Norme di Progetto

Dream Corp.

04-12-2018

G&B

Informazioni sul documento	
Versione	Versione TBD
Responsabile	TBD
Redattori	TBD
Verificatori	TBD
Uso	Interno
Destinatari	Dream Corp.
	Zucchetti SpA
	Prof. Tullio Vardanega
	Prof. Riccardo Cardin

Versione	Data	Descrizione	Autore	Ruolo
1.0.0	10/12/2018	Approvazione documento	Pietro Casotto	Responsabile
0.5.3	8/12/2018	Verifica processi organizzativi	Davide Liu	Verificatore
0.5.2	7/12/2018	Verifica processi supporto	Marco Davanzo	Verificatore
0.5.1	5/12/2018	Verifica introduzione e processi primari	Davide Liu	Verificatore
0.5.0	3/12/2018	Scrittura processi organizzativi	Gianluca Pegoraro	Amministratore
0.4.0	28/11/2018	Scrittura processi primari	Matteo Bordin	Amministratore
0.3.0	26/11/2018	Scrittura processi di supporto	Gianluca Pegoraro	Amministratore
0.2.0	23/11/2018	Scrittura sottosezione ruoli di progetto	Gianluca Pegoraro	Amministratore
0.1.0	22/11/2018	Scrittura introduzione	Matteo Bordin	Amministratore
0.0.1	21/11/2018	Creazione struttura del documento	Matteo Bordin	Amministratore

Indice

Elenco delle tabelle

Elenco delle figure

Informazioni sul documento

1 Introduzione

1.1 Scopo del documento

Questo documento si prefigge lo scopo di garantire a tutti i membri del gruppo un modo comune di lavorare al fine di aumentare l'efficenza$_{\mathbf{G}}$. Verranno descritte le scelte architetturali e i vari software scelti.

1.2 Il prodotto

Il prodotto ha lo scopo di fornire un sistema "smart" di monitoraggio dei sistemi in modo da garantire e migliorare i servizi erogati dall'azienda ai terzi. L'applicativo sarà un estensione scritta in Javascript$_{\mathbf{G}}$per il software Grafana$_{\mathbf{G}}$, si lavorerà inoltre con le reti bayesiane$_{\mathbf{G}}$.

1.3 Glossario

Data la presenza di diversi elementi con significato ambiguo è stato necessario l'utilizzo di un glossario volto a disambiguare tali elementi col loro preciso significato. Questi termini verranno contrassegnati all'interno dei documenti con la lettera **G** a pedice e in grassetto.

1.4 Riferimenti

1.4.1 Normativi

- Standard ISO/IEC 12207:1995
  https://www.math.unipd.it/~tullio/IS-1/2009/Approfondimenti/ISO_12207-1995.pdf
- Capitolato C3
  https://www.math.unipd.it/~tullio/IS-1/2018/Progetto/C3.pdf

1.4.2 Informativi

- Piano di Progetto v. 1.0.0;
- Piano di Qualifica v. 1.0.0;
- GitHub$_{\mathbf{G}}$;

- Javascript$_{\mathbf{G}}$;
- Slide del corso Ingegneria del Software
  https://www.math.unipd.it/~tullio/IS-1/2018/

2 Processi primari

2.1 Accordo di fornitura

In questo paragrafo vengono documentate le norme che i membri devono seguire affinché il gruppo possa diventare committente dei professori Vardanega e Cardin ed essere fornitori dell'azienda Zucchetti$_{\mathbf{G}}$.

2.1.1 Studio di fattibilità

Dopo la presentazione dei capitolati il gruppo si è riunito per discutere la scelta più consona. Dopo aver risolto i dubbi interni ed aver redatto lo studio di fattibilità, documento, ad oggi *in versione 1.0.0*, atto a valutare i pro e i contro di ogni progetto permettendo così una più attenta valutazione, si è optato per la scelta del capitolato *numero 3*.
I punti chiave dell'analisi sono i seguenti:

- **Introduzione:** Viene fatta una breve introduzione del contesto in cui applicare la soluzione;
- **Finalità:** Viene descritto in modo sintetico lo scopo finale da raggiungere a lavoro completato;
- **Tecnologie in uso:** Si descrivono genericamente i software che la proponente intende usare;
- **Conclusioni:** Rappresenta il motivo per cui un capitolato è stato scelto oppure scartato.

2.1.2 Documentazione fornita

Al fine di assicurare massima trasparenza e qualità alla proponente ed ai committenti verranno elencati i documenti forniti con una breve descrizione del loro contenuto:

- **Piano di progetto:** descrive la pianificazione, la consegna e il suo completamento;
- **Analisi dei requisiti:** viene definita l'analisi dei casi d'uso e dei requisiti del gruppo
- **Piano di qualifica:** verifica, validazione e garanzia della qualità dei processi e di prodotto.

2.2 Sviluppo

2.2.1 Analisi dei requisiti

L'Analisi dei Requisiti viene scritta dagli Analisti che hanno il compito di valutare in modo accurato ogni aspetto del progetto. In particolare questo documento è redatto con lo scopo di:

- Descrivere scopo e funzionalità del prodotto;
- Fornire i requisiti e vincoli concordati col cliente;
- Descrivere gli elementi principali che hanno un ruolo chiave nello sviluppo del prodotto;
- Definire tutti i casi d'uso;
- Tracciare in modo dettagliato tutti i requisiti.

L'Analisi dei Requisiti seguirà le specifiche descritte in seguito.

Classificazione casi d'uso Sono elencati in ordine dal più generico al più dettagliato ed è stato scelto il seguente criterio per la loro classificazione:

UCX.Y

- **Codice X:** E' il codice identificativo del caso d'uso generico che potrebbe suddividersi in casi d'uso più specifici. Nel caso non ci siano questi ultimi, il codice risulta univoco.
- **Codice Y:** E' un codice identificativo univoco per il caso d'uso. E' presente solo nel caso in cui il caso d'uso UCX abbia dei sotto casi d'uso più specifici.

X e Y sono numeri progressivi che stanno a indicare la specificità all'interno dei casi d'uso. Ogni caso d'uso è inoltre definito secondo la seguente struttura:

- **ID:** il codice del caso d'uso secondo la convenzione specificata poco sopra;
- **Nome:** titolo del caso d'uso;
- **Descrizione:** breve descrizione del caso d'uso;

- **Precondizione:** condizioni assunte come vere prima del verificarsi degli eventi del caso d'uso;
- **Postcondizione:** condizioni assunte come vere dopo il verificarsi degli eventi del caso d'uso;
- **Attori:** attori principali e secondari (se presenti) del caso d'uso;
- **Scenario Principale:** flusso degli eventi rappresentato attraverso una lista numerata.

Nella *Figura 1* viene riportato un esempio di caso d'uso:

5.3.2 UC1.1: Inserimento della definizione della rete bayesiana sotto forma di file .json

Descrizione: L'utente inserisce la definizione della rete bayesiana sotto forma di file .json all'interno di Grafana.
Precondizione: L'utente deve trovarsi nell'interfaccia principale e possedere un file .json contenente una definizione di rete bayesiana che vuole inserire.
Postcondizione: Viene inserita nell'applicativo la definizione di rete presente nel file .json.
Attore primario: Utente.
Contestualizzazione / Scenario principale:

1. L'utente preme il pulsante per l'upload del file .json contenente la definizione di rete
2. L'utente sceglie il file
3. Upload del file
4. La rete viene caricata

Estensioni:

1. C'è stato un problema con l'interpretazione della rete bayesiana **UC10.1**

Figura 1: Esempio di caso d'uso

Classificazione dei requisiti Tutti i requisiti ottenuti dopo una profonda analisi degli Analisti possono essere ricavati da tre diverse fonti:

- **Interno:** il requisito proviene da una decisione del gruppo DreamCorp, generalmente emersa durante un incontro e riportata in un verbale;
- **Capitolato:** il requisito proviene dalle richieste del capitolato;

- **Esterno:** il requisito proviene da un incontro con la proponente.

Il codice utilizzato per indicizzare univocamente i requisiti è il seguente:

R+(F|Q|V|P)+(C|O)+(X(.Y)*)

- **R:** Requisito;
- **F|Q|V|P:**
  - F: Requisito funzionale che descrive nel dettaglio i servizi che verranno forniti dal sistema agli attori;
  - Q: Requisito di qualità;
  - V: Requisito di vincolo;
  - P: Requisito prestazionale;
- **C|O:**
  - C: Compulsory (obbligatorio);
  - O: Optional (opzionale);
- **X.Y:** Numeri naturali concatenati con un punto per descrivere un sottorequisito.

I requisiti di vincolo, di qualità e prestazionali fanno parte dei requisiti non funzionali che descrivono i vincoli sul sistema e sul suo processo di sviluppo. Ad ogni requisito verranno infine associate la sua priorità, una breve descrizione e le sue fonti come nella *Figura 2*.

Codice	Priorità	Descrizione	Fonte
RFC1	Compulsory	Inserimento della definizione della rete bayesiana	Capitolato
RFC1.1	Compulsory	Inserimento della definizione della rete bayesiana sotto forma di file .json	Interno

Figura 2: Esempio di requisito

Tracciamento Infine, per facilitare la lettura e la visualizzazione dei requisiti, questi verranno indicizzati in due modalità specifiche:

- **Tracciamento Priorità-Requisito:** il focus è orientato sulla priorità;
- **Tracciamento Tipologia-Requisito:**il focus è orientato sulla tipologia.

Per una lettura immediata non sono riportate le descrizioni per le quali si rimanda alle sezioni apposite nel documento "Analisi dei Requisiti". Infine viene riportata una tabella riassuntiva che permette di avere un quadro generale della distribuzione dei requisiti.

2.2.2 Qualità di Processo

Per facilitare il tracciamento dei processi viene utilizzata una rappresentazione contratta formulata come segue:

PX

dove P sta per *processo* e X è un numero intero progressivo.

2.2.3 Qualità del software

Per qualità del software si intende la misura in cui un prodotto software soddisfa un certo numero di aspettative rispetto sia al suo funzionamento che alla struttura interna. I parametri verranno classificati in:

- **Interni:** qualità percepita dagli sviluppatori;
- **Esterni:** qualità percepita dall'utente finale.

Al fine di rendere piu' facile il tracciamento dei parametri viene utilizzata una codifica formulata come segue:

- **IX:** parametri interni;
- **EY:** parametri esterni.

Dove X e Y sono numeri interi progressivi indipendenti.

2.2.4 Progettazione

L'attività di Progettazione consiste nel descrivere una soluzione al problema che sia soddisfacente per tutti gli stakeholders$_{\mathbf{G}}$. Ciò serve a garantire che il prodotto sviluppato soddisfi le qualità, le proprietà e i bisogni nell'attività di analisi permettendo cosi di:

- Garantire la qualità del prodotto;
- Ripartire il problema originale in maniera ricorsiva facilitando cosi la codifica delle componenti;
- Ottimizzare.

Uso di diagrammi Al fine di essere il più comprensibili possibile sarà necessario far uso su larga scala di diagrammi UML$_{\mathbf{G}}$2.0

2.2.5 Codifica

In questa sotto-sezione vengono elencate le norme alle quali i Programmatori devono attenersi durante l'attività di programmazione ed implementazione. Ogni norma e' rappresentata da un paragrafo contenente un titolo, una breve descrizione e, se necessario, un esempio esplicativo. Alcune di esse possono anche contenere una lista di possibili eccezioni d'uso. L'uso di norme e convenzioni e' fondamentale per permettere la generazione di codice leggibile e uniforme, agevolare le fasi di manutenzione, verifica e validazione e migliorare la qualità di prodotto.

Indentazione 1: I blocchi innestati devono essere correttamente indentati usando quattro spazi per ciascun livello.
Esempio:

```
for (int i=0; i<100; i++){
    var++; //SI!
}

for (int i=0; i<100; i++){
var++; //NO!
}
```

Figura 3: Indentazione 1

Indentazione 2: E' vietato l'uso di tabulazioni che possono non essere uniformi tra diversi editor o IDE. Al loro posto vanno usati gli opportuni spazi.

Indentazione 3: Il codice che fa parte di un blocco deve essere innestato allo stesso livello di quel blocco.
Esempio:

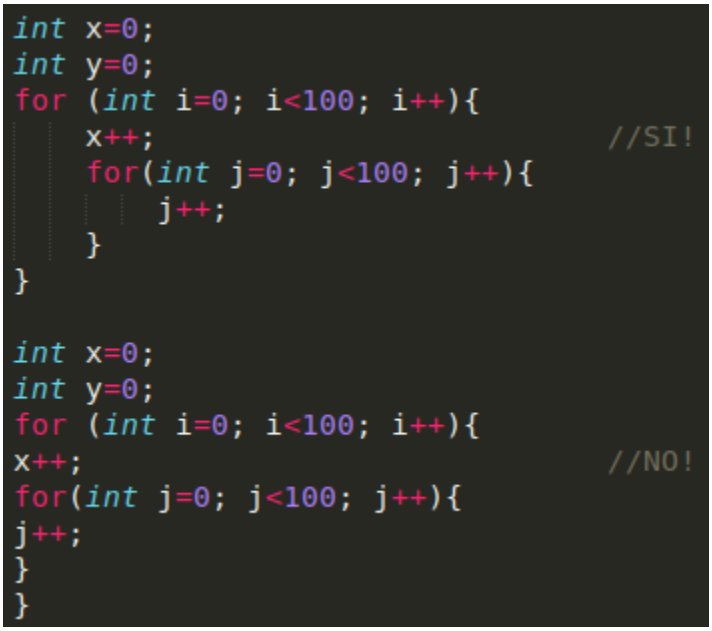

Figura 4: Indentazione 3

Parentesizzazione 1: I blocchi di codice sono sempre racchiusi fra parentesi graffe.
Eccezione: se il blocco e' composto da un'unica istruzione le parentesi possono essere omesse.

Parentesizzazione 2: Le parentesi graffe iniziano sempre nella stessa linea del codice.

Nomenclatura: Classi, metodi e variabili devono avere un nome univoco e quanto piu' descrittivo possibile. Inoltre devono rispettare le buone norme di scrittura in modo da distinguere immediatamente se si tratta di un nome di una classe, un metodo o una variabile. Nello specifico:

- **Classi:** cominciano sempre con la lettera maiuscola;
- **Metodi:** cominciano sempre con una lettera minuscola e se sono composti da più parole le successive iniziano con una lettera maiuscola;
- **Variabili:** tutte le lettere sono minuscole ed e' consentito l'utilizzo del simbolo underscore "_".

La lingua utilizzata deve essere l'inglese.

TODO: Un commento TODO va utilizzato per descrivere codice non definitivo, migliorabile e per soluzioni a breve termine. Può essere anche usato per eventi futuri specificando accuratamente la data.
Esempio:

```
// TODO: Improve after the ValueList has been checked in
// TODO: Remove the constant and add an Integer
// TODO: Fix by January 11
```

Figura 5: Esempi di TODO

Visibilità: Le variabili vanno dichiarate strettamente dove sono necessarie, ovvero nel blocco più interno che comprenda il loro utilizzo. Le variabili di ciclo vanno dichiarate nello statement a meno di una motivazione valida e opportunamente giustificata per non fare ciò. Le variabili globali vanno evitate in tutti i casi.

Inizializzazione: Le variabili vanno inizializzate il prima possibile, nel migliore dei casi subito dopo la loro dichiarazione.

Lunghezza metodi e codice: E' da evitare una lunghezza eccessiva sia nei metodi che nelle righe di codice. Nel primo caso una giusta suddivisione rende più manutenibili i vari metodi e più chiaro il loro scopo, nel secondo rende il codice più scorrevole

3 Processi di Supporto

3.1 Documentazione

3.1.1 Descrizione

In questo capitolo sono presenti le norme adottate per redigere, verificare e approvare la documentazione ufficiale prodotta da DreamCorp. Tutti documenti sono elencati nella sezione *3.1.5* denominata "Lista documenti".

3.1.2 Divisione dei documenti

Per una maggiore formalità ogni documento è stato classificato in Interno o Esterno in base alle seguenti caratteristiche:

- **Interno:** ha utilità interna al team, ovvero contiene tutte le informazioni significative per componenti del gruppo in fase di sviluppo;
- **Esterno:** condiviso anche con i Committenti e la Proponente, espone le informazioni utili per spiegare il metodo di lavoro seguito per lo sviluppo del progetto.

3.1.3 Nomenclatura

I documenti formali seguono uno standard preciso per la loro nomenclatura. Di seguito un esempio esplicativo:

DocumentoDiEsempio.for

- **DocumentoDiEsempio:** indica il nome del documento che viene scritto utilizzando la lettera maiuscola per ogni parola presente senza spazi al suo interno;
- **.for:** indica il formato del documento. Quest'ultimo sarà *.tex* dalla fase di creazione fino alla sua approvazione e da quel momento in poi verrà creato il file *.pdf* contenente la versione definitiva di quel documento.

Codice per il versionamento La versione del documento e' specificata all'interno del documento stesso in una tabella dove sono segnate in modo preciso tutte le modifiche fatte fino ad arrivare alla major release in formato PDF. Il sistema utilizzato verrà spiegato in modo più dettagliato successivamente nella sezione *3.2.2* di questo documento.

3.1.4 Ciclo di vita documentazione

Per ogni documento formale sono previste tre fasi obbligatorie:

- **Redazione:** questa fase dura dalla creazione del documento fino alla scrittura completa di tutti i contenuti previsti. Durante questo processo, il Responsabile di Progetto assegna ai Redattori i contenuti da aggiungere e una volta che quest'ultimi saranno esauriti potrà approvare il passaggio alla fase successiva;
- **Verifica:** in questa fase il documento viene controllato in ogni sua parte attraverso le opportune procedure dai Verificatori che una volta ultimato il lavoro daranno il loro feedback al Responsabile di Progetto; questo può dunque approvare il documento che passa quindi alla fase successiva oppure può farlo tornare alla fase di Redazione per correggere eventuali imperfezioni riscontrate dai Verificatori;
- **Approvato:** In questa fase il documento e' pronto per essere rilasciato e la sua versione viene aggiornata ad una major.

3.1.5 Lista documenti

- **Analisi dei Requisiti:** [Esterno]
  Il suo scopo è di analizzare ed esporre i requisiti del progetto. Contiene l'analisi dei casi d'uso che riguardano il prodotto in fase di sviluppo e i diagrammi di interazione con l'utente a prodotto ultimato;
- **Norme di Progetto:** [Interno]
  Contiene tutte le regole e le convenzioni adottate dai membri del gruppo DreamCorp per uno sviluppo metodico del progetto;
- **Piano di Progetto:** [Esterno]
  Il suo scopo è quello di descrivere gli obiettivi del progetto e gli elementi necessari per il loro raggiungimento. Inoltre vien presentato come il gruppo DreamCorp amministra il proprio tempo e i membri al suo interno;
- **Piano di Qualifica:** [Esterno]
  Consiste in una spiegazione dettagliata di come il team DreamCorp vuole soddisfare i requisiti di qualità del progetto;
- **Studio di Fattibilità:** [Interno]
  Il suo obiettivo è di mostrare come è stato analizzato ogni capitolato elencando per ognuno punti a favore e sfavore in modo da evidenziare tutti i dubbi sorti in fase di decisione.

- **Glossario:** [Esterno]
  Contiene tutti i termini presenti nei documenti formali che il gruppo DreamCorp ha ritenuto opportuno avessero bisogno di una spiegazione o chiarimento per facilitarne la comprensione. E' unico per tutti i documenti.

3.1.6 Norme tipografiche

Le norme scritte in questa sezione devono essere seguite in tutti i documenti prodotti dal team DreamCorp:

- **Date:** sono scritte seguendo la regola anno-mese-giorno (DD-MM-YYYY);
- **Voci nel Glossario:** per segnalare una parola che si trova nel glossario viene posta una **G** a pedice della parola;
- **Link interni:** i collegamenti che rimandano a una sezione interna al documento sono scritti in corsivo e sottolineati;
- **Link esterni:** i collegamenti che rimandano a una pagina web esterna al documento sono scritti in colore blu;
- **Elenchi puntati:** ogni elemento della lista è caratterizzato in generale da una prima parola o serie di parole in grassetto seguite da due punti *":"* e successivamente dal suo contenuto, ma può anche essere formato solamente dal contenuto nei casi in cui non sia necessaria una particolare specifica iniziale. Alla fine è posto un punto e virgola *";"* tranne per l'ultimo elemento che è concluso da un punto *"."*;
- **Citazioni:** le citazioni sono scritte in *corsivo*.

3.1.7 Struttura dei documenti

Tutti i documenti utilizzano una struttura di base contenuta nel file *stile.tex* in modo da non ripetere ogni volta gli elementi comuni.

Frontespizio E' la prima pagina del documento dove si trovano:

- **Logo:** immagine identificativa del gruppo;
- **Nome del progetto:** G&B;
- **Nome del documento**;

- **Data:** data in cui è stato approvata la versione corrente in cui è stato approvato il documento;
- **Versione:** versione corrente del documento;
- **Nome del gruppo:** DreamCorp.

In seguito in forma tabellare sono riportati:

- **Responsabile:** persona a capo del progetto nel momento in cui è stato approvato;
- **Redattori:** persone incaricate nella stesura del documento;
- **Verificatori:** persone incaricate di controllare la qualità del documento;
- **Uso:** Interno o Esterno;
- **Destinatari:** persone alle quali è destinate il documento.

Diario delle modifiche Si trova a partire dalla seconda pagina del documento e riporta sotto forma di tabella tutte le modifiche apportate al documento. La tabella è composta da quattro colonne che contengono rispettivamente descrizione della modifica, autore della modifica, data della modifica e versione a cui è arrivato il documento.

Indice E' posto sempre dopo il Diario delle Modifiche e contiene l'elenco dei capitoli e sottocapitoli in cui è diviso il documento.

Elenco delle Tabelle Contiene la lista delle tabelle presenti nel documento; è presente solo quando necessaria, ovvero se nel documento si trova almeno una tabella.

Elenco delle Figure Contiene la lista delle immagini presenti nel documento; è presente solo se nel documento c'è almeno un'immagine.

Contenuto Il documento è completato dal suo contenuto. Ogni pagina è composta come di seguito:

- **Logo del team:** in alto a sinistra nell'intestazione;
- **Capitolo:** in alto a destra nell'intestazione;
- **Contenuto:** occupa l'intera pagina;

- **Nome Documento:** in basso a sinistra nel piè di pagina;
- **Numero Pagina:** in basso a destra nel piè di pagina.

L'intestazione e il piè di pagina sono separate dal resto del documento attraverso due righe nere orizzontali. Ogni nuovo capitolo (identificato in LaTeX con il comando *\section*) comincia sempre in una nuova pagina.

3.1.8 Strumenti di supporto

Il formato scelto per la stesura della documentazione è il LaTeX che permette una maggiore precisione e uniformità fra tutti i membri del gruppo. L'ambiente di sviluppo scelto è **TeXstudio$_{\mathbf{G}}$** il quale è risultato il più completo in termini di funzioni e praticità.

3.2 Versionamento

E' stata scelta la tecnologia Git per le parti del progetto e della documentazione che necessitano di un versionamento e il servizio utilizzato è GitHub. Per tutte le altre parti lo strumento di condivisione di cui si serve il gruppo DreamCorp è una cartella su Google Drive$_{\mathbf{G}}$.

3.2.1 Comandi di base

In seguito vengono descritti i comandi principali di cui si è servito il team per l'uso di Git all'interno del progetto:

- **git clone:** crea una copia in locale del repository$_{\mathbf{G}}$;
- **git pull:** aggiorna la cartella in locale col contenuto del repository in remoto;
- **git checkout:** permette di cambiare branch$_{\mathbf{G}}$su cui si sta lavorando;
- **git add:** aggiunge uno o più file a seconda del comando alla lista dei file tracciati da Git. Aggiungendo il simbolo "." si possono aggiungere tutti i file modificati con un solo comando;
- **git commit -m "*Messaggio*":** fa il commit$_{\mathbf{G}}$delle modifiche effettuate in locale di tutti i file che sono stati aggiunti attraverso il comando *git add*. Il messaggio da inserire non è opzionale e deve descrivere lo scopo del commit;

- **git push:** carica in remoto le modifiche effettuate in locale;
- **git status:** permette di consultare lo stato del repository locale mostrando i file tracciati e non.

Inoltre, per migliorare il workflow generale è stato utilizzato GitFlow che permette una gestione più accurata dei branch.

3.2.2 Numerazione della versione

La versione di tutti i documenti è rappresentata nella forma X.Y.Z seguendo le seguenti regole:

- **X:** è il numero di una major release che avviene una volta che il documento passa allo stato di Approvato;
- **Y:** numero di una release normale nella quale sono state apportate modifiche o aggiunte sostanziali al documento;
- **Z:** è il numero di una minor release nella quale vengono apportate piccole modifiche o correzioni.

L'aumento di una cifra comporta l'azzeramento di quelle alla sua destra.

3.3 Verifica

3.3.1 Analisi statica

L'analisi statica della documentazione e del codice è divisa in due fasi, Walkthrough e Inspection:

- **Walkthrough:** questa fase consiste nella lettura di tutto il documento da esaminare e nella creazione di una lista degli errori trovati da utilizzare nella fase successiva.
- **Inspection:** questa fase consiste nella rilettura del documento attraverso la lista creata precedentemente per un'analisi mirata.

Gli strumenti utilizzati per l'Analisi statica si possono trovare nella sezione *3.1.8* di questo documento.

3.3.2 Analisi dinamica

Essendo applicato solo al codice del software, il processo di Analisi dinamica deve ancora essere definito in modo definitivo; in generale consisterà nella creazione e esecuzione di vari test.

3.3.3 Strumenti di verifica del software

Gli strumenti di verifica del software verranno definiti in fase di programmazione e codifica del software stesso.

3.4 Metriche per la Qualità di Processo

Verranno utilizzate le seguenti metriche per valutare l'efficienza e l'efficacia dei processi.

3.4.1 Schedule Variance (SV)

a

3.4.2 Budget Variance (BV)

a

3.4.3 Function Points

a

3.4.4 Code Coverage

a

- **Line coverage:** primitiva rispetto alle successive, fornisce un'idea generale
- **Functional coverage:**

- **Path coverage:**
- **Condition coverage:**
- **Branch coverage:**

3.4.5 Servizi esterni non raggiungibili

a

3.4.6 Rischi non calcolati

a

3.5 Metriche per la Qualità di Prodotto

Verranno utilizzate le seguenti metriche per valutare l'efficienza e l'efficacia dei prodotti.
Per i documenti sono usate le seguenti metriche:

3.5.1 Gulpease Index

a

3.5.2 Errori sintattici

a

3.5.3 Gunning Fog Index

a

3.5.4 Simple Measure Of Gobbledygook (SMOG)

a

Per il software sono usate le seguenti metriche:

3.5.5 Percentuale requisiti fondamentali soddisfatti

a

3.5.6 Percentuale requisiti opzionali soddisfatti

a

3.5.7 Mean time between failures (MTBF)

a

3.5.8 Blocco operazioni non corrette

a

3.5.9 Test conclusi in failure

a

3.5.10 Tempo di risposta

a

3.5.11 Impatto nuove aggiunte

a

4 Processi organizzativi

4.1 Comunicazione

In questa sezione vengono presentate le norme per la comunicazione tra le parti che aderiscono al progetto.

4.1.1 Comunicazioni interne

Vengono di seguito esposte le metodologie di comunicazione all'interno del team DreamCorp. Per le comunicazioni all'interno del gruppo viene utilizzato *Slack*$_{\mathbf{G}}$, un'applicazione di messaggistica multipiattaforma adatto per i gruppi di lavoro. Attraverso questo strumento è possibile suddividere lo spazio di comunicazione in canali tematici, ricercare messaggi, condividere file di grosse dimensioni e tracciare i messaggi.
All'interno di *Slack* sono stati predisposti dei canali tematici al fine di rendere più efficiente lo scambio di messaggi:

- **general**: Canale utilizzato per le comunicazioni e l'organizzazione di carattere generale;
- **github-notificaton**: Canale a cui è stato aggiunto un bot di GitHub il quale permette di inviare un messaggio nel momento in cui un membro del team apre/chiude issues, effettua un merge o una *push*;
- **automatismi**: Dove si discute dei software da utilizzare per creare automatismi;
- random: Canale generico dove discutere di tutto ciò che non è inerente al progetto.

Inoltre sono stati predisposti dei canali per la redazione dei documenti, uno per ogni documento, in modo da facilitare la discussione degli stessi:

- **norme_di_progetto**: Utilizzato per discutere delle norme di progetto da seguire durante lo sviluppo;
- **piano_progetto**: Canale riservato alla redazione del documento *Piano di progetto*;
- **studio_di_fattibilita**: Canale per la raccolta delle considerazioni sui capitolati e per la scrittura del documento;
- **piano_di_qualifica**: Utilizzato per discutere del raggiungimento della qualità e per l'organizzazione della redazione del documento;

- **analisi_requisiti**: Per discutere dei casi d'uso, requisiti e utilizzo del software per la redazione del documento *Analisi_dei_requisiti*;
- **glossario**: Dove viene confrontato il contenuto dei glossari personali.

I membri del team sono tenuti a comunicare rispettando i topic, nel caso si volesse portare all'attenzione tutti i membri del gruppo sono presenti i comandi:

- *@everyone* per notificare tutti i membri del gruppo;
- *@channel* per notificare i membri di un specifico canale.

Altre modalità di comunicazione previste sono:

- Comunicazioni orali informali: per parlare di qualsiasi problematica, strategia, utilizzo di strumenti, consigli, dubbi;
- Riunioni: Incontri di persona precedentemente organizzati.

4.1.2 Comunicazioni esterne

Questa sezione è inerente alle modalità di comunicazione da seguire con i membri esterni al gruppo di lavoro.
Allo scopo di garantire un costante miglioramento della qualità di prodotto, sono stati individuati le seguenti entità esterne:

- La proponente **Zucchetti S.p.a**, rappresentata da Gregorio Piccoli, board member e CTO dell'azienda, con il quale si vuole stabilire un rapporto di collaborazione per definire bisogni e requisiti per la realizzazione del prodotto;
- I Committenti **Prof. Tullio Vardanega** e **Prof. Riccardo Cardin**, ai quali verrà consegnata tutta la documentazione in ciascuna fase di revisione, con i quali si vuole stabilire un rapporto utile al miglioramento dei processi e delle strategie.

Comunicazioni esterne scritte Devono essere effettuate tassativamente attraverso l'uso dell'indirizzo mail del gruppo:

dreamcorp.swe@gmail.com

Ogni email ricevuta a questo indirizzo verrà inoltrata automaticamente alla casella postale di ogni membro del gruppo attraverso l'utilizzo dei filtri di Gmail$_{\mathbf{G}}$, inoltre il testo delle mail verrà riportate come messaggio su *Slack* attraverso l'uso di un bot.

Composizione email Viene di seguito presentata la modalità di scrittura delle email rivolte a soggetti esterni.

- **Email verso la Proponente Zucchetti S.p.a**: nel capitolato d'appalto viene fornita la mail del Dr. Gregorio Piccoli.

  gregorio.piccoli@zucchetti.it

  Non sono state date direttive per la composizione dei messaggi, il gruppo si impegna a mantenere un formalismo nella comunicazione con l'azienda proponente.

- **Email verso i Committenti**: ci si propone di utilizzare un oggetto che descriva in modo più accurato possibile il contenute del messaggio, ai committenti ci si rivolgerà con il Voi e con il Lei.

Nella possibilità in cui un messaggio debba essere mandato a più persone, nel campo "A:" va indicato il principale destinatario, mentre nel campo "Cc:" vanno indicati tutti gli altri. Viene posta particolare attenzione nella modalità di risposta e di inoltro delle email. In questi casi sono da utilizzare i tasti "Rispondi a tutti" e "Inoltra a" che formatteranno in automatico il corpo del messaggio includendo le clausole "Re:" e "I:".

4.2 Riunioni

In questa sottosezione vengono presentate le modalità di riunione, sia interne che esterne. Allo scopo di far rispettare l'ordine del giorno, redigere il *Verbale di riunione* e prendere nota degli argomenti trattati e delle decisioni prese, all'interno di ogni riunione dovrà essere nominato a turno tra i membri del gruppo DreamCorp un Segretario.

4.2.1 Verbali di riunione

Nello svolgimento delle riunioni è compito del Segretario redigere il documento *Verbale di riunione* secondo il seguente schema:

- **Verbale Esterno DATA**, incluso nel frontespizio. Per DATA si intende la data in cui è stata effettuata la riunione, scritta secondo le *Norme Tipografiche*;
- **Informazioni sulla riunione**: sezione contente:
  - **Motivo**: Descrizione sintetica del motivo che ha portato ad indirre una riunione;
  - **Luogo e Data**: ad esempio *Padova, Giovedì 23 Febbraio 2019*;

  - **Ora inizio**: nel formato ventiquattro ore;
  - **Ora fine**: nel formato ventiquattro ore;
  - **Partecipanti**: vengono elencati i partecipanti della riunione, iniziando dal Proponente/Committente nel caso la riunione sia esterna. Nel caso un membro sia costretto a lasciare la riunione prima del previsto, deve essere annotata l'ora di uscita e il motivo.

- **Ordine del giorno**: Un elenco puntato degli argomenti discussi;
- **Resoconto**: sezione contente le annotazioni circa le decisioni prese con motivazioni e gli argomenti discussi e non.

Nomenclatura I verbali esterni dovranno avere nome: VER-DATA
I verbali interni saranno nominati: VIR-DATA
Viene usata questa nomenclatura al fine di avere una facile consultazione.

Conservazione dei verbali I verbali interni ed esterni verranno redatti in LaTeXe caricati nel repository del progetto.

4.2.2 Riunioni interne

La partecipazione delle riunioni interne è permessa ai soli membri del gruppo DreamCorp.

Compiti del Responsabile di Progetto Di seguito vengono elencati i compiti che deve eseguire il Responsabile di Progetto:

- Fissare la data delle riunioni interne, previa discussione con i membri all'interno del canale *general* su *Slack*;
- Stabilire l'ordine del giorno;
- Valutare le richieste di riunione dei membri del gruppo e decidere se accettarle;
- Comunicare data e ordine del giorno attraverso il comando **@everyone** all'interno di *Slack* con almeno un giorno di anticipo;
- Verificare ed approvare il verbale;

- Confermare spostare od annullare le riunioni sempre con le modalità di notifica a tutti i membri del gruppo.

Compiti dei partecipanti

- Comunicare tempestivamente assenze e/o ritardi;
- Presentarsi puntualmente alle riunioni;
- Partecipare attivamente alle discussioni;
- Mantenere un atteggiamento educato.

Approvazione delle decisioni Le decisioni vengono approvate con la regola della maggioranza dei partecipanti alla riunione. Affinché una riunione sia valida devono essere presenti almeno cinque membri del gruppo.

4.2.3 Riunioni esterne

4.3 Ruoli di progetto

Nonostante il progetto sia collaborativo, all'interno del gruppo sono stati assegnati dei ruoli corrispondenti alle figure aziendali, essi sono:

- **Responsabile di progetto**;
- **Amministratore**;
- **Progettista**;
- **Verificatore**;
- **Programmatore**.

I ruoli verranno assegnati inizialmente verranno ruotati secondo un calendario prescritto nel documento *Piano di Progetto* Si cercherà evitare situazioni di conflitto d'interesse, come per esempio attuare la verifica di un documento da parte del redattore dello stesso.

4.3.1 Responsabile di Progetto

La figura di "Project Manager", o Responsabile di progetto, è responsabile unico dell'avvio, pianificazione, esecuzione, controllo e chiusura di un progetto facendo ricorso a tecniche e metodi di project management. Egli partecipa al progetto dalla sua fine.
I compiti principali del Project Manager sono:

- Elaborare la pianificazione del lavoro;
- Organizzare efficientemente ed efficacemente le risorse umane a sua disposizione;
- Favorire la comunicazione e l'affiatamento del team di progetto;
- Distribuire le risorse sulle attività e monitorarne lo svolgimento;
- Prendere tutte le iniziative volte a prevenire i rischi;
- Mantenere i contatti con gli utenti di riferimento e gli utenti finali pianificandone il coinvolgimento nelle varie attività del progetto;
- Produrre la documentazione di sua competenza e supervisionare quella prodotta dal team di progetto;
- Controllare la qualità dei prodotti parziali ed assicurarsi che gli standard di qualità adottati siano rispettati;
- Avere sempre un'attenzione particolare al miglioramento dei processi produttivi del progetto.

4.3.2 Amministratore

L'Amministratore è colui che controlla l'ambiente di lavoro e che si preclude l'obiettivo di aumentare la qualità del lavoro dando strumenti al gruppo di lavoro. I principali compiti dell'Amministratore sono:

- Amministrare le infrastrutture di supporto;
- Risolvere i problemi riguardo la gestione dei processi;
- Assicurare che la documentazione sia corretta, verificata, approvata, aggiornata e versionata;
- Controllare le versioni e configurazioni dei prodotti;
- Dare regole e procedure per lo svolgimento del lavoro.

4.3.3 Progettista

Il Progettista è colui che redige il Progetto, ha un proprio bagaglio culturale e una congrua esperienza. Egli:

- Concorre in prima persona allo sviluppo del progetto;
- Supervisiona lo sviluppo software;
- Controlla le attività svolte dal gruppo di lavoro;
- Forma il personale e soggetti esterni.

4.3.4 Verificatore

I verificatori sono figure con competenze tecniche, esperienza professionale e conoscenza delle norme. Il Verificatore è presente per tutto lo svolgimento del progetto. I compiti sono:

- Assicurare che le *Norme di Progetto* siano rispettate;
- Controllare che ogni stadio di vita del prodotto sia conforme al *Piano di Qualifica*;
- Segnala al Responsabile di Progetto eventuali situazioni di conflitto di interesse che violino la pianificazione stabilita nel *Piano di Progetto*.

4.3.5 Programmatore

Il Programmatore è una figura tecnica. Egli partecipa alla parte di codifica della realizzazione del prodotto. Egli:

- Scrive codice documentato e secondo le Norme di Progetto;
- Predispone le componenti di supporto atte a verificare e validare il codice prodotto attraverso dei test;
- Si occupa della stesura del *Manuale Utente*.

4.4 Formazione del personale

I membri del gruppo DreamCorp sono tenuti ad apprendere le tecnologie utilizzate per ogni ambito del progetto in modo autonomo:

- **Grafana**
- **JavaScript**
- **JsBayes**

4.5 Ambiente di lavoro

4.5.1 Coordinamento

Versionamento Per il versionamento si è scelto di utilizzare Git, soprattutto per il fatto che è stato utilizzato almeno una volta da tutti i componenti del gruppo. Ci si appoggia inoltre al sistema di hosting GitHub, che presenta un IssueTrackingSystem comodo ed efficace, oltre ad essere configurabile come bot si *Slack*.

Pianificazione Per la pianificazione delle risorse è stato utilizzato GanttProject. I motivi che ci hanno portato a questa scelta sono:

- Creare tasks e milestones;
- Creare baselines;
- Organizzare i task con una struttura analitica.

4.5.2 Documentazione

LaTeX Per la scrittura di documenti si è scelto di utilizzare LaTeX, un linguaggio di markup che permette di scrivere articoli ben formattati, oltre ad essere molto comodo per scrivere documenti in collaborazione con più persone, permettendo l'inclusione di più file all'interno come un vero e proprio linguaggio di programmazione.

Editor Come editor per la documentazione si è scelto di utilizzare TexStudio, software open source che permette svariate funzionalità.

4.5.3 Ambiente di sviluppo

Sistemi operativi I membri del gruppo sono autorizzati ad utilizzare i più importanti sistemi operativi poiché supportano gli strumenti per la realizzazione del progetto.

IDE Poiché il linguaggio da utilizzare è JavaScript, si è scelto di utilizzare come IDE WebStorm essendo quello riconosciuto più adatto al linguaggio in questione.

4.5.4 Ambiente di verifica

Documenti Per la verifica della correttezza della grammatica si è dovuto scaricare un pacchetto per TexStudio in modo da abilitare la correzione in lingua italiana.

4.5.5 Ticketing

Attraverso l'Issue Tracking System di GitHub è possibile, in modo semplice ed intuitivo, assegnare tasks ai membri del gruppo.

Struttura delle issues Le issues in GitHub hanno la seguente struttura:

- **Title**: Il titolo della issues che descrive il problema in generale;
- **Write**: La descrizione dettagliata del problema;
- **Assignes**: L'utente a cui è assegnato il compito di risolvere l'issues;
- **Labels**: Etichetta con cui specificare la natura del problema;
- **Projects**: Opzione che permette di assegnare una issue ad un progetto con relativa board;
- **Milestone**: La milestone a cui assegnare l'issue.