SpaghettiCode

spaghetti.code.g6@gmail.com

Norme di Progetto

Versione	v1.0.0
Approvazione	Paparazzo Giorgia
Redazione	Kostadinov Samuel Masevski Martin
Verifica	Kostadinov Samuel Masevski Martin
Uso	Interno
Destinato a	prof. Vardanega Tullio prof. Cardin Riccardo SpaghettiCode

Descrizione

Documento contenente tutte le convenzioni, regole e strumenti adottati che il gruppo *SpaghettiCode* ha deciso di adottare per lo sviluppo del capitolato *HD Viz*.

Registro delle modifiche

Versione	Nominativo	Ruolo	Data	Descrizione
v1.0.0	Paparazzo Giorgia	Responsabile	2021-01-05	Verifica e approvazione documento
v0.3.0	Kostadinov Samuel	Verificatore	2021-01-05	Verifica capitolo §4
v0.2.0	Masevski Martin	Verificatore	2021-01-04	Verifica capitoli §2 e §3
v0.1.2	Kostadinov Samuel	Amministratore	2021-01-03	Fine stesura capitolo §3
v0.1.1	Kostadinov Samuel	Amministratore	2021-01-01	Stesura metriche capitolo §2
v0.1.0	Masevski Martin	Verificatore	2021-01-01	Verifica §1
v0.0.7	Masevski Martin	Amministratore	2020-12-31	Fine stesura §4
v0.0.6	Kostadinov Samuel	Amministratore	2020-12-30	Inizio stesura capitolo §3
v0.0.5	Masevski Martin	Amministratore	2020-12-29	Stesura capitolo §4
v0.0.4	Kostadinov Samuel	Amministratore	2020-12-28	Fine stesura §2
v0.0.3	Kostadinov Samuel	Amministratore	2020-12-27	Continuazione stesura §2
v0.0.2	Kostadinov Samuel	Amministratore	2020-12-26	Stesura §1 e inizio stesura §2
v0.0.1	Pagotto Manuel	Analista	2020-12-17	Creazione del documento

Indice

Elenco delle figure

Elenco delle tabelle

1 Introduzione

1.1 Scopo del documento

Questo documento ha lo scopo di fissare il way of working$_G$ che tutti i membri del gruppo *SpaghettiCode* dovranno seguire per tutto lo svolgimento del progetto$_G$. Le varie attività$_G$ descritte all'interno del documento andranno a formare processi$_G$ che prendono ispirazione dallo standard *ISO/IEC 12207:1995*. Il documento segue uno sviluppo di tipo incrementale, perciò sarà sottoposto continuamente a modifiche, rimozioni o aggiunte; al completamento di una qualsiasi di queste azioni, tutti i membri del gruppo dovranno essere notificati. Il documento è, infatti, ancora incompleto e verrà aggiornato riguardo alle regole specifiche per ogni processo del progetto$_G$.

1.2 Scopo del prodotto

Il capitolato$_G$ C4 - "HD Viz" si pone come obbiettivo la realizzazione di una web application$_G$ che permetta di visualizzare dati con molte dimensioni in grafici che devono supportare l'utente nella fase di Exploratory Data Analysis$_G$ (EDA). I dati che saranno visualizzati dovranno essere caricati nell'applicazione web tramite file CSV o prelevati da database$_G$ interni o esterni.

1.3 Glossario

Al fine di eliminare qualsiasi equivocità nei termini presenti all'interno del documento e quindi dell'insorgere di incomprensioni, viene fornito il Glossario v1.0.0, documento nel quale vengono definiti i termini che presentano una "G" come pedice.

1.4 Riferimenti

1.4.1 Riferimenti normativi

- **Capitolato d'appalto C4 - "HD Viz"**:
  https://www.math.unipd.it/~tullio/IS-1/2020/Progetto/C4.pdf;
- **Ulteriori informazioni sul capitolato C4**:
  https://www.dropbox.com/s/nslvtrq2wcycoqw/HD%20Viz.mp4?dl=0.

1.4.2 Riferimenti informativi

- **Standard ISO/IEC 12207:1995**:
  https://www.math.unipd.it/~tullio/IS-1/2009/Approfondimenti/ISO_12207-1995.pdf;
- **Documentazione Git$_G$**:
  https://git-scm.com/docs;
- **Documentazione GitHub**:
  https://docs.github.com/en/free-pro-team@latest;
- **Documentazione LaTeX**:
  https://www.latex-project.org/help/documentation/;
- **Documentazione JavaScript**:
  https://developer.mozilla.org/en-US/docs/Web/JavaScript;
- **Documentazione libreria 3D.js**:
  https://github.com/d3/d3/wiki.

2 Processi primari

2.1 Fornitura

2.1.1 Scopo

Il processo di fornitura consiste nelle seguenti attività:

- Analisi di strumenti e competenze fondamentali e individuazione di rischi e criticità$_G$ per il completamento del progetto. Questa analisi viene redatta nel documento Studio di Fattibilità;
- Stabilire ed esporre l'organizzazione del lavoro che il gruppo seguirà per la realizzazione del prodotto; queste possono essere trovate nel Piano di Progetto$_G$. Il Piano di Progetto sarà, come questo documento, ancora non del tutto completo in quanto non è facile pianificare a lungo termine, perciò quella che viene presentata al momento è una versione che verrà poi aggiornata successivamente in base ad eventuali anticipi e/o ritardi sulle scadenze presenti al momento;
- Verificare la qualità del materiale elaborato, sia per quanto riguarda i documenti, sia per quanto riguarderà, più avanti nel tempo, le attività di progettazione e verifica. Le linee guida per la gestione della qualità sono descritte nel documento Piano di Qualifica.

Questo processo è composto dalle seguenti fasi:

- Avvio;
- Preparazione di risposte alle richieste;
- Contrattazione;
- Pianificazione;
- Esecuzione e controllo;
- Revisione e valutazione;
- Consegna e completamento.

2.1.2 Descrizione

Vengono qui descritte e trattare tutte norme a cui il gruppo *SpaghettiCode* deve attenersi, con lo scopo di diventare i fornitori del prodotto *HD Viz* del proponente *Zucchetti S.p.A.* e dei committenti *prof. Tullio Vardanega* e *prof. Riccardo Cardin*.

2.1.3 Aspettative

Questo processo si pone gli obbiettivi di mantenere un confronto costante con il proponente$_G$ *Zucchetti S.p.A.* e nello specifico con il referente *dott. Gregorio Piccoli* al fine di stimare le tempistiche di lavoro, verificare in modo continuo quanto prodotto dal gruppo, determinare i requisiti del prodotto e infine chiarire eventuali dubbi. Successivamente all'avvenuta consegna, il gruppo *SpaghettiCode*, non seguirà la fase di manutenzione del prodotto, salvo eventuali accordi.

2.1.4 Attività

2.1.4.1 Studio di Fattibilità

Nel documento Studio di Fattibilità$_G$, redatto dagli *analisti* dopo aver deciso la prima scelta del gruppo tra i vari progetti disponibili, viene fornita un'analisi generale di tutti i capitolati proposti, e le motivazioni che hanno spinto il gruppo *SpaghettiCode* a proporsi o meno come fornitore$_G$ di uno specifico capitolato. Questo documento è il prodotto dell'omonima attività, la quale è strutturata come segue:

- **Raccolta di informazioni generali**: raccolta delle informazioni basilari riguardanti il capitolato che comprendono nome, proponente e committente;
- **Comprensione delle caratteristiche**: si studiano e comprendono le caratteristiche del prodotto che deve essere sviluppato;
- **Comprensione dello scopo del progetto**: si studiano e comprendono i possibili fini del progetto. Questi possono essere molteplici;
- **Comprensione delle tecnologie interessate**: si studia e si determina quali sono le tecnologie interessate dal capitolato, se imposte o suggerite dal proponente;
- **Valutazione degli aspetti positivi**: vengono individuati gli aspetti positivi di ogni capitolato;
- **Valutazione dei rischi**: vengono individuati tutti i possibili rischi del capitolato proposto;
- **Conclusioni**: Vengono messi sui piatti della bilancia gli aspetti positivi e gli aspetti negativi, traendo quindi le conclusioni sulla sua fattibilità.

Questo si riflette poi in un documento con la seguente struttura:

- **Informazioni generali**
- **Descrizione**
- **Finalità del progetto**
- **Tecnologie interessate**
- **Aspetti positivi**
- **Rischi**
- **Conclusioni**

2.1.4.2 Piano di Progetto

Il gruppo deve quindi creare un PIANO DI PROGETTO, ovvero pianificare come si svolgerà il progetto e considerare le tempistiche con cui ogni altra attività verrà eseguita. Vista la natura del piano di progetto, questo sarà in continua evoluzione, visto che è sempre possibile sbagliare le stime, soprattutto sui tempi del progetto e sul lungo termine. Quest'attività si formalizza nell'omonimo documento, redatto dagli *amministratori*, sotto la supervisione del *responsabile di progetto*$_G$. Vista il continuo aggiornamento del piano, anche il relativo documento deve essere redatto e aggiornato per tutta la durata del progetto. La sua struttura è la seguente:

- **Analisi dei rischi**: sezione in cui vengono analizzati i rischi che possono presentarsi nel corso del progetto. Vengono fornite anche le modalità con cui vengono risolti o ridimensionati questi rischi. Un'analisi più esaustiva si troverà nel documento Analisi dei requisiti$_G$;
- **Modello di sviluppo$_G$**: sezione in cui viene descritto il modello di sviluppo$_G$ scelto dal gruppo e le motivazioni che hanno portato a scegliere quel determinato modello;
- **Pianificazione**: vengono descritte e pianificate le attività da eseguire nelle vari fasi del progetto, stabilendo i termini temporali (deadline$_G$) per il loro completamento. Queste deadline non sono però rigide perché, come accennato prima, non è facile pianificare sul lungo termine e proprio per questo motivo il piano di progetto sarà sempre soggetto a modifiche e aggiornamenti;
- **Preventivo e consuntivo**: viene stimata la quantità di lavoro necessaria per ogni processo del ciclo di vita$_G$ del progetto. Viene quindi esposto un preventivo$_G$ e un successivo consuntivo$_G$, entrambi relativi ad un dato periodo.

2.1.4.3 Piano di Qualifica

Il Piano di Qualifica$_G$ raccoglie tutte le regole e le linee guida per garantire che i materiali prodotti siano di qualità. Questo insieme di regole deve essere, come per il piano di progetto, formalizzato in un omonimo documento redatto dai *verificatori*. Questo documento è strutturato nel seguente modo:

- **Qualità di processo**: sono individuati i processi dagli standard di processo$_G$, definiti degli obbiettivi, strategie per attuarli e metriche$_G$ per controllarli e misurarli;
- **Qualità di prodotto**: vengono individuate le caratteristiche più importati del prodotto, gli obbiettivi necessari per raggiungerle e le metriche per misurarle;
- **Specifiche dei test**: vengono definiti dei test$_G$ attraverso cui il prodotto deve passare per garantire il soddisfacimento dei requisiti;
- **Standard di qualità**: descritti gli standard di qualità$_G$ selezionati;
- **Resoconto delle attività di verifica**: vengono esposti i risultati dei test eseguiti durante il periodo di revisione; le metriche usate per l'ottenimento di questi risultati sono redatte nel documento;
- **Lista di controllo**: Lista che contiene gli errori riscontrati. Questa sezione è in continua fase di aggiornamento per l'intera durata del progetto.
- **Valutazioni per il miglioramento**: vengono elencati i problemi riscontrati durante lo sviluppo del progetto e vengono proposte delle soluzioni che potrebbero portare alla risoluzione o alla mitigazione dei problemi individuati.

In questo documento si fa uso di un codice identificativo dei rischi. Il codice avrà questa forma:

[Tipologia][Codice]

Dove:

- **Tipologia**: indica la tipologia di rischio e può assumere 3 valori:
  - **RR**: Indica un rischio legato ai requisiti;
  - **RT**: Indica un rischio legato alle tecnologie;
  - **RO**: Indica un rischio legato all'organizzazione.
- **Codice**: Il codice è un numero progressivo univoco all'interno della tipologia. Questo permette di identificare univocamente il rischio.

Ogni rischio avrà, oltre al codice, due informazioni:

- **Occorrenza**: La frequenza con cui occorre il rischio;
- **Gravità**: Indica la gravità del rischio.

2.1.5 Strumenti

Saranno riportati di seguito gli strumenti che il gruppo ha deciso di utilizzare durante il progetto.

2.1.5.1 GitHub

Il gruppo ha deciso, di comune accordo, di affidarsi a GitHub come strumento di condivisione dei file e versionamento$_G$, sia per quanto riguarda il codice, sia per quanto riguarda i documenti. La scelta è ricaduta su questo strumento per la sua integrazione con un sistema di issue tracking, che viene sfruttato, la possibilità di lavorare in più branch e anche per la sua grande diffusione.

2.1.5.2 Google Docs/Google Drive

Questo strumento viene utilizzato dal gruppo per la sua estrema semplicità, che permette una veloce creazione e stesura di bozze di documenti che verranno poi formalizzati. Nelle cartelle di Google Drive si troveranno quindi dei fogli di documenti di Google, ovvero documenti informali generalmente bozze di documenti poi formalizzati o scalette dei documenti che devono essere scritti.

2.1.5.3 LaTeX

Il gruppo ha di comune accordo deciso, dopo aver valutato l'opzione, di usare lo strumento LaTeXper la scrittura formale dei documenti, perciò ha predisposto su GitHub un repository$_G$ in cui mantenere versionati i file .tex in modo da avere uno storico dei documenti e gestire in maniera più semplice le modifiche.

2.1.5.4 Discord

Usato dal gruppo come canale di comunicazione principale, per organizzare gli incontri, per condividere risorse tra i membri e per suddividere in vari canali, corrispondenti ai diversi ruoli, così che ci sia una maggiore organizzazione e divisione dei ruoli stessi.

2.1.5.5 Telegram

Si è deciso di creare un gruppo Telegram$_G$ per le comunicazioni di minore importanza e per avere un canale di comunicazione alternativo in quanto Discord non è un'applicazione che tutti hanno sempre sotto mano.

2.1.5.6 Trello e GitHub Projects

Il gruppo aveva inizialmente deciso di provvedere all'organizzazione tramite Trello, tuttavia successivamente è stato preferito il sistema dei Projects di GitHub. Entrambe sono delle applicazioni web per la creazione di list in stile Kanban.

2.2 Sviluppo

2.2.1 Scopo

Seguendo quanto specificato nello standard *ISO/IEC 12207:1995*, lo sviluppo consiste nella descrizione le varie attività di analisi, progettazione, codifica, integrazione, test, installazione e accettazione.

2.2.2 Descrizione

Questo processo è formato da tre attività principali:

- **Analisi dei requisiti**
- **Progettazione architettura**
- **Codifica del software**

Ognuna di queste verrà descritta meglio nella sezione dedicata.

2.2.3 Aspettative

Le aspettative dello sviluppo sono principalmente le seguenti:

- Stabilire gli obbiettivi del prodotto;
- Stabilire i requisiti tecnologici;
- Stabilire i vincoli di design;
- Realizzare un prodotto che soddisfi le richieste del proponente.

2.2.4 Analisi dei requisiti

2.2.4.1 Scopo

Lo scopo dell'analisi dei requisiti è di redarre un documento che raccolga tutti requisiti che il proponente richiede per il progetto.
I requisiti hanno le seguenti finalità:

- Descrivere lo scopo del lavoro;
- Fornire le indicazioni necessarie ai *progettisti*;
- Fissare le funzionalità concordate con il cliente;
- Fornire una base per un miglioramento continuo;
- Dare ai verificatori un modo per misurare le attività di controllo;
- Dare dei riferimenti per poter fare una stima del lavoro necessario.

Tale documento viene redatto dagli *analisti*.

2.2.4.2 Descrizione

I requisiti, parte fondamentale di questo documento, si possono ricavare da varie fonti:

- **Capitolato d'appalto**: la prima descrizione del prodotto messa a disposizione dal proponente. Da qui è possibile estrarre alcuni dei requisiti del progetto;
- **Incontri interni**: è possibile che emergano requisiti durante una discussione ad uno degli incontri interni;
- **Incontri esterni**: è possibile che emergano requisiti durante una discussione ad uno degli incontri esterni con il proponente;
- **Casi d'uso**: infine è possibile che il requisito emerga durante la stesura dei casi d'uso perché si presenta una necessità riguardante quello specifico caso d'uso.

2.2.4.3 Aspettative

Lo scopo dell'analisi dei requisiti è scrivere un documento che raccolga tutti i requisiti individuati dagli *analisti*.

2.2.4.4 Struttura del documento

La struttura del documento è contraddistinta dalle seguenti parti:

- **Introduzione**: contiene informazioni riguardanti lo scopo del documento, lo scopo del progetto, il glossario, i riferimenti (normativi e informativi);
- **Descrizione generale**: contiene una descrizione del documento di analisi dei requisiti, che riporta informazioni a proposito del prodotto ed in particolare i suoi obbiettivi e le sue funzioni, sugli utenti, nello specifico le loro caratteristiche,, le architetture e le tecnologie e sui vincoli generali;
- **Casi d'uso**: sezione in cui è possibile trovare la struttura e gli attori primari e secondari dei vari casi d'uso, seguita da una lista dei casi d'uso individuati;
- **Requisiti**: in questa sezione del documento è possibile trovare una lista dei requisiti divisi tra funzionali, di qualità e vincoli.

2.2.4.5 Classificazione dei requisiti

La rappresentazione dei requisiti avviene secondo un codice, non variabile, concordato internamente che si presenta nella seguente forma:

R[Tipologia][Importanza][Codice]

Questa notazione, spiegata nel dettaglio di seguito, permette di identificare in modo univoco ogni requisito del capitolato.

- **Tipologia**: i requisiti possono avere diversi tipi, questa parte del codice identificativo permette di identificare, a colpo d'occhio, la tipologia del requisito. I possibili valori sono:
  - **V**: rappresenta il fatto che il requisito è un vincolo imposto dal proponente sui servizi offerti dal prodotto software all'utilizzatore finale;
  - **F**: rappresenta il fatto che il requisito è di tipo funzionale, quindi viene usato per rappresentare le funzioni del prodotto;
  - **P**: rappresenta il fatto che il requisito è di tipo prestazionale, quindi viene usato per dare delle limitazioni sulle prestazioni del prodotto software;
  - **Q**: rappresenta il fatto che il requisito è di qualità, quindi un vincolo sulla qualità del prodotto.
- **Importanza**: i requisiti hanno diversa importanza all'interno del progetto. Questa parte del codice fa si che sia possibile vedere subito quale sia l'importanza di ogni requisito. I possibili valori sono:
  - **O**: indica che il requisito è obbligatorio, quindi questo requisito dovrà essere necessariamente soddisfatto;
  - **D**: indica che il requisito è desiderabile, quindi eventualmente negoziabile con il proponente. Anche se non viene vincolata la loro presenza il proponente vorrebbe che questi venissero implementati in quanto fornirebbero al prodotto una maggiore completezza;
  - **F**: indica che il requisito è facoltativo. I requisiti classificati come tali, anche se portano un valore aggiunto al prodotto finito, molto probabilmente richiedono molto tempo e lavoro al fine di una piccola miglioria.
- **Codice**: il codice identificativo vero e proprio. Questo codice si trova nella forma:

  [CodiceBase](.[CodiceSottocaso])*

  Il codice così formato permette di riferire uno specifico caso d'uso (la cui identificazione è spiegata nel paragrafo successivo).

Oltre al codice ogni requisito avrà associate una serie di informazioni, quali:

- **Descrizione**: una breve descrizione del requisito
- **Fonte**: la fonte da cui è stato estrapolato (capitolato d'appalto, use case, verbali interni o verbali esterni)

2.2.4.6 Classificazione dei casi d'uso

Analogamente ai requisiti, anche i casi d'uso hanno un codice immutabile che li identifica univocamente all'interno del progetto. Tale codice ha la forma:

UC[CodiceBase](.[CodiceSottoCaso])*

Il codice base permette di identificare il caso d'uso generale, mentre il codice del sotto caso fa riferimento ad eventuali sottocasi del caso generale.
Ogni caso d'uso ha, inoltre, una struttura ben definita, riportata a seguito:

- **Descrizione**: breve descrizione del caso d'uso;

- **Attore primario**: Entità che interagisce direttamente con il prodotto;
- **Eventuali attori secondari**: Entità che aiutano l'attore primario a portare a raggiungere il suo scopo. Non è necessariamente presente;
- **Precondizione**: Condizione in cui è il sistema prima che si verifichi degli eventi previsti dal caso d'uso in esame;
- **Input**: Ciò che l'attore porta all'interno del sistema. Non necessariamente presente, serve a specificare con precisione cosa l'attore deve introdurre nel sistema;
- **Postcondizione**: Condizione in cui è il sistema dopo che si sono verificati gli eventi previsti dal caso d'uso in esame;
- **Output**: Ciò che il prodotto darà come risultato alla fine del flusso di eventi dello use case in esame. Non è necessariamente presente;
- **Scenario principale**: Rappresentazione del flusso degli eventi previsti dal caso d'uso;
- **Scenari alternativi**: Rappresentazioni alternative del flusso degli eventi. Non necessariamente presenti;
- **Estensioni**: Parametri opzionali che servono a modellare gli scenari alternativi;
- **Inclusioni**: Utilizzati quando ci sono più casi d'uso collegati. Non sono necessariamente presenti;
- **Generalizzazioni**: Rappresentano delle possibili specializzazioni del caso d'uso. Non necessariamente presenti.

2.2.4.7 Tracciamento dei requisiti e dei casi d'uso

Per tenere traccia dei casi d'uso e dei requisiti il gruppo ha deciso di usare come strumento l'issue tracking system$_G$ di GitHub, strumento che tramite il workflow$_G$ delle issue$_G$ stesse, diviso tra "To do", "In progress" e "Done", allo stesso tempo tiene sempre aggiornati i membri del gruppo e spinge i componenti del gruppo stesso ad aggiornarne lo stato per far si che il resto del gruppo sia informato sullo stato di avanzamento di tutti i requisiti e dei casi d'uso.

2.2.4.8 Metriche

Il gruppo ha deciso di tenere traccia del completamento del progetto in base ai requisiti. A questo scopo è stato deciso di calcolare una percentuale di completamento del progetto come il rapporto tra il numero di requisiti implementati e dei requisiti totali moltiplicato per 100. Questo da un metodo molto immediato per capire lo stato di avanzamento del progetto ed è facile da calcolare. Il valore di questo indice varia tra 0 e 100, dove 100 indica che il progetto è completato, mentre 0 indica che deve ancora essere iniziato.
NOTA: Il gruppo ha deciso di considerare, inizialmente, il numero totale di requisiti uguale al numero totale di requisiti obbligatori e si riserva la possibilità di aumentare il numero totale di requisiti in seguito ad eventuali negoziazioni con il proponente.

2.2.5 Progettazione

2.2.5.1 Scopo

Lo scopo di quest'attività è quello di individuare le caratteristiche che il prodotto deve avere per soddisfare nel modo migliore possibile le caratteristiche del proponente in risposta ai requisiti individuati dall'analisi dei requisiti. In quest'attività bisogna:

- Garantire la qualità del prodotto seguendo un principio di correttezza costruttivo
- Organizzare, suddividere i compiti in modo da diminuire la complessità del problema e riducendolo via via in sottoproblemi sempre più elementari fino ad arrivare ai singoli componenti
- Ottimizzare l'uso di risorse

2.2.5.2 Descrizione

La progettazione è divisa in due parti fondamentali:

- Technology baseline$_G$: Contiene le specifiche ad alto livello della progettazione del software, i relativi diagrammi UML$_G$ e dei test;
- Product baseline$_G$: Arricchisce di dettagli quanto specificato nella Technology baseline e definisce i test necessari.

2.2.5.3 Aspettative

La progettazione è un'attività svolta dai *Progettisti*, volta a produrre l'architettura logica del prodotto. L'architettura deve essere formata da componenti chiari, riusabili e utilizzabili in modo che ci sia coesione tra le parti. Inoltre è necessario rimanere entro i costi fissati.
L'architettura dovrà necessariamente:

- Soddisfare i requisiti individuati dall'analisi dei requisiti;
- Adattarsi in caso i requisiti evolvano;
- Deve riuscire a gestire situazioni erronee
- Risultare affidabile anche in situazioni sfavorevoli come temporanee mancanze
- Garantire un certo livello di sicurezza rispetto ai malfunzionamenti
- Presentare solo il minimo intervallo possibile di indisponibilità durante i periodi di manutenzione
- Impiegare efficientemente le risorse
- Garantire la riusabilità delle sue parti anche in altri applicativi
- Presentare componenti semplici e con basso livello di accoppiamento

2.2.5.4 Design pattern

La scelta dei design pattern$_G$ da utilizzare è lasciata ai *progettisti*, i quali dovranno assicurarsi che le loro scelte portino a una soluzione che sia flessibile e lasci una certa libertà ai *Programmatori*. Ogni design pattern utilizzato andrà spiegato e rappresentato in modo da poterne esporre significato e struttura.

2.2.5.5 Diagrammi UML

Il gruppo ha scelto di utilizzare, allo scopo di rendere più chiare le scelte compiute in ambito di progettazione, dei diagrammi UML. Tra questi spiccano i diagrammi delle attività e quelli di sequenza. I primi vengono usati per descrivere il flusso di operazioni di un'attività, i secondi per illustrare sequenze di azioni.
Ci potranno essere, oltre ai due tipi già menzionati, diagrammi di altro tipo se i *progettisti* lo riterranno utile.

2.2.5.6 Test

Come specificato all'inizio, la definizione dei test è parte dell'attività di progettazione, quindi ogni *progettista* dovrà definire i test necessari. Le regole di nomenclatura da seguire sono le stesse valide per la nomenclatura dei metodi, descritte nella sezione "Stile di codifica" della parte relativa alla codifica.

2.2.5.7 Nota

Il gruppo si riserva di modificare e in particolar modo di ampliare, in caso fosse necessario, la progettazione nelle fasi successive alla sua definizione.

2.2.6 Codifica

2.2.6.1 Scopo

Quest'attività, svolta dai *programmatori*, ha lo scopo di scrivere del codice che traduca l'architettura pensata dai *progettisti*. Quest'attività è soggetta alle regole descritte in seguito (al paragrafo "Stile di codifica") per far si che il codice sia più leggibile possibile.

2.2.6.2 Descrizione

La scrittura del codice dovrà tradurre l'architettura pensata dai *progettisti* mantenendo lo standard qualitativo richiesto e descritto nel *piano di qualifica*.

2.2.6.3 Aspettative

L'obbiettivo dell'attività di codifica è creare un prodotto software che permetta di soddisfare le richieste del proponente e che al contempo mantenga un certo livello di qualità, al fine di:

- Garantire la leggibilità del codice;
- Agevolare manutenzione, verifica e validazione;

2.2.6.4 Stile di codifica

Al fine di garantire uniformità nel codice prodotto, si è deciso di stabilire delle regole nella scrittura del codice:

- Indentazioni: i blocchi di codice innestati, ad esclusione dei commenti, devono presentare 4 spazi di rientro rispetto al livello precedente;
- Parentesi: inserire le parentesi sulla stessa riga del costrutto che le usa;
- Struttura dei metodi: la struttura dei metodi deve sempre avere alcune caratteristiche:
  - I nomi dei metodi devono rispettare la convenzione snake_case;
  - I metodi devono essere i più brevi possibili e non possono essere contenute più istruzioni nella stessa riga;
  - Deve essere presente una spaziatura tra la parentesi tonda che contiene i parametri dei metodi e la parentesi graffa di apertura deve essere inserita una spaziatura.
- Univocità dei nomi: le variabili e i metodi devono avere un nome univoco e rappresentativo che permetta di identificarli univocamente e che eviti ambiguità;
- Costanti: i nomi delle costanti devono essere scritte in stampatello maiuscolo;
- Lingua: la lingua utilizzata per i nomi delle variabili e dei metodi devono essere scritti in inglese.

Il gruppo si riserva la possibilità di cambiare, prima di iniziare l'attività di codifica, le norme qui specificate in caso se ne sentisse la necessità.

2.2.6.5 Metriche

Il gruppo ritiene prematuro stabilire una metrica definitiva per misurare la leggibilità del codice, perciò ha definito un indice indicativo che verrà integrato nei successivi stadi del progetto.
La formula momentaneamente adottata è:

$$Leggibilità = \frac{numero\ di\ linee\ di\ commento}{numero\ di\ linee\ di\ codice}$$

2.2.6.6 Strumenti

Sono riportati di seguito gli strumenti utilizzati nel processo di sviluppo.

2.2.6.6.1 HTML, CSS

HTML e CSS sono due linguaggi usati nello sviluppo di pagine web. Vista la natura di web application di HD Viz, saranno necessari, anche se non come parte principale.

2.2.6.6.2 Database

Il proponente vuole lasciare aperta la possibilità di caricare dati da un database, senza tuttavia fornire dei vincoli a tale proposito. Sarà perciò compito dei *progettisti* scegliere il database più adatto al progetto.

2.2.6.6.3 JavaScript

Linguaggio richiesto dal proponente, è un linguaggio di scripting per web application. Viene richiesto questo linguaggio per la possibilità di utilizzare D3.js.

2.2.6.6.4 D3.js

Libreria open source scritta in JavaScript con lo scopo di facilitare la visualizzazione dei dati in grafici. Richiesta dal proponente, è lo strumento principale per la realizzazione del prodotto HD Viz.

3 Processi di supporto

3.1 Documentazione

3.1.1 Scopo

Lo scopo della documentazione è quello di scrivere in uno o più documenti tutto ciò che è necessario sapere per utilizzare correttamente il prodotto software creato alla fine del progetto.

3.1.2 Descrizione

La produzione della documentazione è un processo che si svolge durante tutto il corso temporale del progetto. Consiste nello scrivere dei documenti di varia natura (riportati nella sezione "Documenti prodotti"), ognuno dei quali ha una specifica funzione. In questa sezione del documento sono riportate le norme per una buona scrittura della documentazione.

3.1.3 Aspettative

Questo processo ha come aspettativa principale la stesura di documenti, funzionali al loro scopo, scritti in maniera chiara e che rispettino le regole di seguito definite.

3.1.4 Ciclo di vita di un documento

Il ciclo di vita deve seguire passi:

- **Creazione del documento**: il documento viene creato a partire da un template, viene già fornito del registro delle modifiche e degli elenchi di figure e tabelle;
- **Stesura**: il documento viene scritto da più membri del gruppo, riportando i vari incrementi nel registro delle modifiche;
- **Revisione**: dopo la stesura il documento viene revisionato da una o più persone che valutano se il documento rispetta o meno le norme decise. È obbligatorio che un redattore non verifichi il suo stesso documento, nel caso in cui il documento sia stato scritto da più persone un redattore può essere anche verificatore a patto che non verifichi una sezione da lui scritta;
- **Approvazione**: dopo la revisione il *responsabile* deve approvare il documento. Da quel momento il documento è considerato completo.

3.1.5 Template

Il gruppo ha deciso di utilizzare, per il progetto, il linguaggio LaTeXper la stesura dei documenti. Allo scopo di uniformare i documenti e velocizzare il processo di creazione e strutturazione del documento stesso è stato definito un template su cui ogni documento si basa.

3.1.6 Documenti prodotti

I documenti prodotti sono diversi e si dividono in due categorie:

- **Informali**: In questa categoria rientrano i documenti non soggetti a versionamento. In particolare, fanno parte di questa categoria i verbali. Questi si dividono in:
  - **Interni**: A questa categoria appartengono i verbali degli incontri interni del gruppo. Riportano in breve dei resoconti degli incontri;
  - **Esterni**: Questa, al contrario della precedente, è la categoria a cui appartengono i verbali degli incontri con membri esterni al gruppo, quali potrebbero essere i committenti e/o i proponenti.

- **Formali**: In questa categoria rientrano tutti i documenti soggetti a versionamento e regolano l'intera attività del gruppo. Questi documenti devono essere approvati dal *responsabile* di progetto. In alcuni casi è possibile che un documento abbia più versioni, in questo caso il *responsabile* deve dare l'approvazione ad ognuna e si considera quella corrente la più recente tra le versioni approvate.
  - **Interni**: I documenti appartenenti a questa categoria sono destinati ad uso interno del gruppo e non sono quindi di interesse primario per i committenti e il proponente;
  - **Esterni**: I documenti appartenenti a questa categoria sono, al contrario, destinati ad essere consegnati al proponente e ai committenti. Verrà consegnata l'ultima versione approvata.

  Di seguito viene riportato un elenco dei documenti formali prodotti:
  - **Norme di progetto**: documento che racchiude tutte le norme che regolamentano il progetto nella sua interezza. Questo documento rientra nella categoria dei documenti interni;
  - **Glossario**: documento che elenca tutti i termini che, secondo il gruppo, necessitano di una spiegazione esplicita. Questo documento rientra nella categoria dei documenti esterni;
  - **Studio di fattibilità**: documento che raccoglie i singoli studi di fattibilità sui capitolati secondo lo schema descritto in questo documento al paragrafo "Struttura dei documenti". Questo documento rientra nella categoria dei documenti interni;
  - **Piano di progetto**: documento che espone la pianificazione delle attività di progetto previste dal gruppo. In allegato a questo documento, che ricade nella categoria di quelli esterni, viene presentato un preventivo sia degli impegni orari che dei costi;
  - **Piano di qualifica**: documento che espone i criteri di valutazione della qualità del progetto. Questo documento rientra nella categoria dei documenti esterni;
  - **Analisi dei requisiti**: documento che raccoglie tutti i requisiti del progetto e che il prodotto software deve avere. Questo documento rientra nella categoria dei documenti esterni.

3.1.7 Cartella dei documenti

Ogni cartella contenente un documento prende il nome dal documento stesso e utilizza la convenzione snake_case. All'interno della cartella si trova un file .tex principale con lo stesso nome del documento, anch'esso scritto con la convenzione snake_case. In caso il documento sia strutturato attraverso la creazione di più sottofile, tutti i sottofile si devono trovare all'interno di una cartella detta componenti, la quale conterrà tutti i sottofile in formato .tex, nominati con una numerazione progressiva seguita dal nome della sezione che quel file rappresenta. Anche qui la convenzione usata è lo snake_case.

3.1.8 Struttura dei documenti

Tutti i documenti prodotti si rifanno ad un template predefinito scritto in formato .tex che ne definisce la struttura e le parti comuni. Ogni documento è caratterizzato, nella prima pagina, che forma il frontespizio, da una serie di elementi. Dall'alto verso il basso ci sono:

- Il logo a colori del gruppo;
- Il nome del gruppo;
- Il contatto e-mail del gruppo;
- Il nome del documento;
- Una tabella riportante la versione, l'approvatore, i redattori, i verificatori, l'uso e i destinatari;
- Una breve descrizione del contenuto del documento.

A seguire c'è una pagina che riporta il registro delle modifiche, in cui vengono specificati:

- Versione del documento;

- Cognome e nome di chi ha apportato la modifica;
- Ruolo all'interno del gruppo;
- Data della modifica;
- Descrizione della modifica.

Di seguito si trova una sezione in cui c'è l'indice del documento ed eventualmente, se necessarie, una lista di tabelle e una lista di figure.
A seguire c'è il corpo del documento, una sezione molto variabile a seconda del documento considerato. Nonostante questa variabilità la struttura della singola pagina rimane fissa ed è così strutturata:

- In alto a sinistra ci sono il nome del gruppo e il nome del documento;
- In alto a destra c'è il logo del gruppo nella versione colorata di nero;
- Al di sotto di questi due elementi c'è una linea nera che li separa dal contenuto della pagina;
- Il contenuto della pagina;
- Nel piè di pagina si trova, a destra, il numero della pagina corrente e il numero di pagine totali.

Ai documenti informali, ovvero ai verbali, non viene applicato nessun versionamento, questo fa si che il registro delle modifiche abbia sempre 3 righe: una corrispondente alla stesura, una corrispondente alla verifica e una corrispondente all'approvazione. I verbali hanno la seguente struttura:

- Frontespizio;
- Registro delle modifiche;
- Indice;
- Informazioni generali formate da: luogo dell'incontro, data dell'incontro, orario dell'incontro, partecipanti;
- Ordine del giorno;
- Resoconto;
- Conclusione.

3.1.9 Norme tipografiche

3.1.9.1 Attribuzione del nome

Per l'attribuzione del nome si segue la convenzione detta snake_case. La convenzione scelta prevede la scrittura di tutti i nomi con stampatello minuscolo e in caso di multiple parole di separarle con l'underscore.

3.1.9.2 Stili di testo

Di seguito sono indicati gli stili del testo:

- **Grassetto**: Stile utilizzato per singole parole da enfatizzare;
- **Corsivo**: Utilizzato per i nomi propri o per evidenziare frasi complete;
- **Monospace**: utilizzato per riportare frammenti di codice;
- **Stampatello maiuscolo**: utilizzato per le sigle.

3.1.9.3 Elenchi puntati

Per gli elenchi puntati la convenzione utilizzata è il punto. Nel caso in cui l'elenco sia annidato si usa il trattino alto (-).

3.1.9.4 Formati di dato

Le date vengono indicate con il formato:

[YYYY]-[MM]-[DD]

Secondo lo standard ISO 8601. In questa notazione YYYY indica l'anno, MM indica il mese e DD indica il giorno.

3.1.9.5 Sigle

Le sigle utilizzate sono riportate di seguito. Per quanto riguarda i documenti, le sigle utilizzate sono:

- Analisi dei requisiti: AdR;
- Piano di progetto: PdP;
- Piano di qualifica: PdQ;
- Studio di fattibilità: SdF;
- Norme di progetto: NdP;
- Verbali interni: VI;
- Verbali esterni: VE.

Per quanto riguarda le revisioni di progetto le sigle utilizzate sono:

- Revisione dei requisiti: RR;
- Revisione di progettazione: RP;
- Revisione di qualifica: RQ;
- Revisione di accettazione: RA.

3.1.10 Elementi grafici

3.1.10.1 Immagini

Le immagini devono essere centrate nella pagina e devono avere un'opportuna didascalia.

3.1.10.2 Grafici UML

I grafici UML sono inseriti come immagini.

3.1.10.3 Tabelle

Ogni tabella, ad eccezione del registro delle modifiche, è accompagnata da una didascalia e deve essere centrata nella pagina.

3.1.11 Strumenti

3.1.11.1 LaTeX

Questo strumento è stato scelto dal gruppo per far si che ci fosse uniformità nella stesura dei documenti e al contempo per avere uno strumento che permettesse grande flessibilità nella stesura stessa.

3.1.11.2 Draw.io

Applicazione web molto versatile per disegnare grafici UML che offre la possibilità di collaborare con più persone costantemente.

3.1.12 Metriche

3.1.12.1 Indice Gulpease$_G$

Quest'indice riporta il grado di leggibilità di un documento. Viene usato il sito https://farfalla-project.org/readability_static/ che permette di calcolarlo semplicemente incollando il testo nell'apposita casella dedicata.

3.1.12.2 Correzione errori ortografici

Per la correzione di errori ortografici viene utilizzato lo strumento di correzione automatica dell'editor usato per la stesura del documento.

3.2 Gestione delle configurazioni

3.2.1 Scopo

Questo processo ha lo scopo di gestire in maniera ordinata e sistematica la produzione di documenti e di codice.

3.2.2 Descrizione

La gestione della configurazione ha lo scopo di raggruppare, normare e organizzare tutti gli strumenti necessari alla produzione di documenti e di codice.

3.2.3 Aspettative

Le aspettative per questo processo sono le seguenti:

- Sistematizzare la produzione di codice e documenti;
- Uniformare gli strumenti utilizzati.

3.2.4 Versionamento

3.2.4.1 Codice di versione

Un documento ha sempre associato un codice di versione, che si presenta nella forma:

[X].[Y].[Z]

In questa notazione i significati sono i seguenti:

- La **X** indica l'ultima versione approvata dal *responsabile*;
- La **Y** indica l'ultima versione verificata da uno dei *verificatori*. La numerazione ricomincia da 0 quando cambia **X**
- La **Z** indica l'ultima versione stesa dai *redattori*. La numerazione ricomincia da 0 quando cambia **X** o **Y**

3.2.4.2 Tecnologie adottate

Per il versionamento dei file è stato deciso di adottare il sistema di versionamento *Git*. Ci sono inoltre due repository remoti su *GitHub*.

3.2.4.3 Repository

Il gruppo ha creato, per il progetto, un'organizzazione su GitHub in cui sono presenti 2 repository:

- **Documenti**: Al link https://github.com/Spaghetti-Code-G6/Documenti si trova il repository di tutti i documenti prodotti dal gruppo;
- **Codice**: Al link https://github.com/Spaghetti-Code-G6/Codice è possibile, invece, trovare il repository con il codice prodotto dal gruppo.

3.2.4.4 Struttura del repository

Il repository Codice è ancora inutilizzato, mentre il repository Documenti è così strutturato:

- **Documenti esterni**: questa directory contiene tutti i documenti destinati ad uso esterno, ovvero indirizzati al proponente e ai committenti. All'interno ogni documento si trova all'interno dell'omonima directory, la quale a sua volta segue le regole di struttura descritte nel paragrafo "Directory dei documenti" nella sezione dedicata alla documentazione;
- **Documenti interni**: questa directory ha una struttura analoga a quella dei documenti esterni. Raggruppa i documenti destinati ad uso interno;
- **Presentazione**: in questa directory è presente solo la presentazione preparata per la RR;
- **Risorse**: in questa directory è presente una sottodirectory che raggruppa tutte le immagini che il gruppo utilizza in tutti i documenti prodotti. Inoltre, sono presenti i file *config.tex* e *template.tex* che vengono utilizzati nel template dei documenti;
- **.gitignore**
- **Readme**

3.2.4.5 Tipi di file

Nella directory dei documenti sono presenti i seguenti tipi di file:

- File **.tex**: sono i file che il gruppo usa per scrivere i documenti. File sorgenti per LaTeX;
- File **.pdf**: sono i risultati della compilazione dei file .tex;
- Immagini da inserire nei documenti;
- File **.md**: il formato del file README;
- **.gitignore**: contiene un riferimento a tutti i file non versionati.

3.2.4.6 Comandi Git

La repo dei documenti ha diversi branch: un branch main e diversi branch per la stesura dei singoli documenti. Per lavorare su un documento si eseguono i seguenti passi:

- Bisogna per prima cosa posizionarsi sul repository locale;
- Aprire il terminale;
- Spostarsi sul branch in cui si vuole lavorare con il comando git branch seguito dal nome del branch;
- Eseguire il comando git pull per sincronizzare eventuali cambiamenti del repository remoto;
- Modificare il file;
- Sul terminale, eseguire il comando git add seguito dai nomi dei file modificati;
- Lanciare il comando git commit -m, seguito da un commento che riassume i cambiamenti tra doppi apici;
- Lanciare il comando git push per aggiornare il repository remoto;
- Se la stesura è completa e il documento è stato verificato, si esegue una pull request che verrà accettata soltanto quando il *responsabile* approva il documento.

3.2.4.7 Gestione delle modifiche

Ogni membro del gruppo può modificare i file nel repository dei documenti, tranne i file presenti sul ramo master, per i quali serve fare una pull request$_G$ che deve essere approvata.
È possibile trovarsi in una situazione in cui è necessario modificare un documento già approvato. In questo caso si deve necessariamente contattare il *responsabile* e chiedere l'autorizzazione prima di apportare qualsiasi modifica.

3.3 Gestione della qualità

3.3.1 Scopo

La gestione della qualità ha lo scopo di garantire qualità del prodotto, ovvero il fatto che il prodotto rispetti gli standard di qualità imposti e che le esigenze siano soddisfatte.

3.3.2 Descrizione

Alla gestione della qualità è dedicato il documento "Piano di Qualifica", in cui vengono descritte le metriche usate per valutare la qualità dei prodotti e dei processi. Questo documento contiene informazioni sui seguenti temi:

- Qualità del prodotto: definizione del modello$_G$ seguito e delle metriche adottate;
- Specifiche dei test.

3.3.3 Aspettative

Le aspettative sono le seguenti:

- Conseguimento della qualità del prodotto;
- Prova oggettiva della qualità;
- Raggiungimento della soddisfazione del proponente.

3.3.4 Attività

Il processo di gestione della qualità è suddiviso nelle seguenti parti:

- **Pianificazione**: Vengono stabiliti gli obbiettivi dei controlli di qualità;
- **Esecuzione**: Vengono messe in pratica le regole prefissate nella pianificazione;
- **Valutazione**: Confronto dei valori ottenuti cono quelli attesi.

Se i valori attesi non sono quelli specificati nelle norme il gruppo si impegna ad apportare le modifiche necessarie perché i valori siano conformi a quelli attesi.

3.3.5 Denominazione delle metriche

Le metriche seguono la seguente denominazione:

M[Categoria][Numero]

In questa notazione:

- La categoria indica a quale categoria la metrica appartiene;
- Il numero indica l'identificativo numerico della metrica.

Il gruppo si riserva la possibilità di integrare le metriche, in caso fosse necessario, successivamente.

3.3.6 Istanziazione di un processo

Il gruppo ha deciso di seguire delle regole per l'istanziazione dei processi in modo da perseguire in maniera più efficiente la qualità e rendere più semplice l'esecuzione del processo di gestione della qualità. Le regole sono le seguenti:

- Ogni processo deve avere un solo obbiettivo;
- L'obbiettivo di un processo non deve sovrapporsi agli obbiettivi di altri processi;
- L'organizzazione delle risorse affidate a un processo deve tenere conto delle risorse già assegnate agli altri processi, allo scopo di massimizzare l'efficienza$_G$ dell'impiego delle risorse umane;
- Ogni processo deve essere pianificato;
- Deve essere nota la durata dei processi e questi devono essere costantemente monitorati per far si che eventuali anticipi o ritardi vengano segnalati immediatamente.

3.4 Verifica

3.4.1 Scopo

Lo scopo della verifica è quello di avere dei prodotti che siano completi e coesi, oltre che corretti.

3.4.2 Descrizione

Questo processo prende in input un prodotto finito e lo restituisce conforme alle norme stabilite. Se il prodotto è già conforme allora il processo lo restituisce inalterato.

3.4.3 Aspettative

La verifica rispetta i seguenti punti:

- Viene effettuata seguendo passi precisi e predefiniti;
- I criteri di verifica sono chiari, oggettivi e affidabili;
- Tutti i prodotti vengono verificati, in ognuna delle fasi che attraversano;
- La verifica lascia il prodotto o, nel caso di prodotti molto corposi, una sua sezione stabile;
- Solo se il prodotto è interamente validato può passare ad essere validato.

3.4.4 Attività

3.4.4.1 Analisi

3.4.4.1.1 Analisi statica

Questo tipo di analisi si effettua sia sui documenti che sul codice e ne valuta la correttezza e la conformità alle regole. Questo tipo di analisi può essere sia manuale che automatico. Un esempio di controllo automatico è il controllo sugli errori ortografici o sugli errori di battitura, un esempio di controllo manuale è il controllo sul lessico utilizzato in un documento.

3.4.4.1.2 Analisi dinamica

Questo tipo di analisi, applicata al solo prodotto software, prevede la sua esecuzione. Viene effettuata tramite test.

3.4.4.2 Test

I test sono il cuore dell'analisi dinamica. Il loro scopo è mostrare che il prodotto software funziona come richiesto. Per definire un test bisogna definire una serie di parametri:

- **Ambiente**: Sistema in cui il test viene eseguito;
- **Stato iniziale**: Stato iniziale dal quale il test viene eseguito;
- **Input**: I dati in ingresso che il test richiede;
- **Output**: I dati che sono attesi;
- **Eventuali istruzioni aggiuntive**: Ulteriori specifiche necessarie per l'esecuzione del test o sull'interpretazione dei risultati ottenuti

Il gruppo si riserva di elencare successivamente i tipi di test previsti in quanto ritiene prematuro vincolare i test prima della progettazione.

3.4.4.3 Codice identificativo dei test

I test vengono descritti da un codice identificativo univoco che permette di distinguerli. Inoltre viene fornita una descrizione e uno stato. I valori che lo stato può assumere sono:

- Implementato;
- Non implementato;
- Non eseguito;
- Superato;
- Non superato.

Il codice identificativo si presenta nella forma:

T[Tipologia][Id]

La tipologia del test varia tra alcuni valori identificativi della tipologia di test, tuttavia, non essendo ancora i tipi di test definiti, queste saranno elencate successivamente.
L'id invece è un campo numerico che permette di identificare il test in base ad una numerazione progressiva.

3.4.5 Metriche

3.4.5.1 Densità degli errori

Questo è un indice che permette di capire quanto un prodotto software è capace di resistere agli errori. La formula adottata è:

$$M = \frac{numero\ test\ con\ errori}{numero\ test\ eseguiti} * 100$$

Il gruppo ritiene sia prematuro stabilire un valore soglia al di sotto del quale non scendere, perciò si riserva di ampliare questa sezione successivamente.

3.5 Validazione

3.5.1 Scopo

Lo scopo della validazione è stabilire se il prodotto soddisfa il compito per il quale è stato creato. A seguito della validazione è garantito che il prodotto rispetti i requisiti imposti.

3.5.2 Descrizione

Il processo prende in input il risultato della verifica e lo restituisce con la garanzia che rispetti i requisiti imposti dal committente e dal proponente. Questo compito è svolto dal *responsabile*

3.5.3 Aspettative

Ci si aspetta, da questo processo, un modo per avere la garanzia della correttezza e della completezza del prodotto rispetto ai requisiti imposti.

3.5.4 Attività

Il *responsabile* ha il compito di controllare il prodotto e decidere se approvare il prodotto o se rigettarlo chiedendo una nuova verifica.

4 Processi organizzativi

4.1 Processi di coordinamento

4.1.1 Scopo

In questo paragrafo vengono raccolte le modalità di coordinamento adottate dal gruppo per le comunicazioni e gli incontri con i vari soggetti coinvolti durante tutta la vita del prodotto. Si svolge questa attività per avere dei canali d'interazione comuni e non escludere nessuno o rischiare di perdere informazioni importanti. Rispetto agli anni passati, questo processo assume un ruolo di assoluta importanza, date le circostanze di questo periodo nel quale è impossibile incontrasi di persona.

4.1.2 Comunicazione

Durante lo svolgimento di questo progetto verranno usati più strumenti per la comunicazione a seconda del soggetto interessato. I soggetti divisi per ruolo sono:

- Proponente: *Gregorio Piccoli* rappresentante dell'azienda *Zucchetti S.p.A.*;
- Committente: prof. *Vardanega Tullio* e prof. *Cardin Riccardo*;
- *SpaghettiCode*: tutti i membri del gruppo;
- Altri gruppi: attualmente i gruppi candidati allo stesso capitolato sono Gruppo 5,10 e 15.

Tutte le comunicazioni saranno svolte per via telematica, data l'impossibilità di incontrarsi di persona.

4.1.2.1 Interna

I membri del gruppo si sono accordati per usare Discord come canale principale per le comunicazioni e Telegram come canale secondario. Su Discord sono stati creati dei canali appositi divisi a seconda delle tematiche che vengono trattate (es. generale, note-risorse, incontri, gestione-gruppo), inoltre ogni ruolo ha a disposizione un canale apposito nel quale scrivere tutto ciò che riguarda i loro compiti. Il gruppo di Telegram viene usato per comunicazioni molto informali e più istantanee.

4.1.2.2 Esterna

Le comunicazioni con gli altri gruppi dello stesso capitolato avverranno tramite un gruppo Telegram. Le comunicazioni con i committenti avverranno tramite video-chiamate con Zoom e messaggi di posta elettronica. Le comunicazioni con il proponente avverranno tramite video-chiamate su Zoom, Skype e messaggi di posta elettronica.

4.1.2.3 Riunioni

Tutte le riunioni saranno svolte tramite i canali di comunicazione scelti per il soggetto interessato.

4.1.2.4 Strumenti

- Discord: applicazione VoIP e di messaggistica istantanea, facile da usare e versatile;
- Telegram: servizio di messaggistica istantanea;
- Zoom: servizio di teleconferenze;
- Skype: software di messaggi istantanea e VoIP.

4.2 Processi di pianificazione

4.2.1 Scopo

Nei prossimi paragrafi si descriverà come il gruppo *SpaghettiCode* intende lavorare. Si tratteranno i ruoli e la loro divisione tra i membri, i compiti di ciascun membro e l'assegnazione di essi. Prendendo come esempio da seguire lo standard ISO/IEC 12207, il processo di pianificazione sarà strutturato nel seguente modo:

- Ruoli con relativi compiti;
- Assegnazione ruoli;
- Divisione del lavoro.

4.2.2 Ruoli di progetto

4.2.2.1 Responsabile

Il **responsabile** rappresenta il progetto ed è il punto di riferimento per le comunicazioni con il committente. Per poter pianificare ed anticipare l'evoluzione del progetto deve possedere capacità tecniche e delle competenze pregresse, deve essere in grado di gestire le risorse e tracciare i progressi. Ha la responsabilità di scelta e approvazione su gran parte del progetto e partecipa per tutta la durata di esso.
In particolare, ha il compito di:

- Redigere l'organigramma;
- Redigere il Piano di Progetto;
- Coordinare i membri del gruppo, le attività e le risorse a disposizione;
- Gestire le criticità;
- Approvare i vari documenti;
- Approvare l'offerta del committente.

4.2.2.2 Amministratore

L'**amministratore$_G$** è responsabile dell'efficienza e dell'operatività dell'ambiente di lavoro, deve assicurarsi che le risorse siano sempre presenti e operanti.
In particolare, ha il compito di:

- Gestire il controllo della configurazione del prodotto;
- Gestire il versionamento;
- Gestire la documentazione del progetto;
- Redigere le Norme di Progetto;
- Collaborare alla redazione del Piano di Progetto;
- Redazione e attuazione di piani e procedure di Gestione della Qualità;
- Risolvere i problemi legati alla gestione dei processi.

4.2.2.3 Analista

L'**analista$_G$** ha una notevole esperienza professionale e vasta conoscenza del dominio del problema, si occupa di esporre il problema in maniera chiara con un linguaggio simile a quello usato dal proponente. Ci possono essere più **analisti** contemporaneamente, il loro lavoro ha un grande impatto sulla riuscita del progetto, ma sono generalmente pochi e non seguono il progetto fino alla sua fine.
In particolare, ha il compito di:

- Redigere lo Studio di Fattibilità;
- Redigere l'Analisi dei Requisiti.

4.2.2.4 Progettista

Il **progettista$_G$** è una persona con competenze tecniche e tecnologiche avanzate e con un'ampia esperienza professionale. Si occupa dello sviluppo della soluzione al problema presentato tramite le attività di progettazione, spesso assumendo anche responsabilità di scelta e gestione. Possono essercene di più contemporaneamente, ma sono comunque pochi e possono seguire il progetto fino alla manutenzione.
In particolare, ha il compito di:

- Redigere la Specifica Tecnica;
- Redigere la Definizione di Prodotto;
- Redigere la parte programmatica del Piano di Qualifica.

4.2.2.5 Programmatore

Il **programmatore$_G$** ha competenze tecniche specifiche, ma responsabilità limitate, che si occupa di implementare la soluzione trovata dal **progettista** tramite attività di codifica del prodotto e dei test di ausilio alla verifica. Rimane a lungo all'interno del progetto, partecipando anche alla manutenzione.
In particolare, ha il compito di:

- Implementare la Specifica Tecnica tramite codifica;
- Implementare i test d'ausilio necessari per l'esecuzione delle prove di verifica e validazione.

4.2.2.6 Verificatore

Il **verificatore$_G$** ha competenze tecniche, esperienze di progetto e conoscenza delle norme, oltre che a capacità di giudizio e relazione. Si occupa di attività di verifica e validazione, partecipa all'intero ciclo di vita assicurandosi che quanto fatto sia conforme alle attese. Illustra nel Piano di Qualifica l'esito e la completezza delle verifiche e delle prove effettuate.
In particolare, ha il compito di:

- Esaminare i prodotti in fase di revisione tramite le tecniche e gli strumenti descritti nelle Norme di Progetto;
- Segnalare eventuali errori o modifiche necessari ai diretti interessati, in modo che possano correggerli.

4.2.3 Assegnazione dei ruoli

I vari ruoli verranno assegnati ai membri del gruppo a rotazione ad ogni raggiungimento di una scadenza terminale. Ogni membro dovrà ricoprire più ruoli durante l'intero ciclo di sviluppo, per un periodo significativo, abbastanza lungo da non interrompere la continuità delle attività in corso. Inizialmente i ruoli sono stati assegnati casualmente, perché nessuno dei membri del gruppo aveva conoscenze pregresse. Si prevede di fare scelte più mirate alla prossima rotazione, tenendo in considerazione le conoscenze acquisite nell'ultimo periodo e gli interessi sviluppati da parte dei membri verso le tecnologie usate.

4.2.4 Assegnazione dei compiti

Ogni membro dovrà svolgere i suoi compiti in base al ruolo assegnatogli. Per tenere traccia di cosa è stato fatto e di cosa bisogna fare si è deciso di usare il sistema di Issue Tracking offerto da GitHub. Ogni membro potrà creare delle Issue con il lavoro da svolgere, dovranno essere piccoli task che potranno essere svolti anche da una sola persona. Se il lavoro da svolgere non è strettamente legato ad un ruolo preciso sarà prenotabile da qualsiasi membro.

4.2.5 Metriche

Durante i meeting tra i membri del gruppo ci si dovrà accordare con delle scadenze e verranno fissate delle Milestone$_G$ su GitHub. Tramite le Milestone si potrà conoscere l'andamento dei lavori e la percentuale di completamento, infatti, ogni Issue creata dovrà essere assegnata ad una Milestone.

4.2.6 Gestione dei rischi

Per gestire i rischi, il gruppo si è accordato nell'usare un sistema di tag tramite Label delle Issue. Alla creazione di una nuova Issue sarà possibile assegnare un livello di priorità, che andrà da minore, normale, importante e critica a seconda dell'urgenza con la quale deve essere svolto un compito. Così ogni membro del gruppo potrà vedere se ci sono attività che necessitano più attenzione rispetto ad altre e potrà intervenire, assegnandosi la Issue oppure sollecitando i diretti interessati.

4.3 Formazione

4.3.1 Scopo

Lo scopo della formazione è quello di uniformare le capacità tecniche e le conoscenze tra i vari membri del gruppo in modo da poter lavorare e comunicare in sintonia.

4.3.2 Descrizione

Per ogni membro di *SpaghettiCode* è prevista la formazione tramite studio autonomo delle varie tecnologie che vengono adoperate o che sono state richieste da Zucchetti S.p.A. durante la presentazione del capitolato e durante gli incontri successivi. In caso di difficoltà il gruppo è disponibile a fare formazione tramite incontri su Discord.

4.3.3 Aspettative

Ci si aspetta che tutti i membri del gruppo acquisiscano familiarità con le seguenti tecnologie:

- LaTeX;
- Git e GitHub;
- JavaScript;
- Libreria D3.js.