



Università degli Studi di Padova



BugPharma - *Login Warrior*

E-mail: bugpharma10@gmail.com

## Piano di Qualifica

<b>Versione</b>	1.0.0
<b>Approvazione</b>	Lorenzo Piran
<b>Redazione</b>	Nicla Faccioli, Lorenzo Piran
<b>Verifica</b>	Silvia Giro, Sara Nanni Andrea Salmaso
<b>Stato</b>	Approvato
<b>Uso</b>	Esterno
<b>Distribuzione</b>	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	15/02/2022	Verifica del documento	Andrea Salmaso	Verificatore
0.2.2	14/02/2022	Stesura appendice B	Michele Masetto	Amministratore
0.2.1	14/02/2022	Stesura appendice A	Sara Nanni	Verificatore
0.2.0	21/01/2022	Verifica del documento	Sara Nanni	Verificatore
0.1.2	08/01/2022	Stesura \$4	Lorenzo Piran	Responsabile
0.1.1	05/01/2022	Stesura \$2.2.2.2 e \$2.2.2.3	Nicla Faccioli	Responsabile
0.1.0	03/01/2022	Verifica del documento	Silvia Giro	Verificatore
0.0.7	27/12/2021	Stesura \$2.3.3.1-1, \$2.3.3.1-2 e \$2.3.3.1-3, \$2.3.3.1-4	Lorenzo Piran	Amministratore
0.0.6	26/12/2021	Stesura \$3.3.3, \$3.3.4 e \$3.3.5, \$3.3.6, \$3.3.7	Lorenzo Piran	Amministratore
0.0.5	23/12/2021	Stesura \$2.2.2.3, \$2.3.2 e \$2.3.3	Nicla Faccioli	Responsabile
0.0.4	08/12/2021	Stesura \$3	Lorenzo Piran	Amministratore
0.0.3	07/12/2021	Stesura \$2	Nicla Faccioli	Responsabile
0.0.2	06/12/2021	Stesura \$1	Nicla Faccioli	Responsabile
0.0.1	05/12/2021	Creazione documento	Nicla Faccioli	Responsabile

## Indice

<b>1</b>	<b>Introduzione</b>	<b>6</b>
1.1	Scopo del documento . . . . .	6
1.2	Scopo del prodotto . . . . .	6
1.3	Glossario . . . . .	6
1.4	Maturità del Documento . . . . .	6
1.5	Riferimenti . . . . .	7
1.5.1	Riferimenti normativi . . . . .	7
1.5.2	Riferimenti informativi . . . . .	7
<b>2</b>	<b>Qualità di processo</b>	<b>8</b>
2.1	Scopo . . . . .	8
2.2	Processi primari . . . . .	8
2.2.1	Fornitura . . . . .	8
2.2.1.1	Metriche . . . . .	8
2.2.1.2	Obiettivi . . . . .	9
2.2.2	Sviluppo . . . . .	9
2.2.2.1	Progettazione architetturale . . . . .	9
2.2.2.1.1	Metriche . . . . .	9
2.2.2.1.2	Obiettivi . . . . .	10
2.2.2.2	Progettazione di dettaglio . . . . .	10
2.2.2.2.1	Metriche . . . . .	10
2.2.2.2.2	Obiettivi . . . . .	10
2.2.2.3	Codifica . . . . .	10
2.2.2.3.1	Metriche . . . . .	10
2.2.2.3.2	Obiettivi . . . . .	11
2.3	Processi di supporto . . . . .	11
2.3.1	Documentazione . . . . .	11
2.3.1.1	Metriche . . . . .	11
2.3.1.2	Obiettivi . . . . .	11
2.3.2	Gestione della qualità . . . . .	12
2.3.2.1	Metriche . . . . .	12
2.3.2.2	Obiettivi . . . . .	12
2.3.3	Verifica . . . . .	12
2.3.3.1	Metriche . . . . .	12
2.3.3.2	Obiettivi . . . . .	13
<b>3</b>	<b>Qualità di prodotto</b>	<b>14</b>
3.1	Scopo . . . . .	14
3.2	Obiettivi prefissati . . . . .	14
3.3	Funzionalità . . . . .	14
3.3.1	Metriche . . . . .	14
3.3.2	Obiettivi . . . . .	15
3.4	Affidabilità . . . . .	15
3.4.1	Metriche . . . . .	15
3.4.2	Obiettivi . . . . .	15
3.5	Efficienza . . . . .	16
3.5.1	Metriche . . . . .	16

3.5.2	Obiettivi . . . . .	16
3.6	Usabilità . . . . .	16
3.6.1	Metriche . . . . .	16
3.6.2	Obiettivi . . . . .	16
3.7	Manutenibilità . . . . .	16
3.7.1	Metriche . . . . .	16
3.7.2	Obiettivi . . . . .	17
3.8	Compatibilità . . . . .	17
3.8.1	Metriche . . . . .	17
3.8.2	Obiettivi . . . . .	18
<b>4</b>	<b>Specifica dei test</b>	<b>19</b>
4.1	Test di unità . . . . .	19
4.2	Test di integrazione . . . . .	19
4.3	Test di sistema . . . . .	19
4.4	Test di accettazione . . . . .	19
4.5	Test di regressione . . . . .	20
4.6	Sviluppo . . . . .	20
<b>A</b>	<b>Resoconto delle attività di verifica</b>	<b>21</b>
A.1	Fornitura . . . . .	21
A.1.1	MPC-AC e MPC-ETC: Actual Cost e Estimated to Completion . . . . .	21
A.1.2	MPC-EV e MPC-PV: Earned Value e Planned Value . . . . .	21
A.1.3	MPC-SV: Schedule Variance . . . . .	22
A.1.4	MPC-CV: Cost Variance . . . . .	22
A.1.5	MPC-EAC: Estimated at Completion . . . . .	23
A.2	Documentazione . . . . .	23
A.2.1	MPC-IG: Indice Gulpease . . . . .	23
A.2.2	MPC-CO: Correttezza Ortografica . . . . .	24
<b>B</b>	<b>Valutazioni per il miglioramento</b>	<b>25</b>
B.1	Scopo . . . . .	25
B.2	Valutazioni sull'organizzazione . . . . .	25
B.3	Valutazioni sui ruoli . . . . .	25
B.4	Valutazioni sugli strumenti di lavoro . . . . .	25

**Elenco delle figure**

1	Grafo Complessità Ciclomatica . . . . .	17
2	Modello a V . . . . .	19
3	Actual Cost e Estimated to Completion . . . . .	21
4	Earned Value e Planned Value . . . . .	21
5	Schedule Variance . . . . .	22
6	Cost Variance . . . . .	22
7	Estimated at Completion . . . . .	23
8	Indice di Gulpease . . . . .	23
9	Errori ortografici . . . . .	24

## Elenco delle tabelle

1	Metriche e obiettivi per la fornitura . . . . .	9
2	Metriche e obiettivi per la progettazione . . . . .	10
3	Metriche e obiettivi per la progettazione di dettaglio . . . . .	10
4	Metriche e obiettivi per la codifica . . . . .	11
5	Metriche e obiettivi per la documentazione . . . . .	11
6	Metriche e obiettivi per la gestione della qualità . . . . .	12
7	Metriche e obiettivi per la verifica . . . . .	13
8	Obiettivi prefissati per la qualità di prodotto . . . . .	14
9	Metriche e obiettivi per le funzionalità . . . . .	15
10	Metriche e obiettivi per l'affidabilità . . . . .	15
11	Metriche e obiettivi per l'efficienza . . . . .	16
12	Metriche e obiettivi per l'usabilità . . . . .	16
13	Metriche e obiettivi per la manutenibilità . . . . .	17
14	Metriche e obiettivi per la compatibilità . . . . .	18
15	Problemi relativi all'organizzazione . . . . .	25
16	Problemi relativi ai ruoli . . . . .	25
17	Problemi relativi agli strumenti di lavoro . . . . .	25

## 1 Introduzione

### 1.1 Scopo del documento

Il presente documento ha come scopo la presentazione dei metodi di *verifica<sub>G</sub>* e *validazione<sub>G</sub>* adottati dal gruppo *BugPharma* al fine di garantire la qualità di prodotto e di processo. In esso verranno inoltre riportati i risultati delle verifiche effettuate sul prodotto, in modo da poter correggere in breve tempo eventuali problemi rilevati e minimizzare lo spreco di risorse.

### 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<sub>G</sub>* (Total Cost of Ownership), in quanto funzionano con un semplice *browser<sub>G</sub>* 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<sub>G</sub>*.

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<sub>G</sub>* (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<sub>G</sub>* che potrà essere caricato nell'applicazione. Tale applicazione sarà fruibile attraverso un browser in grado di supportare le tecnologie *HTML5<sub>G</sub>*, *CSS<sub>G</sub>* e *JavaScript<sub>G</sub>*.

### 1.3 Glossario

Al fine di evitare possibili ambiguità relative al linguaggio utilizzato nei documenti, viene fornito il *Glossario v1.0.0* 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 C5 - Login Warrior:  
<https://www.math.unipd.it/~tullio/IS-1/2021/Progetto/C5.pdf>

### 1.5.2 Riferimenti informativi

- Slide T12 del corso di Ingegneria del Software - Qualità di prodotto:  
<https://www.math.unipd.it/~tullio/IS-1/2021/Dispense/T12.pdf>.
- Slide T13 del corso di Ingegneria del Software - Qualità di processo:  
<https://www.math.unipd.it/~tullio/IS-1/2021/Dispense/T13.pdf>.
- Indice *Gulpease*<sub>G</sub>:  
[https://it.wikipedia.org/wiki/Indice\\_Gulpease](https://it.wikipedia.org/wiki/Indice_Gulpease).
- Metriche di progetto:  
[https://it.wikipedia.org/wiki/Metriche\\_di\\_progetto](https://it.wikipedia.org/wiki/Metriche_di_progetto).



## 2 Qualità di processo

### 2.1 Scopo

La  $qualità_G$  del  $prodotto_G$  finale è influenzata dalla qualità dei singoli  $processi_G$  che lo compongono. È quindi necessario avere delle metriche che permettano di valutare tali processi per garantire che essi raggiungano gli obiettivi di qualità fissati.

In questa sezione si espongono le metriche scelte ed i livelli di qualità accettabili e ottimali per ciascuna di esse.

### 2.2 Processi primari

#### 2.2.1 Fornitura

##### 2.2.1.1 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<sub>G</sub>* 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.

$$AC + ETC$$

## 2.2.1.2 Obiettivi

Metrica	Descrizione	Valore di accettazione	Valore ideale
MPC-AC	Actual Cost	$\geq 0$	$\leq EAC$
MPC-EV	Earned Value	$\geq 0$	$\leq EAC$
MPC-PV	Planned Value	$\geq 0$	$\leq BAC$
MPC-SV	Schedule Variance	$\geq -10\%$	$\geq 0\%$
MPC-CV	Cost Variance	$\geq -5\%$	$\geq 0\%$
MPC-ETC	Estimated to Completion	$\geq 0\%$	$\leq EAC$
MPC-EAC	Estimated at Completion	$EAC \geq BAC - 2,5\%$ e $EAC \leq BAC + 2,5\%$	$= BAC$

Tabella 1: Metriche e obiettivi per la fornitura

## 2.2.2 Sviluppo

## 2.2.2.1 Progettazione architettuale

## 2.2.2.1.1 Metriche

1. **MPC-SFIN: Structural Fan-in**

Indice di utilità, indica quante componenti utilizzano un dato modulo. Un valore alto indica un alto riuso della componente.

2. **MPC-SFOUT: Structural Fan-out**

Indice di dipendenza, indica quante componenti vengono utilizzate dalla componente in esame. Un valore alto indica un alto accoppiamento della componente.

#### 2.2.2.1.2 Obiettivi

Il valore ideale e il valore di accettazione per queste metriche verranno specificati nelle prossime versioni del *Piano di Qualifica*.

Metrica	Descrizione	Valore di accettazione	Valore ideale
MPC-SFIN	Structural Fan-in	-	-
MPC-SFOUT	Structural Fan-out	-	-

Tabella 2: Metriche e obiettivi per la progettazione

#### 2.2.2.2 Progettazione di dettaglio

##### 2.2.2.2.1 Metriche

1. **MPC-NM:** Number of Methods

Indica il numero medio di metodi per package. Un numero eccessivo potrebbe indicare la necessità di refactoring.

##### 2.2.2.2.2 Obiettivi

Il valore ideale e il valore di accettazione per questa metrica sono provvisori e potrebbero subire modifiche nelle prossime versioni del *Piano di Qualifica*.

Metrica	Descrizione	Valore di accettazione	Valore ideale
MPC-NM	Number of Methods	3-11	3-8

Tabella 3: Metriche e obiettivi per la progettazione di dettaglio

#### 2.2.2.3 Codifica

##### 2.2.2.3.1 Metriche

1. **MPC-BLC:** Bugs for Line of Code

Indice del numero di righe di codice contenenti *bug<sub>G</sub>* ed errori al proprio interno.

2. **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.

### 2.2.2.3.2 Obiettivi

Il valore ideale e il valore di accettazione per queste metriche sono provvisori e potrebbero subire modifiche nelle prossime versioni del *Piano di Qualifica*.

Metrica	Descrizione	Valore di accettazione	Valore ideale
MPC-BLC	Bugs for Line of Code	0-70	0-25
MPC-VNUD	Variabili Non Utilizzate e Non Definite	0	0

Tabella 4: Metriche e obiettivi per la codifica

## 2.3 Processi di supporto

### 2.3.1 Documentazione

#### 2.3.1.1 Metriche

1. **MPC-IG:** *Indice Gulpease<sub>G</sub>*

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.

#### 2.3.1.2 Obiettivi

Metrica	Descrizione	Valore di accettazione	Valore ideale
MPC-IG	Indice Gulpease	40 - 100	60 - 100
MPC-CO	Correttezza Ortografica	0	0

Tabella 5: Metriche e obiettivi per la documentazione

### 2.3.2 Gestione della qualità

#### 2.3.2.1 Metriche

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

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

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.

#### 2.3.2.2 Obiettivi

Metrica	Descrizione	Valore di accettazione	Valore ideale
MPC-QMS	Quality Metrics Satisfied	$\geq 90\%$	100%

Tabella 6: Metriche e obiettivi per la gestione della qualità

### 2.3.3 Verifica

#### 2.3.3.1 Metriche

1. **MPC-CC**: 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-SC**: Statement Coverage  
Lo  $statement_G$  coverage è una tecnica di test del codice di tipo white box che prevede almeno un'esecuzione di tutte le istruzioni presenti nel codice sorgente. Questa  $metrica_G$  è utilizzata per calcolare e misurare il numero di istruzioni che sono state eseguite almeno una volta.
3. **MPC-BC**: Branch Coverage  
Indice di quante diramazioni del codice vengono eseguite dai test. Un "ramo" è uno dei possibili percorsi di esecuzione che il codice può seguire dopo che un'istruzione decisionale (ad esempio un'istruzione if) viene valutata.
4. **MPC-DCC**: Decision/Condition Coverage  
Il Decision/Condition Coverage è un criterio di copertura del codice utilizzato nei test del software.

### 2.3.3.2 Obiettivi

Il valore ideale e il valore di accettazione per queste metriche sono provvisori e potrebbero subire modifiche nelle prossime versioni del *Piano di Qualifica*.

Metrica	Descrizione	Valore di accettazione	Valore ideale
MPC-CC	Code Coverage	>70%	>90%
MPC-SC	Statement Coverage	>70%	>85%
MPC-BC	Branch Coverage	>50%	>75%
MPC-DCC	Decision/Condition Coverage	-	-

Tabella 7: Metriche e obiettivi per la verifica

### 3 Qualità di prodotto

#### 3.1 Scopo

Facendo riferimento allo standard ISO/IEC 9126:2001, è qui riportato un insieme di caratteristiche che il *prodotto<sub>G</sub>* deve avere per essere considerato di *qualità<sub>G</sub>*. Viene inoltre riportato un insieme di *metriche<sub>G</sub>* che hanno lo scopo di definire un metodo per valutare se il prodotto finale possieda o meno tali caratteristiche.

#### 3.2 Obiettivi prefissati

Obiettivo	Descrizione
Funzionalità	Si vuole svolgere un lavoro il più completo possibile, perseguendo correttezza e soddisfacendo tutti i requisiti descritti all'interno dell' <i>Analisi dei Requisiti v1.0.0</i> .
Affidabilità	Si vuole creare un prodotto che sia sempre disponibile e tollerante agli errori, cercando in primo luogo di evitare la manifestazione di questi ultimi.
Efficienza	Si vuole creare un prodotto che raggiunga e soddisfi gli obiettivi prefissati con il minor utilizzo di risorse possibile.
Usabilità	Si vuole creare un prodotto di facile apprendimento, comprensione ed utilizzo da parte di tutte le tipologie di utenti. L'obiettivo è quindi quello di creare una user experience di alta qualità.
Manutenibilità	Si vuole creare un prodotto riusabile e aperto a modifiche e miglioramenti.
Compatibilità	Si vuole creare un applicativo accessibile al maggior numero di utenti possibile, garantendo la compatibilità con i browser più diffusi (L'elenco dei browser supportati è consultabile alla sezione §4.3 dell' <i>Analisi dei Requisiti v1.0.0</i> ).

Tabella 8: Obiettivi prefissati per la qualità di prodotto

#### 3.3 Funzionalità

##### 3.3.1 Metriche

###### 1. MPC-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.

### 3.3.2 Obiettivi

Metrica	Descrizione	Valore di accettazione	Valore ideale
MPC-RC	Requirements Coverage	100% di $RC_{obb}$	100% di $RC_{tot}$

Tabella 9: Metriche e obiettivi per le funzionalità

## 3.4 Affidabilità

### 3.4.1 Metriche

#### 1. MPD-FD: Failure Density

Indica l'affidabilità di un prodotto software. Si ricava dal rapporto tra i  $test_G$  eseguiti sul prodotto ed i test che esso ha fallito.

$$FD = \frac{T_f}{T_e} \cdot 100$$

Dove:

- $T_f$  : numero di test falliti;
- $T_e$  : numero di test eseguiti.

### 3.4.2 Obiettivi

Il valore ideale e il valore di accettazione per questa metrica verranno specificati nelle prossime versioni del *Piano di Qualifica*.

Metrica	Descrizione	Valore di accettazione	Valore ideale
MPC-FD	Failure Density	-	-

Tabella 10: Metriche e obiettivi per l'affidabilità



### 3.5 Efficienza

#### 3.5.1 Metriche

1. **MPD-TR:** Tempo medio di Risposta

Tempo impiegato dall'applicativo ad elaborare i dati ed a fornire un risultato all'utente.

#### 3.5.2 Obiettivi

Il valore ideale e il valore di accettazione per questa metrica verranno specificati nelle prossime versioni del *Piano di Qualifica*.

Metrica	Descrizione	Valore di accettazione	Valore ideale
MPC-TR	Tempo medio di Risposta	-	-

Tabella 11: Metriche e obiettivi per l'efficienza

### 3.6 Usabilità

#### 3.6.1 Metriche

1. **MPD-FA:** Facilità di Apprendimento

Numero di minuti necessari all'utente per comprendere come utilizzare le funzionalità del prodotto.

#### 3.6.2 Obiettivi

I valori ideale e di accettazione per questa metrica verranno specificati nelle prossime versioni del *Piano di Qualifica*.

Metrica	Descrizione	Valore di accettazione	Valore ideale
MPC-FA	Facilità di Apprendimento	-	-

Tabella 12: Metriche e obiettivi per l'usabilità

### 3.7 Manutenibilità

#### 3.7.1 Metriche

1. **MPD-CC:** Complessità Ciclomatica

Indica la complessità di un programma.

La Complessità Ciclomatica di una sezione del codice sorgente è data dal numero di percorsi linearmente indipendenti al suo interno.

*Esempio:*

- Se il codice sorgente non contenesse istruzioni condizionali o punti di decisione sul flusso di controllo, la complessità varrebbe 1 poiché ci sarebbe un unico percorso per attraversarlo;

- Se il codice avesse invece un'istruzione **if** a condizione singola, la complessità sarebbe 2 dato che ci sarebbero due percorsi: uno in cui l'istruzione **if** restituisce **TRUE** e un altro in cui restituisce **FALSE**;
- Due **if** a condizione singola nidificati o un **if** con due condizioni produrrebbero una complessità di valore 3.

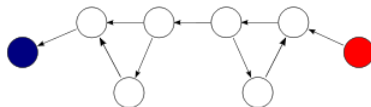


Figura 1: Grafo Complessità Ciclomantica

## 2. MPD-NPM: Numero Parametri per Metodo

Indica il numero dei parametri passati in *input<sub>G</sub>* ad un metodo. Un valore troppo grande può indicare un metodo troppo complesso.

### 3.7.2 Obiettivi

Il valore ideale e il valore di accettazione per queste metriche sono provvisori e potrebbero subire modifiche nelle prossime versioni del *Piano di Qualifica*.

Metrica	Descrizione	Valore di accettazione	Valore ideale
MPD-CC	Complessità Ciclomantica	0-30	0-10
MPD-NPM	Numero Parametri per Metodo	0-8	0-3

Tabella 13: Metriche e obiettivi per la manutenibilità

## 3.8 Compatibilità

### 3.8.1 Metriche

#### 1. MPD-VS: Versioni del *browser<sub>G</sub>* 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.

3.8.2 Obiettivi

Metrica	Descrizione	Valore di accettazione	Valore ideale
MPD-VS	Versioni del browser Supportate	80%	100%

Tabella 14: Metriche e obiettivi per la compatibilità

## 4 Specifica dei test

L'analisi dinamica si occupa di studiare il comportamento del programma in un insieme finito di casi che siano rappresentativi di tutte le esecuzioni possibili. Essa richiede quindi l'esecuzione dell'oggetto di  $verifica_G$ , il quale verrà sottoposto ad un insieme di  $test_G$ .

Come già riportato nelle *Norme di Progetto v1.0.0*, i test possono essere di vario tipo, ognuno con lo scopo di individuare difetti software differenti.

Il gruppo *BugPharma* ha deciso che, per perseguire la correttezza del  $prodotto_G$  e facilitare la  $fase_G$  di  $validazione_G$ , svolgerà la verifica in parallelo allo sviluppo (*Modello a V<sub>G</sub>*). Per evitare che il testing del codice rallenti la produzione, è necessario rendere i test il più automatici possibile.

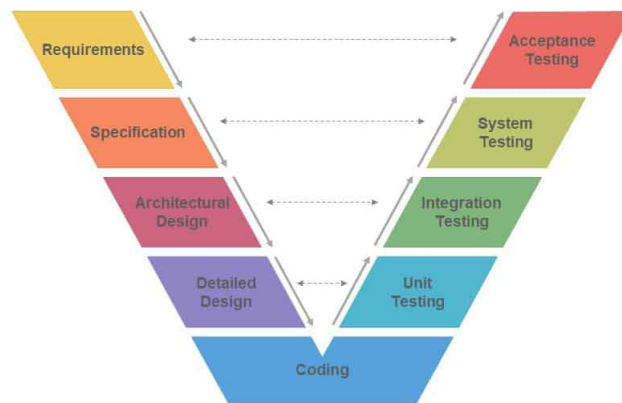


Figura 2: Modello a V

### 4.1 Test di unità

I test di  $unità_G$  si occupano di isolare ciascuna unità del prodotto per mostrarne completezza e correttezza nell'implementazione. In questa prima versione del *Piano di Qualifica* il gruppo non è ancora in grado di individuare i test di unità, non avendo iniziato il processo di progettazione e sviluppo software.

### 4.2 Test di integrazione

I test di integrazione sono eseguiti quando due o più unità già testate vengono assemblate incrementalmente. Essi si occupano di verificare che le parti lavorino insieme nel modo corretto e abbiano il comportamento atteso. I test di integrazione permettono inoltre di individuare possibili difetti nella progettazione architetturale o una bassa  $qualità_G$  dei test di unità.

In questa prima versione del *Piano di Qualifica* il gruppo non è ancora in grado di stabilire i test di integrazione poiché non sono ancora state individuate o testate le componenti del prodotto.

### 4.3 Test di sistema

Il test di sistema si occupa di verificare il comportamento del sistema completo rispetto ai requisiti software individuati.

### 4.4 Test di accettazione

Il test di accettazione si svolge alla presenza del  $committente_G$  ed ha lo scopo di accertare il soddisfacimento di tutti i requisiti utente richiesti dal  $capitolato_G$ .

#### 4.5 Test di regressione

I test di *regressione<sub>G</sub>* hanno lo scopo di verificare che le correzioni o le estensioni effettuate su specifiche unità già testate non danneggino il resto del sistema. Essi consistono nella ripetizione selettiva di test di unità, integrazione e sistema.

#### 4.6 Sviluppo

Le specifiche dei test verranno definite nelle prossime versioni del *Piano di Qualifica*.

## A Resoconto delle attività di verifica

### A.1 Fornitura

#### A.1.1 MPC-AC e MPC-ETC: Actual Cost e Estimated to Completion

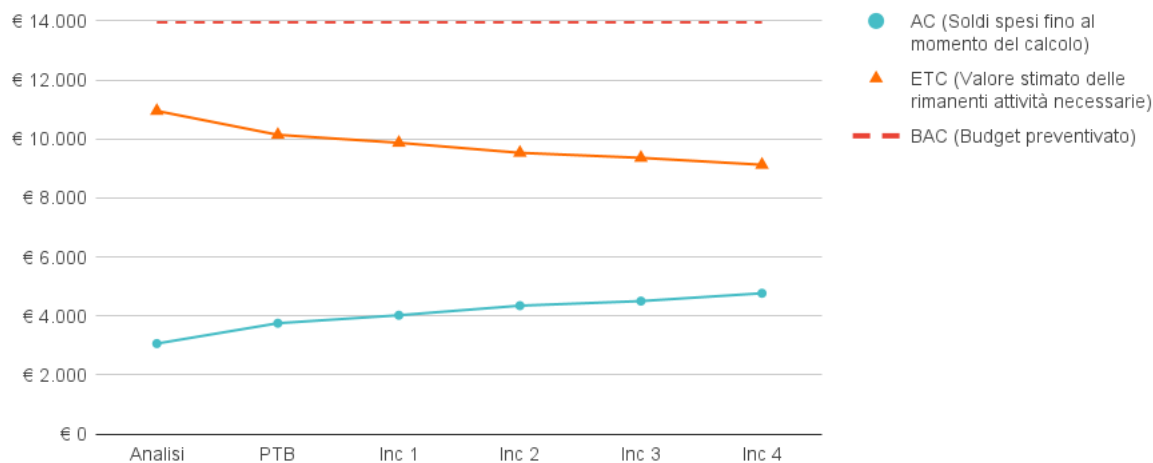


Figura 3: Actual Cost e Estimated to Completion

#### A.1.2 MPC-EV e MPC-PV: Earned Value e Planned Value

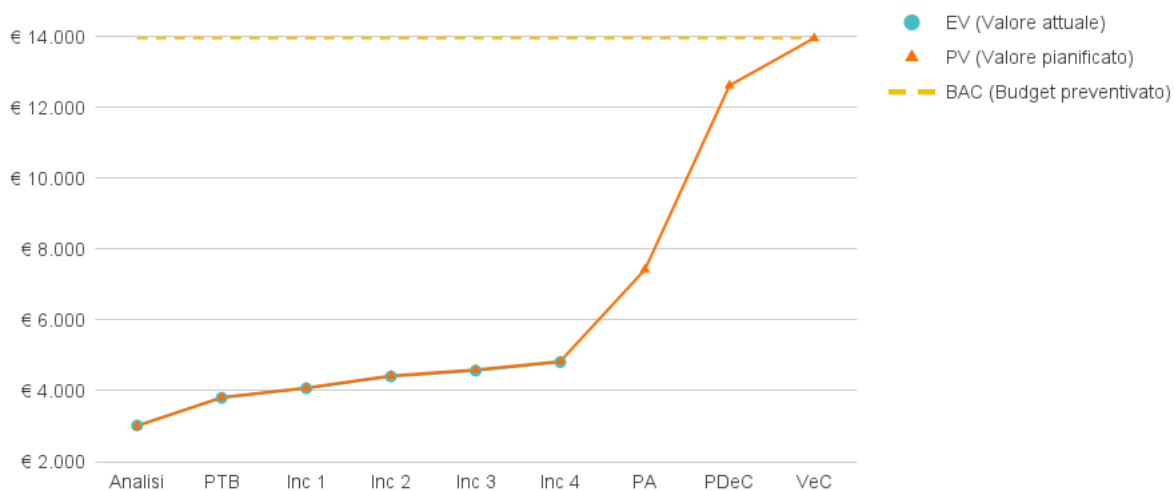


Figura 4: Earned Value e Planned Value

A.1.3 MPC-SV: Schedule Variance

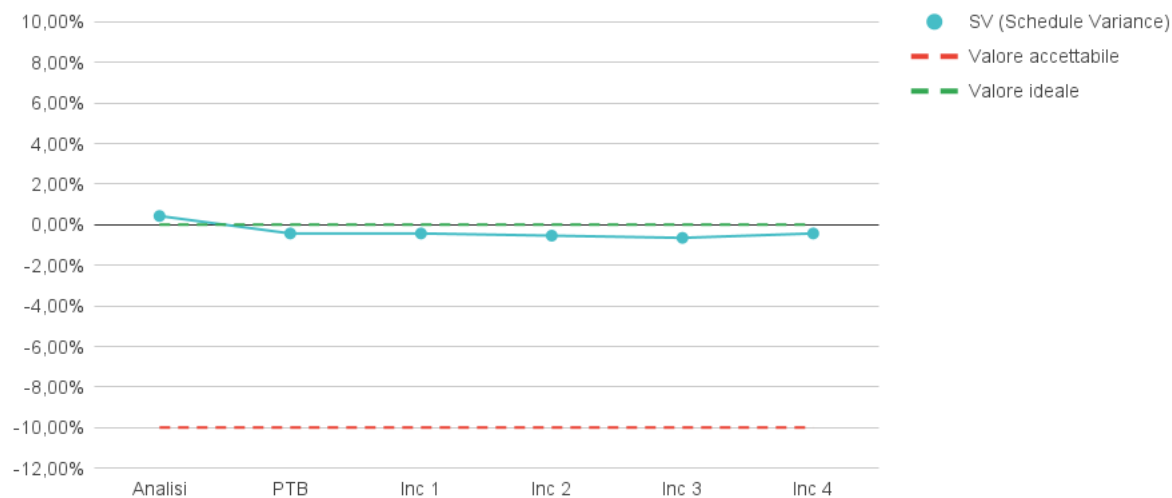


Figura 5: Schedule Variance

A.1.4 MPC-CV: Cost Variance

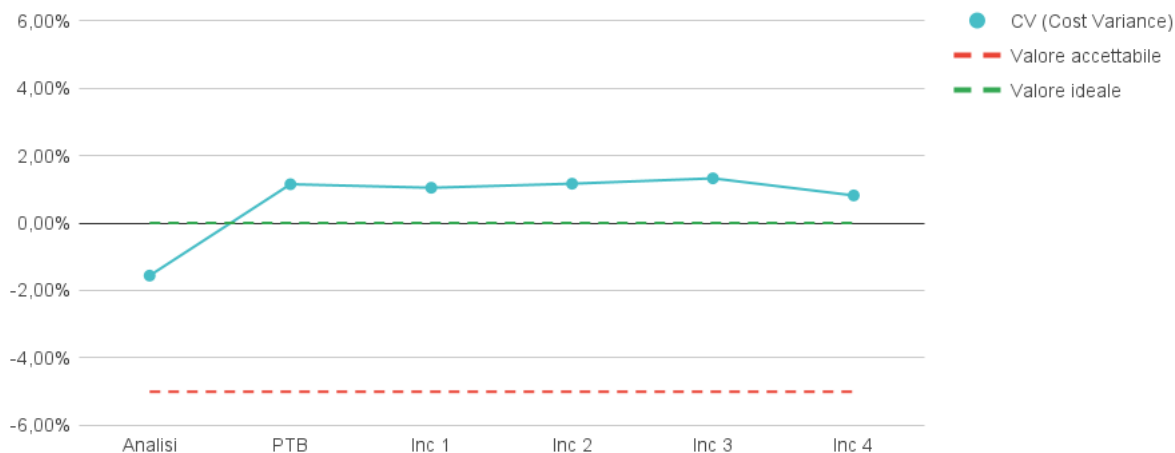


Figura 6: Cost Variance

A.1.5 MPC-EAC: Estimated at Completion

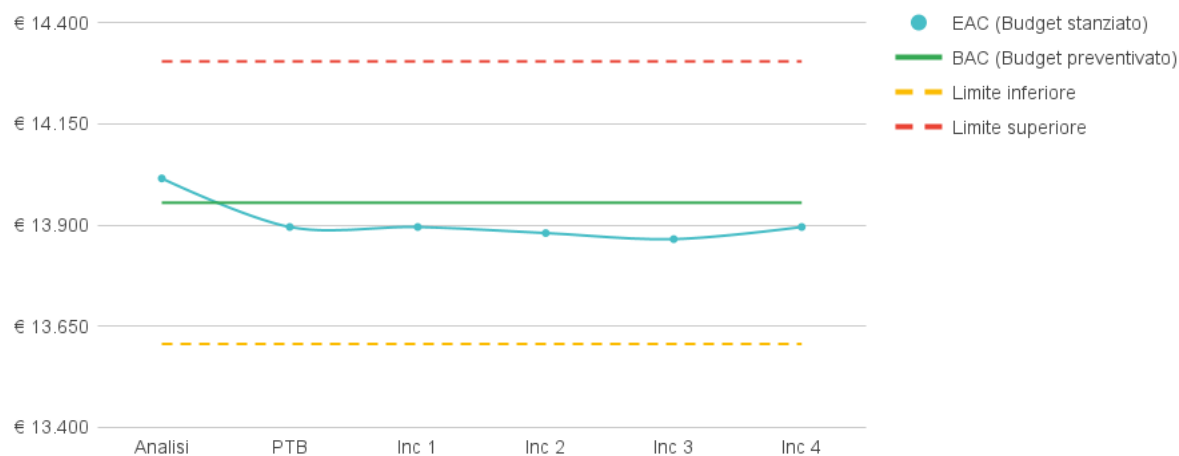


Figura 7: Estimated at Completion

A.2 Documentazione

A.2.1 MPC-IG: Indice Gulpease

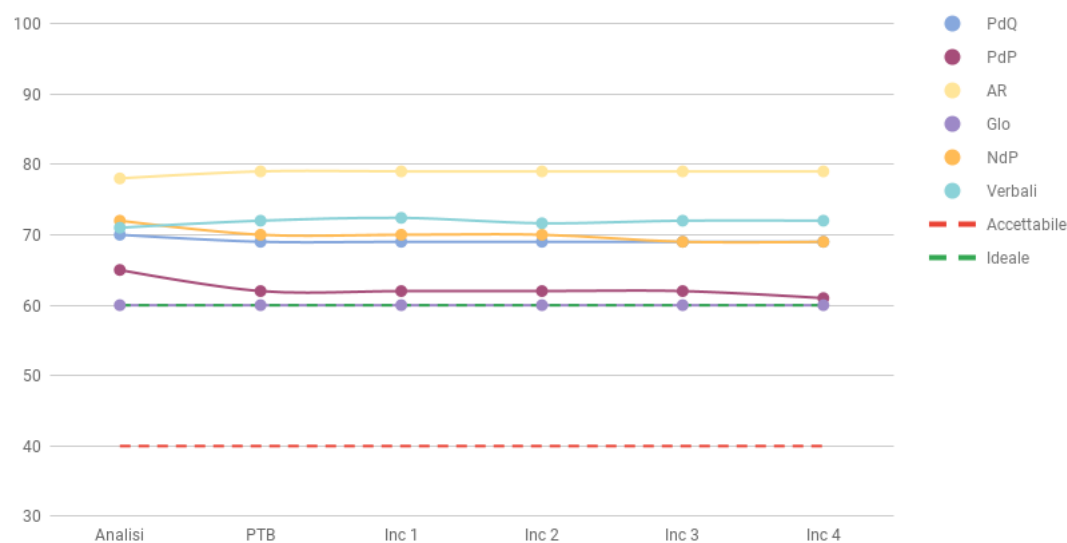


Figura 8: Indice di Gulpease



A.2.2 MPC-CO: Correttezza Ortografica

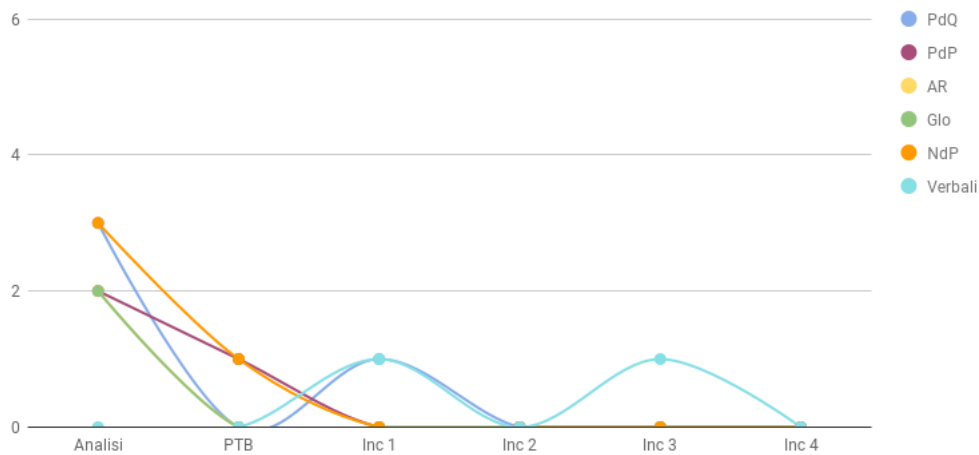


Figura 9: Errori ortografici

## B Valutazioni per il miglioramento

### B.1 Scopo

In questa sezione viene riportata la valutazione del lavoro svolto fino ad ora. Lo scopo è quello di individuare i problemi sorti, procedere alla loro risoluzione e diminuire la probabilità che essi si verifichino nuovamente.

### B.2 Valutazioni sull'organizzazione

Problema	Descrizione	Gravità	Soluzione
Meeting di gruppo	Difficoltà a trovare orari in cui tutti i membri del gruppo siano disponibili.	Bassa	Utilizzo di <i>Google Calendar</i> <sub>G</sub> per tracciare gli impegni personali dei membri del gruppo.

Tabella 15: Problemi relativi all'organizzazione

### B.3 Valutazioni sui ruoli

Problema	Descrizione	Gravità	Soluzione
Responsabile di Progetto	A causa dell'inesperienza all'inizio non era ben chiaro quali fossero le attività assegnate alla figura del Responsabile di Progetto.	Bassa	Approfondimento sulla figura del Responsabile di Progetto e redistribuzione del carico di lavoro.

Tabella 16: Problemi relativi ai ruoli

### B.4 Valutazioni sugli strumenti di lavoro

Problema	Descrizione	Gravità	Soluzione
Poca esperienza con le tecnologie scelte	Lo sviluppo del prodotto richiede tecnologie poco conosciute dai membri del gruppo.	Alta	Suddivisione dello studio delle tecnologie tra i membri del gruppo per un approfondimento mirato, con conseguente incontro di condivisione delle nozioni apprese.

Tabella 17: Problemi relativi agli strumenti di lavoro