

# 1. Processi di supporto

## 1.1 Documentazione

### 1.1.1 Descrizione

Questo capitolo descrive le scelte intraprese per la stesura, verifica e approvazione riguardante la documentazione ufficiale. Tali norme sono tassative per tutti i documenti formali. Tali documenti sono elencati successivamente nella sotto-sezione "Documenti correnti".

### 1.1.2 Ciclo di vita documentazione

Ogni documento formale deve passare gli stadi di "Sviluppo", "Verifica" e "Approvato".

- **Sviluppo:** inizia con la creazione del documento e termina con la conclusione della stesura di tutte le sue parti. In questa fase i Redattori aggiungono le parti assegnate tramite ticket. Il passaggio alla fase di Verifica è automatizzato con la segnalazione al Responsabile;
- **Verifica:** il documento entra nella fase di verifica dopo l'assegnazione da parte del Responsabile. I Verificatori effettueranno le procedure di controllo dello stesso elencate nella sezione  
Al termine del controllo in caso positivo il documento entra automaticamente in fase di "Approvato", altrimenti i loro riscontri vengono consegnati al Responsabile di Progetto, che provvederà ad assegnare nuovamente il documento ad un Redattore attraverso una nuova fase di Sviluppo;
- **Approvato:** l'approvazione di un documento coincide con il superamento positivo da parte di un Verificatore dello stadio di Verifica e diventa una versione ufficiale.

### 1.1.3 Separazione documenti interni ed esterni

Ogni documento formale dovrà essere classificato come Interno oppure Esterno con le seguenti differenze:

- **Interno:** ha utilizzo interno al gruppo 353, redatto in lingua Italiana;
- **Esterno:** verrà condiviso con la Proponente ed i Committenti, nel caso sia utile per il deploy dell'applicazione web o per il suo utilizzo da parte degli utenti finali, deve essere redatto in lingua Inglese.

Un documento formale farà sempre parte di una delle due categorie elencate.

### 1.1.4 Nomenclatura documenti

Tutti i documenti formali tranne “...” seguiranno questo sistema di nomenclatura:

- **NomeDocumento:** indica il nome del documento senza spazi e lettere maiuscole all'inizio di ogni parola;
- **vX.Y.Z:** indica il numero di versionamento con X,Y e Z numeri interi e non negativi.

Di seguito la spiegazione che assumono le versione del documento:

- **X:** rappresenta il numero di pubblicazioni ufficiali del documento: ogni qual volta il documento viene pubblicato il valore di Y e Z viene azzerato, quello di X incrementato di uno;
- **Y:** rappresenta il numero di verifiche completate con successo, in caso di mancato superamento della fase di verifica il valore non va modificato, mentre se superata viene incrementato di uno e si azzerà il valore della Z;
- **Z:** rappresenta il numero di modifiche effettuate al documento durante il suo sviluppo.

Formato dei file: ogni documento si trova nel formato .tex durante il suo ciclo di vita. Dopo lo stato di “Approvato” per il documento viene quindi creato un PDF contenente la versione approvata dal Responsabile.

### 1.1.5 Documenti correnti

Qui di seguito si presentano i documenti formali in ordine alfabetico, classificati per appartenenza (Interno ed Esterno):

- **Analisi dei Requisiti:** utilizzo Esterno, sigla (AR)  
Documento per esporre e scomporre i requisiti<sub>G</sub> del progetto contenente casi d'uso relativi al prodotto e diagrammi di interazione con l'utente. Viene scritto dagli Analisti dopo aver analizzato il capitolato e interagendo con il Proponente in riunioni esterne;
- **Glossario:** utilizzo Esterno, sigla (GL)  
Documento per raccogliere le definizioni dei termini o concetti che saranno usati nei documenti formali per facilitarne la comprensione;
- **Norme di Progetto:** utilizzo Interno, sigla (NdP)  
Documento per mostrare le direttive e gli standard utilizzati all'interno del gruppo di lavoro 353 per lo sviluppo del progetto;
- **Piano di Progetto:** utilizzo Esterno, sigla (PdP)  
Documento per l'analisi e la pianificazione della gestione delle risorse di tempo e umane;
- **Piano di Qualifica:** utilizzo Esterno, sigla (PdQ)  
Documento per descrivere standard e obiettivi che il gruppo dovrà raggiungere per garantire la qualità di prodotto e processo;
- **Studio di Fattibilità:** utilizzo Interno, sigla (SdF)  
Documento per indicare le riflessioni, punti di forza e caratteristiche sfavorevoli per ogni capitolato che ha portato alla scelta finale del gruppo.

### 1.1.6 Norme

#### 1.1.6.1 Struttura dei documenti

Ogni documento è realizzato a partire da una disposizione prestabilita che dovrà essere conforme per ogni documento ufficiale ad eccezione dei verbali:

- **Frontespizio:** questa sezione si troverà nella prima pagina di ogni documento e conterrà:
  - Logo del gruppo;
  - Titolo del documento;
  - Versione del documento con l'ultima data di modifica;

- Nome del gruppo;
- Nome del progetto.
- **Informazioni sul documento:** conterrà la lista di responsabili, redattori, verificatori del documento e infine lo stato e la tipologia di uso vedi sotto sezione Separazione documenti interni ed esterni;
- **Diario delle modifiche:** Questo diario sarà presente nella seconda pagina del documento sotto forma di tabella in ordine di versione decrescente con righe composte da: versione, data, descrizione modifiche, autore, ruolo;
- **Indice delle sezioni:** L'indice delle sezioni conterrà l'elenco di tutti gli argomenti trattati nel documento con la seguente struttura: titolo, argomento e numero pagina;
- **Indice delle tabelle:** Sezione contenente l'elenco delle tabelle con struttura identica all'indice delle sezioni;
- **Indice delle figure:** Sezione contenente l'elenco delle figure con struttura identica all'indice delle sezioni;
- **Introduzione:** scopo del documento, info glossario e riferimenti utili?
- **Contenuto del documento:** il resto del documento è occupato dal contenuto.

#### 1.1.6.2 Norme tipografiche

- **Intestazione** ogni pagina dopo frontespizio presenterà sulla sinistra il logo del gruppo e a destra il nome del capitolo corrente;
- **Piè pagina:** a sinistra è presente il nome del documento corrente e a destra il numero di pagina;
- **Virgolette:** alte singole ' ' per singolo carattere, alte doppie " " per racchiudere stringhe mentre quelle basse '<<' '>>' per racchiudere citazioni;
- **Parentesi:** tonde per descrivere esempi e fornire sinonimi o precisazioni, quadre per rappresentare uno standard ISO, uno stato relativo a un ticket o un riferimento ad un codice definito all'interno del documento stesso;
- **Punteggiatura:** ogni segno di punteggiatura deve essere seguito da uno spazio e non avere spazi precedenti al segno stesso;
- **Stile del testo:**

- Corsivo: Per dare enfasi ad una parola, un concetto o per indicare il nome di un tool/tecnologia;
- Grassetto: Per i titoli, sottotitoli ed elementi di elenchi e liste;
- Sottolineato: Per indicare dei collegamenti ipertestuali.
- **Elenchi:** ogni elenco avrà la prima parola di ogni elemento maiuscola seguita dal carattere 'due punti' (:) seguito dalla sua descrizione, mentre al termine dell'elemento si inserirà sempre il carattere 'punto e virgola' (;) tranne per l'ultimo elemento della lista per cui si userà il carattere 'punto' (.)
- **Note a Piè pagina:** seguono le seguenti regole:
  - Devono presentare una numerazione progressiva all'interno del documento;
  - Devono essere scritte solo una volta;
  - Il primo carattere di ogni nota deve essere maiuscolo. Fanno eccezione i casi in cui la parola sia un acronimo.
- **Formati:**
  - Date: scritte con lo standard DD-MM-YYYY dove YYYY indica l'anno, MM il mese e DD il giorno, inseribili nel formato corretto attraverso il comando `\nData`;
  - Grassetto: lo stile grassetto va utilizzato per i titoli dei paragrafi e per i titoli degli elementi di un elenco;
  - URI: lo stile utilizzato per un URI è il corsivo di colore blu, in modo da mantenere una continuità con lo standard web attuale, utilizzabile col comando personalizzato latex `\nURI`.
- **Ruoli/Fasi/Revisioni di Progetto**
- **Nomi:** sono stati realizzati dei comandi personalizzati per poter richiamare la visualizzazione dei seguenti termini:
  - nome gruppo: `\gruppo`, visualizza "353";
  - nomi propri: `\NomeProprioPersona`;
  - nome progetto: `\progetto`, visualizza "Marvin";
  - nome di un file: `\nFile`;
  - nome di un documento: `\nDoc`;
  - percorso cartelle: `\nPath`.

- **Componenti grafiche:**

- immagini: i formati ammessi sono PNG e PDF
- tabelle: devono rispettare lo stile del template realizzato

### 1.1.7 Struttura documentazione

E' stato creato un template di documento utilizzabile per tutti i documenti ufficiali in modo tale da favorire lo sviluppo dei documenti.

Il template è basato sulle norme di documentazione elencate nella sezione precedente, inoltre è stata scritta una pagina di showcase per facilitare il mantenimento corretto delle strutture dei documenti.

### 1.1.8 Gestione termini Glossario

Il glossario è un documento unico per tutti i documenti, esso conterrà tutte le definizioni, in ordine lessicografico crescente, dei termini inerenti al tema del progetto, che possono essere fraintesi o non ben comprensibili. I termini inseriti nel glossario saranno contrassegnati da una G pedice all'interno dei documenti.

Ovviamente prima di inserire un nuovo termine bisognerà assicurarsi che non sia già presente, in modo da evitare duplicati.

Il comando `\citGloss` da utilizzare per contrassegnare un termine da glossario all'interno dei documenti è `\citGloss`, mentre per l'inserimento di una nuova parola all'interno del glossario viene utilizzato `\glossDef`.

La scelta di creare un comando apposito per una operazione "elementare" è scaturita dall'agevolazione che porta alla stesura della documentazione: avendo un modo univoco di riconoscere i termini all'interno del glossario è possibile automatizzare la verifica del soddisfacimento delle norme all'interno dei documenti.

### 1.1.9 Strumenti a supporto della documentazione

La stesura dei documenti deve essere effettuata utilizzando il linguaggio di markup `LaTeX` e l'ambiente `TeXstudio` con dizionario italiano ed inglese installati.

## 1.2 Qualità

### 1.2.1 Descrizione

### 1.2.2 Metriche

- **Metriche per processi:** ogni processo dovrà avere uno standard di qualità elevato definito come unione di:
  - Tempo: richiesto per completamento
  - Risorse: uomo o software richieste
  - Occorrenze: ossia il numero di volte in cui si presenterà un particolare evento come il numero di difetti di una caratteristica del prodotto.
- **Metriche per i documenti:** per la verifica dei documenti è stata scelto l'indice di leggibilità Gulpease che viene calcolato tramite funzione Latex. I risultati sono compresi tra 0 e 100 dove il valore 100 indica una leggibilità massima, si è scelto di considerare un documento sufficientemente leggibile all'ottenimento del valore X;
- **Metriche per il software:** al fine di perseguire gli obiettivi qualitativi prefissati nel Piano Di Qualifica è doveroso definire delle metriche come copertura del codice tramite test automatici;
- **Metriche per i feedback di miglioramento:** i feedback gestiti come spiegato nelle sezioni successive presentano un indice di occorrenze che viene incrementato in automatico ad ogni nuova segnalazione.

## 1.3 Configurazione

### 1.3.1 Controllo di versione

#### 1.3.1.1 Descrizione

Per le parti versionabili del progetto e per i documenti ufficiali si è scelto l'utilizzo della tecnologia Git, ospitata sul servizio di hosting GitHub<sub>G</sub>. La condivisione dei documenti informali e delle parti non versionabili è invece effettuata tramite l'uso di una cartella Google Drive condivisa.

### 1.3.1.2 Struttura delle repository

E' stata realizzata solo una repository<sub>G</sub> durante la fase di RR ossia quella relativa alla documentazione denominata "Documentazione 353", si prevede inoltre di creare ulteriori repository<sub>G</sub> per suddividere lo sviluppo e la codifica dell'applicazione del progetto.

I file interni al repository<sub>G</sub> "Documentazione 353" sono organizzati secondo questa struttura:

- **Esterni:**
  - AnalisiDeiRequisiti;
  - Glossario;
  - PianoDiProgetto;
  - PianoDiQualifica;
  - Verbali esterni.
- **Interni:**
  - NormeDiProgetto;
  - Glossario;
  - StudioDiFattibilita;
  - Verbali interni.

All'interno di ogni cartella è stato definito un file LaTeX principale che assume il nome del documento stesso ed il file diariomodifiche.tex per mantenere traccia delle modifiche effettuate.

Nella root della repository<sub>G</sub> è stato messo invece un file .sh gestito tramite Travis CI che controlla automaticamente gli errori in tutti i documenti presenti e compila un log con i risultati ottenuti.

### 1.3.1.3 Ciclo di vita dei branch

Per sfruttare il parallelismo nello sviluppo di uno stesso documento sono stati creati appositamente dei branch denominati con il nome del membro o dei membri del gruppo che devono lavorare su una determinata parte, mentre i documenti baseline saranno invece contenuti nel master branch.

Il merge col master avviene quindi solamente quando un documento si trova in stato di "Approvato".



#### 1.3.1.4 Aggiornamento della repository

Per l'aggiornamento della repository<sub>G</sub> è previsto il seguente sotto processo motivato dalla sezione precedente "ciclo di vita":

- Verificare di trovarsi sul branch personale con "git branch" (quella selezionata presenta un asterisco \*) e in caso di riscontro negativo cambiare branch con "git checkout"
- 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" per ottenere ed applicare i commit mancanti;
  - Dare il comando "git stash apply" per ripristinare le modifiche.

In questo modo il repository<sub>G</sub> locale risulta aggiornato rispetto il repository<sub>G</sub> remoto, mantenendo le modifiche personali apportate;

- Dare il comando "git add \*" , che aggiungerà i file modificati e quelli nuovi;
- Dare il comando "git commit" e successivamente riassumere le modifiche effettuate, in caso sia utile si può aggiungere un messaggio esteso di descrizione;
- Dare il comando "git push" per completare l'operazione e fornire le modifiche agli altri membri del gruppo.

## 1.4 Verifica

### 1.4.1 Descrizione

Un processo fondamentale per il proseguimento e l'evoluzione di un progetto è la verifica su ogni suo sottoprodotto che porta alla creazione di un singolo componente.

In questa sezione si descriveranno gli strumenti e i metodi che verranno usati per la verifica del codice e dei documenti durante la loro realizzazione.

Per quanto riguarda questa prima parte di progetto la verifica si è concentrata essenzialmente su documenti e diagrammi.

### 1.4.2 Analisi statica

L'analisi statica è una tecnica di analisi applicabile sia alla documentazione che al codice e permette di effettuare la verifica di quanto prodotto individuando errori ed anomalie.

Essa può essere svolta in due modi diversi:

- **Walkthrough:** tecnica applicata quando non si sanno le tipologie di errori o problemi che si stanno cercando e quindi prevede una lettura da cima a fondo del codice o documento per trovare anomalie di qualsiasi tipo.
- **Inspection:** tecnica da applicare quando si ha idea delle possibili problematiche che si stanno cercando e si attua leggendo in modo mirato il documento/codice sulla base di una lista di possibili errori precedentemente stilata.

Sono stati utilizzati degli strumenti e sotto processi per velocizzare e rendere più veloce questa analisi con l'utilizzo di funzionalità integrate nell'editor TexStudio oltre all'utilizzo di diverse checklist online per garantire l'esecuzione di ogni controllo da parte del Verificatore.

### 1.4.3 Analisi dinamica

Il processo di analisi dinamica consiste nella realizzazione ed esecuzione di una serie di test sul codice del software.

### 1.4.4 Verifica Diagrammi UML

I Verificatori devono controllare tutti i diagrammi UML prodotti rispettino lo standard UML e che siano corretti semanticamente.

### 1.4.5 Strumenti usati per la verifica

- **Software:** verranno usati per la verifica Mocha ed Enzyme per React, Jest per Redux, mentre per i contratti Solidity useremo Mocha e Chai;
- **Documenti:** Per il controllo dei documenti prodotti utilizzeremo le funzionalità di Textstudio assieme a script bash eseguiti automaticamente da un Bot<sub>G</sub> per controllare che non ci siano errori nei file LaTeX, per assicurarsi che il Glossario non presenti mancanze o duplicati e verificare l'assenza di errori ortografici;

- **Gestione processi e feedback:** è stato scelto di utilizzare il sistema integrato di issues presente su GitHub<sub>G</sub> per permettere un dialogo maggiore per ogni singola issue.