



Università degli Studi di Padova



BugPharma - *Login Warrior*

E-mail: bugpharma10@gmail.com

Norme di Progetto

Versione	2.0.0
Approvazione	Michele Masetto
Redazione	Andrea Salmaso, Nicla Faccioli Nicholas Sertori, Sara Nanni Michele Masetto, Lorenzo Piran
Verifica	Nicla Faccioli, Silvia Giro
Stato	Approvato
Uso	Interno
Distribuzione	Prof. Vardanega Tullio Prof. Cardin Riccardo Zucchetti S.p.A. Gruppo BugPharma

Registro delle modifiche

Versione	Data	Descrizione	Autore	Ruolo
2.0.0	20/06/2022	Approvazione del documento	Michele Masetto	Responsabile
1.1.0	18/06/2022	Revisione del documento	Nicla Faccioli, Silvia Giro	Verificatori
1.0.7	16/06/2022	Aggiornamento metriche rispetto a <i>Piano di Qualifica</i> e verifica	Andrea Salmaso, Nicla Faccioli	Amministratore, Verificatore
1.0.6	03/06/2022	Modifica sezione §3.4.6 e verifica	Nicla Faccioli, Lorenzo Piran	Amministratore, Verificatore
1.0.5	06/05/2022	Modifica sezione §3.4.4.3 e verifica	Lorenzo Piran, Silvia Giro	Amministratore, Verificatore
1.0.4	20/04/2022	Modifica sezione §2.2.4.3 e verifica	Michele Masetto, Sara Nanni	Amministratore, Verificatore
1.0.3	17/04/2022	Modifica sezione §3.1.2.2 (Piano di Progetto) e verifica	Andrea Salmaso, Sara Nanni	Amministratore, Verificatore
1.0.2	01/04/2022	Modifica sezione §3.2.4.1 e verifica	Nicla Faccioli, Sara Nanni	Amministratore, Verificatore
1.0.1	17/03/2022	Ristrutturazione generale del documento e verifica	Nicholas Sertori, Silvia Giro, Nicla Faccioli, Sara Nanni	Responsabili, Amministratore, Verificatore
1.0.0	16/02/2022	Approvazione del documento	Lorenzo Piran	Responsabile
0.3.0	13/02/2022	Revisione del documento	Nicla Faccioli, Michele Masetto	Verificatori
0.2.3	27/01/2022	Approfondimento §2	Sara Nanni	Amministratore
0.2.2	13/01/2022	Stesura appendice B	Nicholas Sertori	Amministratore
0.2.1	12/01/2022	Stesura appendice A	Nicholas Sertori	Amministratore
0.2.0	28/12/2021	Revisione del documento	Michele Masetto	Verificatore
0.1.3	20/12/2021	Correzione errori tipografici	Nicla Faccioli	Responsabile
0.1.2	16/12/2021	Stesura §3.5	Nicla Faccioli	Responsabile
0.1.1	13/12/2021	Stesura §3.3 e §3.4	Nicla Faccioli	Responsabile
0.1.0	12/12/2021	Revisione complessiva del documento	Michele Masetto	Verificatore

Versione	Data	Descrizione	Autore	Ruolo
0.0.6	11/12/2021	Correzione errori tipografici	Andrea Salmaso	Responsabile
0.0.5	02/12/2021	Aggiornamento \$2	Nicla Faccioli	Amministratore
0.0.4	01/12/2021	Stesura \$3	Nicholas Sertori	Amministratore
0.0.3	01/12/2021	Stesura \$4	Andrea Salmaso	Responsabile
0.0.2	29/11/2021	Stesura \$1 e \$2	Nicla Faccioli	Amministratore
0.0.1	29/11/2021	Creazione documento	Nicla Faccioli	Amministratore

Indice

1	Introduzione	7
1.1	Scopo del documento	7
1.2	Scopo del prodotto	7
1.3	Glossario	7
1.4	Maturità del Documento	7
1.5	Riferimenti	8
1.5.1	Riferimenti normativi	8
1.5.2	Riferimenti informativi	8
2	Processi primari	9
2.1	Fornitura	9
2.1.1	Scopo	9
2.1.2	Aspettative	9
2.1.3	Attività	9
2.1.3.1	Inizializzazione	9
2.1.3.2	Preparazione di una risposta	9
2.1.3.3	Pianificazione	9
2.1.3.4	Esecuzione e controllo	9
2.1.3.5	Revisione e valutazione	10
2.1.3.6	Collaudo e rilascio del prodotto	10
2.1.4	Metriche	10
2.1.5	Strumenti	11
2.2	Sviluppo	11
2.2.1	Scopo	11
2.2.2	Aspettative	11
2.2.3	Attività	12
2.2.3.1	Implementazione del Processo	12
2.2.3.2	Analisi dei Requisiti	12
2.2.3.2.1	Scopo	12
2.2.3.2.2	Aspettative	12
2.2.3.2.3	Requisiti	12
2.2.3.2.4	Denominazione dei Requisiti	12
2.2.3.2.5	Casi d'uso	13
2.2.3.2.6	Denominazione dei Casi d'uso	13
2.2.3.3	Progettazione	13
2.2.3.3.1	Scopo	13
2.2.3.3.2	Aspettative	14
2.2.3.3.3	Descrizione	14
2.2.3.3.4	Technology Baseline	14
2.2.3.3.5	Product Baseline	14
2.2.3.4	Codifica	14
2.2.3.4.1	Scopo	15
2.2.3.4.2	Aspettative	15
2.2.3.4.3	Descrizione	15
2.2.4	Metriche	15
2.2.4.1	Progettazione di dettaglio	15
2.2.4.2	Codifica	16

2.2.4.3	Funzionalità	16
2.2.4.4	Usabilità	16
2.2.4.5	Manutenibilità	16
2.2.4.6	Compatibilità	17
2.2.5	Strumenti	17
3	Processi di supporto	18
3.1	Documentazione	18
3.1.1	Scopo	18
3.1.2	Attività	18
3.1.2.1	Tracciamento del ciclo di vita del documento	18
3.1.2.2	Sviluppo e Design	18
3.1.2.3	Definizione delle norme tipografiche e delle convenzioni	23
3.1.2.4	Inserimento di elementi grafici	25
3.1.3	Metriche	26
3.1.4	Strumenti	26
3.2	Gestione della configurazione	27
3.2.1	Scopo	27
3.2.2	Attività	27
3.2.2.1	Gestione del repository	27
3.2.2.2	Gestione delle modifiche e del caricamento dei file	27
3.2.3	Metriche	27
3.2.4	Strumenti	27
3.3	Gestione della qualità	28
3.3.1	Scopo	28
3.3.2	Aspettative	28
3.3.3	Descrizione	28
3.3.4	Attività	28
3.3.4.1	Denominazione metriche	28
3.3.5	Metriche	28
3.3.6	Strumenti	28
3.4	Verifica	29
3.4.1	Scopo	29
3.4.2	Aspettative	29
3.4.3	Descrizione	29
3.4.4	Attività	29
3.4.4.1	Analisi statica	29
3.4.4.2	Analisi dinamica	29
3.4.4.3	Denominazione dei Test	30
3.4.5	Metriche	31
3.4.6	Strumenti	32
3.5	Validazione	33
3.5.1	Scopo	33
3.5.2	Aspettative	33
3.5.3	Descrizione	33

4	Processi organizzativi	34
4.1	Gestione dei processi	34
4.1.1	Scopo	34
4.1.2	Aspettative	34
4.1.3	Descrizione	34
4.1.4	Attività	34
4.1.4.1	Ruoli di progetto	34
4.1.4.1.1	Responsabile di Progetto	34
4.1.4.1.2	Amministratore di Progetto	35
4.1.4.1.3	Analista	35
4.1.4.1.4	Progettista	35
4.1.4.1.5	Verificatore	35
4.1.4.1.6	Programmatore	36
4.1.4.2	Gestione delle comunicazioni	36
4.1.4.2.1	Comunicazioni interne	36
4.1.4.2.2	Comunicazioni esterne	36
4.1.4.3	Gestione degli incontri	36
4.1.4.3.1	Incontri interni	37
4.1.4.3.2	Incontri esterni	37
4.1.4.3.3	Verbal degli incontri	38
4.1.4.4	Gestione degli strumenti di coordinamento	38
4.1.4.4.1	Ticketing	38
4.1.4.5	Gestione dei rischi	39
4.1.4.5.1	Codifica dei rischi	40
4.1.5	Metriche	40
4.1.6	Strumenti	40
4.2	Formazione	40
4.2.1	Scopo	40
4.2.2	Aspettative	41
4.2.3	Descrizione	41
4.2.4	Attività	41
4.2.4.1	Materiale per la formazione	41
4.2.4.1.1	Linguaggi di programmazione	41
4.2.4.1.2	Tecnologie	41
4.2.5	Metriche	41
4.2.6	Strumenti	41

Elenco delle figure

1	Ciclo di vita dei documenti	18
2	Tipi di test del codice	30
3	Incontri interni	37
4	Incontri esterni	38
5	Gestione dei rischi	39

1 Introduzione

1.1 Scopo del documento

Il presente documento ha come scopo la definizione delle *best practices*_G e del *way of working*_G che ogni componente del gruppo *BugPharma* ha obbligo di rispettare durante l'intero svolgimento del progetto *Login Warrior*. In questo modo si cercherà di garantire, quanto più possibile, omogeneità e coesione in ogni aspetto del suddetto.

1.2 Scopo del prodotto

Le applicazioni cloud in tecnologia web stanno gradualmente sostituendo le applicazioni tradizionali "on premise": portano diversi vantaggi, soprattutto in termini di *TCO*_G (Total Cost of Ownership), in quanto funzionano con un semplice browser web e con una connessione ad Internet veloce e con bassa latenza. Nonostante questo, presentano la criticità di poter subire attacchi da parte di script kiddies, hacker e criminali informatici: essendo infatti sistemi esposti in rete, possono ricevere per loro natura connessioni da tutto il mondo, anche da utenti che non sono esattamente coloro per cui sono state rilasciate. Nasce quindi la necessità di distinguere un utente malintenzionato da uno legittimo nel suo accedere ad applicazioni di questo tipo, così da migliorarne l'esperienza d'uso, attraverso un sistema di analisi esplorativa dei dati ottenuti dai *login*_G.

Il capitolato C5, *Login Warrior*, pone come obiettivo la realizzazione di un'applicazione di visualizzazione di dati di login a supporto della fase esplorativa, *EDA*_G (Exploratory Data Analysis), attraverso grafici di varia tipologia, quali:

- Scatter Plot;
- Parallel Coordinates;
- Force-Directed Graph;
- Sankey Diagram.

L'utente dovrà quindi essere in possesso di un file **.CSV** contenente il *dataset*_G che potrà essere caricato nell'applicazione. Tale applicazione sarà fruibile attraverso un *browser*_G in grado di supportare le tecnologie *HTML5*_G, *CSS*_G e *JavaScript*_G.

1.3 Glossario

Al fine di evitare possibili ambiguità relative al linguaggio utilizzato nei documenti, viene fornito il *Glossario v2.0.0* nel quale sono contenute le definizioni di termini aventi uno specifico significato. Tali termini, ove necessario, sono segnati in corsivo e marcati con una *G* a pedice.

1.4 Maturità del Documento

Il presente documento è redatto con un approccio incrementale al fine di poter trattare nuove o ricorrenti questioni in modo rapido ed efficiente, sulla base di decisioni concordate tra tutti i membri del gruppo. Non può pertanto essere considerato definitivo nella sua attuale versione.

1.5 Riferimenti

1.5.1 Riferimenti normativi

- *Capitolato d'appalto*_G C5 - Login Warrior:
<https://www.math.unipd.it/~tullio/IS-1/2021/Progetto/C5.pdf>

1.5.2 Riferimenti informativi

- Slide T03 del corso di Ingegneria del Software - Processi di ciclo di vita del software:
<https://www.math.unipd.it/~tullio/IS-1/2021/Dispense/T03.pdf>
- Slide T09 del corso di Ingegneria del Software - Progettazione Software:
<https://www.math.unipd.it/~tullio/IS-1/2021/Dispense/T09.pdf>
- Slide T14 del corso di Ingegneria del Software - Verifica e Validazione: introduzione:
<https://www.math.unipd.it/~tullio/IS-1/2021/Dispense/T14.pdf>
- Slide T14 del corso di Ingegneria del Software - Verifica e Validazione: analisi statica:
<https://www.math.unipd.it/~tullio/IS-1/2021/Dispense/T15.pdf>
- Slide T14 del corso di Ingegneria del Software - Verifica e Validazione: analisi dinamica:
<https://www.math.unipd.it/~tullio/IS-1/2021/Dispense/T16.pdf>
- ISO/IEC 12207:1997:
https://www.math.unipd.it/~tullio/IS-1/2009/Approfondimenti/ISO_12207-1995.pdf
- ISO/IEC 12207:2017:
<https://www.iso.org/standard/63712.html>
- ISO/IEC 9126-1:2001:
http://www.colonese.it/00-Manuali_Pubblicatii/07-ISO-IEC9126_v2.pdf.
- ISO/IEC 25010:2011:
<https://www.iso.org/standard/35733.html>.

2 Processi primari

2.1 Fornitura

2.1.1 Scopo

Il processo di Fornitura ha lo scopo di definire e trattare le norme e i termini che i membri del gruppo *BugPharma* sono tenuti a rispettare per rivestire adeguatamente il ruolo di *fornitore_G* nei confronti dell'azienda *proponente_G* *Zucchetti S.p.A.* e dei *committenti_G* Prof. Tullio Vardanega e Prof. Riccardo Cardin.

2.1.2 Aspettative

Nel corso dell'intero progetto, il gruppo intende instaurare con *Zucchetti S.p.A.*, in particolare con il relativo referente Dr. Gregorio Piccoli, un rapporto di costante collaborazione al fine di:

- Determinare aspetti chiave per soddisfare i bisogni del proponente;
- Determinare eventuali vincoli sui requisiti e sui *processi_G*;
- Effettuare una stima, in tempo e in denaro, dei costi;
- Garantire che il *prodotto_G* soddisfi le richieste, concordandone la qualifica.

2.1.3 Attività

Il processo di Fornitura si compone principalmente delle seguenti *attività_G*:

2.1.3.1 Inizializzazione

Al fine di decidere per quale tra i *capitolati_G* proposti candidarsi, il *Responsabile di Progetto* ha il compito di organizzare incontri tra i membri del gruppo, dedicati alla discussione dei suddetti. Per ciascuno dei capitolati, il documento di *Valutazione dei Capitolati* espone i fattori critici individuati. Per quello scelto vengono inoltre elencati gli aspetti positivi che hanno portato alla candidatura per tale appalto.

2.1.3.2 Preparazione di una risposta

Il gruppo deve redigere una *lettera di Presentazione* con cui dichiara l'impegno nella realizzazione del prodotto proposto dal capitolato scelto. I contenuti di tale lettera corrispondono a quelli che vengono esposti nella presentazione per l'aggiudicazione dell'appalto in data 19 Novembre 2021. In concomitanza alla preparazione della lettera di presentazione, viene redatto il documento *Preventivo Costi e Impegni Orari*.

2.1.3.3 Pianificazione

Il gruppo deve redigere un piano per la gestione del progetto e della qualità. Nel capitolo apposito dedicato alla Documentazione vengono descritte le sezioni che compongono i due documenti e quindi i loro contenuti.

2.1.3.4 Esecuzione e controllo

Il gruppo deve seguire ed implementare il piano di progetto e controllare l'avanzamento, i costi e la qualità dei prodotti in tutto il loro ciclo di vita.

2.1.3.5 Revisione e valutazione

Il gruppo deve coordinare le revisioni delle attività svolte, deve eseguire la verifica e la validazione del prodotto garantendo che questo rispetti le aspettative del proponente.

2.1.3.6 Collaudo e rilascio del prodotto

Il gruppo deve garantire la correttezza, completezza e affidabilità del prodotto realizzato in vista del suo collaudo dimostrativo. In seguito al superamento del suddetto, il *Responsabile di Progetto* deve presentare al *committente* il consuntivo finale e può procedere alla consegna del codice sorgente e della relativa documentazione. Ad avvenuto rilascio del prodotto, non è prevista attività di *manutenzione*_G dello stesso.

2.1.4 Metriche

Parametri per comprendere meglio le metriche successive:

- **Budget at Completion (BAC):** Numero intero, costo totale di progetto preventivato.

Metriche:

1. **MPC-AC:** Actual Cost

Soldi spesi per il progetto fino al momento del calcolo.

2. **MPC-EV:** Earned Value

Valore del lavoro fatto fino al momento del calcolo (denaro che si è guadagnato fino a quel momento).

$$(\% \text{ lavoro svolto}) \cdot EAC$$

3. **MPC-PV:** Planned Value

Lavoro che si era pianificato di svolgere fino al momento del calcolo.

$$(\% \text{ lavoro pianificato}) \cdot BAC$$

4. **MPC-SV:** Schedule Variance

Stato di anticipo o ritardo rispetto alla pianificazione. Un valore negativo indica che si è in ritardo rispetto a quanto era stato pianificato.

$$EV - PV$$

Esempio: Se si ha un PV di €20.000 e un EV di €40.000, utilizzando la formula sopra riportata si ottiene:

$$40.000(EV) - 20.000(PV) = 20.000(SV).$$

La variazione di programma quindi è di €20.000. Ciò significa che il progetto è in anticipo rispetto al programma.

Per rappresentare questa cifra in percentuale basta dividere per il valore di EV:

$$\frac{SV}{EV} = \frac{20.000}{40.000} = 0,5$$

Quindi in questo caso il progetto risulta essere in anticipo del 50% rispetto al programma.

5. **MPC-CV**: Cost Variance

Differenza tra il budget a disposizione per il progetto e quello effettivamente utilizzato. Un valore negativo indica che si sta spendendo più di quello che si sta guadagnando.

$$EV - AC$$

Per rappresentare questa cifra in percentuale:

$$\frac{CV}{EV}$$

Una percentuale negativa indica che il progetto ha superato il budget.

6. **MPC-ETC**: Estimated to Completion

Valore stimato per la realizzazione delle rimanenti *attività_G* necessarie.

7. **MPC-EAC**: Estimated at Completion

Indica alla data della misurazione qual è la stima del costo finale che si sta prefigurando, revisione del BAC.

$$AC + ETC$$

2.1.5 Strumenti

Gli strumenti di cui abbiamo fatto uso a supporto delle attività del processo di fornitura sono i seguenti:

- **Google Sheets**: Utilizzato per creare grafici da inserire nella documentazione di progetto;
- **Google Slides**: Utilizzato per creare le presentazioni da mostrare in vari colloqui con proponente e committenti;
- **L^AT_EX**: Utilizzato per la stesura dei documenti di pianificazione e per la lettera di presentazione;
- **Excel**: Utilizzato per creare i diagrammi di Gantt relativi alla pianificazione.

2.2 Sviluppo

2.2.1 Scopo

Il processo di Sviluppo ha lo scopo di definire i *compiti_G* e le attività che il gruppo ha il compito di svolgere al fine di produrre il software richiesto dal proponente.

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_G* 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 Attività

Il processo di Sviluppo, in accordo con lo standard ISO/IEC 12207, si compone delle seguenti attività:

2.2.3.1 Implementazione del Processo

Il gruppo deve scegliere quale modello di sviluppo adottare, in conformità con la dimensione e la complessità del progetto. Il modello scelto informerà la distribuzione, all'interno del ciclo di vita, delle attività da svolgere.

2.2.3.2 Analisi dei Requisiti

2.2.3.2.1 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, quali per 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 contenente tali informazioni è l'*Analisi dei Requisiti*.

2.2.3.2.2 Aspettative

Obiettivo della presente attività è l'individuazione di tutti i requisiti richiesti dal proponente.

2.2.3.2.3 Requisiti

Ogni requisito è strutturato come segue:

- Codice;
- Classificazione;
- Descrizione;
- Fonti.

2.2.3.2.4 Denominazione dei Requisiti

Ogni requisito è classificato e identificato univocamente attraverso un codice, composto secondo il seguente schema:

R[Priorità]-[Tipologia]-[Identificativo]

Dove:

- **R**: indica che si tratta di un requisito;
- **Priorità**: può assumere i seguenti valori:
 - **O**: requisito obbligatorio;
 - **F**: requisito facoltativo;

- **D**: requisito desiderabile.
- **Tipologia**: può assumere i seguenti valori:
 - **F**: funzionale;
 - **P**: prestazionale;
 - **Q**: qualitativo;
 - **V**: vincolo.
- **Identificativo**: numero progressivo per contraddistinguere il requisito, in forma gerarchica padre-figlio, strutturato come segue:

[codicePadre].[codiceFiglio]

2.2.3.2.5 Casi d'uso

Ogni *caso d'uso*_G è strutturato come segue:

- Denominazione;
- Titolo;
- Diagramma *UML*_G (se necessario);
- Attore primario;
- Attori secondari (se presenti);
- Precondizioni;
- Postcondizioni;
- Scenario principale;
- Estensioni (se presenti);
- Generalizzazioni (se presenti).

2.2.3.2.6 Denominazione dei Casi d'uso

Ogni caso d'uso è classificato e identificato univocamente attraverso un codice, composto secondo il seguente schema:

UC[codiceCaso].[codiceSottocaso]

2.2.3.3 Progettazione

2.2.3.3.1 Scopo

L'attività di Progettazione avviene ad opera dei *Progettisti*, i quali hanno il compito di definire le caratteristiche essenziali del prodotto software richiesto, in funzione di quanto esposto nell'*Analisi dei Requisiti*.

2.2.3.3.2 Aspettative

Obiettivo della presente attività è l'individuazione dell'architettura software più adatta, la quale dovrà:

- Soddisfare tutti i requisiti individuati in fase di analisi e riportati nell'*Analisi dei Requisiti*;
- Risultare comprensibile e modulare, garantendo una facile manutenzione;
- Perseguire correttezza per costruzione per garantire la correttezza del prodotto finale.

2.2.3.3.3 Descrizione

La presente attività si articola nelle seguenti fasi:

- **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 di un *PoC_G* e di una conseguente *Technology Baseline*;
- **Progettazione Architettuale:** nella quale viene eseguita una definizione ad alto livello dell'architettura del prodotto e delle sue componenti, insieme alla definizione dei test di integrazione;
- **Progettazione di Dettaglio:** nella quale viene eseguita una definizione delle specifiche di dettaglio dell'architettura del prodotto e delle sue componenti, scomposte in unità, insieme ai diagrammi atti a descriverle e ai test di verifica. Tali informazioni costituiranno la *Product Baseline*.

2.2.3.3.4 Technology Baseline Definisce e motiva le tecnologie, i framework, e le librerie selezionate per la realizzazione del prodotto, includendo:

- Tecnologie adottate;
- Tracciamento delle componenti;
- *Proof Of Concept_G*;

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

- Desing patterns;
- Definizione delle classi;
- Tracciamento delle classi;
- Diagrammi UML, quali:
 - Diagrammi delle classi;
 - Diagrammi dei package;
 - Diagrammi delle attività;
 - Diagrammi di sequenza.
- Test di unità su ogni componente.

2.2.3.4 Codifica

2.2.3.4.1 Scopo

L'attività di Codifica avviene ad opera dei *Programmatori*, i quali hanno il compito di concretizzare quanto prodotto nell'attività di Progettazione realizzando il prodotto software finale attraverso la programmazione.

2.2.3.4.2 Aspettative

- Sviluppare un prodotto software conforme alle richieste del proponente;
- Produrre codice uniforme e leggibile, al fine di agevolare la verifica e la validazione dello stesso.
- Garantire la qualità del prodotto.

2.2.3.4.3 Descrizione

Per soddisfare le aspettative di cui sopra, i *Programmatori* hanno il compito di adottare le regole e le convenzioni di seguito riportate.

Sintassi

- **Indentazione:** tutti i blocchi di codice annidati devono avere un'indentazione di 4 spazi, come illustrato dal seguente esempio:

```
rows.forEach(elt => {  
  const row = elt.split(',');  
  array.push(row);  
});
```

- **Metodi:** il nome dei metodi deve adottare lo stile camelCase, mantenendo minuscola la lettera iniziale della prima parola e maiuscole le iniziali delle successive;
- **Classi:** il nome delle classi deve adottare lo stile PascalCase, mantenendo maiuscole le lettere iniziali delle parole di cui è composto;
- **Variabili:** il nome delle variabili deve adottare lo stile camelCase, mantenendo minuscola la lettera iniziale della prima parola e maiuscole le iniziali delle successive;
- **File:** il nome dei file deve iniziare con la lettera maiuscola e deve essere identificativo del contenuto;
- **Commenti:** i commenti vanno inseriti prima di ogni componente al fine di fornire per esso una breve e sintetica descrizione che ne spieghi le specifiche.

Ricorsione

Ove possibile è necessario che i *Programmatori* non facciano uso della ricorsione, evitando così un'eccessiva allocazione di memoria.

2.2.4 Metriche

2.2.4.1 Progettazione di dettaglio

1. MPC-NM: Number of Methods

Indica il numero medio di metodi per classe. Un numero eccessivo potrebbe indicare la necessità di *refactoring*_G.

2.2.4.2 Codifica

1. **MPC-VNUD**: Variabili Non Utilizzate e Non Definite

Numero di variabili non utilizzate e non definite. Esse rappresentano un errore di programmazione e una fonte comune di bug nel software: le prime sporcano il codice e fanno allocare memoria inutilmente, le seconde sono dichiarate ma non inizializzate ad un valore noto definito prima di essere utilizzate.

2.2.4.3 Funzionalità

1. **MPD-RC**: Requirements Coverage

Indica la percentuale dei requisiti soddisfatti. Per il calcolo del valore di accettazione si considerano solo i requisiti obbligatori.

Per il valore di accettazione:

$$RC_{obb} = \frac{N_{os}}{N_{ot}} \cdot 100$$

Dove:

- N_{os} : numero di requisiti obbligatori soddisfatti;
- N_{ot} : numero di requisiti obbligatori totali.

Per il valore ideale:

$$RC_{tot} = \frac{N_s}{N_t} \cdot 100$$

Dove:

- N_s : numero di requisiti soddisfatti;
- N_t : numero di requisiti totali.

2.2.4.4 Usabilità

1. **MPD-PAA**: Profondità Albero delle Azioni

Numero massimo di azioni effettuate dall'utente prima di poter arrivare al suo obiettivo.

2. **MPD-PAP**: Profondità Albero delle Pagine

Numero massimo di pagine visitate dall'utente prima di poter arrivare al suo obiettivo.

2.2.4.5 Manutenibilità

1. **MPD-NCS**: Numero di Code Smell

Indica il numero di *Code Smell*_G, e quindi il numero di best practices mancate, rilevati.

2. **MPD-PDC**: Percentuale di Duplicazione del Codice

Indica la percentuale di codice duplicato nel corso dello sviluppo.

2.2.4.6 Compatibilità

1. **MPD-VS:** Versioni del *browser_G* Supportate

Misura la percentuale di versioni di browser supportate dal prodotto in relazione a quelle pianificate.

$$VS = \frac{B_s}{B_p} \cdot 100$$

Dove:

- B_s : versioni di browser supportate;
- B_p : versioni di browser che si era pianificato di supportare.

2.2.5 Strumenti

Gli strumenti di cui abbiamo fatto uso a supporto delle attività del processo di sviluppo sono i seguenti:

- **D3.js:** libreria per realizzazione di diversi tipi di grafici a partire da dati reali;
- **DruidJS:** libreria per la riduzione dimensionale. Fornisce molti tipi diversi di algoritmi da cui scegliere, tra cui quelli richiesti dal proponente;
- **React:** libreria *JavaScript* che facilita la creazione di interfacce utente dinamiche, con migliori tempi di renderizzazione, e ne aumenta la manutenibilità;
- **IDE e Editor di Testo:**
 - *JetBrain WebStorm*;
 - *Microsoft Visual Studio*;
 - *GitHub Atom*;
 - *Microsoft Visual Studio Code*.

3 Processi di supporto

3.1 Documentazione

3.1.1 Scopo

Scopo del presente processo è la definizione degli standard e degli strumenti necessari alla stesura di tutti i documenti del progetto.

3.1.2 Attività

Il processo di Documentazione si compone principalmente delle seguenti *attività*_G:

3.1.2.1 Tracciamento del ciclo di vita del documento

Ogni documento segue le fasi del seguente *ciclo di vita*_G:

- **Redazione:** il documento viene scritto utilizzando un approccio incrementale. Viene considerato redatto una volta scritto nella sua interezza;
- **Verifica:** il documento è prodotto in parte o interamente e rimane in attesa di verifica. Ogni sezione viene verificato da almeno una persona, diversa dal *Redattore*_G dello stesso. Viene considerato verificato quando i *verificatori* incaricati completano il loro lavoro e le modifiche necessarie sono state portate a termine;
- **Approvazione:** il *Responsabile di Progetto* dichiara che il documento è completo in ogni sua parte e pronto per essere rilasciato, marcandolo come approvato.

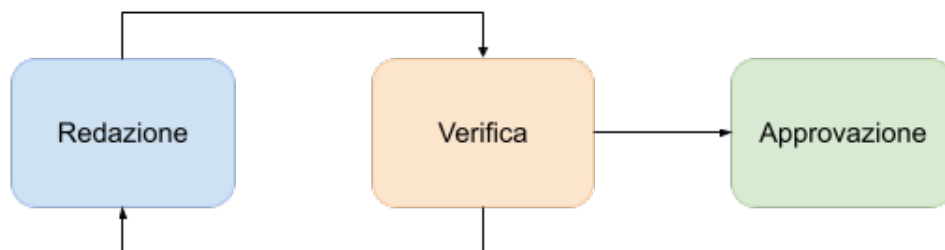


Figura 1: Ciclo di vita dei documenti

3.1.2.2 Sviluppo e Design

Classificazione dei documenti

I documenti redatti sono raggruppabili nelle due classi descritte qui di seguito.

- **Documenti ad uso interno**
Sono documenti ad uso interno quelli pensati unicamente per l'uso da parte di membri del gruppo:
 - Tutti i *Verbali Interni*;
 - Le *Norme di Progetto*.
- **Documenti ad uso esterno**
Sono documenti ad uso esterno quelli destinati ad essere divulgati al di fuori del gruppo:

- Tutti i *Verbali Esterni*;
- L'*Analisi dei Requisiti*;
- Il *Piano di Progetto*;
- Il *Piano di Qualifica*;
- Il *Glossario*.

Struttura dei documenti

Tutti i documenti ufficiali che vengono prodotti devono seguire una struttura precisa per consentire omogeneità e coesione. Segue un elenco delle norme comuni a tutti i documenti, contenente delle sezioni specifiche per ognuno dei documenti prodotti nel corso del progetto.

Frontespizio

Il frontespizio è la prima pagina del documento ed è così strutturato:

- **Logo dell'Università e didascalia:** logo di *UniPd* posizionato in alto a centro pagina. Subito sotto, la dicitura "Università degli Studi di Padova";
- **Logo del gruppo e didascalia:** nome del gruppo, visibile centralmente subito sotto il logo dello stesso, seguito dal recapito e-mail;
- **Nome del progetto:** nome del progetto (*Login Warrior*) in corsivo, posizionato di fianco al nome del gruppo e separato da esso attraverso un trattino;
- **Recapito:** recapito email, posizionato centralmente;
- **Titolo:** nome del documento, posizionato centralmente in grassetto;
- **(Solo per i verbali interni) Ordine del giorno:** breve elenco degli argomenti trattati nell'incontro cui fa riferimento il verbale;
- **Tabella descrittiva:** posizionata sotto il titolo, centrale e contenente le seguenti informazioni:
 - **Versione:** numero di versione del documento;
 - **Approvazione:** nome e cognome dei membri del gruppo che hanno eseguito l'approvazione del documento;
 - **Redazione:** nome e cognome dei membri del gruppo che hanno eseguito la redazione del documento;
 - **Verifica:** nome e cognome dei membri del gruppo che hanno eseguito la verifica del documento;
 - **Stato:** stato del ciclo di vita in cui si trova attualmente il documento;
 - **Uso:** destinazione d'uso del documento (interno o esterno);
 - **Distribuzione:** destinatari del documento.

Registro delle modifiche

Tutti i documenti soggetti a continue integrazioni e modifiche devono contenere un registro delle modifiche dopo la prima pagina. Tale registro deve essere aggiornato con le suddette modifiche mano a mano che queste vengono apportate, indicando per ognuna:

- **Versione:** versione cui giunge il documento in seguito alla modifica;
- **Data:** data della modifica;

- **Descrizione:** descrizione breve e concisa della modifica apportata;
- **Autore:** nome e cognome dei membri del gruppo direttamente coinvolti nella modifica;
- **Ruolo:** ruolo ricoperto dai membri del gruppo coinvolti nella modifica.

Fanno eccezione i verbali interni ed esterni, i quali vengono redatti contemporaneamente agli incontri e non sono soggetti a modifiche successive.

Indice

Tutti i documenti, eccezion fatta per i verbali che seguono una struttura leggermente diversa, devono contenere un indice, per agevolare la consultazione e permettere una visita non sequenziale del documento stesso:

- Ogni indice inizia dal numero 1;
- Ogni sottosezione deve riportare il numero della sezione genitrice seguito da un punto e dal numero della sottosezione stessa, ripartendo sempre da 1.

Intestazione e Piè di pagina

Fatta eccezione per la prima pagina, tutte le altre devono contenere:

- **Intestazione**, composta da:
 - A sinistra: nome del gruppo;
 - A destra: numero e nome della sezione.
- **Piè di pagina**, composto da:
 - A sinistra: nome e versione del documento;
 - A destra: pagina corrente sul totale di pagine del documento.

Valutazione dei Capitolati

Il documento espone per ogni capitolato:

- Le informazioni generali, quali:
 - Nome;
 - Proponente;
 - Committente.
- Una breve descrizione;
- Il dominio applicativo e il dominio tecnologico;
- I fattori critici individuati;
- Le conclusioni cui il gruppo è giunto.

Inoltre, per il capitolato scelto:

- Aspetti positivi in esso individuati.

Analisi dei Requisiti

L'*Analisi dei Requisiti* è formata dalle seguenti sezioni:

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

Piano di Progetto

All'interno del *Piano di Progetto*, il *Responsabile di Progetto* e gli *Amministratori* hanno il compito di pianificare le *attività_G* da svolgere durante tutte le *fasi_G* del progetto, organizzando e distribuendo le risorse lungo vari periodi. Tale documento espone:

- **Analisi dei rischi:** ove vengono identificati e classificati, per pericolosità e probabilità di occorrenza, i rischi che potrebbero insorgere durante lo sviluppo del progetto;
- **Modello di sviluppo:** ove vengono definiti i dettagli del *modello di sviluppo_G* adottato;
- **Pianificazione:** ove vengono pianificate, a livello temporale, le attività da svolgere nel corso del progetto. I periodi e gli *incrementi_G*, caratteristica fondamentale del modello di sviluppo adottato, sono esibiti in modo simile.

Periodi

I periodi vengono utilizzati per fare riferimento ai lassi temporali costituenti le fasi che non prevedono codifica, ma unicamente avanzamenti incrementali della documentazione o altre attività. Vengono esibiti nel seguente modo:

- **Identificativo:** codice univoco composto secondo il seguente schema:

PER-[Numero][Fase]

Con:

- * **PER:** sigla identificativa di un periodo;
- * **Numero:** numero progressivo, circoscritto alla fase cui fa riferimento (che quindi si azzerà all'inizio di ogni fase);
- * **Fase:** codice identificativo della fase, in particolare:
 - **A:** indica la fase di Analisi;
 - **PTB:** indica la fase di Progettazione della Technology *Baseline_G*;
 - **ARC:** indica la fase di Progettazione Architettuale;
 - **VC:** indica la fase di Validazione e Collaudo.
- **Data inizio:** data di inizio prevista;
- **Data fine:** data di fine prevista;
- **Pre-condizione:** attività o incrementi necessari per l'inizio delle attività previste per il periodo;
- **Post-condizione:** condizioni decretanti il completamento delle attività previste per il periodo;
- **Ruoli attivi:** ruoli lavorativi previsti per lo svolgimento delle attività;
- **Descrizione:** descrizione generale del periodo;
- **Attività previste:** lista delle attività da svolgere nel periodo.

Incrementi

Gli incrementi vengono utilizzati per fare riferimento ai lassi temporali costituenti le fasi che prevedono la codifica del prodotto. Vengono esibiti nel seguente modo:

- **Identificativo:** codice univoco composto secondo il seguente schema:

INC-[Numero]

Con:

- * **INC:** sigla identificativa di un incremento;
 - * **Numero:** numero progressivo, in ambito globale (che quindi non si azzera all’inizio di ogni fase);
 - **Data inizio:** data di inizio prevista;
 - **Data fine:** data di fine prevista;
 - **Pre-condizione:** attività o incrementi necessari per l’inizio delle attività previste per l’incremento;
 - **Post-condizione:** condizioni decretanti il completamento delle attività previste per l’incremento;
 - **Ruoli attivi:** ruoli lavorativi previsti per lo svolgimento delle attività;
 - **Obiettivo:** obiettivi da raggiungere associati all’incremento (rimossi a partire dalla fase di Progettazione di Dettaglio e Codifica perché ridondanti con le attività previste);
 - **Attività previste:** lista delle attività da svolgere nell’incremento.
- **Preventivo:** ove viene stimata, sulla base della pianificazione, la quantità di lavoro necessaria al completamento di ogni fase e viene esibito un preventivo per il costo totale del progetto;
 - **Consuntivo:** ove viene tracciato l’andamento rispetto a quanto preventivato.

Piano di Qualifica

All’interno del *Piano di Qualifica*, i *Verificatori*, hanno il compito di scegliere le strategie da adottare per la *verifica_G* e la *validazione_G* del materiale prodotto, necessarie per garantirne la *qualità_G*. Il documento è strutturato come segue:

- **Qualità di processo:** ove vengono stabilite le metriche per il controllo della qualità di processo;
- **Qualità di prodotto:** ove vengono stabilite le metriche per il controllo della qualità di prodotto;
- **Specifiche dei test:** ove vengono specificati i test da eseguire sul prodotto, necessari a garantire il soddisfacimento dei requisiti;
- **Resoconto delle attività di verifica:** ove vengono riportati i risultati dell’attività di verifica.

Norme di Progetto

All’interno delle *Norme di Progetto* gli *Amministratori* e il *Responsabile di Progetto* si occupano di stabilire il *Way of Working_G* per il progetto nella sua interezza. Il documento è suddiviso in base a tre macrogruppi di processi:

- **Processi primari:** costituiti dalle attività principali di un progetto, senza le quali questo non esisterebbe;
- **Processi di Supporto:** costituiti da attività di appoggio, utili appunto a supportare i processi primari;

- **Processi Organizzativi:** costituiti da attività che hanno un ruolo "orizzontale", volto a facilitare quello "verticale" dei processi primari.

Glossario

Il *Glossario* presenta una sezione per ogni lettera dell'alfabeto.

Verbali

I *verbali* (interni ed esterni) non seguono la struttura classica definita per gli altri documenti. Non necessitano di indice, al posto del quale viene definita una sezione apposita denominata "Informazioni Generali", che contiene, nell'ordine:

- **Luogo e data dell'incontro:** luogo, fisico o virtuale, nel quale si è svolto l'incontro, con orario di inizio e fine dello stesso;
- **Presenze**, nel quale sono riportati:
 - Il numero di membri del gruppo presenti all'incontro;
 - Il nome e cognome dei membri del gruppo presenti all'incontro;
 - Il nome e cognome dei membri del gruppo assenti all'incontro;
 - (Per i verbali esterni) nome e cognome dei partecipanti esterni.

I verbali contengono inoltre una tabella riassuntiva che riepiloga le decisioni principali che sono state prese, assegnando ad ognuna di esse un codice identificativo necessario al tracciamento di tali decisioni. Tale codice è scritto come segue:

[Nome Verbale]-[X]

Dove:

- **Nome Verbale:** nome del verbale, che segue le regole specificate al capitolo §3.1.2.3 di questo documento;
- **X:** numero intero che parte da 1 per la prima decisione tracciata all'interno del verbale ed è incrementale per le decisioni a seguire.

Eccezioni

Viene fatta un'eccezione per i documenti redatti precedentemente alla scrittura delle *Norme di Progetto* e all'assegnazione degli appalti, in quanto seguono una struttura lievemente più permissiva e non presentano il nome del progetto nella prima pagina, per ovvi motivi.

3.1.2.3 Definizione delle norme tipografiche e delle convenzioni

Nome dei file e versionamento

I nomi dei file seguono la notazione *CamelCase*_G, tenendo la lettera iniziale sempre maiuscola. Il nome così scritto deve essere seguito da un'indicazione della versione dello stesso, la cui sigla è strutturata nel seguente modo:

_v.[X].[Y].[Z]

Con:

- **X:** numero intero che parte da 0 e viene incrementato dal responsabile di progetto all'approvazione del documento. Il valore di **X** è limitato superiormente dal numero di revisioni;

- **Y**: numero intero che parte da 0 e viene incrementato dal verificatore ad ogni verifica. Il suo valore non ha limiti superiori, ma viene riportato a 0 quando **X** viene incrementato;
- **Z**: numero intero che parte da 0 e viene incrementato dal redattore ad ogni modifica. Anche **Z** non ha limite superiore, ma viene riportato a 0 quando **Y** viene incrementato.

Fanno eccezione i verbali, che seguono una nomenclatura diversa. Essi non prevedono *versionamento*_G, poiché non subiscono modifiche successive alla loro redazione. Devono seguire la seguente nomenclatura:

V[T]_[AAMMGG]_[Numero]

Con:

- **V**: tipologia di documento, nonché verbale;
- **T**: tipologia di incontro per il quale è stato scritto il verbale, in particolare:
 - **I**: indica un incontro interno;
 - **E**: indica un incontro esterno.
- **AAMMGG**: data di svolgimento dell'incontro cui si riferisce il verbale;
- **Numero**: indicazione sull'ordine temporale di svolgimento degli incontri, nel caso se ne svolga più di uno alla stessa data, costituito da zero o più numeri interi progressivi.

Stile del testo

- **Grassetto**: viene applicato ai titoli delle sezioni e sottosezioni e agli elementi degli elenchi puntati che ne riassumono il contenuto;
- **Corsivo**: Viene utilizzato solo per le seguenti occorrenze:
 - Nome del gruppo, dell'azienda *proponente*_G e del Progetto;
 - Ruoli ricoperti dai componenti del gruppo;
 - Strumenti;
 - Citazioni;
 - Parole inserite nel glossario;
 - Riferimenti ad altri documenti;
 - Riferimenti ai repository;
 - Riferimenti alle revisioni;
 - Parole particolari, poco usate o poco conosciute.
- **Monospazio**: per riferirsi alle estensioni dei file e al codice;
- **Maiuscolo**: consentito unicamente per gli acronimi.

Elenchi puntati

Gli elenchi puntati possono essere utilizzati per esprimere in modo più accessibile un pensiero, o per esprimere un concetto di natura sequenziale e non necessariamente ordinata.

Le voci degli elenchi puntati sono identificate da forme diverse in base al livello di profondità in cui si trovano, in particolare da:

- Un pallino al primo livello.
 - Un trattino al secondo livello.
 - * Un asterisco al terzo livello.
 - Un punto al quarto livello.

Ogni voce deve iniziare con la lettera maiuscola e terminare con un punto e virgola, ad eccezione dell'ultima voce di ogni livello che termina con un punto.

Glossario

I termini del glossario presenti nel testo dei documenti sono segnati in corsivo e marcati con una _G (maiuscola) a pedice. Vengono formattati in questo modo solo i termini che compaiono nel contenuto di un documento, escludendo quindi didascalie di immagini o tabelle, titoli di sezioni e via dicendo.

Sigle

Di seguito una lista di sigle che è possibile trovare all'interno dei documenti, con il relativo significato:

- Documentazione:
 - **AdR**: indica il documento *Analisi Dei Requisiti*;
 - **NdP**: indica il documento *Norme Di Progetto*;
 - **PdP**: indica il documento *Piano Di Progetto*;
 - **PdQ**: indica il documento *Piano Di Qualifica*;
 - **MU**: indica il documento *Manuale Utente*;
 - **MdM**: indica il documento *Manuale del Manutentore*.
- Revisioni di progetto:
 - **RTB**: indica la prima revisione, costituita dalla Requirements and Technology Baseline;
 - **PB**: indica la seconda revisione, costituita dalla Product Baseline;
 - **CA**: indica la terza revisione, costituita dalla Customer Acceptance.
- Ruoli:
 - **Re**: indica il *Responsabile di Progetto*;
 - **Am**: indica l'*Amministratore di Progetto*;
 - **An**: indica l'*Analista*;
 - **Pt**: indica il *Progettista*;
 - **Pr**: indica il *Programmatore*;
 - **Ve**: indica il *Verificatore*.

3.1.2.4 Inserimento di elementi grafici

Tabelle

Tutte le tabelle devono essere centrate orizzontalmente alla pagina e devono essere accompagnate da una didascalia che riporta il numero della tabella, incrementale all'interno del documento, e una breve descrizione.

Per semplificarne la consultazione, le righe di ogni tabella sono colorate con un pattern alternato, nel modo seguente:

- **Intestazione:** caratteri neri su sfondo blu (HEX #7fb3e1);
- **Righe dispari:** caratteri neri su sfondo beige (HEX #f7f3ed);
- **Righe pari:** caratteri neri su sfondo azzurro (HEX #d7e8f6).

Immagini

Come le tabelle, anche le immagini devono essere centrate orizzontalmente nella pagina e devono essere accompagnate da una didascalia che indichi il numero dell'immagine, incrementale all'interno del documento, e una breve descrizione. Eventuali grafici, *diagrammi di Gantt*_G e diagrammi *UML*_G verranno inseriti nei documenti come immagini in un formato adeguato.

3.1.3 Metriche

1. MPC-IG: *Indice Gulpease*_G

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 delle lettere.

$$IG = 89 + \frac{300 * N_f - 10 * N_l}{N_p}$$

Dove:

- N_f : numero di frasi;
- N_l : numero di lettere;
- N_p : numero di parole.

In generale, testi con indice:

- < 80 : difficili da leggere per chi ha la licenza elementare;
- < 60 : difficili da leggere per chi ha la licenza media;
- < 40 : difficili da leggere per chi ha un diploma superiore.

2. MPC-CO: Correttezza Ortografica

Numero di errori grammaticali o ortografici per documento.

3.1.4 Strumenti

Le attività di questo processo si appoggiano ai seguenti strumenti:

- **L^AT_EX**: Per redigere i documenti si è scelto il *linguaggio di markup*_G L^AT_EX. Il gruppo lascia la libertà ai singoli membri di scegliere l'editor a loro più comodo per la stesura e successiva modifica di tali documenti;
- **Google Docs**: Per la stesura di brevi bozze, documenti ufficiosi e note e per la creazione di grafici e diagrammi, si è deciso di utilizzare la suite *Google Docs*. Rimane strettamente proibito l'uso di *Google Docs* per la redazione di qualsiasi documento destinato in un futuro a diventare ufficiale, o che comunque necessiti di versionamento;
- **StarUML**: Si è scelto di utilizzare *StarUML* per la creazione, modifica, salvataggio ed esportazione dei diagrammi *UML* che saranno inclusi nei documenti ufficiali.

3.2 Gestione della configurazione

3.2.1 Scopo

La gestione della configurazione definisce le norme utili a semplificare e unificare la conservazione dei documenti e delle componenti del software.

3.2.2 Attività

3.2.2.1 Gestione del repository

Per una migliore organizzazione dei contenuti, si è deciso di utilizzare tre repository separati:

- **Repository *Documenti***, viene utilizzato per immagazzinare in modo ordinato i file sorgente (`.tex`) dei documenti di progetto, dai verbali (interni ed esterni) ai documenti necessari per lo sviluppo del progetto, insieme al prodotto della loro compilazione in formato `.pdf`. Pensato per la lavorazione in itinere dei file in esso contenuti, viene utilizzato come repo *privato* interno al gruppo;
- **Repository *BugPharma-Documentazione***, rappresenta un'interfaccia tra il gruppo ed i proponenti e contiene esclusivamente i documenti destinati alla consultazione da parte di esterni, in formato `.pdf`;
- **Repository *BugPharma-Login-Warrior***, rappresenta un'interfaccia tra il gruppo ed i proponenti e contiene il codice del prodotto;
- **Repository *Poc_G***, contiene tutti i file sorgente relativi al *Proof of Concept_G* che dovrà essere consegnato in occasione della prima revisione.

3.2.2.2 Gestione delle modifiche e del caricamento dei file

All'interno del repo *Documenti*, è stato creato un *branch_G* per ogni tipologia di documento, così da consentire una migliore gestione delle modifiche da parte dei membri del gruppo. In caso di necessità di correzioni di tipo grammaticale ed ortografico, sarà necessario contattare direttamente il redattore tramite il canale *Slack* relativo al documento interessato. Quest'ultimo si occuperà di apportare le modifiche.

Nelle cartelle del repo *Documenti* saranno presenti solo i file con estensione `.tex`, `.pdf` e `.png`. I file rimanenti, generati automaticamente durante la compilazione, vengono esclusi dal tracciamento inserendo la relativa estensione all'interno del file `.gitignore` presente nella root del repo.

3.2.3 Metriche

Il processo di gestione della configurazione non fa uso di metriche qualitative particolari.

3.2.4 Strumenti

Per il versionamento e il controllo di gestione vengono utilizzati i seguenti strumenti a supporto delle attività menzionate:

- **GitHub**: Per gestire i repository *Git_G* in modo comodo e semplice.

3.3 Gestione della qualità

3.3.1 Scopo

Scopo del processo di Gestione della qualità è di stabilire metriche precise per tutte le *attività*_G nell'ambito della *verifica*_G e della *validazione*_G. In tal modo è possibile garantire il rispetto del livello di *qualità*_G fissato. Per una trattazione approfondita delle metriche utilizzate si rimanda al *Piano di Qualifica v2.0.0*.

3.3.2 Aspettative

Obiettivo del presente processo è:

- Avere un livello quantificabile della qualità dei *processi*_G attuati;
- Avere un controllo continuo della qualità del *prodotto*_G, in modo da poter verificare che esso rispetti le aspettative del proponente.

3.3.3 Descrizione

Nel documento *Piano di Qualifica v2.0.0* vengono descritte le metriche che il gruppo ha deciso di adottare per la valutazione della qualità, con le relative soglie e i range di valori accettabili.

3.3.4 Attività

3.3.4.1 Denominazione metriche

Le metriche utilizzate sono identificate tramite un codice univoco così composto:

M[Utilizzo]-[Acronimo]

Dove:

- **Utilizzo:** indica se la *metrica*_G è per:
 - **PC:** processo;
 - **PD:** prodotto.
- **Acronimo:** acronimo del nome della metrica.

3.3.5 Metriche

1. **MPC-QMS:** Quality Metrics Satisfied
Percentuale di metriche di qualità soddisfatte.

$$QMS = \frac{NQMS}{TQM} \cdot 100$$

Dove:

- **NQMS** (Number of Quality Metrics Satisfied): numero di metriche di qualità soddisfatte;
- **TMQ** (Total number of Quality Metrics): numero di metriche di qualità totali.

3.3.6 Strumenti

Il processo di gestione della qualità non fa uso di strumenti particolari.

3.4 Verifica

3.4.1 Scopo

Scopo del processo di Verifica è l'individuazione di errori introdotti all'interno del prodotto durante la sua produzione. Sono soggetti a verifica sia il software che i documenti.

3.4.2 Aspettative

Obiettivo del presente processo è:

- Verificare ogni $fase_G$ rispettando i criteri stabiliti, in modo da facilitare le attività successive;
- Rendere il processo di verifica il più automatico possibile;
- Rispettare gli obiettivi riportati nel *Piano di Qualifica v2.0.0*.

3.4.3 Descrizione

Il processo di verifica si svolge in due forme:

- **Analisi statica:** non richiede l'esecuzione dell'oggetto di verifica.
- **Analisi dinamica:** richiede l'esecuzione dell'oggetto di verifica.

3.4.4 Attività

3.4.4.1 Analisi statica

Non richiedendo l'esecuzione dell'oggetto di verifica, l'analisi statica può essere applicata sia al codice che ai documenti, sin dalle prime fasi del progetto. Esistono due metodi per effettuare analisi statica:

- **Walkthrough:** lettura a largo spettro per individuare eventuali problematiche. Utilizzabile nei casi in cui non si abbia ben chiaro dove tali difetti potrebbero trovarsi;
- **Inspection:** lettura mirata dell'oggetto di verifica, solo nei punti in cui è noto che potrebbero trovarsi degli errori. Utilizzabile nel caso in cui sia presente un criterio per capire cosa potrebbe generare problemi.

3.4.4.2 Analisi dinamica

L'analisi dinamica è applicabile solo a codice che può essere eseguito e consiste nell'attività di $test_G$. I test a cui viene sottoposto il software devono essere:

- **Ripetibili:** per permettere di verificare che gli errori individuati siano stati effettivamente rimossi;
- **Automatizzati:** tramite l'utilizzo di appositi strumenti.

Segue un elenco delle tipologie di test utilizzati durante il progetto.

Test di unità

I test di $unità_G$ vengono eseguiti su singole unità di codice per verificarne la correttezza. Poiché le unità sono piccole per loro natura, è possibile utilizzare strumenti quali $Stub_G$ e $Driver_G$ per simulare rispettivamente un'unità chiamante e un'unità chiamata.

Test di integrazione

I test di integrazione vengono eseguiti per verificare che le unità si interfaccino tra loro nel modo corretto. Al superamento dei test si formano agglomerati di unità sempre più grandi, fino al raggiungimento della dimensione totale del sistema.

Test di sistema

Il test di sistema viene effettuato quando tutte le sue unità sono state integrate e dopo che tutti i test di integrazione hanno riportato buon esito. Esso si occupa di verificare che tutti i *requisiti*_G individuati nell'*Analisi dei Requisiti* siano stati rispettati.

Test di regressione

Il test di *regressione*_G viene effettuato successivamente a una modifica per correzione, per verificare che non siano stati introdotti nuovi errori.

Test di accettazione - collaudo

Il test di accettazione viene effettuato con la presenza del *committente*_G e serve ad accertare che siano stati rispettati tutti i requisiti utente. Il suo superamento garantisce che il software sia pronto per essere rilasciato.

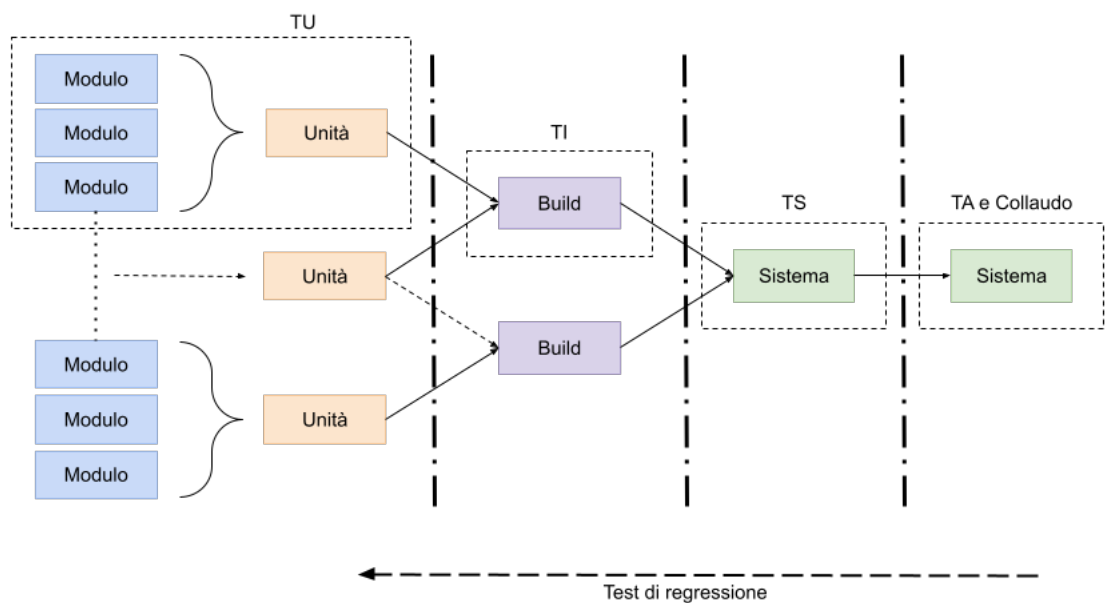


Figura 2: Tipi di test del codice

3.4.4.3 Denominazione dei Test

Per garantire facilità di utilizzo e reperibilità, tutti i test sono identificati con un codice, generato come segue:

T[Tipologia]-[Identificativo]

Dove:

- **Tipologia:** indica se il test è di:
 - **U:** Unità;
 - **I:** Integrazione;

- **R**: Regressione.

- **Identificativo**: attributo numerico sequenziale.

Fanno eccezione i test di sistema e quelli di accettazione, la cui denominazione segue regole personalizzate per poter tenere traccia dei requisiti che si vuole testare. Il codice di tali test viene generato come segue:

T[Tipologia]-[Importanza requisito][Tipologia requisito]-[Id requisito]

Dove:

- **Tipologia**: indica se il test è di:
 - **S**: Sistema;
 - **A**: Accettazione.
- **Importanza requisito**: indica la priorità del requisito che si sta testando o del requisito principale del gruppo di requisiti che il test verifica:
 - **O**: Obbligatorio;
 - **D**: Desiderabile;
 - **F**: Facoltativo.
- **Tipologia requisito**: indica il tipo di requisito, tra:
 - **F**: Funzionale;
 - **Q**: Qualità;
 - **V**: Vincolo.
- **Id requisito**: il numero identificativo del requisito che si sta testando o del requisito principale del gruppo di requisiti che il test verifica.

Per i test di Sistema si è deciso di usare l'id del requisito e non un numero sequenziale. Questa scelta è stata presa per evitare confusione nel far corrispondere ad ogni requisito il proprio test e viceversa. All'interno delle tabelle dei test nel *Piano di Qualifica* vengono inoltre sfruttate le seguenti sigle per il tracciamento dello stato dei suddetti:

- **S**: test superato;
- **NI**: test non implementato;
- **NS**: test non superato.

3.4.5 Metriche

1. **MPC-COV**: Code Coverage

Misura del grado di esecuzione del codice sorgente di un programma quando viene eseguita una particolare suite di $test_G$. Una percentuale alta indica che il codice sorgente eseguito è molto e che vi è quindi una minore probabilità di contenere bug non rilevati. Al contrario, una percentuale bassa suggerisce un'alta probabilità di presenza di bug.

2. MPC-PTCP: Passed Test Cases Percentage

Indica la percentuale di test passati con successo fino alla data corrente.

$$PTCP = \frac{T_p}{B_e} \cdot 100$$

Dove:

- B_s : numero di test passati;
- B_p : numero di test eseguiti.

3.4.6 Strumenti

Analisi statica del codice

- **Prettier**: Strumento per la formattazione automatica del codice;
- **SonarCloud**: Servizio web per eseguire controlli di qualità del codice presente all'interno del repository, riportando tutti i risultati in un cruscotto. Questo strumento è stato integrato con il repository del gruppo in modo tale che ad ogni caricamento venga effettuata l'analisi;
- **SonarLint**: Estensione per IDE che permette di individuare problematiche del codice mentre lo si scrive.

Analisi dinamica del codice

- **Jest**: Framework di testing utilizzato per l'analisi dinamica del codice JavaScript;
- **React Testing Library**: libreria utilizzata in collaborazione con Jest per testare le componenti scritte in React;
- **Puppeteer**: libreria di Node.js per effettuare test di sistema di tipo E2E;
- **Codecov**: servizio utilizzato per stabilire la percentuale del codice coperto dai test. Questo strumento è stato integrato con il repository del gruppo in modo tale che ad ogni caricamento venga effettuata l'analisi;
- **GitHub Actions**: Strumento fornito da GitHub che permette di definire workflows personalizzati all'interno di repository.

Verifica ortografica

La verifica ortografica della documentazione viene effettuata utilizzando i plug-in offerti dagli editor di L^AT_EX utilizzati dal gruppo (*TeXstudio*, *Texmaker*, *Overleaf*).

Indice di Gulpease

È stato creato uno script in *Python* che, dato in $input_G$ un file in formato pdf, restituisce il suo *indice di Gulpease_G*. Il calcolo di tale valore viene effettuato escludendo:

- Frontespizio;
- Registro delle modifiche;
- Indice;
- Elenco delle tabelle (se presente);
- Elenco delle figure (se presente).

3.5 Validazione

3.5.1 Scopo

Scopo del processo di Validazione è normare le modalità tramite le quali il gruppo *BugPharma* ha deciso di attuare il processo di validazione. L'esito positivo della validazione garantisce la conformità del prodotto rispetto alle aspettative.

3.5.2 Aspettative

Obiettivo del presente processo è:

- Accertare la correttezza delle attività di verifica svolte durante tutto il ciclo di vita del progetto;
- Accertare l'efficacia del prodotto finale;
- Assicurare che il prodotto finale soddisfi tutti i requisiti individuati all'interno dell'*Analisi dei Requisiti v3.0.0*.

3.5.3 Descrizione

Il processo di validazione accerta che il prodotto finale sia conforme alle richieste del proponente. Una buona verifica durante l'intero progetto porta al successo della validazione: essa infatti viene eseguita sul prodotto finale, in modo da effettuare un test completo su di esso.

4 Processi organizzativi

4.1 Gestione dei processi

4.1.1 Scopo

Scopo del processo è la creazione del documento denominato *Piano di Progetto*, necessario ai membri del gruppo per l'organizzazione e la gestione dei ruoli di ogni componente.

4.1.2 Aspettative

Le principali aspettative del processo organizzativo di gestione dei processi consistono nella:

- Redazione del documento *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 organizzativo di gestione dei processi sono raccolte nel *Piano di Progetto*, la cui redazione è a carico 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;
- Rischi.

4.1.4 Attività

4.1.4.1 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 qui di seguito descritti.

4.1.4.1.1 Responsabile di Progetto

È la figura professionale punto di riferimento sia per il *committente* sia per il *fornitore* e assume il ruolo di intermediario tra i due. Si assume inoltre la responsabilità delle scelte del gruppo, dopo averle approvate. In particolare, 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;
- Studiare e gestire i rischi.

4.1.4.1.2 Amministratore di Progetto

È la figura professionale incaricata del controllo e dell'amministrazione di tutto l'ambiente di lavoro, con piena responsabilità sulla capacità operativa e sull'efficienza. In particolare, si occupa di:

- 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.

4.1.4.1.3 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 definirne la complessità e i requisiti;
- Redigere l'*Analisi dei Requisiti*.

4.1.4.1.4 Progettista

È la figura professionale che gestisce gli aspetti tecnologici e tecnici del progetto sulla base di competenze costantemente aggiornate. In particolare, si occupa di:

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

4.1.4.1.5 Verificatore

È la figura professionale incaricata del controllo del lavoro svolto dagli altri componenti del gruppo sulla base delle proprie competenze tecniche, esperienza e conoscenza delle norme. In particolare, si occupa di:

- 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.

4.1.4.1.6 Programmatore

È la figura professionale responsabile della codifica del progetto e delle componenti di supporto che serviranno per effettuare le prove di verifica e validazione sul prodotto. In particolare, si occupa di:

- Implementare la *Specifica Tecnica* redatta 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.

4.1.4.2 Gestione delle comunicazioni

4.1.4.2.1 Comunicazioni interne

Le comunicazioni interne riguardano esclusivamente i membri del gruppo e avvengono attraverso i seguenti strumenti:

- *Telegram_G*, il quale:
 - Permette una comunicazione istantanea tra tutti i membri all'interno di un gruppo;
 - Viene utilizzato per la pianificazione degli incontri interni e per discussioni di carattere generale.
- *Slack_G*, il quale:
 - Permette la creazione di uno o più *workspace_G* suddivisibili in canali specifici per ogni attività o task;
 - È integrabile con altri servizi, quali *GitHub_G*, *Jira_G* e *Zoom_G*;
 - Viene utilizzato per le comunicazioni relative al lavoro e, se necessario, per la pianificazione di incontri tra i membri dei singoli canali.

4.1.4.2.2 Comunicazioni esterne

Le comunicazioni esterne sono affidate al *Responsabile di Progetto*, il quale:

- Utilizza, come strumento principale, un'apposita casella di posta elettronica:

bugpharma10@gmail.com

- Ha il compito di tenere informati tutti i componenti del gruppo sugli sviluppi del suo operato attraverso i canali di comunicazione interna precedentemente citati.

4.1.4.3 Gestione degli incontri

4.1.4.3.1 Incontri interni

Gli incontri interni, cui partecipano unicamente i membri del gruppo:

- Possono essere richiesti da ogni componente al *Responsabile di Progetto*;
- Vengono organizzati dal *Responsabile di Progetto*, il quale:
 - Sceglie, in accordo e secondo le disponibilità degli altri membri, data, orario e modalità (fisica o virtuale) dell'incontro;
 - Utilizza i canali di comunicazione interna precedentemente descritti per fissare definitivamente l'evento;
 - Incarica uno o due membri del gruppo alla redazione del verbale dell'incontro secondo le norme previste.
- Possono avvenire in due modalità:
 - **Fisica**: i suddetti si riuniscono di persona, in un luogo fisico, per discutere questioni critiche e conseguentemente prendere decisioni a riguardo;
 - **Virtuale**: i suddetti svolgono chiamate o video chiamate di gruppo per discutere dubbi o eventuali difficoltà sorti nello svolgere determinate attività. Gli incontri di questa tipologia avvengono attraverso lo strumento *DiscordG*, il quale:
 - * Permette la creazione di uno o più server suddivisibili in canali testuali specifici per ogni attività o task, al pari di Slack;
 - * Permette la creazione di canali vocali attraverso i quali i membri possono comunicare con o senza video e, se necessario, condividere lo schermo.

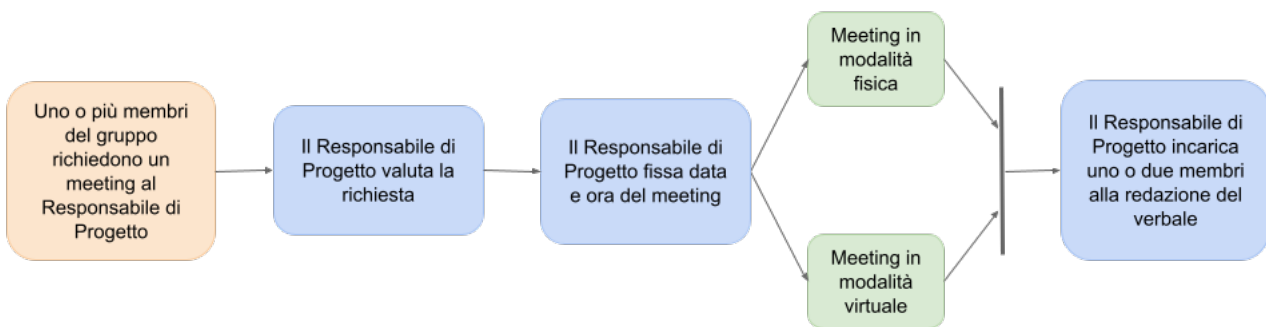


Figura 3: Incontri interni

4.1.4.3.2 Incontri esterni

Gli incontri esterni, cui partecipano i membri del gruppo unitamente al *referente* dell'azienda proponente e/o il *committente*:

- Possono essere richiesti da ogni componente al *Responsabile di Progetto*, il quale valuterà i motivi della richiesta;
- Vengono organizzati e comunicati dal *Responsabile di Progetto*, il quale:
 - Sceglie, in accordo con gli altri membri e con l'ente esterno, e secondo le disponibilità di questi, data, orario e modalità dell'incontro;

- Utilizza i canali di comunicazione interna ed esterna precedentemente descritti per fissare definitivamente l'evento;
 - Incarica uno o due membri del gruppo alla redazione del verbale dell'incontro secondo le norme previste.
- Avvengono in modalità virtuale: i membri del gruppo svolgono video chiamate con il proponente e/o con il committente per discutere dubbi o eventuali difficoltà sorti nello svolgere determinate attività, raccolti principalmente sotto forma di domande. Gli incontri di questa tipologia avvengono attraverso lo strumento *Zoom_G*, il quale permette di effettuare in modo semplice video chiamate con una o più persone.

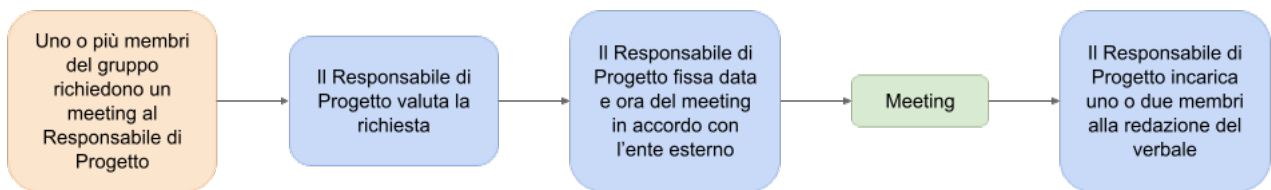


Figura 4: Incontri esterni

4.1.4.3.3 Verballi degli incontri

In seguito ad ogni incontro, di entrambe le tipologie, i membri del gruppo incaricati dal *Responsabile di Progetto* provvedono alla redazione del verbale, secondo le norme descritte in §3.1.

4.1.4.4 Gestione degli strumenti di coordinamento

4.1.4.4.1 Ticketing

Il *Responsabile di Progetto* ha il compito di suddividere il carico di lavoro in task che saranno assegnati ai componenti del gruppo. Tale processo viene svolto attraverso lo strumento *Jira_G*, il quale:

- Permette la visualizzazione di una *roadmap_G* in cui è possibile inserire le principali attività, chiamate *Epic_G*, in cui è suddiviso il progetto. Per ogni *Epic* è possibile visualizzare:
 - **Titolo:** nome identificativo dato all'attività;
 - **Descrizione:** descrizione contenente un breve sunto dell'attività;
 - **Assegnatario e Collaboratori:** persone cui compete lo svolgimento dell'attività;
 - **Etichette:** eventuali etichette per una classificazione rapida dell'attività;
 - **Data di inizio e Data di scadenza:** date entro le quali l'attività deve essere completata;
 - **Commenti:** eventuali commenti lasciati dai componenti del gruppo.

Tale *roadmap* permette di avere una chiara illustrazione dello stato d'avanzamento del progetto nel tempo;

- Permette la visualizzazione di una *board di progetto_G*, in cui è possibile inserire *ticket_G* corrispondenti ai singoli task:
 - Per ogni *ticket* è possibile visualizzare:
 - * **Titolo:** nome identificativo dato al task;

- * **Descrizione:** descrizione contenente un breve sunto del *task*;
 - * **Assegnatario e Collaboratori:** persone cui compete lo svolgimento del *task*;
 - * **Etichette:** eventuali etichette per una classificazione rapida del *task*;
 - * **Priorità:** livello di priorità dato al *task*;
 - * **Stima Originale:** tempo previsto per lo svolgimento del *task*;
 - * **Tracciamento temporale:** tempo effettivamente impiegato per lo svolgimento del *task*;
 - * **Commenti:** eventuali commenti lasciati dai componenti del gruppo.
- È possibile assegnare ad ogni *ticket* uno stato:
- * Da Completare;
 - * In Corso;
 - * Completato.

In seguito alla creazione, assegnazione, modifica o cambiamento di stato di un *ticket*, il sistema provvede ad informare i diretti interessati attraverso una mail ed un messaggio in un apposito canale *Slack*.

4.1.4.5 Gestione dei rischi

Il *Responsabile di Progetto* ha il compito di monitorare i rischi indicati nel *Piano di Progetto* e, in caso ne emergano di nuovi, classificarli e documentarli, prevedendone la gestione. Il processo di gestione dei rischi si sviluppa sulla base dei seguenti punti:

- Verifica periodica dello stato dei rischi già presenti nel *Piano di Progetto*;
- Individuazione dei problemi non considerati in precedenza;
- Aggiunta al *Piano di Progetto* dei nuovi rischi individuati;
- Registrazione degli aggiornamenti su tutti i rischi, sia nuovi sia già presenti;
- Se necessario, ridefinizione delle strategie di progetto per la loro gestione.

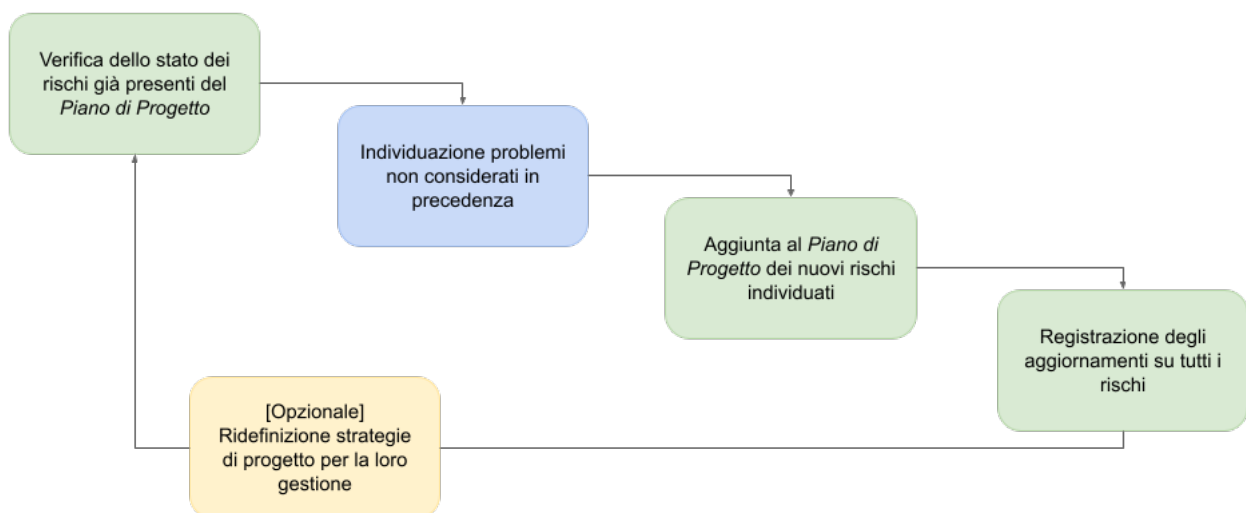


Figura 5: Gestione dei rischi

4.1.4.5.1 Codifica dei rischi

I rischi sono codificati nel modo seguente:

R[Iniziale categoria][Numero progressivo]

Dove:

- **Iniziale categoria:** indica la classificazione del rischio che può appartenere ad una delle seguenti categorie, individuate in base al valore assunto:
 - **T:** rischi correlati alle tecnologie scelte;
 - **I:** rischi correlati ai rapporti interpersonali;
 - **O:** rischi correlati all'organizzazione del lavoro.
- **Numero progressivo:** indica il numero progressivo che identifica univocamente il rischio all'interno di una categoria.

4.1.5 Metriche

Il processo di gestione dei processi non fa uso di metriche qualitative particolari.

4.1.6 Strumenti

I componenti del gruppo lavorano sui seguenti sistemi operativi:

- Windows 10 Home x64;
- Windows 10 Pro x64;
- Ubuntu 20.04 LTS x64;
- Kubuntu 20.04 (con Desktop Environment KDE Plasma 5.18.5) x64.

Fanno inoltre uso dei seguenti strumenti, nelle modalità descritte nei paragrafi precedenti:

- **Telegram;**
- **Slack;**
- **Gmail;**
- **Discord;**
- **Zoom;**
- **Jira;**

4.2 Formazione

4.2.1 Scopo

I membri del gruppo hanno il compito di informarsi e formarsi in modo autonomo sulle tecnologie e sugli strumenti necessari al completamento del progetto.

4.2.2 Aspettative

- Ottenere una buona conoscenza di \LaTeX ;
- Ottenere una buona conoscenza delle librerie, degli strumenti e del linguaggio utilizzati per la codifica.

4.2.3 Descrizione

Il processo di formazione è un processo per fornire e mantenere i componenti del gruppo qualificati.

4.2.4 Attività

4.2.4.1 Materiale per la formazione

Di seguito sono elencate i linguaggi di programmazione e le tecnologie necessarie allo svolgimento del capitolato con i relativi link ai siti ufficiali.

4.2.4.1.1 Linguaggi di programmazione

- \LaTeX : <https://www.latex-project.org/>;
- JavaScript: <https://www.javascript.com/>;
- HTML e CSS: <https://www.w3schools.com/>.

4.2.4.1.2 Tecnologie

- React: <https://it.reactjs.org/>;
- D3.js: <https://d3js.org/>;
- DruidJS: <https://github.com/saehm/DruidJS>.

4.2.5 Metriche

Il processo di formazione non fa uso di metriche qualitative particolari.

4.2.6 Strumenti

Il processo di formazione non fa uso di strumenti particolari.