dovrà obbligatoriamente contenere:

- dove si trova il problema. Questo va specificato tramite sezione e, possibilmente, numero di riga dove inizia il problema;
- una descrizione dettagliata del problema che deve comprendere:
  - tutte gli errori associati al problema. Questi devono essere riportati nella forma di checkbox. Per fare ciò è necessario premettere alla descrizione di ogni errore i caratteri `- [ ]`. Questo permette di facilitare il tracciamento dello stato di avanzamento della  $issue_g$ .
  - una eventuale porzione di codice che potrebbe essere causa del problema. Per fare ciò è consigliabile utilizzare il comando:

```
```nomelinguaggio
codice da riportare
```.
```

Sostituendo “nomelinguaggio” al nome del linguaggio di programmazione utilizzato, si potrà vedere la porzione di codice evidenziata secondo la sintassi del linguaggio stesso, facilitandone la lettura.

- la motivazione per cui si è ritenuto necessario sollevare una  $issue_g$ .

### 1.3.3.3 Procedure

**1.3.3.3.1 Gestione di una issue** Qualora i *Verificatori* dovessero riscontrare delle anomalie, la procedura per la segnalazione e gestione del ticketing di una  $issue_g$  è la seguente:

1. il *Verificatore* che ha riscontrato un’anomalia $_g$  dovrà aprire una nuova  $issue_g$  assegnandole l’etichetta Request;
2. il *Responsabile di progetto* dovrà valutare la  $issue_g$  e decidere se:
  - la  $issue_g$  risulta essere futile o dannosa. In questo caso verrà rimossa l’etichetta Request ed assegnata l’etichetta Rejected. La  $issue_g$  verrà poi chiusa, terminando la procedura.
  - la  $issue_g$  risulta essere appropriata. In questo caso verrà rimossa l’etichetta Request ed assegnata l’etichetta Question.
3. i membri del gruppo dovranno discutere e proporre nuove idee al fine di risolvere la  $issue_g$ . Questa discussione non dovrà protrarsi per più di tre giorni. Qualora non si dovesse essere trovata una soluzione ottimale, il *Responsabile di progetto* dovrà decidere quale sia la soluzione migliore.
4. il *Responsabile di progetto*, trovata la soluzione da applicare, assegnerà, al membro del gruppo da lui ritenuto opportuno, un ticket $_g$ . L’assegnatario del ticket $_g$  è la persona incaricata all’applicazione della soluzione relativa alla  $issue_g$ . Inoltre, il *Responsabile di progetto* dovrà rimuovere dalla  $issue_g$  l’etichetta Question ed assegnare l’etichetta ToDo.
5. quando l’incaricato inizierà a lavorare sulla  $issue_g$  rimuoverà l’etichetta ToDo ed assegnerà l’etichetta Working. Ad ogni nuovo commit, l’incaricato dovrà riferire, nel messaggio del commit stesso, la  $issue_g$  sulla quale sta lavorando. Questo permetterà di tener traccia delle modifiche effettuate per risolvere la  $issue_g$ .

6. una volta che l'incaricato avrà terminato le modifiche, queste dovranno essere caricate nel repository<sub>g</sub> e l'etichetta della issue<sub>g</sub> dovrà essere modificata da Working a Done. Dopo che sono state svolte queste operazioni, la issue<sub>g</sub> verrà chiusa.



sections/in

**Figura 5:** Procedura di gestione di una issue. Parte 1



sections/in

**Figura 6:** Procedura di gestione di una issue. Parte 2

### 1.3.3.4 Strumenti

**1.3.3.4.1 Strumento per l'issue tracking** Lo strumento scelto per fare il tracking delle issue<sub>g</sub> è il servizio Issues messo a disposizione da GitHub<sub>g</sub>.

### 1.3.4 Analisi

Per poter verificare la qualità del prodotto<sub>g</sub> è stato scelto di applicare delle tecniche di analisi sui documenti e sul codice.

**1.3.4.1 Analisi statica** Per analisi statica si intende la valutazione di un sistema basata su struttura, contenuto e documentazione senza che questo venga eseguito. Questa tecnica è applicabile sia al codice, che alla documentazione stessa. L'analisi statica può avvenire con due modalità:

- **Walkthrough:** questa tecnica dev'essere applicata quando non si sa che errori o problematiche si stanno cercando. La tecnica consiste nel leggere il codice sorgente o il documento da cima in fondo per trovare anomalie di qualsiasi tipo;
- **Inspection:** questa tecnica dev'essere applicata quando si ha idea della problematica che si sta cercando e consiste in una lettura mirata del documento/codice, sulla base di una lista degli errori precedentemente stilata.

La tecnica walkthrough è molto onerosa e deve essere utilizzata soprattutto nelle prime fasi del progetto, quando non è già presente una lista degli errori comuni, oppure non si è ancora sufficientemente preparati riguardo un aspetto del progetto. Avanzando nelle fasi del progetto sarà utile stilare una lista quanto più possibile completa di errori comuni, in modo tale da evitare la tecnica walkthrough e applicare l'inspection.

**1.3.4.2 Analisi dinamica** L'analisi dinamica è una forma di valutazione di un sistema software<sub>g</sub>, oppure di qualche suo componente, basato sull'osservazione del suo comportamento durante l'esecuzione. Questa tecnica non è applicabile, quindi, per trovare errori nella documentazione. I test che devono essere implementati devono essere in numero relativamente ridotto e il più possibile di valore dimostrativo.

## 1.4 Validazione

Il processo di validazione ha l'obiettivo di verificare che:

- il prodotto finale sia conforme a quanto è stato pianificato;
- il prodotto finale sia abile nell'isolare e minimizzare gli effetti degli errori.

### 1.4.1 Responsabilità

Di seguito si elencano le rispettive responsabilità per la validazione del prodotto:

- i *Verificatori* hanno il compito di eseguire con attenzione i test, tracciandone i risultati che andranno poi valutati;
- il *Responsabile di progetto* revisiona i risultati dei test e decide se accettarli o ripeterli. Ha inoltre il compito di informare il committente dell'esecuzione e risultato dei test, fornendo indicazioni sulla possibilità di eseguirli in modo indipendente.

### 1.4.2 Procedure

**1.4.2.1 Procedura per la validazione** I passi per eseguire l'attività di validazione sono i seguenti:

1. i *Verificatori* eseguono manualmente i test sul prodotto finale, tracciando accuratamente i risultati ottenuti;
2. il *Responsabile di progetto* di progetto analizza i risultati e decide se:
  - (a) accettarli;
  - (b) chiedere una ripetizione di alcuni o tutti i test, eventualmente con verificatori diversi.
3. una volta accettati i test, il *Responsabile di progetto* consegna i risultati al proponente, informandolo sulle modalità di esecuzione indipendente della validazione.