



Università degli Studi di Padova



BugPharma - *Login Warrior*

E-mail: bugpharma10@gmail.com

Norme di Progetto

Versione	1.0.0
Approvazione	Lorenzo Piran
Redazione	Andrea Salmaso, Nicla Faccioli Nicholas Sertori, Sara Nanni
Verifica	Michele Masetto, Nicla Faccioli
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
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
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	Valutazione dei Capitolati	9
2.1.3.2	Piano di Progetto	10
2.1.3.3	Piano di Qualifica	11
2.1.3.4	Collaudo e Rilascio del prodotto	11
2.2	Sviluppo	11
2.2.1	Scopo	11
2.2.2	Aspettative	11
2.2.3	Descrizione	12
2.2.3.1	Analisi dei Requisiti	12
2.2.3.1.1	Scopo	12
2.2.3.1.2	Aspettative	12
2.2.3.1.3	Requisiti	12
2.2.3.1.4	Denominazione dei Requisiti	12
2.2.3.1.5	Casi d'uso	13
2.2.3.1.6	Denominazione dei Casi d'uso	13
2.2.3.2	Progettazione	13
2.2.3.2.1	Scopo	13
2.2.3.2.2	Aspettative	14
2.2.3.2.3	Descrizione	14
2.2.3.2.4	Technology Baseline	14
2.2.3.2.5	Product Baseline	14
2.2.3.3	Codifica	14
2.2.3.3.1	Scopo	15
2.2.3.3.2	Aspettative	15
2.2.3.3.3	Descrizione	15
3	Processi di supporto	16
3.1	Documentazione	16
3.1.1	Scopo	16
3.1.2	Ciclo di vita del documento	16
3.1.3	Struttura dei documenti	16
3.1.3.1	Frontespizio	16
3.1.3.2	Registro delle modifiche	17

3.1.3.3	Indice	17
3.1.3.4	Contenuto principale	18
3.1.3.5	Eccezioni	18
3.1.4	Classificazione dei documenti	18
3.1.4.1	Documenti ad uso interno	18
3.1.4.2	Documenti ad uso esterno	18
3.1.5	Norme tipografiche	19
3.1.5.1	Nome dei file e versionamento	19
3.1.5.2	Stile del testo	19
3.1.5.3	Elenchi puntati	20
3.1.5.4	Glossario	20
3.1.5.5	Sigle	20
3.1.6	Elementi Grafici	21
3.1.6.1	Tabelle	21
3.1.6.2	Immagini	21
3.1.7	Strumenti	21
3.1.7.1	L ^A T _E X	21
3.1.7.2	Google Docs	21
3.1.7.3	StarUML	21
3.2	Gestione della configurazione	21
3.2.1	Repository	21
3.2.1.1	Tecnologie e strumenti	21
3.2.1.2	Struttura del repository	22
3.2.1.3	Gestione delle modifiche	22
3.2.1.4	Tipi di file e <code>.gitignore</code>	22
3.3	Gestione della qualità	22
3.3.1	Scopo	22
3.3.2	Aspettative	22
3.3.3	Descrizione	23
3.3.4	Denominazione metriche	23
3.4	Verifica	23
3.4.1	Scopo	23
3.4.2	Aspettative	23
3.4.3	Descrizione	23
3.4.4	Analisi statica	23
3.4.5	Analisi dinamica	24
3.4.5.1	Test	24
3.4.5.1.1	Test di unità	24
3.4.5.1.2	Test di integrazione	24
3.4.5.1.3	Test di sistema	24
3.4.5.1.4	Test di regressione	24
3.4.5.1.5	Test di accettazione - collaudo	24
3.4.6	Strumenti	25
3.4.6.1	Verifica ortografica	25
3.4.6.2	Indice di Gulpease	25
3.5	Validazione	25
3.5.1	Scopo	25
3.5.2	Aspettative	26
3.5.3	Descrizione	26

4	Processi organizzativi	27
4.1	Gestione dei processi	27
4.1.1	Scopo	27
4.1.2	Aspettative	27
4.1.3	Descrizione	27
4.1.4	Ruoli di progetto	27
4.1.4.1	Responsabile di Progetto	27
4.1.4.2	Amministratore di Progetto	28
4.1.4.3	Analista	28
4.1.4.4	Progettista	28
4.1.4.5	Verificatore	28
4.1.4.6	Programmatore	29
4.1.5	Procedure	29
4.1.5.1	Gestione delle comunicazioni	29
4.1.5.1.1	Comunicazioni interne	29
4.1.5.1.2	Comunicazioni esterne	29
4.1.5.2	Gestione degli incontri	29
4.1.5.2.1	Incontri interni	30
4.1.5.2.2	Incontri esterni	30
4.1.5.2.3	Verbali degli incontri	31
4.1.5.3	Gestione degli strumenti di coordinamento	31
4.1.5.3.1	Ticketing	31
4.1.5.4	Gestione dei rischi	32
4.1.5.4.1	Codifica dei rischi	33
4.1.6	Strumenti	33
4.1.7	Formazione	33
A	Standard ISO/IEC 12207:1997	34
A.1	Processi primari	34
A.1.1	Acquisizione	34
A.1.2	Fornitura	34
A.1.3	Sviluppo	35
A.1.4	Operation	35
A.1.5	Manutenzione	35
A.2	Processi di supporto	36
A.2.1	Documentazione	36
A.2.2	Gestione della configurazione	36
A.2.3	Accertamento della qualità	36
A.2.4	Verifica	37
A.2.5	Validazione	37
A.2.6	Revisione congiunta	37
A.2.7	Audit	37
A.2.8	Risoluzione dei problemi	37
A.3	Processi organizzativi	38
A.3.1	Gestione	38
A.3.2	Infrastrutture	38
A.3.3	Miglioramento	38
A.3.4	Formazione	38

B Standard di qualità ISO/IEC 9126	39
B.1 Modello di qualità	39
B.1.1 Qualità esterna ed interna	39
B.1.1.1 Funzionalità	39
B.1.1.2 Affidabilità	39
B.1.1.3 Usabilità	40
B.1.1.4 Efficienza	40
B.1.1.5 Manutenibilità	40
B.1.1.6 Portabilità	41
B.1.2 Qualità in uso	41
B.2 Metriche per la qualità esterna	41
B.3 Metriche per la qualità interna	41
B.4 Metriche per la qualità in uso	41

Elenco delle figure

1	Ciclo di vita dei documenti	16
2	Tipi di test del codice	25
3	Incontri interni	30
4	Incontri esterni	31
5	Gestione dei rischi	32

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 v1.0.01.0.0* 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>
- ISO/IEC 12207:1997:
https://www.math.unipd.it/~tullio/IS-1/2009/Approfondimenti/ISO_12207-1995.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 9126:
http://www.colonese.it/00-Manuali_Pubblicatii/07-ISO-IEC9126_v2.pdf

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*, in parte identificate da documenti:

2.1.3.1 Valutazione dei Capitolati

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. Il documento contenente le informazioni estrapolate da tali incontri, chiamato *Valutazione dei Capitolati*, è redatto da uno o più membri ricoprenti il ruolo di *Analista*, ed 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.

Per il capitolato scelto, oltre ai suddetti, la *Valutazione dei Capitolati* espone anche gli aspetti positivi in esso individuati, i quali hanno spinto i componenti del gruppo alla scelta definitiva.

2.1.3.2 Piano di Progetto

Il *Responsabile di Progetto*, con l'aiuto degli *Amministratori*, ha il compito di pianificare le *attività_G* necessarie alla guida del processo durante tutte le *fasi_G* del progetto, organizzando e distribuendo le risorse lungo vari periodi. Il documento contenente tali informazioni è il *Piano di Progetto*, il quale 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. Essi sono identificati univocamente dal seguente codice:

R[Iniziale categoria][Numero progressivo].

Con:

- **R:** indica che si tratta di un requisito;
- **Iniziale categoria:** iniziale della categoria a cui appartiene. In particolare:
 - * **T** : Tecnologie scelte;
 - * **I** : Rapporti Interpersonali;
 - * **O** : Organizzazione del lavoro.
- **Numero progressivo.**
- **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 attraverso:
 - **Identificativo:** codice univoco composto secondo il seguente schema:

[Categoria]-[Numero][Fase]

Con:

- * **Categoria:** categoria in cui viene classificato il lasso temporale, in particolare:
 - **PER:** indica un periodo;
 - **INC:** indica un incremento.
- * **Numero:** numero progressivo, differente per fasi e incrementi. In particolare ha:
 - Se associato ad una fase: ambito circoscritto alla Fase cui fa riferimento;
 - Se associato ad un incremento: ambito globale in riferimento a tutti gli incrementi.
- * **Fase:** codice identificativo della fase, in particolare:
 - **A:** indica la fase di Analisi;
 - **PTB:** indica la fase di Progettazione della Technology *Baseline_G*;
 - **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 o incremento attuale;
- **Post-condizione:** condizioni decretanti il completamento delle attività previste per il periodo o incremento attuale;

- **Ruoli attivi:** ruoli lavorativi previsti per lo svolgimento delle attività;
 - **Descrizione/Obiettivo:** descrizione generale del periodo o obiettivi da raggiungere associati all'incremento;
 - **Attività previste.**
- **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.

2.1.3.3 Piano di Qualifica

I membri del gruppo ricoprono il ruolo di *Verificatore*, 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 contenente tali informazioni è il *Piano di Qualifica*, il quale espone:

- **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.

2.1.3.4 Collaudo e Rilascio del prodotto

Il gruppo avrà il compito di 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* presenterà al *committente* il consuntivo finale e potrà procedere alla consegna del codice sorgente e della relativa documentazione. Ad avvenuto rilascio del prodotto, non è prevista attività di *manutenzione_G* dello stesso.

2.2 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 Descrizione

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

2.2.3.1 Analisi dei Requisiti

2.2.3.1.1 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*, il quale espone:

- **Descrizione generale del prodotto:** 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.

2.2.3.1.2 Aspettative

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

2.2.3.1.3 Requisiti

Ogni requisito è strutturato come segue:

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

2.2.3.1.4 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]

2.2.3.1.5 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).

2.2.3.1.6 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]

2.2.3.2 Progettazione

2.2.3.2.1 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*.

2.2.3.2.2 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.

2.2.3.2.3 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 di un *PoC_G* e di una conseguente *Technology Baseline*;
- **Progettazione Architetture:** nella quale verrà 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 verrà 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*.

2.2.3.2.4 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*;

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

- Design 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.

2.2.3.3 Codifica

2.2.3.3.1 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.

2.2.3.3.2 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.

2.2.3.3.3 Descrizione

Per soddisfare le aspettative di cui sopra, i *Programmatori* dovranno adottare le regole e convenzioni che verranno in futuro descritte e approfondite qui di seguito.

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 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 verificata 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.

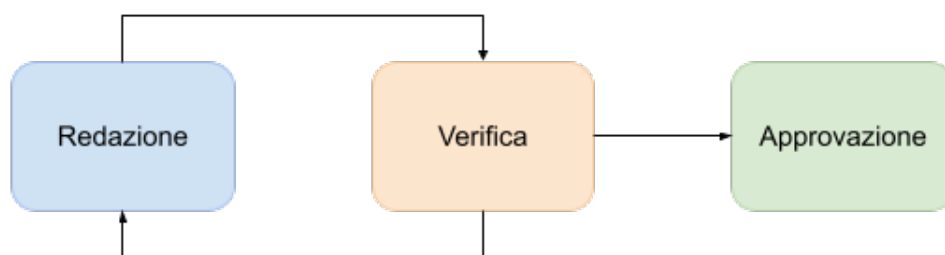


Figura 1: Ciclo di vita dei documenti

3.1.3 Struttura dei documenti

Tutti i documenti ufficiali che vengono prodotti devono seguire una struttura precisa per consentire omogeneità e coesione.

3.1.3.1 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.

3.1.3.2 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.

3.1.3.3 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.

3.1.3.4 Contenuto principale

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.

3.1.3.5 Eccezioni

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.

Un'altra eccezione riguarda 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.

3.1.4 Classificazione dei documenti

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

3.1.4.1 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*.

3.1.4.2 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*.

3.1.5 Norme tipografiche

3.1.5.1 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.

3.1.5.2 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.

3.1.5.3 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.

3.1.5.4 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.

3.1.5.5 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*.

3.1.6 Elementi Grafici

3.1.6.1 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).

3.1.6.2 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.

3.1.7 Strumenti

3.1.7.1 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.

3.1.7.2 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.

3.1.7.3 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 Repository

3.2.1.1 Tecnologie e strumenti

Per il progetto viene utilizzato il sistema di controllo di versione distribuito *Git*_G, nello specifico con il servizio *GitHub*_G. Per fare uso del *repository*_G (anche abbreviato in *repo*), i membri del gruppo sono stati lasciati liberi di usare lo strumento a loro più consono, a patto di mantenere la struttura interna qui di seguito descritta.

3.2.1.2 Struttura 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 *Progetto-SWE-2021***, rappresenta un'interfaccia tra il gruppo ed i proponenti e contiene esclusivamente i documenti destinati alla consultazione da parte di esterni, in formato *.pdf*, insieme al 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.

3.2.1.3 Gestione delle modifiche

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.

3.2.1.4 Tipi di file e *.gitignore*

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.

3.3 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 v1.0.0*.

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 v1.0.0* 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 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.

3.4 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 v1.0.0*.

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 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.

3.4.5 Analisi dinamica

L'analisi dinamica è applicabile solo a codice che può essere eseguito e consiste nell'attività di $test_G$.

3.4.5.1 Test

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.

3.4.5.1.1 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.

3.4.5.1.2 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.

3.4.5.1.3 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.

3.4.5.1.4 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.

3.4.5.1.5 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.

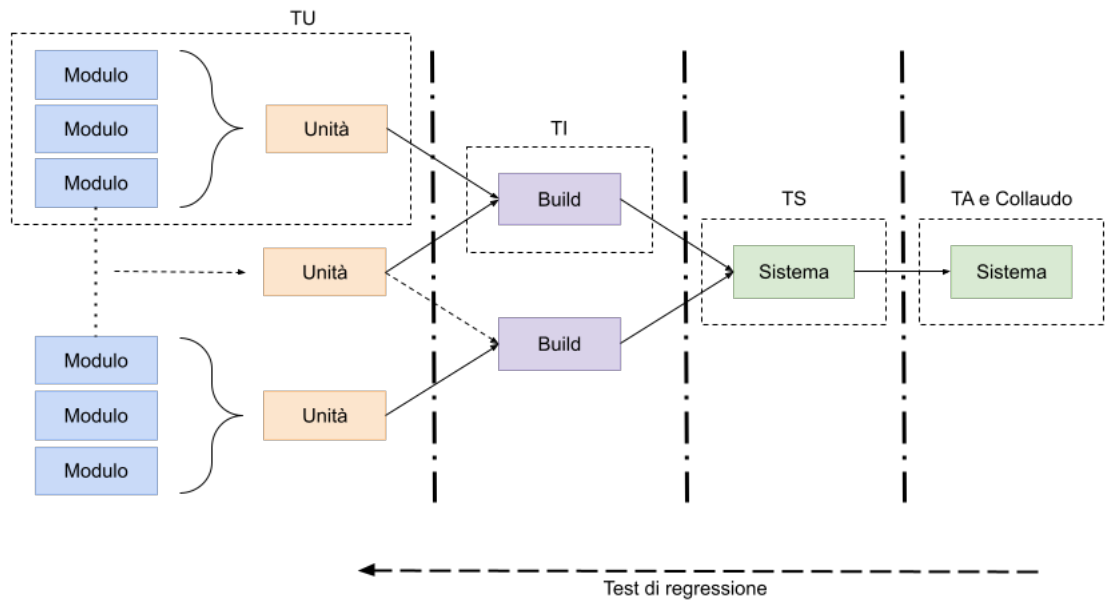


Figura 2: Tipi di test del codice

3.4.6 Strumenti

Non sono ancora stati definiti gli strumenti da utilizzare per la verifica del codice.

Di seguito sono quindi riportati esclusivamente gli strumenti utilizzati per la verifica della documentazione.

3.4.6.1 Verifica ortografica

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

3.4.6.2 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 v1.0.0*.

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 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 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.

4.1.4.2 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.

4.1.4.3 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*.

4.1.4.4 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*.

4.1.4.5 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.

4.1.4.6 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.

4.1.5 Procedure

4.1.5.1 Gestione delle comunicazioni

4.1.5.1.1 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.

4.1.5.1.2 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.

4.1.5.2 Gestione degli incontri

4.1.5.2.1 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.

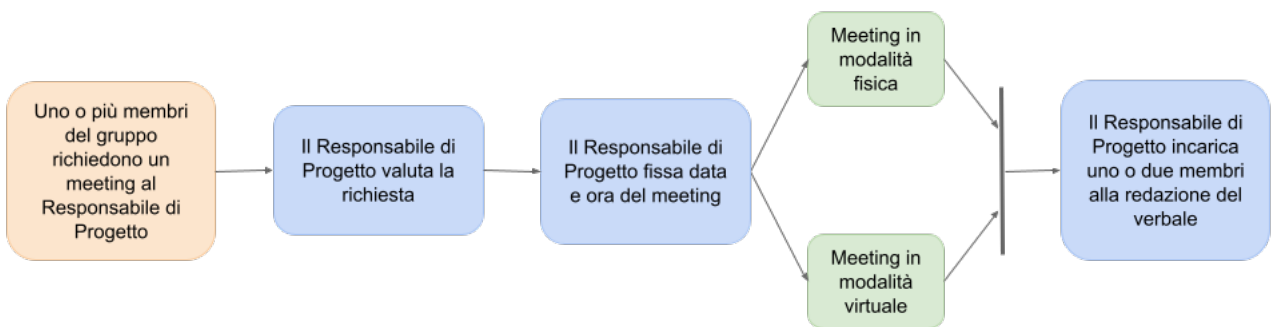


Figura 3: Incontri interni

4.1.5.2.2 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.

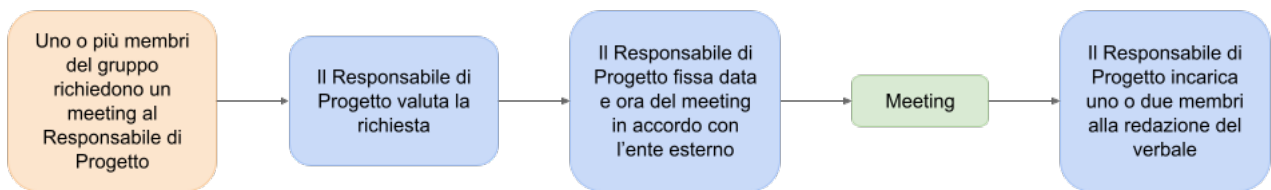


Figura 4: Incontri esterni

4.1.5.2.3 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.

4.1.5.3 Gestione degli strumenti di coordinamento

4.1.5.3.1 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*.

4.1.5.4 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.

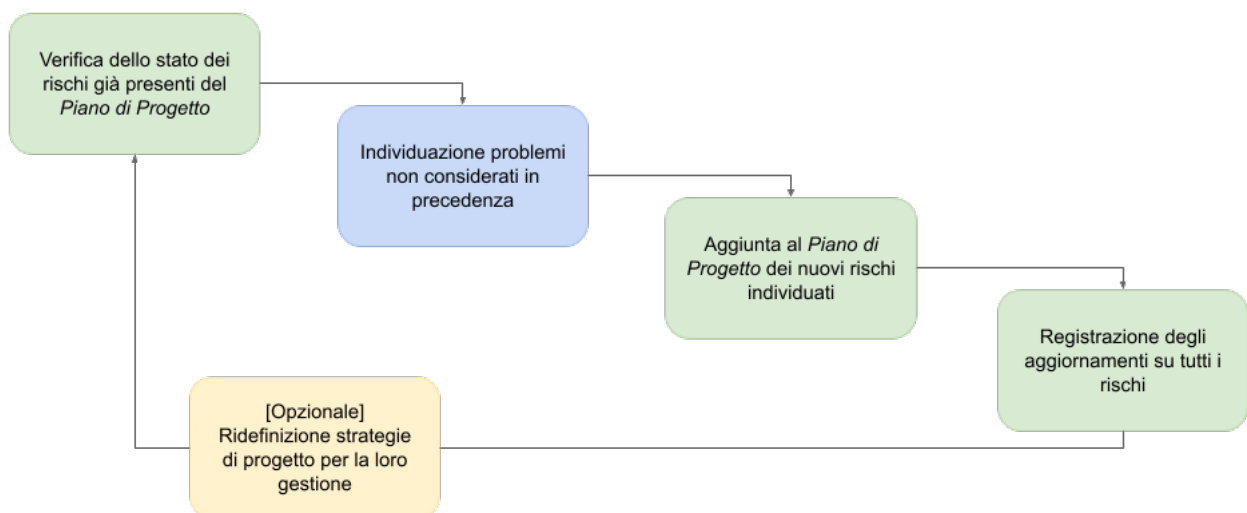


Figura 5: Gestione dei rischi

4.1.5.4.1 Codifica dei rischi

I rischi sono codificati nel modo seguente:

R[Iniziale categoria][Numero]

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:** indica il numero progressivo che identifica univocamente il rischio all'interno di una categoria.

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;**

4.1.7 Formazione

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

A Standard ISO/IEC 12207:1997

ISO/IEC 12207 è uno standard internazionale che definisce tutti i *processi*_G necessari per lo sviluppo e la manutenzione di sistemi software. Per ogni processo esso stabilisce inoltre le *attività*_G da svolgere ed i risultati da produrre.

I processi sono divisi in tre categorie:

- Processi primari;
- Processi di supporto;
- Processi organizzativi.

A.1 Processi primari

A.1.1 Acquisizione

Il processo descrive le attività dell'acquirente. Esso ha inizio con la necessità di un sistema, un *prodotto*_G software o un servizio.

Si compone delle seguenti attività:

- Iniziazione;
- Preparazione della richiesta di proposta;
- Preparazione e aggiornamento del contratto;
- Monitoraggio dei fornitori;
- Accettazione e completamento.

A.1.2 Fornitura

Il processo descrive le attività del *fornitore*_G. Esso può avere inizio in due occasioni differenti:

- Decisione di presentare una proposta in risposta alla richiesta del acquirente;
- Al momento della firma del contratto per la fornitura di un sistema, prodotto software o servizio.

Si compone delle seguenti attività:

- Iniziazione;
- Preparazione di una risposta;
- Contratto;
- Pianificazione;
- Esecuzione e controllo;
- Revisione e valutazione;
- Consegna e completamento.

A.1.3 Sviluppo

Il processo descrive le attività dello sviluppatore. Esso norma lo sviluppo di un sistema software che risponda alle necessità del cliente.

Si compone delle seguenti attività:

- Implementazione del processo;
- Analisi dei Requisiti di sistema;
- Progettazione architetturale del sistema;
- Analisi dei requisiti software;
- Progettazione architetturale del software;
- Progettazione di dettaglio del software;
- Codifica e testing del software;
- Integrazione software;
- Collaudo del software;
- Integrazione del sistema;
- Collaudo del sistema;
- Installazione software.

A.1.4 Operation

Il processo si svolge in contemporanea alla *fase_G* di Manutenzione. Ha lo scopo di mantenere operativo il sistema e di fornire supporto agli utenti.

Si compone delle seguenti attività:

- Implementazione del processo;
- Operational testing;
- System operation;
- Supporto dell'utente.

A.1.5 Manutenzione

Il processo descrive le attività del manutentore. Esso ha inizio quando il prodotto software viene sottoposto a modifiche del codice o della documentazione associata a causa di difetti o miglioramenti. Questo processo si conclude con il ritiro del prodotto software.

Si compone delle seguenti attività:

- Implementazione del processo;
- Analisi del problema o della modifica;
- Implementazione della modifica;
- Revisione/accettazione della manutenzione;
- Migrazione;
- Ritiro del software.

A.2 Proessi di supporto

I processi di supporto hanno lo scopo di aiutare tutti gli altri processi attivi. Essi possono essere attivati da un processo primario o da un altro processo di supporto.

A.2.1 Documentazione

Il processo ha lo scopo di registrare le informazioni prodotte da un'attività o da un processo del *ciclo di vita*_G di un prodotto software.

Si compone delle seguenti attività:

- Implementazione del processo;
- Progettazione e sviluppo;
- Produzione;
- Manutenzione.

A.2.2 Gestione della configurazione

Il processo ha lo scopo di definire e mantenere l'integrità di tutti i componenti della configurazione e di renderli accessibili a chi ne ha diritto.

Si compone delle seguenti attività:

- Implementazione del processo;
- Identificazione della configurazione;
- Controllo della configurazione;
- Registrazione dello stato della configurazione;
- Valutazione della configurazione;
- Gestione del rilascio e distribuzione.

A.2.3 Accertamento della qualità

Il processo ha lo scopo di assicurare che tutti i prodotti e i processi del ciclo di vita siano conformi con gli standard ed i *requisiti*_G definiti.

Si compone delle seguenti attività:

- Implementazione del processo;
- Accertamento del prodotto;
- Accertamento del processo;
- Accertamento dei sistemi di *qualità*_G.

A.2.4 Verifica

Il processo ha lo scopo di determinare se i prodotti software di un'attività soddisfano i requisiti o le condizioni imposte dalle attività precedenti. Il processo di *Verifica_G* dovrebbe essere integrato nei processi di Sviluppo, Fornitura e Manutenzione il prima possibile.

Si compone delle seguenti attività:

- Implementazione del processo;
- Verifica.

A.2.5 Validazione

Il processo ha lo scopo di accertarsi che il prodotto software o il sistema rispettino tutti i requisiti individuati.

Si compone delle seguenti attività:

- Implementazione del processo;
- *Validazione_G*.

A.2.6 Revisione congiunta

Il processo ha lo scopo di revisionare lo stato ed i prodotti delle attività rispetto agli obiettivi definiti negli accordi, sia a livello di progetto che a livello tecnico. Tali revisioni avvengono durante l'intero ciclo di vita. Ad esse partecipano le parti interessate: solo il team se si revisiona un singolo componente o con la partecipazione del *committente_G* per revisioni sull'intero prodotto.

Si compone delle seguenti attività:

- Implementazione del processo;
- Revisioni della gestione del progetto;
- Revisioni tecniche.

A.2.7 Audit

Il processo ha lo scopo di determinare se vi è conformità dei prodotti e dei processi rispetto a requisiti, pianificazione e accordi. L'attività di auditing è svolta da personale che non ha partecipato direttamente allo sviluppo dei prodotti, servizi o sistemi oggetto delle revisioni.

Si compone delle seguenti attività:

- Implementazione del processo;
- Audit.

A.2.8 Risoluzione dei problemi

Il processo ha lo scopo di analizzare e risolvere i problemi che vengono individuati in modo responsabile e documentato.

Si compone delle seguenti attività:

- Implementazione del processo;
- Risoluzione del problema.

A.3 Processi organizzativi

I processi organizzativi sono di responsabilità dell'organizzazione che li attiva. Essi hanno lo scopo di stabilire e implementare una struttura e gestire i processi del ciclo di vita e del personale. Inoltre sono utili per la gestione del continuo miglioramento dei processi e della struttura stessa.

A.3.1 Gestione

Il processo contiene le seguenti attività generiche per la gestione dei processi:

- Inizializzazione e definizione del raggio d'azione;
- Pianificazione;
- Esecuzione e controllo;
- Revisione e valutazione;
- Terminazione.

A.3.2 Infrastrutture

Il processo ha lo scopo di istituire e mantenere le infrastrutture necessarie a qualsiasi altro processo. L'infrastruttura può comprendere: hardware, software, strumenti, tecniche, standard e funzionalità per sviluppo, operazioni o manutenzione.

Si compone delle seguenti attività:

- Implementazione del processo;
- Istituzione dell'infrastruttura;
- Manutenzione dell'infrastruttura.

A.3.3 Miglioramento

Il processo ha lo scopo di istituire, valutare, misurare, controllare e migliorare un processo di ciclo di vita del software.

Si compone delle seguenti attività:

- Istituzione del processo;
- Valutazione del processo;
- Miglioramento del processo.

A.3.4 Formazione

Il processo ha lo scopo di fornire e mantenere personale istruito. L'acquisizione, la fornitura, lo sviluppo e molti altri processi dipendono da personale esperto e competente.

Si compone delle seguenti attività:

- Implementazione del processo;
- Sviluppo del materiale per la formazione;
- Implementazione del piano di formazione.

B Standard di qualità ISO/IEC 9126

ISO/IEC 9126 descrive un modello di *qualità_G* del software, definisce le caratteristiche che la determinano e propone *metriche_G* per la misurazione.

Questo standard si divide in quattro parti:

B.1 Modello di qualità

Questa prima parte del modello si compone a sua volta di due parti:

B.1.1 Qualità esterna ed interna

In questa sezione vengono descritte sei caratteristiche e varie sotto-caratteristiche per la qualità esterna ed interna, di seguito riportate.

B.1.1.1 Funzionalità

Capacità del *prodotto_G* software di fornire funzioni che soddisfino esigenze espresse ed implicite, necessarie ad operare in specifiche condizioni.

- **Adeguatezza:** capacità del prodotto software di fornire un appropriato insieme di funzioni per specifici task e obiettivi dell'utente;
- **Accuratezza:** capacità del prodotto software di fornire i risultati o effetti corretti o stabiliti con un certo grado di precisione;
- **Interoperabilità:** capacità del prodotto software di interagire con uno o più sistemi specificati;
- **Sicurezza:** la capacità del prodotto software di proteggere informazioni e dati, negandone l'accesso e la modifica a persone e sistemi non autorizzati ma permettendola a chi è abilitato;
- **Conformità:** capacità del prodotto software di aderire a standard, convenzioni e regolamenti di carattere legale o simili che siano attinenti alle funzionalità offerte.

B.1.1.2 Affidabilità

Capacità del prodotto software di mantenere livelli di performance specificati quando utilizzato sotto specifiche condizioni.

- **Maturità:** capacità del prodotto software di evitare che si verifichino errori o siano prodotti risultati non corretti;
- **Tolleranza agli errori:** capacità del prodotto software di mantenere livelli predeterminati di prestazioni anche in presenza di malfunzionamenti o usi scorretti del prodotto;
- **Recuperabilità:** capacità del prodotto software di ripristinare il livello appropriato di prestazioni e di recuperare le informazioni rilevanti in seguito a un malfunzionamento. A seguito di un errore, il software può risultare non accessibile per un determinato periodo di tempo. Questo arco di tempo è valutato proprio dalla caratteristica di recuperabilità;
- **Aderenza:** capacità del prodotto software di aderire a standard, regole e convenzioni relative all'affidabilità.

B.1.1.3 Usabilità

Capacità del prodotto software di essere comprensibile, apprendibile, usabile e attraente per l'utente quando utilizzato sotto condizioni specifiche.

- **Comprensibilità:** capacità del prodotto software di rendere l'utente in grado di capire le sue funzionalità e di come poterle utilizzare per svolgere determinati task.
- **Apprendibilità:** capacità del prodotto software di permettere all'utente di imparare l'applicazione;
- **Operabilità:** capacità del prodotto software di permettere agli utenti di utilizzarlo e controllarlo;
- **Attrattività:** capacità del prodotto software di risultare attraente per l'utente;
- **Aderenza all'usabilità:** capacità del software di aderire a standard o convenzioni relative all'usabilità.

B.1.1.4 Efficienza

Capacità del prodotto software di fornire prestazioni appropriate, relativamente alla quantità di risorse usate, sotto determinate condizioni.

- **Comportamento rispetto al tempo:** capacità del prodotto software di fornire adeguati tempi di risposta e di elaborazione sotto specifiche condizioni;
- **Utilizzo delle risorse:** capacità del prodotto software di utilizzare un numero e tipo di risorse appropriato quando esegue le funzionalità previste sotto determinate condizioni di utilizzo;
- **Conformità:** capacità del prodotto software di aderire a standard e convenzioni relative all'efficienza.

B.1.1.5 Manutenibilità

Capacità del prodotto software di essere modificato, includendo correzioni, miglioramenti o adattamenti.

- **Analizzabilità:** capacità di poter effettuare la diagnosi sul prodotto software ed individuare le cause di errori o malfunzionamenti;
- **Modificabilità:** capacità del prodotto software di permettere lo sviluppo di modifiche al prodotto originale (codice, progettazione, documentazione);
- **Stabilità:** capacità del prodotto software di evitare effetti indesiderati derivanti da modifiche del software;
- **Testabilità:** capacità del prodotto software di consentire la *verifica_G* e *validazione_G* del software modificato;
- **Aderenza alla manutenibilità:** capacità del prodotto software di aderire a standard e convenzioni relative alla *manutenibilità_G*.

B.1.1.6 Portabilità

Capacità del prodotto software di essere trasportato da un ambiente di lavoro ad un altro.

- **Adattabilità:** capacità del prodotto software di essere adattato a differenti ambienti senza richiedere azioni specifiche diverse da quelle previste dal software per tali attività;
- **Installabilità:** capacità del prodotto software di essere installato in uno specifico ambiente;
- **Coesistenza:** capacità del prodotto software di coesistere con altre applicazioni indipendenti in ambienti comuni e condividere risorse;
- **Sostituibilità:** capacità del prodotto software di sostituire un altro software specifico per svolgere gli stessi compiti nello stesso ambiente;
- **Aderenza alla portabilità:** capacità del prodotto software di aderire a standard e convenzioni relative alla portabilità.

B.1.2 Qualità in uso

In questa sezione vengono specificate quattro caratteristiche per la qualità in uso.

- **Efficacia:** capacità del prodotto software di permettere all'utente di raggiungere gli obiettivi specifici con accuratezza e completezza;
- **Produttività:** capacità del prodotto software di permettere all'utente di impegnare un numero definito di risorse, in relazione all'efficacia raggiunta, in uno specifico contesto d'uso;
- **Sicurezza fisica:** rappresenta la capacità del prodotto software di raggiungere un livello accettabile di rischio per i dati, le persone, il business, le proprietà o gli ambienti in uno specifico contesto di utilizzo;
- **Soddisfazione:** capacità del prodotto software di soddisfare gli utenti in uno specifico contesto di utilizzo.

B.2 Metriche per la qualità esterna

La seconda parte dello standard definisce le metriche per misurare quantitativamente la qualità esterna del prodotto software rispetto alle caratteristiche definite nella prima parte. Con qualità esterna si intendono i comportamenti del prodotto software rilevabili dai *test_G* e dall'osservazione durante la sua esecuzione.

B.3 Metriche per la qualità interna

La terza parte dello standard definisce le metriche per misurare quantitativamente la qualità interna del prodotto software rispetto alle caratteristiche definite nella prima parte. Le metriche interne si applicano al software non eseguibile durante le fasi di progettazione e codifica e misurano qualità intrinseche del prodotto. Esse permettono di individuare eventuali problemi che potrebbero influire sulla qualità finale del prodotto prima che sia realizzato il software eseguibile.

B.4 Metriche per la qualità in uso

La quarta parte dello standard definisce le metriche per misurare quantitativamente la qualità in uso del prodotto software rispetto alle caratteristiche definite nella prima parte. Esse misurano il grado con cui il prodotto software permette agli utenti di svolgere le proprie attività con efficacia, produttività, sicurezza e soddisfazione nel contesto operativo previsto. La qualità in uso rappresenta la vista esterna che l'utente ha del prodotto ed è misurata in base ai risultati ottenuti dal suo utilizzo.