



SpaghettiCode

spaghetti.code.g6@gmail.com

NORME DI PROGETTO

Versione	v0.0.4
Approvazione	Giorgia Paparazzo
Redazione	Samuel Kostadinov
Verifica	PLACEHOLDER
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
v0.0.4	Samuel Kostadinov	Amministratore	2020-12-28	Fine stesura §2
v0.0.3	Samuel Kostadinov	Amministratore	2020-12-27	Continuazione stesura §2
v0.0.2	Samuel Kostadinov	Amministratore	2020-12-26	Stesura §1 e inizio stesura §2
v0.0.1	Manuel Pagotto	Analista	2020-12-17	Creazione del documento



Indice

1	Introduzione	1
1.1	Scopo del documento	1
1.2	Scopo del prodotto	1
1.3	Glossario	1
1.4	Riferimenti	1
1.4.1	Riferimenti normativi	1
1.4.2	Riferimenti informativi	1
2	Processi primari	2
2.1	Fornitura	2
2.1.1	Scopo	2
2.1.2	Descrizione	2
2.1.3	Aspettative	2
2.1.4	Attività	2
2.1.4.1	Studio di Fattibilità	2
2.1.4.2	Piano di Progetto	3
2.1.4.3	Piano di Qualifica	4
2.1.5	Strumenti	4
2.1.5.1	gitHub	4
2.1.5.2	Google docs/Google drive	4
2.1.5.3	L ^A T _E X	4
2.1.5.4	Discord	4
2.1.5.5	Telegram	4
2.1.5.6	Trello	5
2.2	Sviluppo	5
2.2.1	Scopo	5
2.2.2	Descrizione	5
2.2.3	Aspettative	5
2.2.4	Analisi dei requisiti	5
2.2.4.1	Scopo	5
2.2.4.2	Descrizione	6
2.2.4.3	Aspettative	6
2.2.4.4	Struttura del documento	6
2.2.4.5	Classificazione dei requisiti	6
2.2.4.6	Classificazione dei casi d'uso	7
2.2.4.7	Tracciamento dei requisiti e dei casi d'uso	8
2.2.4.8	Metriche	8
2.2.5	Progettazione	8
2.2.5.1	Scopo	8
2.2.5.2	Descrizione	8
2.2.5.3	Aspettative	8
2.2.5.4	Design pattern	9
2.2.5.5	Diagrammi UML	9
2.2.5.6	Test	9
2.2.5.7	Nota	9
2.2.6	Codifica	9
2.2.6.1	Scopo	9
2.2.6.2	Descrizione	9
2.2.6.3	Aspettative	9
2.2.6.4	Stile di codifica	9
2.2.6.5	Metriche	10
2.2.6.6	Strumenti	10



2.2.6.6.1	HTML, CSS	10
2.2.6.6.2	Database	10
2.2.6.6.3	JavaScript	10
2.2.6.6.4	d3.js	10

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 conformi con lo 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.

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_G 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 v0.0.1, 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 cul 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;



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 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 QUALIFICAZIONE_G.

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 trattate tutte norme a cui il gruppo *SpaghettiCode* deve attenersi, con lo scopo di diventare i fornitori_G 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 obiettivi di mantenere un confronto costante con il proponente *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, redatto dagli *analisti* dopo aver deciso la prima scelta del gruppo tra i vari progetti disponibili, viene fornita un'analisi generale di tutti i capitoli proposti, e le motivazioni che hanno spinto il gruppo *SpaghettiCode* a proporsi o meno come fornitore di uno specifico capitolo. 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_G*, 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 (deadlines_G) per il loro completamento. Queste deadlines non sono però rigide perchè, 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 del progetto. Viene quindi esposto un preventivo e un successivo consuntivo, entrambi relativi ad un dato periodo.



2.1.4.3 Piano di Qualifica

Il PIANO DI QUALIFICA 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_G*. Questo documento è strutturato nel seguente modo:

- **Qualità di processo:** sono individuati i processi dagli standard di processo_G, definiti degli obiettivi, strategie per attuarli e metriche per controllarli e misurarli;
- **Qualità di prodotto:** vengono individuate le caratteristiche più importanti del prodotto, gli obiettivi necessari per raggiungerle e le metriche per misurarle;
- **Specifiche dei test:** vengono definiti dei test 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;
- **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.

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, 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 issue tracking system, 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 L^AT_EX

Il gruppo ha di comune accordo deciso, dopo aver valutato l'opzione, di usare lo strumento L^AT_EX per la scrittura formale dei documenti, perciò ha predisposto su gitHub un repository 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, viene utilizzato per organizzare gli incontri, per condividere risorse tra i membri e per suddividere in vari canali, corrispondenti ai diversi ruoli, la comunicazione, così che ci sia una maggiore organizzazione e divisione dei ruoli stessi. Discord viene utilizzata per le comunicazioni importanti

2.1.5.5 Telegram

Il gruppo ha deciso di utilizzare un gruppo Telegram come canale di comunicazione per le comunicazioni di importanza inferiore 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

Il gruppo aveva inizialmente deciso di provvedere all'organizzazione tramite Trello, tuttavia successivamente è stato preferito utilizzare le issue di gitHub.

2.2 Sviluppo

2.2.1 Scopo

Aderendo a 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:** una possibilità è che un requisito emerga durante uno degli incontri interni, quindi quelli in cui partecipano i soli membri del gruppo, durante una discussione;
- **Incontri esterni:** un'ulteriore possibilità è che un requisito emerga in un incontro esterno, quindi in un incontro tra il gruppo e il referente dell'azienda proponente;
- **Casi d'uso:** infine è possibile che il requisito emerga durante la stesura dei casi d'uso perchè 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, sul glossario, sui riferimenti (normativi e informativi);
- **Descrizione generale:** Contiene una descrizione del documento di analisi dei requisiti, che riporta informazioni a proposito del prodotto (in particolare i suoi obiettivi e le sue funzioni), sugli utenti (nello specifico le loro caratteristiche), le architetture e le tecnologie (PLACEHOLDER), 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 sì 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_G:** 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_G:** 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 di gitHub, strumento che tramite il workflow delle issue 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 sì 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

(?)

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 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 erranee
- 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

(?)

2.2.5.6 Test

Come specificato all'inizio, ogni la definizione dei test è parte dell'attività di progettazione, quindi ogni progettista dovrà definire i test necessari. Le regole di nomenclatura saranno specificate nel in (PLACEHOLDER)

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 PLACEHOLDER).

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'obiettivo 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:
- Parentesi:
- Struttura dei metodi:
- 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:
- Lingua:



2.2.6.5 Metriche

(?)

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

(?)

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.