

# Piano di qualifica

Filippo Sabbadin

---

## Indice

<b>1. Introduzione .....</b>	<b>1</b>
1.1. Scopo del documento .....	1
1.2. Glossario .....	1
1.3. Riferimenti .....	1
1.3.1. Riferimenti normativi .....	1
1.3.2. Riferimenti informativi .....	1
<b>2. Metriche di qualità .....</b>	<b>3</b>
2.1. Introduzione .....	3
2.2. Qualità di processo .....	3
2.2.1. Fornitura .....	3
2.2.2. Sviluppo .....	3
2.2.2.1. Codice .....	3
2.2.2.2. Grafica 3D .....	4
2.2.3. Documentazione .....	4
2.2.4. Verifica .....	4
2.2.5. Gestione della qualità .....	4
2.3. Qualità del prodotto .....	4
2.3.1. Funzionalità .....	4
2.3.2. Affidabilità .....	5
2.3.3. Efficienza .....	5
2.3.3.1. Specifiche macchina .....	5
<b>3. Metodologie di testing .....</b>	<b>6</b>
3.1. Tipologie di test .....	6
3.1.1. Test di unità .....	6
<b>4. Cruscotto di valutazione delle metriche .....</b>	<b>7</b>

---

## Lista di immagini

---

## Lista di tabelle

<b>Tabella 1 Componenti macchina 1 .....</b>	<b>5</b>
--	----------

---

# 1. Introduzione

## 1.1. Scopo del documento

Il presente documento ha lo scopo di definire le metriche di qualità che verranno utilizzate per valutare il prodotto software, e le modalità di verifica e validazione del prodotto.

## 1.2. Glossario

Per facilitare la comprensione del documento, è stato creato un glossario che contiene i termini utilizzati nel documento e le loro definizioni. I termini presenti nel glossario sono colorati di blu e seguiti da un'asterisco: [esempio\\*](#).

Il glossario è accessibile tramite il link <https://github.com/fliposab/ProgettoStage/blob/main/Documentazione/Glossario.pdf> oppure consultando il rispettivo documento all'interno della stessa cartella.

## 1.3. Riferimenti

### 1.3.1. Riferimenti normativi

- Norme di progetto:

<https://github.com/fliposab/ProgettoStage/blob/main/Documentazione/Norme-di-progetto.pdf>

### 1.3.2. Riferimenti informativi

- Slide T08 - Qualità di processo:

<https://www.math.unipd.it/~tullio/IS-1/2024/Dispense/T08.pdf>

- Slide T09 - Verifica e validazione:

<https://www.math.unipd.it/~tullio/IS-1/2024/Dispense/T09.pdf>

- Glossario:

---

<https://github.com/fliposab/ProgettoStage/blob/main/Documentazione/Glossario.pdf>

---

## 2. Metriche di qualità

### 2.1. Introduzione

Le metriche di qualità sono utilizzate per valutare la qualità del prodotto software, e per identificare eventuali problemi o aree di miglioramento.

### 2.2. Qualità di processo

La qualità di processo si riferisce alla qualità delle attività svolte durante lo sviluppo del prodotto software.

Le metriche di qualità di processo sono utilizzate per valutare l'efficacia e l'efficienza delle attività svolte, e per identificare eventuali problemi o aree di miglioramento.

#### 2.2.1. Fornitura

Per il processo di fornitura, vengono indicate tutte le scelte operative fatte in fase di sviluppo. Viene usato l'acronimo MPC (Minimum Predictive Capability).

In questo caso, il MPC è il valore minimo da raggiungere per essere considerato accettabile.

- **MPC-CT - Completion Time:** tempo totale previsto per completare il progetto, idealmente non deve superare quello pianificato nel Piano di lavoro.

#### 2.2.2. Sviluppo

##### 2.2.2.1. Codice

- **MPC-RSI - Requirements Stability Index:** indice di stabilità dei requisiti. Indica la percentuale di requisiti che sono stati modificati rispetto al totale dei requisiti. Un valore alto indica che i requisiti sono stabili e non soggetti a modifiche frequenti.

- **MPC-TD - Technical Debt Ratio:** rapporto tra il tempo necessario per risolvere i problemi tecnici e il tempo necessario per sviluppare nuove funzionalità. Un valore basso indica che il codice è ben strutturato e non presenta problemi tecnici.

---

### 2.2.2.2. Grafica 3D

Le seguenti metriche vengono applicate principalmente al [modello 3D\\*](#) principale del giocatore, visto che gli altri modelli utilizzati sono semplicemente cubi o sfere:

- **MPC-MVC - Model Vertex Count:** un modello 3D è costituito da un certo numero di triangoli e vertici. Ovviamente, maggiore è il numero di vertici, più risorse sono richieste per il [rende-ring\\*](#) di esso, rendendo inoltre più complicato successive modifiche. Un numero giusto (né troppo alto né troppo basso) è consigliato per garantire una buona qualità del modello ed evitare possibili complicazioni.
- **MPC-MBC - Model Bones Count:**
- **MPC-UIC - UV Islands Count:** avere un numero minore di [UV Islands\\*](#) permette una gestione della [texture\\*](#) più semplice.
- **MPC-UIS - UV Islands Space (in percentuale):** percentuale della texture occupata e utilizzata dalle UV Islands. Uno spazio maggiore utilizza più pixel dell'immagine e garantisce una maggiore qualità.

### 2.2.3. Documentazione

- **MPC-IG - Indice di Gulpease:**
- **MPC-CO - Correttezza ortografica:**

### 2.2.4. Verifica

- **MPC-CCO - Code Coverage:**
- **MPC-TSP - Test superati (in percentuale):**

### 2.2.5. Gestione della qualità

- **MPC-SQM - Satisfaction of Quality Metrics:**

## 2.3. Qualità del prodotto

### 2.3.1. Funzionalità

- **MPD-RO - Copertura requisiti obbligatori:**
- **MPD-OP - Copertura requisiti opzionali:**



---

## 2.3.2. Affidabilità

- **MPD-CC - Code Coverage:**

## 2.3.3. Efficienza

- **MPD-TF - Target FPS:** indica gli [fps\\*](#) obiettivo da mantenere durante l'esecuzione del gioco.
- **MPD-LS - Lag Spikes:** indica il numero di [lag spikes\\*](#) che occorrono durante l'esecuzione del gioco. Non vengono contati durante un [caricamento\\*](#) tra un livello e un altro e vengono contati solo durante i livelli.

### 2.3.3.1. Specifiche macchina

Molto importante è specificare le componenti della macchina su cui viene testato il gioco, dato che macchine diverse offrono prestazioni diverse:

Componente	Dettagli
<a href="#">CPU*</a>	AMD® Ryzen 5 4500U
<a href="#">GPU*</a>	Integrata alla CPU
<a href="#">RAM*</a>	8GB DDR4
<a href="#">SSD*</a>	
Sistema Operativo	Pop_OS

Tabella 1: Componenti macchina 1

In sintesi, la macchina su cui viene testato il gioco offre prestazioni sulla fascia media-bassa, quindi si ritiene che se il gioco offre delle buone prestazioni sulla macchina di testing, offrirà in media buone prestazioni su tutte le macchine con un sistema operativo supportato.

---

## **3. Metodologie di testing**

### **3.1. Tipologie di test**

#### **3.1.1. Test di unità**

I test di unità sono utilizzati per verificare il corretto funzionamento delle singole unità del prodotto software, come ad esempio le funzioni o i metodi.

I test di unità sono eseguiti durante lo sviluppo del prodotto, e sono utilizzati per identificare eventuali problemi o errori nelle singole unità del prodotto.

---

## **4. Cruscotto di valutazione delle metriche**