Norme di Progetto

A.A. 2022-2023

Componenti del gruppo:

Andrea Crocco, matr. 1226135
Elena Fabris, matr. 2008072
Sonia Franco, matr. 1224437
Andrea Stecca, matr. 2016104
Filippo Tonini, matr. 2008080
Enrico Zangrando, matr. 2000547
e-mail: *next.team.swe@gmail.com*

Università
degli Studi
di Padova

Registro delle modifiche

Versione	Descrizione	Verificatore	Autore	Data
2.0	Approvato per il rilascio	non richiesto	Andrea Stecca	2023-08-18
1.1	aggiunte le norme per la codifica	Sonia Franco	Enrico Zangrando	2023-04-10
1.0	approvato	non richiesto	Filippo Tonini	2023-03-21
0.5	concluso §3	Andrea Crocco	Sonia Franco	2023-02-19
0.4	aggiunto §4	Sonia Franco	Enrico Zangrando	2022-12-20
0.3	aggiunto §2	Sonia Franco	Filippo Tonini	2022-12-07
0.2	inizio stesura §3	Andrea Crocco	Andrea Stecca	2022-11-12
0.1	aggiunto §1	Andrea Crocco	Andrea Stecca	2022-11-11

Indice

1 Introduzione

1.1 Scopo del documento

In questo documento vengono definite le regole che ogni membro del gruppo deve seguire durante lo svolgimento dell'intero progetto. Queste regole servono a garantire un livello di omogeneità e coesione accettabile per il suddetto.

1.2 Descrizione del progetto

L'obiettivo del capitolato è la creazione di un sistema di illuminazione con rilevamento della presenza di persone attraverso l'utilizzo di *sensori*$_G$ e lo sviluppo di un'applicazione web *responsive*$_G$ in grado di monitorare un sistema di illuminazione pubblico. In particolare, il sistema deve essere in grado di:

- Rilevare la presenza di persone in prossimità della fonte luminosa
- Aumentare o ridurre l'intensità luminosa di un singolo lampione o di intere aree
- Rilevare automaticamente il guasto di un *impianto di illuminazione*$_G$
- Permettere la segnalazione manuale dei guasti di un impianto di illuminazione
- Permettere l'inserimento e la gestione di un impianto luminoso

1.3 Riferimenti

- Regolamento del progetto didattico:
  https://www.math.unipd.it/ tullio/IS-1/2022/Dispense/PD02.pdf
- I processi di ciclo di vita del software:
  https://www.math.unipd.it/ tullio/IS-1/2022/Dispense/T02.pdf
- Gestione di progetto:
  https://www.math.unipd.it/ tullio/IS-1/2022/Dispense/T04.pdf;
- Amministrazione di progetto:
  https://www.math.unipd.it/ tullio/IS-1/2022/Dispense/FC2.pdf
- ISO/IEC 12207;
- ISO/IEC 9126;

2 Processi primari

2.1 Fornitura

2.1.1 Scopo

Il processo fornitura ha lo scopo di definire e trattare le norme e i termini che i membri del gruppo *Next Team* sono tenuti a rispettare per rivestire adeguatamente il ruolo di fornitore nei confronti dell'azienda proponente *Imola Informatica* e dei committenti Prof. Tullio Vardanega e Prof. Riccardo Cardin.

2.1.2 Aspettative

Durante il progetto il gruppo intende instaurare con l'azienda *Imola Informatica* e in particolare con il referente Dr. Lorenzo Patera un rapporto di collaborazione per poter:

- Identificare le principali esigenze del cliente per garantire la loro soddisfazione;
- Identificare eventuali limitazioni o restrizioni sui requisiti e sui processi;
- Calcolare una stima dei costi necessari in termini di tempo e denaro;
- Accertarsi che il prodotto finale soddisfi le richieste del cliente e concordare i criteri di qualifica.

2.1.3 Attività

Di seguito le attività che compongono il processo di fornitura.

Valutazione dei capitolati

Al fine di decidere il capitolato per il quale candidarsi ogni componente del gruppo deve esprimere la proprie preferenze e motivare tali scelte. Le preferenze vengono poi confrontate per decidere il capitolato che più rispecchia gli interessi del gruppo.

Piano di Progetto

Il *Responsabile di Progetto*, con l'aiuto degli *Amministratori*, ha il compito di pianificare le attività necessarie alla guida del processo durante tutte le sue fasi, organizzando e distribuendo le risorse necessarie ai vari periodi di tempo. Il documento contenente tali informazioni è il *Piano di Progetto* che deve essere redatto prima dell'inizio del progetto e deve essere aggiornato periodicamente. Il *Piano di Progetto* deve contenere:

Analisi dei rischi: dove vengono elencati i possibili rischi che possono verificarsi durante il progetto e le relative azioni da intraprendere per mitigarli. I rischi sono identificati univocamente dal seguente codice: **R[Iniziale categoria][Numero progressivo]** con **R** che indica che si tratta di un rischio, **Iniziale categoria** che indica la categoria del rischio, in particolare:

- **T**: tecnologie scelte;
- **I**: rapporti interpersonali;
- **O**: organizzazione del lavoro;
- **R**: requisiti;

Pianificazione: dove vengono pianificate, a livello temporale, le attività da svolgere nel corso del progetto. La pianificazione deve essere divisa in macroperiodi che hanno inizio al seguito di una revisione (o aggiudicazione capitolato) e finiscono a quella dopo. Ogni macroperiodo è suddiviso in *sprint*$_G$ della durata media di 2 settimane. Per ogni sprint viene fatta una pianificazione (a priori) e una retrospettiva (a posteriori) per evidenziare eventuali problemi e miglioramenti per gli sprint futuri. **Pianificazione**

- **Attività**: Lista con breve descrizione delle attività da svolgere durante lo sprint
- **Obbiettivi**: Obbiettivi da raggiungere entro fine sprint
- **Gantt a priori**: *Diagramma di Gantt*$_G$ che mostra stime e collocazione temporale delle attività

- **Preventivo**: Stima di costi per lo sprint

Retrospettiva:

- **Obbiettivi raggiunti**: Obbiettivi raggiunti rispetto a quelli prefissati, ritardi e le loro cause
- **Gantt a posteriori**: Diagramma di Gantt simile al precedente che però mostra il tempo impiegato per le varie attività
- **Consuntivo**: Costo effettivo dello sprint
- **Note**: eventuali note migliorative

Preventivo del macroperiodo: dove viene stimata, sulla base della pianificazione, la quantità di lavoro necessaria al completamento del macroperiodo;

Consuntivo del macroperiodo: dove viene tracciato l'andamento rispetto a quanto preventivato.

Aggiornamenti:
Gli aggiornamenti al piano di progetto saranno fatti con frequenza pari agli sprint. Una volta che le attività dello spint sono terminate il Responsabile dovrà:
Pianificare il prossimos sprint:

- il responsabile determinerà i compiti e obiettivi per lo sprint
- aggiungerà i compiti in *YouTrack*$_{G}$ come *issues*$_{G}$
- assegnerà gli issues ai membri nella dashboard si YouTrack
- li collocherà temporalmente e creerà un diagramma di *Gantt*$_{G}$ per il periodo
- stimerà costi orari e ruoli impiegati per creare il preventivo di periodo
- aggiungerà la pianificazione al piano di progetto seguendo la struttura descritta sopra

Retrospettiva dello sprint giunto a termine

- il responsabile tramite cruscotto e feedback dei membri raccoglierà informazioni sullo sprint passato
- riporterà ore di lavoro e lo stato dei compiti in un Gantt aggiornato
- farà un resoconto delle ore e costi crando il consuntivo di periodo
- aggiungerà la retrospettiva al piano di progetto seguendo la struttura descritta sopra

Strumenti:
per svolgere tali compiti il responsabile userà Youtrack, per più informazioni sull'uso si veda la sezione dedicata §3.5.

Piano di qualifica

I membri del gruppo ricoprenti il ruolo di *Verificatore*, hanno il compito di scegliere le strategie da adottare per la verifica e la validazione del materiale prodotto, necessarie per garantirne la qualità. Il documento contente tali informazioni è il *Piano di Qualifica*, il quale espone:

- **Qualità di processo**: dove vengono stabilite le *metriche*$_{G}$ per il controllo della qualità del processo;
- **Qualità di prodotto**: dove vengono stabilite le metriche per il controllo della qualità del prodotto;
- **Specifica dei test**: dove vengono definiti i test da effettuare garantire il soddisfacimento dei requisiti;
- **Resoconto delle attività di verifica**: dove vengono rendicontate le attività di verifica effettuate;

Aggiornamenti: Gli aggiornamenti al resoconto di attività di verifica saranno fatti con frequenza pari agli sprint. Finite le attività di verifica dello sprint i verificatori dovranno:

- raccogliere gli esiti dei diversi compiti di verifica nell'apposito documento *Google Sheets*$_{G}$

- Aggiornare i grafici del resoconto delle attività di verifica
- aggiungere note ai grafici e riportare al gruppo segnalazioni degne di nota

I verificatori dovranno anche specificare e riportare i test da eseguire sul prodotto.
Strumenti: Verrà usato Google Sheets per il resoconto della attività di verifica, nell'apposito file condiviso i verificatori dovranno rimepire le tabelle già prdisposte con i dati sulla verifica e aggiornare i grafici del medesimo file.

Collaudo e rilascio del prodotto

In seguito al superamento del collaudo, il *Responsabile di Progetto* potrà consegnare al committente il consuntivo finale, il codice sorgente e la relativa documentazione. Non è prevista alcuna attività di manutenzione dopo il rilascio del prodotto.

2.2 Sviluppo

2.2.1 Scopo

Il processo di sviluppo ha lo scopo di definire i compiti e le attività che andranno svolti dal gruppo.

2.2.2 Aspettative

Una corretta implementazione di tale processo si basa sulle seguenti aspettative:

- Realizzazione di un prodotto finale:
  - Che sia conforme alle richieste del proponente;
  - Che superi i test di verifica descritti nel *Piano di Qualifica*;
  - Che superi i test di validazione descritti nel *Piano di Qualifica*;
- Fissaggio dei vincoli:
  - Tecnologici;
  - Di design.
- Fissaggio degli obiettivi di sviluppo.

2.2.3 Descrizione

Seguono le attività di cui si compone il processo di sviluppo secondo lo standard ISO/IEC 12207.

Analisi dei requisiti

Scopo Gli Analisti hanno il compito di individuare ed elencare in modo formale i requisiti del capitolato, i quali possono essere estrapolati da più fonti, ad esempio:

- Documenti di specifica del capitolato d'appalto;
- Confronto interno tra i membri del gruppo;
- Confronto esterno con il proponente;
- Casi d'uso.

Il documento contente tali informazioni è l'*Analisi dei Requisiti*, il quale deve esporre:

- **Descrizione generale del prodotto**: dove vengono definiti gli obiettivi del prodotto e i requisiti minimi e opzionali estrapolati dal capitolato d'appalto;
- **Casi d'uso**: dove vengono identificati i casi d'uso individuati sulla base delle potenziali funzionalità dell'applicativo;
- **Requisiti**: dove vengono elencati i requisiti individuati.

Aspettative L'obiettivo dell'analisi è individuare tutti i requisiti richiesti dal proponente.

Requisiti Ogni requisito è strutturato come segue:

- Codice identificativo;
- Descrizione;
- Fonte;

Il campo fonte indica il caso d'uso da cui esso deriva e se questo è di origine interna o del capitolato.

Codice del requisito Ogni requisito è identificato da un codice univoco, formato utilizzando il seguente schema:

R[Numero]-[Tipologia]-[Priorità]

Dove:

- **R**: indica che il codice identifica un requisito;
- **Numero**: numero progressivo che segue la struttura dei documenti;
- **Tipologia**: indica la tipologia del requisito, può essere:
  - **F**: funzionalità;
  - **Q**: qualità;
  - **V**: vincolo;
- **Priorità**: indica la priorità del requisito, può essere:
  - **0**: opzionale;
  - **1**: desiderabile;
  - **2**: obbligatorio;

Casi d'uso Ogni caso d'uso dovrà essere accompagnato da sottocasi, la scomposizione in sottocasi procede fino ad avere casi d'uso atomici, non ulteriormente scomponibili. In casi particolari è possibile evitare la scomposizione ove il costo supera il beneficio.
Ogni caso d'uso è strutturato come segue:

- Denominazione;
- Titolo;
- Diagramma UML_{G} (se necessario);
- Attore primario;
- Attori secondari (se presenti);
- Pre-condizioni;
- Post-condizioni;
- Scenario principale;
- Estensioni (se presenti);
- Generalizzazioni (se presenti);

Denominazione dei Casi d'uso Ogni caso d'uso è identificato da un codice univoco, formato utilizzando il seguente schema:

UC[Codice caso].[Codice sottocaso]

Modifiche casi d'uso e requisiti : Nell'aggiungere, rimuovere, modificare un caso d'uso o un requisito è necesario:

- confrontarsi con un altro analista (se non disponibile col responsabile) e accordarsi sulle modifiche.
- avvisare il responsabile
- nelle modifiche all'analisi dei requsiti seguire la struttura descritta sopra
- documentare le modifiche in *Google Sheets*$_G$

Strumenti : Per modifiche nell'analisi dei requisiti verranno usati: *starUML*$_G$ per la creazione dei diagrammi dei casi d'uso rispettando la sintassi UML; *Google Sheets*$_G$ per riportare il numero di aggiunte/rimozioni/modifiche dei casi d'uso a fini di reportistica.

Progettazione

Scopo L'attività di progettazione avviene ad opera dei *Progettisti*, i quali hanno il compito di definire le caratteristiche fondamentali del prodotto in funzione di quanto esposto nell'*Analisi dei requisiti*.

Aspettative L'obiettivo della progettazione è individuare l'architettura software più adatta, che dovrà:

Soddisfare tutti i requisiti individuati in fase di analisi e riportati nell'*Analisi dei Requisiti*;

Risultare comprensibile e modulare in modo da poter garantire una facile manutenzione;

Perseguire correttezza per costruzione in modo da garantire che il prodotto finale sia corretto.

Descrizione La presente attività si articola nelle seguenti sotto-attività:

- **Progettazione della Technology Baseline**: nella quale viene eseguita una prima analisi ad alto livello delle tecnologie che verranno coinvolte nello sviluppo del prodotto, la quale porta alla produzione del *Proof of Concept*$_G$ (PoC) e di una conseguente *Technology Baseline*.
- **Progettazione Architetturale**: nella quale verrà eseguita una definizione ad alto livello dell'architettura del prodotto e della sue componenti, insieme alla definizione dei test di integrazione;
- **Progettazione di dettaglio**: nella quale verrà eseguita una definizione di dettaglio dell'architettura del prodotto e della sue componenti, insieme alla definizione dei test di integrazione. Le informazioni ottenute in questa attività costituiranno la *Product Baseline*.

Technology Baseline La *Technology Baseline* ha come obiettivo la definizione e la motivazione delle tecnologie che verranno utilizzate nello sviluppo del prodotto. La *Technology Baseline* è composta da:

- Tecnologie adottate;
- Tracciamento delle componenti;
- Proof of Concept;

Product Baseline Definisce e illustra la baseline architetturale (design e coding) del prodotto, includendo:

- Design patterns;
- Definizione delle classi;
- Tracciamento delle classi;
- Diagrammi UML;
- Test di unità su ogni componente;

Specifica tecnica : L'attività di prgettazione verrà documentata nella *Specifica Tecnica*, che sarà divisa in:

- **Architettura** dove vengono descirtti strutture e pattern usati
- **Diagrammi delle classi** dove vengono commentate i diagrammi e le corrispettive classi
- **Requisiti implementati** dove vengono riportati i requsiti implementati

Progettazione di un nuovo componente : La progettazione avverrà tra gruppi da due progettisiti che dovranno:

- Usare best practices e pattern noti per garantire le funzionalità definite nell'*Analisi dei requisit*
- Procedere in modalità bottom up, parendo dalle signgole classi per poi arrivare al diagramma completato
- Documentare le proprie scelte e progressi aggiornando le opportune sezioni della specifica tecnica

Strumenti : Verrà usato *starUML*$_G$ per la creazione dei diagrammi dei casi d'uso.

Codifica

Scopo L'attività di codifica avviene ad opera dei *Programmatori*, i quali hanno il compito di implementare il prodotto in funzione di quanto esposto nella progettazione.

Aspettative L'obiettivo della codifica è l'implementazione di un prodotto che sia conforme alle richieste del proponente il cui codice sorgente risulti leggibile e facilmente comprensibile in modo da facilitarne la verifica, la validazione e la manutenzione.

Descrizione Di seguito verranno scritte le regole di scrittura del codice. Le regole scritte di seguito sono quelle che vengono consigliate in relazione alle tecnologie utilizzate.

Stile del codice scritto in Dart Nella documentazione di Dart si possono trovare delle regole da seguire per scrivere del codice pulito e consistente. Ecco le regole da seguire:

- I nomi dei tipi devono essere scritti in UpperCamelCase;
- I nomi di pacchetti, cartelle e file devono essere scritti in lowercase_with_underscore;
- I nomi degli identificatori che non sono tipi vanno scritti usando lowerCamelCase, anche per le costanti;
- Per formattare il codice va utilizzato il tool fornito con il dart *SDK*$_G$ che può essere richiamato tramite il comando *dart format* eseguito alla base del progetto;
- Le linee non devono superare gli 80 caratteri. Il formattatore di dart divide automaticamente le righe più lunghe ma non le stringhe per cui se una stringa fa eccedere i caratteri della riga oltre gli 80 andrà divisa manualmente;
- Utilizzare le parentesi graffe per tutte le istruzioni di controllo del flusso tranne nel caso in cui ci sia un *if* che ci sta in una riga e non sia seguito da un *else*;

Stile del codice scritto in Typescript Typescript non prevede delle regole ufficiali per lo stile del codice. Noi useremo le seguenti:

- I nomi dei tipi devono essere scritti in UpperCamelCase;
- I nomi degli identificatori che non sono tipi vanno scritti usando lowerCamelCase;
- Le costanti vanno scritte in UPPERCASE_WITH_UNDERSCORE;
- Andare a capo dopo aver aperto le parentesi graffe per eventuali funzioni, cicli o controlli;
- Usare i punti e virgola per terminare uno statement;

Aggiornamenti nel codice Il programmatore dovrà:

- scrivere codice solo a seguito della prgettazione e della specifica dei test del componente in questione.
- Seguire fedelemente i diagrammi UML prodotti in progettazione.
- Garantire il passaggio dei test maggiore o uguale alle iterazioni precedenti.
- Scrivere codice confome alle norme riportate sopra e garantire massima leggibilità.

Strumenti : I membri sono liberi di usare gli strumenti che preferiscono per la struttura del codice. Per la codivisione del codice verrà usato $GitHub_{G}$ e verranno seguite le norme descitte in §3.2.

3 Processi di supporto

3.1 Documentazione

3.1.1 Scopo

Questo processo è necessario per definire gli standard e gli strumenti necessari alla stesura di tutti i documenti del progetto.

3.1.2 Ciclo di vita del documento

Ogni documento segue le fasi del seguente ciclo di vita:

- **Redazione**: Viene fatta dai membri con il ruolo adeguato per la scrittura dei documenti (vedi sezione sui ruoli di progetto §4.1.4), la modifica o l'espansione del documento vengono assegnati ai membri tramite assegnazione di *issue*$_G$ su *YouTrack*$_G$;
- **Verifica**: il documento viene sottoposto ad una verifica tramite una *pull request*$_G$ per eseguire il *merge*$_G$ delle modifiche apportate dal redattore. Il verificatore corregge errori lievi come quelli sintattici o di battitura. In caso di errori logici nella stesura viene richiesta una correzione da parte dei redattori.
- **Approvazione**: superate le fasi di verifica, se ce ne sono in sospeso, il documento verrà inviato al responsabile di progetto che deciderà se approvarlo o meno per la pubblicazione.

3.1.3 Struttura dei documenti

Verrà ora definita una struttura generale che dovrà essere seguita nella produzione di tutti i documenti al fine di ottenere omogeneità e coesione.

Frontespizio Il frontespizio (prima pagina di ogni documento) è strutturato come segue, ogni elemento dovrà essere centrato orizzontalmente:

- **Logo del team**: il logo del team accompagnato da una didascalia contente il nome del gruppo;
- **Nome del documento**: scritto in grassetto;
- **Anno**;
- **Elenco dei componenti del gruppo**: la didascalia *Componenti del gruppo* seguita dall'elenco dei componenti, ognuno dei quali identificato da Nome Cognome, numero di matricola;
- **Recapito**: didascalia *email* seguita dall'indirizzo email del gruppo: next.team.swe@gmail.com;
- **Logo dell'università**;

Registro delle modifiche Tutti i documenti devono contenere un registro delle modifiche dopo la prima pagina. Tale registro deve essere aggiornato ogni qualvolta delle modifiche vengono verificate. Il *versionamento*$_G$ segue lo schema **a.v** dove:

- **a** rappresenta la versione approvata dal responsabile di progetto. È il valore più importante e permette la pubblicazione del documento.
- **v** rappresenta il livello di verifica del documento, viene incrementato ogni volta che una verifica termina con esito positivo.

Indice Tutti i documenti devono contenere un indice, per agevolare la consultazione e permettere una visita non sequenziale del documento stesso.

Contenuto principale Fatta eccezione per il frontespizio, tutte le pagine dei documenti devono contenere:

- **Intestazione**, composta da:
  - a sinistra: il logo del gruppo;
  - a destra: il nome del documento;
- Piè di pagina che contiene, al centro, il numero della pagina seguito dalle pagine totali: *Pagina corrente* di *Pagine totali*;

Verbali I verbali non seguono la struttura precedentemente definita per tutti gli altri documenti.

Frontespizio verbali il frontespizio dei verbali conterrà:

- **Logo**: il logo del gruppo con relativo nome;
- **Recapito**: *email*: l'indirizzo email del gruppo;
- **Data**: *Verbale* data del verbale;
- **Ruoli**: vengono indicati responsabile, redattore e verificatore;
- **Presenti**: l'elenco dei presenti all'incontro;

Al frontespizio seguono due sezioni:

- **Ordine del giorno**: in questa sezione vanno indicati i principali argomenti di discussione che sono stati affrontanti durante l'incontro;
- **Resoconto**: un resoconto che indica come sono stati affrontanti i vari argomenti e in che modo si è giunti ad una soluzione;

Glossario Il *Glossario* non segue il ciclo di vita degli altri documenti poiché redatto in maniera automatica. Per aggiungere una parola al glossario bisognerà utilizzare lo script contenuto nel *repository*$_G$ *glossary-updater* dell'organizzazione, le istruzioni per l'utilizzo sono contenute nel file README.md del suddetto repository. Per generare il documento bisognerà lanciare lo script *creaGlossario.py* presente all'interno del repository *Documents*.

Classificazione dei documenti I documenti redatti possono essere categorizzati come indicato di seguito:

- **Documenti ad uso interno**: tutti i documenti pensati per l'uso da parte di membri del gruppo, essi sono:
  - Tutti i verbali interni;
  - Le *Norme di progetto*;
- **Documenti ad uso esterno**: sono i documenti ad uso esterno quelli destinati ad essere divulgati al di fuori del gruppo, quindi:
  - Tutti i verbali esterni;
  - L'*Analisi dei requisiti*;
  - Il *Piano di progetto*;
  - Il *Piano di qualifica*;
  - Il *Glossario*;

3.1.4 Norme tipografiche

Di seguito le regole per la nomenclatura:

Tipo	Stile
Branch$_G$ di *GitHub*$_G$	snake_case
.tex e .pdf	snake_case
Cartelle	PascalCase
Date	YYYY-MM-GG
Voci del *Glossario*	*Parola*$_G$
Collegamenti ipertestuali	collegamento
Verbali	verbale-YYYY-MM-GG

3.2 Gestione del repository

Per il progetto viene utilizzato il sistema di controllo di versione distribuito *Git*$_G$, utilizzando nello specifico il servizio *GitHub*$_G$.

3.2.1 Struttura del repository

Per organizzare al meglio i contenuti si è deciso di utilizzare diversi repository:

- **Repository Documents**, viene utilizzato per salvare i file riguardanti i documenti, è un repository privato destinato ad uso esclusivamente interno.
- **Repository docs**, viene utilizzato per condividere con componenti esterni al gruppo i documenti redatti e approvati.
- **Repository *PoC*$_G$**, per il PoC vengono utilizzati i seguenti repository: poc_frontend (per la parte frontend), poc-server (per il server).
- **Repository lamp-simulator**, repository contenente il simulatore delle luci dato che quelle fisiche non saranno disponibili.
- **Repository glossary-updater**, repository contente lo script per aggiungere nuove parole al *Glossario*.

3.2.2 Gestione delle modifiche

All'interno del repository *Documents* è presente un *branch*$_G$ per ogni tipologia di documento.
Nei repository contenenti software andrà creato un branch per ogni feature che si svilupperà.
Nei branch di un documento sarà creato un ulteriore branch da parte del redattore (il nome sarà uguale al nome dell'*issue*$_G$ di *YouTrack*$_G$), in questo secondo branch saranno eseguite tutte le modifiche o espansioni. Una volta finito il compito specificato nell'issue il branch verrà chiuso con una *pull request*$_G$ che sarà revisionata e accettata del verificatore, sarà così eseguito il *merge*$_G$ con il branch del documento.
Per quanto riguarda l'approvazione e successivo rilascio verrà sempre usato il sistema delle pull request, effettuando un merge sul main branch del repository, mantenendo però aperti i branch dei vari documenti.

3.2.3 Tipi di file

All'interno del repository *Documents* saranno presenti solo i file .tex, .pdf e .png. I file generati automaticamente durante la compilazione dovranno essere esclusi dal tracciamento inserendo la relativa estensione nel file .gitignore.

3.3 Gestione della qualità

3.3.1 Scopo

L'obiettivo della gestione della qualità è quello di garantire che i processi e il prodotto soddisfino le richieste del proponente e che lo facciano con la migliore qualità possibile.

3.3.2 *Metriche*$_G$

All'interno del documento denominato *Piano di Qualifica* vengono utilizzate diverse metriche per fissare gli obiettivi di qualità di processi e prodotto.
In questa sezione ne verrà data una descrizione breve e verrà spiegato come utilizzarle per eseguire i test di qualità.

Ogni metrica è codificata come segue:

M [Tipologia] [ID numerico]

dove **[Tipologia]** può essere:

- **PC** che indica le metriche di processo
- **PD** che indica le metriche di prodotto

e **[ID numerico]** indica un numero univoco incrementale diverso per le due tipologie.

Metriche di processo

Per comprendere meglio la tabella sottostante, si fornisce la definizione delle seguenti variabili:

- **BAC**: *Budget At Completion*: costo iniziale previsto per la realizzazione del progetto.
- **NoC**: *Number of Changed*: numero dei requisiti variati.
- **NoD**: *Number of Deleted*: numero dei requisiti eliminati.
- **NoA**: *Number of Added*: numero dei requisiti aggiunti.
- **TNIR**: *Total Number of Initial Requirements*: numero totale dei requisiti iniziali.
- $\mathbf{N_f}$: numero di frasi.
- $\mathbf{N_l}$: numero di lettere.
- $\mathbf{N_p}$: numero di parole.
- **NQMS**: *Number of Quality Metrics Satisfied*: numero di metriche di qualità soddisfatte.
- **TQM**: *Total number of Quality Metrics*: numero totale di metriche di qualità.
- **LCE**: *Line of Code Executed*: linee di codice eseguite da algoritmi di test.
- **LC**: *Line of Code*: linee di codice totali.

Metrica	Codice	Descrizione	Formula
Actual Cost (AC)	**MPC3**	Soldi spesi per il progetto fino al momento del calcolo	
Estimate to complete (ETC)	**MPC7**	Valore stimato per la realizzazione delle rimanenti attività necessarie	

Estimated at Completion (EAC)	**MPC1**	Indica alla data della misurazione qual è la stima del costo finale che si sta prefigurando. Si tratta di una revisione del BAC	AC+ETC
Earned Value (EV)	**MPC5**	Valore del lavoro svolto fino al momento del calcolo	$(\%\ lavoro\ svolto) * EAC$
Planned Value (PV)	**MPC6**	Lavoro che si era pianificato di svolgere fino al momento del calcolo	$(\%\ lavoro\ pian) * BAC$
Budget Variance (BV)	**MPC2**	Indica se si stanno rispettando o meno i costi prestabiliti per i processi	$(\frac{EV}{AC} - 1) * 100$
Schedule Variance (SV)	**MPC4**	Stato di anticipo o ritardo rispetto alla pianificazione. Un valore negativo indica ritardo.	$EV - PV$
Requirements stability index (RSI)	**MPC8**	Indicano quanto i requisiti variano nel tempo	$(\frac{(1-NoC*NoD*NoA)}{TNIR}) * 100$
Satisfied obligatory requirements (SOR)	**MPC9**	Numero di requisiti obbligatori soddisfatti	
Indice Gulpease$_{\text{G}}$	**MPC10**	Indice di leggibilità del testo tarato sulla lingua italiana. Considera due variabili linguistiche: la lunghezza della parola e la lunghezza della frase rispetto al numero di lettere	$(89 + \frac{(300*N_f - 10*N_l)}{N_p})$
Correttezza ortografica	**MPC11**	Numero di errori grammaticali o ortografici per il documento	
Quality Metrics Satisfied (QMS)	**MPC12**	Percentuale di metriche di qualità soddisfatte	$\frac{NQMS}{TQM} * 100$
Code Coverage (CC)	**MPC13**	Descrive quanto il codice prodotto è coperto dalla suite di test dinamici	$\frac{LCE}{LC} * 100$
Passed Test Cases Percentage (PTCP)	**MPC14**	Indica il valore percentuale di test passati	$\frac{Test\ passati}{Test\ totali} * 100$

Failed Test Cases Percentage (FTCP)	**MPC15**	Indica il valore percentuale di test falliti	$\frac{Test\ falliti}{Test\ totali} * 100$

Tabella 3.1: Descrizione metriche di processo

Metriche di prodotto

Per comprendere meglio la tabella sottostante, si fornisce la definizione delle seguenti variabili:

- $\mathbf{N_{os}}$: Numero di requisiti obbligatori soddisfatti.
- $\mathbf{N_{tot}}$: Numero di requisiti totali. Nel caso di RC_{obb} si considerano solo i requisiti obbligatori.
- $\mathbf{N_s}$: Numero di requisiti soddisfatti.
- $\mathbf{R_c}$: Righe di codice commentate
- $\mathbf{R_{tot}}$: Righe di codice totali.
- $\mathbf{B_s}$: Numero di $browser_G$ supportati.
- $\mathbf{B_{tot}}$: Numero di browser totali.

Metrica	Codice	Descrizione	Formula
Requirements Coverage	**MPD1**	Indica la percentuale dei requisiti soddisfatti. Per il calcolo del valore accettabile si considerano solo i requisiti obbligatori.	$RC_{obb} = \frac{N_{os}}{N_{tot}} * 100$, $RC_{tot} = \frac{N_s}{N_{tot}} * 100$
Failure Density	**MPD2**	Indica l'affidabilità di un prodotto software. Si ricava dal rapporto tra i test eseguiti sul prodotto ed i test che esso ha fallito.	$\frac{Test\ falliti}{Test\ eseguiti} * 100$
Tempo medio di risposta	**MPD3**	Tempo impiegato dall'applicativo ad elaborare i dati e a fornire un risultato all'utente.	
Complessità ciclomatica	**MPD4**	Indica la complessità di un programma. La complessità ciclomatica di una sezione del codice sorgente è data dal numero di percorsi linearmente indipendenti al suo interno.	

Facilità apprendimento funzionalità	**MPD5**	Numero di minuti necessari all'utente per comprendere come utilizzare le funzionalità del prodotto.	
Comprensione del codice	**MPD6**	Percentuale di righe commentate sulle righe di codice totali.	$(\frac{R_c}{R_{tot}}) * 100$
Versioni di browser supportate	**MPD7**	Percentuale di browser supportati dal software	$(\frac{B_s}{B_{tot}}) * 100$

Tabella 3.2: Descrizione metriche di prodotto

3.4 Verifica e Approvazione

Verifica

Per ogni compito che viene svolto deve esserci una verifica di tale compito, essa avviene in contemporanea allo svolgimento del compito seguendo l'andamento da vicino. Solo per compiti particolarmente veloci è possibile eseguire la verifica una volta terminato il compito. Per questi motivi sia su *YouTrack*$_G$ che nel registro delle modifiche, i compiti svolti e la loro verifica saranno considerati come un unica unità.

Chi fa verifica La verifica è fatta dai verificatori, il membro che è stato incaricato di verificare una modifica o espansione di un documento non può mai essere il membro che ha attuato tale modifica o espansione.

Come fare verifica Il verificatore ha il compito di controllare che quanto prodotto sia conforme a regole prefissate. Dunque ha il compito di verificare che la qualità sia sufficientemente alta eseguendo i controlli descritti nel *Piano di qualifica*. Deve verificare inoltre che quanto scritto rispetti le *Norme di progetto*, che non ci siano incongruenze con il *Piano di progetto*, e che rispetti le decisioni prese nei verbali.
L'esito della verifica può essere:

- **positivo**, nel caso non ci siano problemi o siano velocemente risolvibili
- **negativo**, nel caso in cui ci siano problemi più grandi

In caso di esito negativo la modifica è respinta, viene riassegnata all'autore, i problemi vengono comunicati tramite i canali di comunicazione informali.

Verifica su *GitHub*$_G$ Una volta che il membro incaricato terminerà il compito aprirà una *pull request*$_G$. Il verificatore una volta finito il suo lavoro di verifica farà il *merge*$_G$ della pull request unendo il *branch*$_G$ delle modifiche fatte con quello base.

Verifica su YouTrack Il verificatore ha il compito di modificare lo stato dell'*issue*$_G$ che deve verificare: se il compito è ancora in corso lo stato sarà "In Progress", se il compito è completato ma la verifica no lo stato sarà "Verifying", se la verifica è completata lo stato sarà "Verified".
Nel caso in cui la modifica sia respinta il verificatore deve riassegnare l'issue all'autore originale e cambiare il suo stato in "To be fixed".

Approvazione

L'approvazione viene fatta dal responsabile nel momento in cui l'oggetto dell'approvazione deve essere presentato al proponente o committente. Il responsabile deve quindi apportare le modifiche del branch del documento in questione al branch main tramite merge e pubblicare l'oggetto accettato sulla *repository*$_G$ pubblica.

3.5 *Issue tracking system*$_G$

L'issue tracking system da usare è YouTrack, esso verrà usato per:

- Elencare e descrivere i compiti da svolgere sotto forma di *issue*$_G$
- Assegnare gli issue ai membri
- Pianificare lo svolgimento degli issue nel tempo tramite *diagrammi di Gantt*$_G$
- Monitorare lo stato degli issue, il carico di lavoro per ogni membro, l'andamento dello *sprint*$_G$ nel suo complesso
- Tracciamento delle ore di lavoro dei membri

L'issue tracking system mette a disposizione diverse viste e funzionalità che saranno usate nel seguente modo:

Creazione e modifica degli issue Gli issue saranno concordati dai membri del gruppo in incontri formali o comunicazioni informali. Il responsabile si occuperà di creare gli issue prima dell'inizio dello sprint.
Per ogni issue dovrà obbligatoriamente essere inserire:

- Un nome descrittivo per l'issue
- Una breve descrizione del compito
- Una data entro quale l'issue deve essere risolto
- Il tipo di issue
- Il progetto a cui appartiene
- Eventuali relazioni con altri issue

Durante lo svolgimento dei compiti i singoli membri hanno la responsabilità di aggiungere le ore ed il tipo di lavoro effettuato all'issue di cui si stanno occupando e di indicare i *commit*$_G$ corrispondenti. Le ore di lavoro verranno riportare a fine sprint dal responsabile nel *Piano di progetto*.

Agile *board*$_G$ La board sarà sempre strutturata come segue: nelle colonne gli stati che gli issue possono assumere, nelle righe i membri del team. Per ogni sprint verrà creata una board dal responsabile che selezionerà gli issue li inserirà nella board. Il responsabile assegnerà gli issue a inizio sprint ai membri tenendo in considerazione:

- il ruolo, quindi se il compito può essere svolto da quel ruolo (vedi ruoli di progetto §4.1.4)
- il carico di lavoro per ogni membro, il tempo previsto per gli issue non deve superare di più di 5 ore la media del tempo di lavoro degli altri membri
- impegni personali precedentemente dichiarati dai membri

Durante lo svolgimento dei compiti i singoli membri hanno la responsabilità di modificare lo stato degli issue di cui si stanno occupando.

***Diagrammi di Gantt*$_G$** Il responsabile dovrà creare un diagramma di Gantt per ogni sprint, al suo interno dovrà inserire gli issue dello sprint.
Nel pianificare lo sprint dovrà:

- stimare un numero di ore di lavoro per completare ogni issue
- mantenere un carico di lavoro costante per ogni membro, la differenza tra il numero massimo e minimo di issue svolti in contemporanea non deve essere più di 2
- mantenere un margine di almeno un giorno tra issue dipendenti uno dall'altro
- tenere conto di impegni personali precedentemente dichiarati dai membri

Il responsabile avrà anche il compito di riportare le informazioni del diagramma di Gantt sia all'inizio dello sprint sia alla fine.

4 Processi organizzativi

4.1 Gestione dei processi

4.1.1 Scopo

Lo scopo del processo è la redazione del *Piano di Progetto*, documento che ha il compito di indicare ai membri del gruppo l'organizzazione e la gestione dei ruoli.

4.1.2 Aspettative

Le principali aspettative del processo sono:

- Redazione del *Piano di Progetto*;
- Definizione dei ruoli assunti dai membri del gruppo;
- Definizione di un piano per l'esecuzione dei compiti programmati.

4.1.3 Descrizione

Le attività previste dal processo di gestione dei ruoli sono raccolte nel *Piano di Progetto*, documento redatto da parte del *Responsabile di Progetto*, con la collaborazione di uno o più *Amministratori*. Viene trattata, nello specifico, la gestione dei seguenti argomenti:

- Ruoli all'interno del progetto;
- Comunicazioni;
- Incontri;
- Strumenti di coordinamento;
- Strumenti di *versionamento*$_{G}$;
- Rischi;

4.1.4 Ruoli di progetto

Il *Responsabile di Progetto* ha il compito di suddividere i ruoli e l'assegnazione oraria per i membri del gruppo, garantendo che ognuno di essi assuma, nel corso del progetto, almeno una volta ogni ruolo. I ruoli richiesti dal progetto sono descritti di seguito.

Responsabile di Progetto È il punto di riferimento per il committente e per il fornitore facendo da intermediario tra i due. È sua la responsabilità delle scelte del gruppo che devono essere da lui approvate. Si occupa di:

- Approvare l'emissione della documentazione;
- Approvare l'offerta economica sottoposta al committente;
- Pianificare e coordinare le attività di progetto;
- Gestire le risorse umane;
- Valutare e gestire i rischi;

Amministratore di Progetto Figura professionale che si occupa del controllo e dell'amministrazione dell'ambiente di lavoro. Svolge le seguenti mansioni:

- Ricercare, studiare e mettere in opera risorse per migliorare l'ambiente di lavoro e automatizzarlo ove possibile;
- Risolvere problemi legati alla gestione dei processi;
- Salvaguardare la documentazione di progetto;
- Effettuare il controllo di versioni e configurazioni del prodotto software;
- Redigere e attuare i piani e le procedure per la gestione della qualità;
- Redigere le *Norme di Progetto*;

Analista È la figura professionale che possiede maggiori conoscenze riguardo il dominio applicativo del problema. In particolare, si occupa di:

- Studiare il problema e il relativo contesto applicativo;
- Comprendere il problema e definire la complessità e i requisiti;
- Redigere l'*Analisi dei Requisiti*;

Progettista È la figura professionale che si occupa della progettazione del prodotto software. In particolare, si occupa di:

- Effettuare scelte riguardanti gli aspetti tecnici e tecnologici del prodotto, favorendone l'efficacia e l'efficienza;
- Definire un'architettura del prodotto da sviluppare che miri all'economicità e alla manutenibilità;
- Redigere la *Specifica Tecnica* e la parte pragmatica del *Piano di Qualifica*;

Verificatore Il verificatore si occupa di controllare il lavoro svolto dagli altri membri del gruppo sulla base delle proprie competenze tecniche, esperienza e conoscenza delle norme. In particolare, deve:

- Esaminare i prodotti in fase di revisione, con l'ausilio delle tecniche e degli strumenti definiti nelle *Norme di Progetto*;
- Verificare la conformità dei prodotti ai requisiti funzionali e di qualità;
- Segnalare eventuali errori;

Programmatore Figura professionale che si occupa della codifica del progetto e delle componenti di supporto che serviranno per effettuare i test. Le sue mansioni sono:

- Implementare la *Specifica Tecnica* scritta dal Progettista;
- Scrivere un codice pulito e facile da mantenere che rispetti le *Norme di Progetto*;
- Realizzare gli strumenti per la verifica e la validazione del software;
- Redigere il *Manuale Utente* relativo alla propria codifica.

.

4.1.5 Procedure

Gestione delle comunicazioni

Comunicazioni interne Le comunicazioni interne sono quelle che avvengono tra i membri del gruppo, ed utilizzano i seguenti strumenti:

- Whatsapp utilizzato per la pianificazione degli incontri interni e per discussioni di carattere generale;
- *Discord*$_{G}$ utilizzato per lo svolgimento di riunioni tramite canali vocali e per la condivisione di file nei canali di testo;

Comunicazioni esterne Le comunicazioni esterne sono affidate al *Responsabile di Progetto*, che utilizza la casella di posta:

next.team.swe@gmail.com

Gestione degli incontri Gli incontri vengono organizzati dal *Responsabile di Progetto* il quale:

1. Sceglie in base alla disponibilità di tutti i membri del gruppo e di eventuali membri esterni un giorno e un orario per l'incontro;
2. Comunica a tutti i partecipanti la data e l'orario definitivi per l'incontro;
3. Incarica uno o due membri del gruppo per la redazione del verbale dell'incontro.

Incontri interni Gli incontri interni possono essere richiesti dai membri del gruppo al *Responsabile di Progetto*.

Incontri esterni Gli incontri esterni possono essere richiesti dai membri del gruppo, dal referente aziendale o dal committente al *Responsabile di Progetto*.

Verbali degli incontri In seguito ad ogni incontro, i membri del gruppo incaricati dal *Responsabile di Progetto* redigono il verbale dell'incontro.

Gestione degli strumenti di coordinamento

Assegnazione task Il *Responsabile di Progetto* ha il compito di suddividere il carico di lavoro in task da assegnare ai membri del gruppo utilizzando *YouTrack*$_{G}$. Tramite questo strumento è possibile visualizzare una dashboard contenente l'elenco di task. Per ogni task è possibile visualizzare:

- **Titolo** identificativo del task;
- **Descrizione** indica in dettaglio cosa deve essere fatto;
- **Assegnatari** persone che devono svolgere il task;
- **Tipologia** indica il tipo di task, ad esempio: *bug*, *feature*, *documentation*;
- **Scadenza** entro quando deve essere svolta la task;
- **Commenti** eventuali commenti lasciati dei membri del gruppo;

Gestione dei rischi Il responsabile di progetto ha il compito di monitorare i rischi indicati nel *Piano di Progetto* e se lo ritiene opportuno aggiungere nuovi possibili rischi e prevederne la gestione. Il processo si sviluppa come di seguito:

1. Verifica periodica dello stato dei rischi già presenti nel *Piano di Progetto*;
2. Individuazione dei problemi non considerati in precedenza;
3. Aggiunta al *Piano di Progetto* dei nuovi rischi individuati;
4. Se necessario ridefinizione della gestione di alcuni rischi.

Codifica dei rischi I rischi sono codificati secondo il seguente schema:

R[Iniziale categoria][Numero]

Dove:

- **Iniziale categoria**: indica la tipologia del rischio e appartiene ad una delle seguenti:
  - **T**: rischi collegati alle tecnologie scelte;
  - **I**: rischi correlati ai rapporti interpersonali;
  - **O**: rischi correlati all'organizzazione del progetto;
- **Numero**: indica il numero progressivo del rischio in una determinata categoria.