



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 di cui il gruppo *SpaghettiCode* ha deciso di avvalersi 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

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	3
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	5
2.1.5.6	Trello e GitHub Projects	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 progressi e delle risorse	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
2.2.6	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
3	Processi di supporto	11
3.1	Documentazione	11
3.1.1	Scopo	11
3.1.2	Descrizione	11
3.1.3	Aspettative	11
3.1.4	Ciclo di vita di un documento	11
3.1.5	Template	11
3.1.6	Documenti prodotti	11
3.1.7	Cartella dei documenti	12
3.1.8	Struttura dei documenti	12
3.1.9	Norme tipografiche	13
3.1.9.1	Attribuzione del nome	13
3.1.9.2	Stili di testo	13
3.1.9.3	Elenchi puntati	13
3.1.9.4	Formati di date	13
3.1.9.5	Sigle	14
3.1.10	Elementi grafici	14
3.1.10.1	Immagini	14
3.1.10.2	Grafici UML	14
3.1.10.3	Tabelle	14
3.1.11	Strumenti	14
3.1.11.1	L ^A T _E X	14
3.1.11.2	Draw.io	14
3.1.12	Metriche	14
3.1.12.1	Indice Gulpease _G	14
3.1.12.2	Correzione errori ortografici	15
3.2	Gestione delle configurazioni	15
3.2.1	Scopo	15
3.2.2	Descrizione	15
3.2.3	Aspettative	15
3.2.4	Versionamento	15
3.2.4.1	Codice di versione	15
3.2.4.2	Tecnologie adottate	15
3.2.4.3	Repository	15
3.2.4.4	Struttura del repository	16
3.2.4.5	Tipi di file	16
3.2.4.6	Comandi Git	16
3.2.4.7	Gestione delle modifiche	17
3.3	Gestione della qualità	17
3.3.1	Scopo	17
3.3.2	Descrizione	17
3.3.3	Aspettative	17
3.3.4	Attività	17
3.3.5	Denominazione delle metriche	17
3.3.6	Istanziamento di un processo	18
3.4	Verifica	18
3.4.1	Scopo	18
3.4.2	Descrizione	18
3.4.3	Aspettative	18
3.4.4	Attività	18
3.4.4.1	Analisi	18



3.4.4.1.1	Analisi statica	18
3.4.4.1.2	Analisi dinamica	18
3.4.4.2	Test	19
3.4.4.3	Codice identificativo dei test	19
3.4.5	Metriche	19
3.4.5.1	Densità degli errori	19
3.5	Validazione	19
3.5.1	Scopo	19
3.5.2	Descrizione	20
3.5.3	Aspettative	20
3.5.4	Attività	20
4	Processi organizzativi	21
4.1	Processi di coordinamento	21
4.1.1	Scopo	21
4.1.2	Comunicazione	21
4.1.2.1	Interna	21
4.1.2.2	Esterna	21
4.1.2.3	Riunioni	21
4.1.2.4	Strumenti	21
4.2	Pianificazione	22
4.2.1	Scopo	22
4.2.2	Ruoli di progetto	22
4.2.2.1	Responsabile	22
4.2.2.2	Amministratore	22
4.2.2.3	Analista	23
4.2.2.4	Progettista	23
4.2.2.5	Programmatore	23
4.2.2.6	Verificatore	23
4.2.3	Assegnazione dei ruoli	23
4.2.4	Assegnazione dei compiti	24
4.2.5	Metriche	24
4.2.6	Gestione dei rischi	24
4.3	Formazione	24
4.3.1	Scopo	24
4.3.2	Descrizione	24
4.3.3	Aspettative	24



1 Introduzione

1.1 Scopo del documento

Lo scopo di questo documento è quello di fissare le convenzioni e regole che ciascun membro del gruppo *SpaghettiCode* dovrà rispettare al fine di sviluppare efficientemente ed efficacemente il prodotto richiesto. L'aggregazione delle varie attività_G descritte all'interno del documento genera diversi processi_G, ispirati allo standard ISO/IEC 12207:1995, che permetteranno ai membri del team di effettuare transizioni di stato del progetto il più economicamente possibile.

Il documento segue uno sviluppo di tipo incrementale e perciò sarà sottoposto a continua evoluzione. Al completamento di ciascuna modifica tutti i membri del gruppo verranno notificati, di modo da poter adeguarsi alle norme il più aggiornate possibili. Inoltre il presente è da considerarsi incompleto e verrà aggiornato relativamente alle modifiche apportate alle regole specifiche di ciascun processo del progetto_G.

1.2 Scopo del prodotto

Il capitolato_G *HD Viz* si pone come obiettivo la realizzazione di una web application_G che permetta la visualizzazione di dati pluridimensionali mediante diverse tipologie di grafici, i quali permettono all'utente di individuare strutture e/o modelli intrinseci dei dati stessi, nella fase di esplorazione dei dati (EDA_G).

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 documento GLOSSARIO V1.0.0, nel quale vengono riportate le definizioni specifiche dei termini che presentano una "G" a 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_G:
<https://docs.github.com/en/free-pro-team@latest>;
- Documentazione L^AT_EX_G:
<https://www.latex-project.org/help/documentation/>;
- Documentazione JavaScript_G:
<https://developer.mozilla.org/en-US/docs/Web/JavaScript>;
- Documentazione libreria 3D.js_G:
<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, di competenze fondamentali e individuazione di rischi e criticità_G per il completamento del progetto. I risultati di questa analisi vengono redatti nel documento STUDIO DI FATTIBILITÀ;
- Stabilimento dell'organizzazione del lavoro che il gruppo seguirà durante lo svolgimento del progetto. I relativi risultati sono riportati nel documento PIANO DI PROGETTO. Il PIANO DI PROGETTO sarà, come del resto questo documento, in continua evoluzione, in quanto la pianificazione a lungo termine potrebbe subire modifiche a consuntivo di periodo causate da ritardi o anticipi rispetto a quanto preventivato;
- Stabilimento delle metriche di verifica della qualità del materiale prodotto durante lo svolgimento delle diverse attività. 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

In questa sezione vengono descritte e trattate tutte le norme a cui il gruppo *SpaghettiCode* intende attenersi durante il processo di fornitura, con lo scopo di diventare i fornitori del prodotto *HD Viz* del proponente_G *Zucchetti S.p.A.* e dei committenti *prof. Tullio Vardanega* e *prof. Riccardo Cardin*.

2.1.3 Aspettative

Nello svolgersi di tale processo, il gruppo di lavoro si pone gli obiettivi di mantenere un confronto costante con il proponente *Zucchetti S.p.A.*, 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. All'avvenuta consegna, il gruppo *SpaghettiCode* non seguirà il processo di manutenzione del prodotto salvo eventuali accordi.

2.1.4 Attività

2.1.4.1 Studio di Fattibilità

In seguito alla presentazione ufficiale dei diversi capitolati d'appalto, tenutasi in data 2020-11-05, i vari componenti del gruppo esprimono i propri interessi e dubbi in merito ai diversi capitolati. Successivamente, dopo aver deciso la prima scelta del gruppo tra i vari capitolati disponibili, gli *analisti* raccolgono ed analizzano meglio i punti di forza e le criticità emersi nei diversi capitolati, e riportano quanto riscontrato nel documento STUDIO DI FATTIBILITÀ, fornendo così 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:** si raccolgono tutte le 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;
- **Comprensione delle tecnologie interessate:** si studia e si determina quali siano 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 ponderati gli aspetti positivi e gli aspetti negativi del capitolato, traendo quindi le conclusioni in merito alla fattibilità

2.1.4.2 Piano di Progetto

Il gruppo deve pianificare come si svolgerà il progetto e considerare le tempistiche con cui ogni altra attività verrà eseguita. Quest'attività si formalizza nell'omonimo documento, redatto dagli *amministratori*, sotto la supervisione del *responsabile di progetto*_G. Vista la continua evoluzione del piano di progetto 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 inoltre fornite le modalità con cui vengono risolti o ridimensionati. 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, la pianificazione sarà sempre soggetta a modifiche e/o 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

La presente attività raccoglie tutte le regole e le linee guida per garantire che i materiali prodotti rispettino determinati standard di qualità. Questo insieme di regole deve essere, come per il piano di progetto, formalizzato nell'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 obiettivi, strategie per attuarli e metriche_G 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_G che il prodotto deve superare 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 (sezione 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 degli stessi.

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 tre 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, che permette di identificare univocamente il rischio.

Ogni rischio avrà, oltre al codice, due informazioni:

- **Occorrenza:** indica a 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_G, per la possibilità di lavorare in più branch_G contemporaneamente 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. 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_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

Questo strumento viene utilizzato 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 lo strumento è stato considerato di più veloce accesso rispetto a *Discord* dai membri del gruppo.

2.1.5.6 Trello e GitHub Projects

Il gruppo aveva inizialmente deciso di provvedere all'organizzazione tramite le board di Trello_G, tuttavia successivamente è stato preferito l'utilizzo dei GitHub Projects_G. Entrambe sono delle applicazioni web per la creazione di liste in stile Kanban.

2.2 Sviluppo

2.2.1 Scopo

Secondo quanto specificato nello standard ISO/IEC 12207:1995, lo sviluppo consiste nella descrizione dei diversi compiti ed attività di analisi, progettazione, codifica, integrazione, test, installazione e accettazione relative al prodotto in considerazione.

2.2.2 Descrizione

Questo processo è formato da tre attività principali:

- **Analisi dei requisiti**
- **Progettazione architettuale**
- **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 obiettivi 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 quello di individuare tutti i requisiti richiesti dal proponente o intrinseci nel prodotto stesso. L'attività viene svolta da parte degli *analisti*, e il suo risultato viene in seguito riportato nel documento ANALISI DEI REQUISITI_G redatto dagli *analisti* stessi.

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.



2.2.4.2 Descrizione

I requisiti, parte fondamentale di questo documento, possono essere ricavati 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 specifica.

2.2.4.3 Aspettative

Le aspettative dell'attività di analisi dei requisiti sono quelle di individuare e fissare i requisiti del prodotto e di produrre un documento che li raccolga.

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 che riporta informazioni circa il prodotto, i suoi obiettivi, le sue funzioni, i suoi utenti con le loro caratteristiche, le architetture da adottare, le tecnologie consigliate ed eventuali vincoli;
- **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, di vincolo e prestazionali, oltre che al tracciamento tra fonti e requisiti e viceversa.

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 prodotto.

- **Tipologia:** i requisiti possono avere diversi tipi, questa parte del codice identificativo permette di identificare la tipologia del requisito. I possibili valori sono:
 - **V:** indica che il requisito è di vincolo, ossia una limitazione imposta dal proponente circa i servizi offerti dal prodotto software all'utilizzatore finale;
 - **F:** indica che il requisito è di tipo funzionale, quindi viene usato per rappresentare le funzioni del prodotto;
 - **P:** indica che il requisito è di tipo prestazionale, quindi viene usato per dare delle limitazioni sulle prestazioni del prodotto software;
 - **Q:** indica 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 permette di individuare 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. Il soddisfacimento del requisito verrebbe visto positivamente dal proponente, in quanto fornirebbe al prodotto una maggiore completezza, eppure non ne viene vincolata l'implementazione;
 - **F:** indica che il requisito è facoltativo, dunque anche se porta un valore aggiunto al prodotto, molto probabilmente comporta una piccola miglioria a dispendio di molto tempo e lavoro.
- **Codice:** rappresenta il codice identificativo vero e proprio e 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, caso d'uso, 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 una struttura ben definita, riportata di seguito:

- **Descrizione:** breve descrizione del caso d'uso;
- **Attore primario:** entità che interagisce direttamente con il prodotto;
- **Attori secondari:** entità che supportano l'attore primario nel portare a termine il caso d'uso in esame, questo elemento non è necessariamente presente;
- **Precondizione:** condizione del sistema necessaria affinché possa compiersi il caso d'uso in esame;
- **Postcondizione:** condizione in cui si trova il sistema immediatamente dopo il compimento del caso d'uso in esame;
- **Scenario principale:** rappresentazione del flusso degli eventi previsti dal caso d'uso;
- **Scenari alternativi:** rappresentazioni alternative del flusso degli eventi del caso d'uso, questo elemento non è necessariamente presente;
- **Estensioni:** elemento che indica casi d'uso eseguiti condizionatamente che determinano l'interruzione dell'esecuzione del caso d'uso in esame, questo elemento non è necessariamente presente;
- **Inclusioni:** Elemento che indica casi d'uso eseguiti incondizionatamente successivamente a quello in esame, questo elemento non è necessariamente presente;
- **Generalizzazioni:** rappresentano delle possibili specializzazioni del caso d'uso, questo elemento non è necessariamente presente.



2.2.4.7 Tracciamento dei progressi e delle risorse

Per tenere traccia dell'andamento del lavoro e dei ruoli assegnati, 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 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

Il gruppo ha deciso di tenere traccia del completamento del progetto in base ai requisiti implementati. A questo scopo è stato deciso di calcolare una percentuale di completamento del progetto come il rapporto tra il numero di requisiti implementati e il numero dei requisiti totali, moltiplicato per 100. Tale metrica permette così di individuare lo stato di avanzamento del progetto in maniera immediata. 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 richieste del proponente in risposta ai requisiti individuati dall'analisi dei requisiti. In quest'attività bisogna rispettare i seguenti vincoli:

- Garantire la qualità del prodotto seguendo un principio di correttezza costruttivo;
- Organizzare e suddividere i compiti in modo da diminuire la complessità del problema, 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 rispettare i seguenti punti:

- Soddisfare i requisiti individuati dall'analisi dei requisiti;
- Adattarsi in caso i requisiti evolvano;
- 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 *Programmatici*. 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 diagrammi UML 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 affiancare alla progettazione del sistema la definizione dei relativi test. Le regole di nomenclatura da seguire sono le stesse valide per la nomenclatura dei metodi, descritte nella sezione §2.2.6.4.

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 *programmatici*, ha lo scopo di scrivere del codice che traduca lo schema architetturale prodotto dai *progettisti* in un prodotto eseguibile. Quest'attività è soggetta alle regole descritte nel paragrafo §2.2.6.4, per far sí che il codice sia più leggibile possibile.

2.2.6.2 Descrizione

La scrittura del codice dovrà tradurre lo schema architetturale prodotto 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: 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 il più brevi possibile 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 di blocco.
- 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 scritti in stampatello maiuscolo;
- Lingua: la lingua utilizzata per i nomi delle variabili e dei metodi deve essere l'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:

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

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 HD Viz l'utilizzo di queste tecnologie sarà necessario per il completamento del progetto.

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 tal proposito, sarà perciò compito dei *progettisti* scegliere il database più adatto al progetto.

2.2.6.6.3 JavaScript

Linguaggio di scripting per web application richiesto dal proponente che fornisce la possibilità di utilizzare la libreria 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 del processo di documentazione è quello di riportare, in uno o più documenti, quanto sia necessario all'utilizzo corretto del prodotto.

3.1.2 Descrizione

La produzione della documentazione è un processo che si svolge durante tutto il ciclo di vita del progetto; consiste nel redarre dei documenti di varia natura, riportati nella sezione §3.1.6, ognuno dei quali ricopre una specifica funzione. In questa sezione del documento vengono riportate le norme adottate per ottenere una buona scrittura della documentazione.

3.1.3 Aspettative

Questo processo ha come aspettativa principale la stesura di documenti chiara e che rispetti le regole di seguito definite, così da permettere agli stessi di svolgere al meglio la loro funzione.

3.1.4 Ciclo di vita di un documento

Il ciclo di vita di un documento deve seguire i seguenti passi:

- **Creazione del documento:** il documento viene creato a partire da un template e 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 la stesura dei documenti del progetto, il linguaggio L^AT_EX. Al fine di uniformare i documenti e velocizzare il processo di creazione e strutturazione degli stessi è stato definito un template su cui ogni documento si basa.

3.1.6 Documenti prodotti

I documenti prodotti vengono suddivisi in due macrocategorie:

- **Informali:** In questa macrocategoria rientrano i documenti non soggetti a versionamento, in particolare fanno parte di questa categoria i verbali, i quali fungono da resoconto degli incontri. I verbali vengono a loro volta distinti in verbali interni ed esterni:
 - **Verbali Interni:** a questa categoria appartengono i verbali degli incontri interni del gruppo;
 - **Verbali Esterni:** a questa categoria appartengono i verbali degli incontri con membri esterni al gruppo, quali potrebbero essere i committenti e/o i proponenti.
- **Formali:** In questa macrocategoria rientrano tutti i documenti soggetti a versionamento; questi devono essere approvati ad ogni loro versione dal *responsabile* di progetto e viene considerata come corrente la versione più recentemente approvata. Un'ulteriore distinzione viene fatta tra:



- **Documenti Interni:** Appartengono a questa categoria i documenti destinati ad uso interno al gruppo e non sono quindi di interesse primario per il proponente e i committenti;
- **Documenti Esterni:** Appartengono a questa categoria i documenti destinati ad essere consegnati, nella loro ultima versione approvata, al proponente e ai committenti.

Di seguito viene riportato un elenco dei documenti formali prodotti:

- **Norme di progetto:** documento interno che racchiude tutte le norme che regolamentano il progetto nella sua interezza;
- **Glossario:** documento esterno che elenca tutti i termini che, secondo il gruppo, necessitano di una spiegazione esplicita;
- **Studio di fattibilità:** documento interno che raccoglie i singoli studi di fattibilità sui capitoli, secondo lo schema descritto in questo documento al paragrafo §2.1.4.1;
- **Piano di progetto:** documento esterno che espone la pianificazione delle attività di progetto previste dal gruppo. In allegato a questo documento viene presentato un preventivo sia degli impegni orari che dei costi;
- **Piano di qualifica:** documento esterno che espone i criteri di valutazione della qualità del progetto;
- **Analisi dei requisiti:** documento esterno che raccoglie tutti i requisiti del progetto e che il prodotto software deve avere.

3.1.7 Cartella dei documenti

Per ogni documento viene creata un'omonima cartella e all'interno di questa viene posto un file .tex, rappresentante il documento stesso, assieme ad un'eventuale cartella *componenti* contenente i diversi file di supporto del documento. Ogni singolo elemento deve rispettare le norme di attribuzione dei nomi presenti nel paragrafo §3.1.9.1.

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 nel frontespizio da una serie di elementi, dall'alto verso il basso:

- 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 ci sono una o più pagine che riportano 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.

A seguire si trova la tabella dei contenuti e, se previste dal documento, una lista delle tabelle e una delle figure presenti.

Segue poi il corpo del documento, sezione caratteristica di ciascun documento considerato. Nonostante ciò la struttura di ogni 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 sua versione a colori;
- Al di sotto di questi due elementi c'è una linea nera che li separa dal contenuto della pagina;
- Il contenuto della pagina;
- A piè di pagina si trova, sulla destra, il numero della pagina corrente e il numero di pagine totali.

Ai documenti informali non viene applicato nessun versionamento, questo fa sì che il registro delle modifiche presenti sempre tre righe: una corrispondente alla stesura, una corrispondente alla verifica e una corrispondente all'approvazione. I verbali dunque presentano 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 viene seguita la convenzione `snake_case`; essa prevede la scrittura di tutti i nomi in stampatello minuscolo e la separazione delle parole mediante il carattere `underscore`.

3.1.9.2 Stili di testo

Di seguito sono indicati i differenti stili del testo adottati nei documenti, con il relativo significato:

- **Grassetto**: 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 o per riferirsi a dei documenti.

3.1.9.3 Elenchi puntati

Per indicare gli elementi di un elenco puntato viene utilizzato il punto elenco_G, mentre nel caso in cui l'elenco sia annidato viene utilizzato il trattino_G.

3.1.9.4 Formati di date

Le date vengono indicate secondo lo standard ISO 8601, con il formato:

[YYYY]-[MM]-[DD]

In questa notazione YYYY indica l'anno, MM indica il mese e DD indica il giorno.



3.1.9.5 Sigle

Vengono riportate di seguito le diverse sigle utilizzate. Per quanto riguarda i documenti vengono utilizzate le sigle:

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

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

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

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 vengono inseriti come immagini.

3.1.10.3 Tabelle

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

3.1.11 Strumenti

3.1.11.1 \LaTeX

Questo strumento è stato scelto dal gruppo per far sí 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 contemporaneamente.

3.1.12 Metriche

3.1.12.1 Indice Gulpease_G

Quest'indice riporta il grado di leggibilità di un documento, viene utilizzato 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 degli errori ortografici viene utilizzato lo strumento da linea di comando `AspellG`.

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 cloud *GitHub*, basato sul sistema di versionamento *Git*.

3.2.4.3 Repository

Il gruppo ha creato per il progetto un'organizzazione su *GitHub* e ha provveduto a creare due 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_G* è così strutturato:

- **Documenti esterni:** questa directory contiene tutti i documenti destinati ad uso esterno, ovvero indirizzati al proponente e ai committenti. All'interno della cartella ogni documento si trova nell'omonima directory, la quale a sua volta segue le regole di struttura descritte nel paragrafo §3.1.7;
- **Documenti interni:** questa directory contiene tutti i documenti destinati ad uso interno ed ha una struttura analoga a quella dei documenti esterni;
- **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 definiscono il 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**: file sorgenti L^AT_EX che il gruppo contenenti i contenuti dei documenti;
- File **.pdf**: sono i risultati della compilazione dei file .tex e rappresentano i documenti veri e propri;
- 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 presenta un branch *main* e diversi branch per la stesura dei singoli documenti. Per lavorare su un documento si eseguono i seguenti passi:

- Posizionarsi sul repository locale;
- Aprire il terminale;
- Spostarsi sul branch in cui si vuole lavorare con il comando `git checkout` seguito dal nome del branch;
- Eseguire il comando `git pull` per sincronizzare eventuali cambiamenti del repository remoto;
- Effettuare le modifiche;
- Sul terminale, eseguire il comando `git add` seguito dal nome 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 *main*, per i quali serve aprire una pull request_G che deve poi essere approvata.

Nel caso ci si trovi nella situazione di dover modificare un documento già approvato 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 la 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 obiettivi dei controlli di qualità;
- **Esecuzione:** Vengono messe in pratica le regole prefissate nella pianificazione;
- **Valutazione:** Confronto dei valori ottenuti con quelli attesi.

Se i valori ottenuti non sono quelli specificati nelle norme il gruppo si impegna ad apportare le modifiche necessarie affinché 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 successivamente ulteriori metriche, nel caso si ritenesse necessario.



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 imposte, relative all'istanziazione dei processi, sono le seguenti:

- Ogni processo deve avere un solo obiettivo;
- L'obiettivo di un processo non deve sovrapporsi agli obiettivi 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 sì 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. Essa può avvenire sia manualmente (e.g. controllo sul lessico utilizzato in un documento) che automaticamente (e.g. controllo sugli errori ortografici).

3.4.4.1.2 Analisi dinamica

Questo tipo di analisi, applicabile al solo prodotto software, prevede la sua esecuzione e viene effettuata tramite test.



3.4.4.2 Test

I test sono il cuore dell'analisi dinamica e il loro scopo è mostrare che il prodotto software funzioni 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 ne viene fornita una descrizione ed uno stato. I valori che quest'ultimo può assumere sono i seguenti:

- 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, questi 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 sia capace di resistere agli errori, la formula adottata è:

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

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

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 approvarlo o se rifiutarlo chiedendone 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 il ciclo di vita del progetto. 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 incontrarsi 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: *Piccoli Gregorio* rappresentante dell'azienda *Zucchetti S.p.A.*;
- Committenti: prof. *Vardanega Tullio* e prof. *Cardin Riccardo*;
- *SpaghettiCode*: tutti i membri del gruppo;
- Altri gruppi: 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 poter discutere di argomenti inerenti alle attività in corso. Il gruppo di Telegram, invece, viene usato per comunicazioni molto informali e più istantanee.

4.1.2.2 Esterna

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

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 messaggistica istantanea e VoIP.



4.2 Pianificazione

4.2.1 Scopo

Nei prossimi paragrafi verrà descritto come il gruppo *SpaghettiCode* intende pianificare il lavoro; verranno elencati i ruoli con i loro relativi compiti e l'assegnazione di essi tra i membri del gruppo. Prendendo come esempio 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 documenti;
- Approvare l'offerta del committente.

4.2.2.2 Amministratore

L'**amministratore**_G è il 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;
- Redigere ed attuare i piani e le 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. Possono esserci contemporaneamente più di un analista e il loro lavoro ha un grande impatto sulla riuscita del progetto. Generalmente sono pochi e non seguono il progetto fino al suo completamento.

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 l'attività di progettazione, spesso assumendo anche responsabilità di scelta e gestione. Possono esserci contemporaneamente più progettisti, ma sono comunque pochi e seguono 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, 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 capacità di giudizio e relazione. Si occupa di attività di verifica e validazione, partecipa all'intero ciclo di vita del progetto 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 diversi ruoli verranno assegnati mediante rotazione a ciascun raggiungimento di una scadenza terminale. Ogni membro dovrà ricoprire più ruoli durante l'intero ciclo di vita del progetto, 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 nella 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 quanto è stato fatto e di quanto resta ancora da fare si è deciso di usare il sistema di *Issue Tracking* offerto da *GitHub*. Ogni membro potrà creare delle *Issue* relative al compito 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* 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 il relativo compito. In questo modo ogni membro del gruppo potrà vedere se ci sono attività che necessitano di maggior attenzione e/o risorse 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 nei canali di comunicazione ufficiali.

4.3.3 Aspettative

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

- L^AT_EX;
- Git e GitHub;
- JavaScript;
- Libreria D3.js.