



Progetto: *HD Viz*  
[codebusterswe@gmail.com](mailto:codebusterswe@gmail.com)

## Norme di Progetto

### Informazioni sul documento

<b>Versione</b>	1.0.0
<b>Approvatori</b>	Sassaro Giacomo
<b>Redattori</b>	Pirolo Alessandro Zenere Marco Rago Alessandro Safdari Houssaine
<b>Verificatori</b>	Baldisseri Michele Scialpi Paolo
<b>Uso</b>	Interno
<b>Distribuzione</b>	Prof. Vardanega Tullio Prof. Cardin Riccardo Gruppo <i>CodeBusters</i>

### Descrizione

Questo documento racchiude le regole, gli strumenti e le convenzioni adottate dal gruppo nello svolgimento del progetto HD Viz.

## Registro delle modifiche

Versione	Data	Nominativo	Ruolo	Verificatore	Descrizione
0.0.5	2020-12-19	Rago Alessandro	Analista	-	Stesura §3.2 e in parte §3.3
0.0.4	2020-12-19	Zenere Marco	Analista	-	Processi primari.
0.0.3	2020-12-17	Pirolo Alessandro	Analista	-	Iniziata stesura §3 fino a §3.1.6.3
0.0.2	2020-12-15	Pirolo Alessandro	Analista	-	Stesura §1.
0.0.1	2020-12-14	Zenere Marco	Analista	-	Creazione scheletro documento, introduzione e paragrafi.

# Indice

<b>1</b>	<b>Introduzione</b>	<b>5</b>
1.1	Scopo del documento . . . . .	5
1.2	Scopo del capitolato . . . . .	5
1.3	Glossario . . . . .	5
1.4	Riferimenti . . . . .	5
1.4.1	Riferimenti normativi . . . . .	5
1.4.2	Riferimenti informativi . . . . .	5
<b>2</b>	<b>Processi Primari</b>	<b>7</b>
2.1	Fornitura . . . . .	7
2.1.1	Descrizione . . . . .	7
2.1.2	Scopo della fornitura . . . . .	7
2.1.3	Proponente . . . . .	7
2.1.4	Documenti . . . . .	8
2.1.4.1	Studio di fattibilità . . . . .	8
2.1.4.2	Piano di qualifica . . . . .	8
2.1.4.3	Piano di progetto . . . . .	8
2.1.5	Strumenti . . . . .	9
2.1.5.1	Excel . . . . .	9
2.1.5.2	Microsoft Planner . . . . .	9
2.2	Sviluppo . . . . .	10
2.2.1	Descrizione . . . . .	10
2.2.2	Aspettative . . . . .	10
2.2.3	Attività . . . . .	10
2.2.4	Studio di fattibilità . . . . .	10
2.2.4.1	Finalità . . . . .	10
2.2.4.2	Requisiti . . . . .	11
2.2.4.3	Casi d'uso . . . . .	11
2.2.4.4	Codice identificativo casi d'uso . . . . .	11
2.2.4.5	Struttura dei requisiti . . . . .	11
2.2.4.6	UML . . . . .	12
2.2.5	Progettazione . . . . .	12
2.2.5.1	Scopo . . . . .	12
2.2.5.2	Aspettative . . . . .	12
2.2.5.3	Descrizione . . . . .	12
2.2.5.4	Technology Baseline . . . . .	12
2.2.5.5	Product Baseline . . . . .	13
2.2.6	Codifica . . . . .	13
2.2.6.1	Scopo . . . . .	13
2.2.6.2	Aspettative . . . . .	13

2.2.6.3	Stile della codifica . . . . .	13
<b>3</b>	<b>Processi di Supporto</b>	<b>14</b>
3.1	Documentazione . . . . .	14
3.1.1	Scopo . . . . .	14
3.1.2	Aspettative . . . . .	14
3.1.3	Descrizione . . . . .	14
3.1.4	Ciclo di vita del documento . . . . .	14
3.1.5	Template . . . . .	15
3.1.6	Struttura del documento . . . . .	15
3.1.6.1	Registro delle modifiche . . . . .	16
3.1.7	Convenzioni . . . . .	16
3.1.8	Elementi grafici . . . . .	16
3.1.9	Strumenti . . . . .	16
3.2	Gestione della Configurazione . . . . .	17
3.2.1	Scopo . . . . .	17
3.2.2	Aspettative . . . . .	17
3.2.3	Descrizione . . . . .	17
3.2.4	Versionamento . . . . .	17
3.2.4.1	Codice di versione . . . . .	17
3.2.4.2	Sistemi software utilizzati . . . . .	18
3.2.5	Struttura dei repository . . . . .	18
3.2.5.1	Repository utilizzati . . . . .	18
3.2.5.2	CodeBuster-Docs . . . . .	18
3.2.5.3	Gestione dei cambiamenti in CodeBuster-Docs . . . . .	19
3.3	Gestione della Qualità . . . . .	20
3.3.1	Scopo . . . . .	20
3.3.2	Aspettative . . . . .	20
3.3.3	Descrizione . . . . .	20
3.3.4	Attività . . . . .	20
3.3.5	Strumenti utili . . . . .	21
3.4	Verifica . . . . .	22
3.5	Validazione . . . . .	23
<b>4</b>	<b>Processi Organizzativi</b>	<b>24</b>
4.1	Gestione Organizzativa . . . . .	24
4.1.1	Scopo . . . . .	24
4.1.2	Aspettative . . . . .	24
4.1.3	Descrizione . . . . .	24
4.1.4	Ruoli di Progetto . . . . .	24
4.1.5	Responsabile di progetto . . . . .	25
4.1.6	Amministratore di progetto . . . . .	25
4.1.7	Analista . . . . .	25

4.1.8	Progettista . . . . .	25
4.1.9	Programmatore . . . . .	25
4.1.10	Verificatore . . . . .	25
4.1.11	Procedure . . . . .	25

# 1 Introduzione

## 1.1 Scopo del documento

Questo documento ha lo scopo di definire le linee guida di tutti i processi istanziati dal gruppo *CodeBusters*, inoltre contiene l'organizzazione e l'uso di tutte le risorse di sviluppo e le convenzioni che il gruppo decide di attuare sull'uso delle tecnologie, sullo stile di codifica e di scrittura. Ogni membro del gruppo è obbligato a tenere in considerazione questo documento per garantire maggiore uniformità e coerenza del materiale prodotto.

## 1.2 Scopo del capitolato

Oggigiorno, anche i programmi più tradizionali gestiscono e memorizzano una grande mole di dati e di conseguenza serve un software in grado di eseguire un'analisi e una interpretazione delle informazioni.

Il capitolato<sup>G</sup> C4 ha come obiettivo quello di creare un'applicazione di visualizzazione di dati con numerose dimensioni in un formato comprensibile dall'occhio umano. A questo scopo è necessario utilizzare algoritmi di intelligenza artificiale, o nel caso svilupparne di nuovi, che, agendo sulla distanza dei vari punti del grafico, riescano a sviluppare un modello semplificato che ne evidenzi i cluster<sup>G</sup>. L'applicazione dovrà inoltre agire su questi grafici creati evidenziando i dati ottenuti.

## 1.3 Glossario

Per evitare ambiguità relative alle terminologie utilizzare, è stato compilato il *Glossario 1.0.0*. In questo documento sono riportati tutti i termini di particolare importanza e con un significato particolare. Questi termini sono evidenziati da una 'G' ad apice.

## 1.4 Riferimenti

### 1.4.1 Riferimenti normativi

- Standard ISO/IEC 12207:1995:  
[https://www.math.unipd.it/~tullio/IS-1/2009/Approfondimenti/ISO\\_12207-1995.pdf](https://www.math.unipd.it/~tullio/IS-1/2009/Approfondimenti/ISO_12207-1995.pdf)
- Capitolato d'appalto C4 - HD Viz: visualizzazione di dati multidimensionali:  
<https://www.math.unipd.it/~tullio/IS-1/2020/Progetto/C4.pdf>

### 1.4.2 Riferimenti informativi

- Piano di Progetto v 1.0.0
- Piano di Qualifica v 1.0.0

- Guide to the Software Engineering Body of Knowledge(SWEBOK), 2014
- Software Engineering - Ian Sommerville - 10 th Edition (2010):  
(formato cartaceo);
- Slide T3 del corso Ingegneria del Software - Ciclo di vita del software:  
<https://www.math.unipd.it/~tullio/IS-1/2020/Dispense/L03.pdf>

## 2 Processi Primari

### 2.1 Fornitura

#### 2.1.1 Descrizione

In questa sezione si presenteranno le regole a cui si atterranno i membri del gruppo Codebusters nei processi che comprendono lo studio del capitolato fino alla candidatura a fornitore del progetto HD Viz del proponente Zucchetti e dei committenti Prof. Vardanega Tullio e Prof. Cardin Riccardo.

#### 2.1.2 Scopo della fornitura

Il fine del processo di fornitura è di scegliere le procedure e le risorse atte a perseguire lo sviluppo del progetto. Dopo aver ricevuto le richieste del proponente, il gruppo redige uno studio di fattibilità e la fornitura può essere avviata per completare tali richieste.

Il proponente e il fornitore stipuleranno un contratto per la consegna del prodotto.

Si dovrà poi sviluppare un piano di progetto partendo dalla determinazione delle procedure e delle risorse necessarie. Da quel momento fino alla consegna del prodotto il piano di progetto scaglionerà le attività da svolgere.

Il processo di fornitura è composto dalle seguenti fasi:

1. avvio;
2. approntamento di risposte alle richieste;
3. contrattazione;
4. pianificazione;
5. esecuzione e controllo;
6. revisione e valutazione;
7. consegna e completamento.

#### 2.1.3 Proponente

Codebusters vorrebbe avere un contatto costante con il proponente in modo da avere un riscontro:

- sulle soluzioni utilizzate;
- sulle tempistiche di consegna del prodotto;
- su eventuali dubbi;
- stimare i costi;
- su vincoli e requisiti.



#### **2.1.4 Documenti**

Di seguito sono descritti i documenti che saranno redatti durante questa fase.

##### **2.1.4.1 Studio di fattibilità**

Documento che contiene la stesura dello studio di fattibilità riguardante i sette capitolati proposti, per ciascuno di essi vengono evidenziati i seguenti aspetti:

- Descrizione generale;
- Prerequisiti e tecnologie coinvolte;
- Vincoli;
- Aspetti positivi;
- Aspetti critici.

Infine, per ogni capitolato vengono espone le motivazioni e le ragioni per cui il gruppo ha scelto come progetto il capitolato *C4 HD Viz* a discapito degli altri sei proposti.

##### **2.1.4.2 Piano di qualifica**

Il piano di qualifica, redatto dal verificatore, contiene tutte le misure da adottare per garantire la qualità del prodotto. È suddiviso nelle seguenti parti:

- da inserire

##### **2.1.4.3 Piano di progetto**

Gli amministratori e il responsabile dovranno redigere questo documento che dovrà essere seguito durante tutto il corso del progetto. È suddiviso nelle seguenti sezioni:

- Analisi dei rischi;
- Modello di sviluppo;
- Pianificazione;
- Preventivo;
- Consuntivo;
- Organigramma.

### **2.1.5 Strumenti**

#### **2.1.5.1 Excel**

Strumento utilizzato per creare grafici, fare calcoli e presentare tabelle organizzative.

#### **2.1.5.2 Microsoft Planner**

Per gestire le risorse, le task che ciascun membro del gruppo deve completare e verificare se queste sono in svolgimento, i responsabili hanno utilizzato questo strumento. Permette di assegnare attività a risorse e di verificare che vengano completate in linea con i tempi previsti.

## **2.2 Sviluppo**

### **2.2.1 Descrizione**

Lo scopo del processo di sviluppo è descrivere i task e le attività di analisi, progettazione, codifica, integrazione, test, installazione ed accettazione, relative al prodotto software da sviluppare.

### **2.2.2 Aspettative**

Le aspettative sono le seguenti:

- determinare vincoli tecnologici;
- determinare gli obiettivi di sviluppo;
- determinare vincoli di design;
- realizzare un prodotto finale che superi i test e soddisfi requisiti e richieste del proponente.

### **2.2.3 Attività**

Le attività del processo di sviluppo sono tre:

- Studio di fattibilità;
- Progettazione;
- Codifica.

### **2.2.4 Studio di fattibilità**

Questo documento, redatto dagli analisti, contiene:

- la descrizione degli attori del sistema;
- i casi d'uso individuati a partire dai requisiti;
- una visione del problema più chiara possibile per i progettisti;
- fornire ai verificatori riferimenti per l'attività di controllo dei test;
- calcolare la mole di lavoro per tracciare dei riferimenti per una stima dei costi.

#### **2.2.4.1 Finalità**

Lo scopo dell'attività è redigere tutti i requisiti in un documento.

#### **2.2.4.2 Requisiti**

I requisiti si raccolgono tramite:

- Lettura del capitolato;
- Visione della presentazione del capitolato;
- Confronto con l'azienda;

#### **2.2.4.3 Casi d'uso**

É un diagramma che esprime un comportamento o un modo di utilizzare il prodotto. É costituito da:

- Codice identificativo e titolo;
- Attore primario;
- Precondizioni;
- Postcondizioni;
- Scenario principale;
- Estensioni.

#### **2.2.4.4 Codice identificativo casi d'uso**

Un caso d'uso è così identificato:

**UC[Numero caso d'uso].[caso d'uso figlio] - titolo caso d'uso**

Dove caso d'uso figlio è il sottocaso del caso d'uso principale.

#### **2.2.4.5 Struttura dei requisiti**

- **Codice identificativo:**

**R[Importanza][Tipologia][Codice]**

Per importanza si intende un numero da 1 a 3 che rappresenta:

1. requisito obbligatorio;
2. requisito desiderabile ma non essenziale per il funzionamento;
3. requisito opzionale.

Per tipologia si intende una lettera che rappresenta la natura del requisito:

**V** : Vincolo

**P** : Prestazionale

**Q** : Qualitativo

**F** : Funzionale

- **classificazione:** per rendere la tabella più esplicativa viene riportata nuovamente l'importanza del requisito nonostante sia già scritta nel codice identificativo;
- **descrizione:** una sintetica descrizione del requisito
- **fonti:** come scritto in precedenza, vi sono diverse fonti da cui possono derivare i requisiti. L'origine dei requisiti viene quindi riportata in questa sezione.

#### 2.2.4.6 UML

I diagrammi UML<sup>G</sup> devono essere realizzati usando la versione del linguaggio v2.0.

### 2.2.5 Progettazione

#### 2.2.5.1 Scopo

Questa attività ha la funzione di definire una soluzione al capitolato proposto basandosi sull'analisi dei requisiti. Mentre l'analisi dei requisiti divide il problema nei requisiti da soddisfare, la progettazione incorpora le parti specificando le funzionalità dei sottosistemi e riconducendo ad un'unica soluzione.

#### 2.2.5.2 Aspettative

Riuscire ad arrivare, al termine di questa attività, ad una architettura di sistema.

#### 2.2.5.3 Descrizione

É formata da due parti:

**Technology Baseline:** motiva le tecnologie, i framework, e le librerie selezionate per la realizzazione del prodotto;

**Product Baseline:** illustra la baseline architetturale (design e coding) del prodotto, coerente con la Technology Baseline.

#### 2.2.5.4 Technology Baseline

Sarà il progettista ad occuparsene e dovrà contenere:

- Diagrammi UML delle classi, di attività, di sequenza e dei package;
- Tecnologie adottate, motivando i motivi di tali scelte;

- Design pattern accompagnato da una descrizione e un diagramma che ne esponga la struttura;
- La relazione tra ciascuna componente e il requisito che soddisfa per avere un tracciamento;
- Proof of Concept, ovvero un dimostratore eseguibile per dimostrare che ogni componente funzioni nel modo voluto.

#### **2.2.5.5 Product Baseline**

Il progettista dovrà occuparsi anche di questa parte che conterrà:

- una definizione delle classi, evitando nomi e funzionalità ridondanti;
- tracciamento delle classi, ovvero ciascun requisito deve essere soddisfatto da una classe;
- test di unità su ogni componente in modo da verificare il corretto funzionamento.

### **2.2.6 Codifica**

#### **2.2.6.1 Scopo**

L'attività di codifica ha il fine di concretizzare la progettazione con la programmazione del software vero e proprio.

#### **2.2.6.2 Aspettative**

Questa attività dovrà avere come risultato un prodotto software avente le caratteristiche e i requisiti concordati con il proponente. Il codice generato dovrà rispettare alcune norme per poter essere leggibile e poter facilmente intervenire in seguito nelle attività di manutenzione, modifica, verifica e validazione.

#### **2.2.6.3 Stile della codifica**

## 3 Processi di Supporto

### 3.1 Documentazione

#### 3.1.1 Scopo

Tutti i processi e le attività di sviluppo devono essere documentate. Questa sezione ha lo scopo di definire le norme, le convenzioni e la struttura organizzativa riguardanti la documentazione, oltre che la definizione degli strumenti necessari alla sua stesura.

#### 3.1.2 Aspettative

Le aspettative di questo processo sono:

- avere una chiara struttura per i documenti, in modo da ottenere un risultato uniforme alla fine del suo ciclo di vita;
- avere delle norme e convenzioni ben precise che coprono tutti gli aspetti della stesura di un documento, in modo che tutti membri di *CodeBusters* possano lavorare senza dover interpellare il gruppo per prendere decisioni riguardo un generico aspetto.

#### 3.1.3 Descrizione

La documentazione è un processo per registrare le informazioni prodotte da una attività del ciclo di vita. Il processo contiene una serie di attività che pianificano, progettano, sviluppano, producono, modificano, distribuiscono e mantengono quei documenti necessari a tutti gli interessati, come manager, ingegneri e utenti.

#### 3.1.4 Ciclo di vita del documento

Ogni documento passa per queste fasi:

- **Pianificazione:** il documento viene pensato e vengono organizzate le varie parti. Questo accade soprattutto quando le informazioni sono numerose e complesse;
- **Impostazione:** viene creata la bozza e la struttura del documento;
- **Realizzazione:** viene redatto il contenuto del documento;
- **Verifica:** ogni sezione del documento è soggetta a revisioni da parte dei verificatori per correggere e, di conseguenza, sistemare;
- **Approvazione:** l'approvatore stabilisce che il documento è stato completato ed è pronto per essere rilasciato.

### 3.1.5 Template

Il gruppo ha deciso di creare un template con l'utilizzo di  $\text{\LaTeX}$ , grazie al quale viene standardizzata la struttura del documento. In questo modo i componenti del gruppo si occupano unicamente di redigere il contenuto dei singoli testi senza doversi di . Più precisamente, nel template vengono definite la prima pagina, la struttura del registro delle modifiche e l'indicizzazione delle sezioni e sottosezioni.

### 3.1.6 Struttura del documento

Ogni documento è formato da diverse sezioni, ognuna definita dal proprio file  $\text{\LaTeX}$ . La parte principale è chiamata "*nomedoc.tex*" (dove *nomedoc* sta ad indicare il nome del documento) ed ha il compito di includere le seguenti componenti:

- i file  $\text{\LaTeX}$  delle sezioni, che contengono il contenuto del testo vero e proprio. Se una sezione contiene numerose sottosezioni, allora il file avrà il compito di includere i file delle varie sottosezioni ;
- il registro delle modifiche, che contiene una lista ;
- "*String.tex*", che contiene una serie di comandi  $\text{\LaTeX}$  personalizzati che facilitano la scrittura di parole frequentemente utilizzate;
- "*Comandi.tex*", che contiene una serie di comandi  $\text{\LaTeX}$  personalizzati

#### Prima pagina

La prima pagina di un documento è formata da:

- **Logo:** logo di *CodeBusters* posto in alto e centralizzato;
- **Progetto ed e-mail:** sotto il logo e centralizzato viene scritto il nome del progetto e la mail del gruppo *CodeBusters*;
- **Titolo:** il nome del documento;
- **Informazioni sul documento:** sotto la titolo è presente una tabella con le seguenti informazioni riguardanti il documento:
  - **Versione:** versione del documento;
  - **Approvatori:** nomi dei componenti del gruppo che svolgono il ruolo di approvatore<sup>G</sup>;
  - **Redattori:** nomi dei componenti del gruppo che svolgono il ruolo di redattore<sup>G</sup>;
  - **Verificatori:** nomi dei componenti del gruppo che svolgono il ruolo di verificatore<sup>G</sup>;
  - **Uso:** specifica il tipo di utilizzo che viene fatto di questo documento;
  - **Distribuzione:** specifica a chi il documento verrà distribuito;
- **Descrizione:** una breve descrizione del documento posta sotto la tabella.



### 3.1.6.1 Registro delle modifiche

Il registro delle modifiche è una tabella che riporta ogni modifica effettuata al documento in questione. Una modifica è rappresentata da una riga della tabella avente le seguenti voci:

- **Versione:** versione attuale del documento;
- **Data:** data della modifica;
- **Nominativo:** il nome del redattore<sup>G</sup> della modifica;
- **Ruolo:** il ruolo che il redattore<sup>G</sup> ha all'interno del gruppo;
- **Verificatore:** il nome del componente che si è occupato di verificare la parte modificata;
- **Descrizione:** una breve descrizione sulla modifica effettuata.

### Indice

L'indice rappresenta le sezioni in cui sono divise le diverse parti del documento ed

### 3.1.7 Convenzioni

### 3.1.8 Elementi grafici

### 3.1.9 Strumenti

## 3.2 Gestione della Configurazione

### 3.2.1 Scopo

Lo scopo di questa sezione è definire come il gruppo ha deciso di attuare il processo di supporto di gestione della configurazione sui file prodotti. Quest'ultimo è risultato fin da subito fondamnetale

### 3.2.2 Aspettative

Le aspettative del gruppo *CodeBusters* nell'utilizzo di questo processo sono:

- avere un vantaggio nell'individuazione e risoluzione di possibili conflitti o errori;
- avere il tracciamento di ogni modifica;
- avere il ripristino, se necessario, a una versione precedente;
- avere la condivisione tra i membri del gruppo del materiale configurato.

### 3.2.3 Descrizione

Il processo di gestione della configurazione ha lo scopo di mantenere organizzata e tracciabile la documentazione redatta e il codice sviluppato, creando una storia per ogni file prodotto. Il particolare si vuole gestire la struttura e la disposizione delle varie parti di ogni file all'interno di un repository<sup>G</sup> facilmente accessibile e navigabile. Inoltre il processo si occupa anche di mantenere ordinati i repository<sup>G</sup> utilizzati, favorendo lo sviluppo di un senso dell'orientamento.

### 3.2.4 Versionamento

#### 3.2.4.1 Codice di versione

La storia di ogni documento avanza nel tempo attraverso una successione di modifiche, verifiche e approvazioni. Ciascuna di queste operazioni comporta il cambiamento della versione del documento stesso. Ogni versione è identificata tramite un codice numerico di tre cifre:

**X.Y.Z**

Singolarmente esse rappresentano:

**X** : versione stabile, ossia sottoposta ad approvazione del responsabile del documento;

**Y** : versione controllata, ossia sottoposta a verifica di un verificatore del documento;

**Z** : versione modificata, ossia sottoposta a modifica di un redattore del documento.

Tutte le cifre iniziano dal valore 0.

Ciascuna cifra aumenta di un'unità ogni volta che si compie un'operazione sul documento, inoltre:

**Se la cifra X è modificata** : le cifre Y e Z ritornano a 0 (per esempio da 0.2.6 a 1.0.0);

**Se la cifra Y è modificata** : la cifra Z ritorna a 0 (per esempio da 0.2.6 a 0.3.0).

### 3.2.4.2 Sistemi software utilizzati

Per gestire le versioni è stato deciso di utilizzare un version control system (VCS) distribuito<sup>G</sup> Git<sup>G</sup>. Le motivazioni di questa è scelta si racchiudono nei vantaggi di utilizzo rispetto a VCS centralizzati:

- possibilità di lavorare in locale senza il supporto del nodo centrale remoto;
- possibilità di creare diversi flussi di lavoro (branch<sup>G</sup>) in cui lavorare a documenti differenti;
- miglior gestione dei conflitti, a favore di una migliore collaborazione.

Per gestire i repository<sup>G</sup> Git<sup>G</sup> è stata scelto il servizio GitHub<sup>G</sup> per i seguenti motivi:

- integrazione di un issue tracking system(ITS)<sup>G</sup>;
- possibilità di utilizzarlo tramite browser, applicazione desktop, applicazione mobile o linea di comando;
- buona conoscenza di quest'ultimo da parte di tutti i membri del gruppo.

### 3.2.5 Struttura dei repository

#### 3.2.5.1 Repository utilizzati

Per favorire una migliore organizzazione e divisione del lavoro è stato deciso di creare due repository<sup>G</sup> pubblici distinti:

**CodeBusters-Docs** : per il versionamento dei documenti.

Riferimento: <https://github.com/Panz99/CodeBusters-Docs>

??? : per il versionamento del codice. Il suo utilizzo è successivo alla revisione dei requisiti.

Riferimento: ???

In questa prima versione delle *Norme di Progetto* si tratta la struttura del solo repository<sup>G</sup> CodeBusters-Docs.

#### 3.2.5.2 CodeBuster-Docs

L'organizzazione del lavoro collaborativo è così riassunta:

- un ramo principale *main* in cui è presente la sola documentazione pronta alla revisione;
- un ramo *develop* in cui effettuare periodicamente il merge dai vari rami minori;
- rami derivanti dal *develop* con nomi parlanti, ognuno dedicato alla stesura di uno specifico documento (l'idea fondante è quella del Git feature branch workflow<sup>G</sup>).

Nel *main* i file sono contenuti nella cartella RR (*Revisione dei Requisiti*), con l'obiettivo di crearne una nuova per ogni revisione futura.

Il file *.gitignore* è l'unico esterno a cartelle e dichiara esplicitamente l'estensione dei file automaticamente generati da L<sup>A</sup>T<sub>E</sub>X da non tracciare, poiché poco utili allo scopo del repository<sup>G</sup>.

Gli altri file sono invece organizzati in ulteriori cartelle dentro la RR:

**Doc\_esterna** : contiene l'*Analisi dei Requisiti* , il *Piano di Progetto* , il *Glossario* , i *verbali esterni* ;

**Doc\_interna** : contiene lo *Studio di Fattibilità* , le *Norme di Progetto* , i *verbali interni*;

**Utility** : contiene file di utilità generale come il template dei documenti, comandi L<sup>A</sup>T<sub>E</sub>X per velocizzare la redazione e il logo del gruppo.

### 3.2.5.3 Gestione dei cambiamenti in CodeBuster-Docs

La separazione del flusso di lavoro tra i vari documenti da redarre permette una notevole diminuzione dei conflitti e quindi delle modifiche da apportare per errore. Il punto focale è che il ramo *main* rimanga pulito da ogni tipo di errore, per cui non è utilizzabile da nessun membro del gruppo fino a che ciascun responsabile non abbia dato l'approvazione al corrispettivo documento. Solo in quel momento è permesso il merge del ramo *develop* nel *main*.

I cambiamenti da gestire sui documenti possono essere:

**Modifiche minori** : riguardano errori grammaticali, lessicali o di sintassi, che possono essere corretti dai redattori senza l'approvazione del responsabile;

**Modifiche generali** : riguardano cambiamenti più generali come la struttura del documento o convenzioni da utilizzare, e richiedono il consulto con il responsabile che potrà accettare o declinare la proposta di modifica.

### 3.3 Gestione della Qualità

#### 3.3.1 Scopo

Lo scopo di questa sezione è definire gli obiettivi che il gruppo si è posto nel processo di supporto della gestione della qualità.

#### 3.3.2 Aspettative

Le aspettative del gruppo *CodeBusters* nell'utilizzo di questo processo sono:

- avere un continuo accertamento sulla qualità del prodotto, in modo che sia conforme con quella richiesta dal proponente<sup>G</sup>;
- avere un'organizzazione dei documenti qualitativa, al fine di velocizzare possibili modifiche e manutenzioni future;
- avere un livello quantificabile della qualità dei processi attuati.

#### 3.3.3 Descrizione

Il documento *Piano di Qualifica* racchiude gli standard di qualità che il gruppo si è posto di mantenere, le misurazioni oggettive che descrivono gli stati di avanzamento. In generale per perseguire la qualità di un prodotto bisogna agire in modo sistematico<sup>G</sup>, fornendo per esempio un ruolo a ciascun componente del gruppo, gestendo risorse e procedure per ogni processo in atto, effettuando test statici e dinamici sul codice prodotto in modo non invasivo.

#### 3.3.4 Attività

Le attività del processo sono principalmente tre:

**QC (Quality Control)** : rappresenta tutti i controlli di qualità *product-oriented* che iniziano insieme all'inizio effettivo di produzione di documenti o codice;

**QA (Quality Assurance)** : rappresenta tutti gli accertamenti preventivi *process-oriented* che valutano il way of working<sup>G</sup> adottato, senza rallentare i processi di sviluppo;

**QP (Quality Planning)** : rappresenta una visione generale sugli obiettivi di qualità e sulle risorse necessarie a conseguirli. Nel nostro caso questo corrisponde alla redazione del *Piano di Qualifica* .

Nello specifico ogni membro del gruppo deve:

- comprendere fin da subito l'obiettivo del file che sta scrivendo, che sia documentale o software;
- porsi obiettivi incrementali, per ridurre la possibilità d'errore e perseguire correttezza;

- utilizzare conoscenze personali o di altri componenti del gruppo, creando un avanzamento continuo della propria formazione;
- rispettare gli standard di qualità del *Piano di Qualifica* .

### **3.3.5 Strumenti utili**

Per le attività del processo di gestione della qualità gli strumenti sono stati scelti in base....

## **3.4 Verifica**

## **3.5 Validazione**



## 4 Processi Organizzativi

### 4.1 Gestione Organizzativa

#### 4.1.1 Scopo

In questa sezione vengono espone le modalità di coordinamento adottate dal gruppo che regolano gli incontri(interni o esterni) e le comunicazioni.

- **comunicazione:** interna con i membri del gruppo, quella esterna con l'azienda;
- **incontri:** incontri interni con in membri del gruppo, esterno con l'azienda.

#### 4.1.2 Aspettative

Le attese, riguardo al processo in questione sono i seguenti:

- ottenere una pianificazione ragionevole delle attività da seguire;
- coordinamento dell'attività del gruppo, assegnando loro i ruoli, i compiti e semplificando la comunicazione tra loro;
- adoperare processi per regolare le attività e renderle economiche;

#### 4.1.3 Descrizione

Le attività di gestione sono:

- assegnazione dei ruoli e dei compiti;
- inizio e definizione dello scopo;
- istanziazione dei processi;
- pianificazione e stima di tempi, risorse e costi;
- esecuzione e controllo;
- revisione e valutazione periodica delle attività.

#### 4.1.4 Ruoli di Progetto

Ogni membro del gruppo deve, a turno ricoprire almeno una volta ciascun ruolo di progetto che corrisponde alle figure aziendali. I ruoli che ogni membro del gruppo è tenuto a rappresentare sono descritti di seguito.

#### **4.1.5 Responsabile di progetto**

Il responsabile di progetto è un ruolo fondamentale, in quanto si occupa delle comunicazioni con il proponente e committente. Inoltre, egli deve svolgere i seguenti compiti:

- pianificare;
- gestire;
- controllare;
- coordinare,

#### **4.1.6 Amministratore di progetto**

L'amministratore deve avere il controllo dell'ambiente di lavoro ed essere di supporto,. Inoltre Egli deve:

- dirigere le infrastrutture di supporto;
- controllare versioni e configurazioni;
- risolvere i problemi che riguardano la gestione dei processi;
- gestire la documentazione;

#### **4.1.7 Analista**

L'analista si occupa dell'analisi dei problemi e del dominio applicativo. Questa figura ha le seguenti responsabilità:

- studio del dominio del problema;
- redazione della documentazione: Analisi dei Requisiti e Studio di Fattibilità;
- definizione dei requisiti e la sua complessità.

#### **4.1.8 Progettista**

Il progettista si occupa dell'aspetto tecnico e tecnologico del progetto, segue lo sviluppo e non la manutenzione del prodotto. Inoltre egli deve scegliere:

- un'architettura adatta per il sistema del prodotto in base alle tecnologie scelte;
- il modo più efficiente per ottimizzare l'aspetto tecnico del progetto.

#### **4.1.9 Programmatore**

#### **4.1.10 Verificatore**

#### **4.1.11 Procedure**