



# >Three Way Milkshake\_

---

## Piano di progetto

---

### Three Way Milkshake - Progetto "PORTACS"

threewaymilkshake@gmail.com

<b>Versione</b>	0.0.1
<b>Stato</b>	Non approvato
<b>Uso</b>	Esterno
<b>Approvazione</b>	Nome approvatore
<b>Redazione</b>	Tessari Andrea
<b>Verifica</b>	Nome verificatore
<b>Destinatari</b>	San Marco Informatica Prof. Tullio Vardanega Prof. Riccardo Cardin Three Way Milkshake

### Descrizione

Questo documento contiene la descrizione delle strategie di controllo della qualità del gruppo *Three Way Milkshake* per il progetto PORTACS



>Three Way  
Milkshake\_

Piano di progetto

## Registro delle modifiche

Versione	Descrizione	Data	Nominativo	Ruolo
0.0.3	Modifiche sezione §1	1-1-2020	Crivellari Alberto	<i>Ruolo1</i>
0.0.3	Redazione sezione §3	1-1-2020	Tessari Andrea	<i>Ruolo2</i>
0.0.2	Redazione sezione §1	30-12-2020	Crivellari Alberto	<i>Ruolo1</i>
0.0.1	Impostazione iniziale	27-12-2020	Tessari Andrea	<i>Ruolo1</i>



## 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	Riferimenti . . . . .	6
1.5	Riferimenti normativi . . . . .	6
1.6	Riferimenti informativi . . . . .	6
<b>2</b>	<b>Qualità del processo</b>	<b>7</b>
2.1	Processo di pianificazione . . . . .	7
2.1.1	Obiettivi . . . . .	7
2.2	Metriche di verifica . . . . .	7
2.2.1	Budget at Completion . . . . .	7
2.2.2	Schedule Variance . . . . .	7
2.2.3	Budget Variance . . . . .	7
2.2.4	Actual Cost . . . . .	7
<b>3</b>	<b>Qualità del prodotto</b>	<b>8</b>
3.1	Funzionalità . . . . .	8
3.1.1	Obiettivi . . . . .	8
3.1.2	Metriche . . . . .	8
3.2	Affidabilità . . . . .	8
3.2.1	Obiettivi . . . . .	9
3.2.2	Metriche . . . . .	9
3.3	Efficienza . . . . .	9
3.3.1	Obiettivi . . . . .	9
3.3.2	Metriche . . . . .	9
3.4	Usabilità . . . . .	9
3.4.1	Obiettivi . . . . .	10
3.4.2	Metriche . . . . .	10
3.5	Manutenibilità . . . . .	10
3.5.1	Obiettivi . . . . .	10
3.5.2	Metriche . . . . .	11
3.6	Portabilità . . . . .	11
3.6.1	Obiettivi . . . . .	11
3.6.2	Metriche . . . . .	11
<b>4</b>	<b>Test</b>	<b>12</b>
4.1	Specifica dei test . . . . .	12
4.2	Test di accettazione . . . . .	12
4.3	Test di sistema . . . . .	12
4.4	Test di integrazione . . . . .	12
4.5	Test d'unità . . . . .	12
4.6	Resoconto attività di verifica . . . . .	12
4.6.1	Strategia per l'analisi statica dei documenti . . . . .	12
4.6.2	Esiti dell'indice di Gulpease . . . . .	12
4.6.3	Conclusioni del periodo di analisi . . . . .	12
4.7	Lista di controllo . . . . .	12



4.8	Valutazioni per il miglioramento . . . . .	12
-----	--	----



## **Elenco delle figure**



## **Elenco delle tabelle**



# 1 Introduzione

## 1.1 Scopo del documento

Il presente documento ha lo scopo di documentare le strategie di verifica e validazione che il gruppo **Three Way Milkshake** ha deciso di adottare relativi al progetto **PORTACS**, per raggiungere gli obiettivi di qualità processo e prodotto.

## 1.2 Scopo del prodotto

Questo capitolato si concentra sulla realizzazione di un software che coordini lo spostamento di varie unità in una determinata griglia di movimento. Ogni unità (che può rappresentare un robot, un muletto o un'automobile) ha un punto di partenza nella griglia, una velocità massima e una lista di punti denominati "Points Of Interest" (POI) che deve raggiungere.

## 1.3 Glossario

Nel seguente documento sono presenti termini tecnici, specifici o ambigui. Per semplificare la lettura viene fornito un glossario, reperibile nel file Glossario. I termini presenti in questo documento che verranno spiegati meglio nel Glossario vengono contrassegnati dalla lettera **G** come pedice, per esempio Prova<sub>G</sub>.

## 1.4 Riferimenti

### 1.5 Riferimenti normativi

- *Norme di progetto v\_1.0.0*;
- Specifica tecnico-economica e organigramma:  
<https://www.math.unipd.it/~tullio/IS-1/2020/Progetto/R0.html>
- Regolamento progetto didattico - slide del corso di Ingegneria del Software:  
<https://www.math.unipd.it/~tullio/IS-1/2020/Dispense/P1.pdf>

### 1.6 Riferimenti informativi

- Slide L12 del corso Ingegneria del Software - Qualità del Software: <https://www.math.unipd.it/~tullio/IS-1/2020/Dispense/L12.pdf>
- Slide L13 del corso Ingegneria del Software - Qualità di Processo: <https://www.math.unipd.it/~tullio/IS-1/2020/Dispense/L13.pdf>
- Slide L14 del corso Ingegneria del Software - Verifica e Validazione: introduzione :  
<https://www.math.unipd.it/~tullio/IS-1/2020/Dispense/L14.pdf>



## **2 Qualità del processo**

### **2.1 Processo di pianificazione**

#### **2.1.1 Obiettivi**

### **2.2 Metriche di verifica**

#### **2.2.1 Budget at Completion**

#### **2.2.2 Schedule Variance**

#### **2.2.3 Budget Variance**

#### **2.2.4 Actual Cost**





## 3 Qualità del prodotto

Per valutare la qualità del prodotto, il gruppo Three Way Milkshake ha deciso di avvalersi dello standard ISO/IEC 9126.

Questo modello è mirato a migliorare l'organizzazione e i processi di una società software. Di seguito verrà descritto il modello della qualità del software, in:

- Funzionalità
- Affidabilità
- Efficienza
- Usabilità
- Manutenibilità
- Portabilità

### 3.1 Funzionalità

La funzionalità è la capacità di un prodotto di rispondere ad esigenze specifiche. In questo caso le esigenze vengono descritte nel documento Analisi dei Requisiti.

#### 3.1.1 Obiettivi

- Appropriatezza: Capacità del software di riuscire a svolgere tutte le funzionalità prefissate con l'utente;
- Accuratezza: Capacità del software di svolgere correttamente ciò che era stato precedentemente concordato;
- Interoperabilità: Capacità del software di operare con più sistemi;
- Conformità: Capacità del software di aderire agli standard relativi alla funzionalità;
- Sicurezza: Capacità del software di non permettere alle persone non autorizzate di accedere o modificare dati sensibili dell'utente; consente alle persone autorizzate di accedere ai dati.

#### 3.1.2 Metriche

Viene specificata la completezza del software attraverso la seguente formula:

$$C = (1 - (\text{Funzionalità non implementate} / \text{Funzionalità implementate}))$$

Con i valori:

preferibile = 1;

accettabile = 1.

### 3.2 Affidabilità

L'affidabilità è la capacità di un certo software di mantenere un certo livello di prestazioni in determinate condizioni in un certo periodo.



### 3.2.1 Obiettivi

- **Maturità:** Capacità del prodotto di dare risultati corretti, esenti da malfunzionamenti o errori;
- **Tolleranza agli errori:** Capacità del prodotto di poter essere usabile anche in presenza di malfunzionamenti o casi derivanti un uso scorretto del software;
- **Recuperabilità:** Capacità del prodotto di recuperare almeno le informazioni rilevanti in seguito ad un malfunzionamento;
- **Aderenza:** Capacità del prodotto di aderire a standard inerenti all'affidabilità.

### 3.2.2 Metriche

Viene specificata l'abilità del software di resistere a malfunzionamenti attraverso la seguente formula:

$$R = \text{Numero di errori} / \text{Numero di test eseguiti}$$

Con i valori:

preferibile = 0;

accettabile < 0.15.

## 3.3 Efficienza

L'efficienza è la capacità del software di poter offrire un determinato livello di prestazioni in date condizioni in un certo periodo.

### 3.3.1 Obiettivi

- **Comportamento rispetto al tempo:** Capacità del prodotto di fornire adeguati livelli di elaborazione, velocità e tempi di risposta;
- **Utilizzo delle risorse:** Capacità del prodotto di utilizzare le risorse in maniera adeguata;
- **Conformità:** Capacità del prodotto di aderire a standard sull'efficienza.

### 3.3.2 Metriche

Visto che il proponente non ha incluso dettagli relativi alla qualità dell'efficienza, non verranno proposte metriche per questa sezione.

## 3.4 Usabilità

L'usabilità è la capacità del prodotto di essere compreso ed utilizzato dall'utente senza difficoltà tenendo conto certe condizioni.



### 3.4.1 Obiettivi

- **Comprensibilità:** Capacità del prodotto di visualizzare le varie funzionalità del software e permette all'utente di capire se il software è indicato per le sue esigenze;
- **Apprendibilità:** Capacità del prodotto di aumentare nel tempo l'abilità dell'utente di sfruttare il software;
- **Operabilità:** Capacità del prodotto che permette agli utenti di farne uso per i loro scopi;
- **Attrattiva:** Capacità del prodotto di rendere più piacevole l'utilizzo del software;
- **Conformità:** Capacità del prodotto di aderire a standard relativi all'usabilità.

### 3.4.2 Metriche

Viene specificata la facilità con cui l'utente riesce a raggiungere ciò che vuole attraverso il conteggio del numero di tocchi o click necessari al suo raggiungimento.

Si considera la capacità dell'operatore di visualizzare la propria lista delle task:

Numero di tocchi o click preferibili < 4;  
accettabile < 6.

Viene specificata la facilità con cui l'utente riesce a raggiungere ciò che vuole attraverso il conteggio dei secondi necessari al suo raggiungimento.

Si considera la capacità dell'operatore di visualizzare la propria lista delle task:

valore dei secondi preferibile < 15;  
accettabile < 40.

Viene specificata la profondità gerarchica massima dei collegamenti e delle funzionalità presenti all'interno del software:

valore preferibile < 4;  
accettabile < 6.

## 3.5 Manutenibilità

Capacità del prodotto di essere modificato anche in futuro.

### 3.5.1 Obiettivi

- **Analizzabilità:** Facilità con cui è possibile interpretare il codice del software;
- **Modificabilità:** Capacità per cui risulta non troppo oneroso modificare il codice del software;
- **Stabilità:** Capacità del software di evitare errori inaspettati derivanti da modifiche errate;
- **Testabilità:** Capacità del prodotto di essere testato al fine di validare le modifiche al codice sorgente.



### 3.5.2 Metriche

Viene specificata la leggibilità del software attraverso la seguente formula:

$$L = \text{Numero di linee di codice commentate} / \text{Numero di linee di codice}$$

Con i valori:

preferibile > 0.15;

accettabile > 0.10.

## 3.6 Portabilità

La portabilità è la capacità del software di poter funzionare senza tener conto di uno specifico ambiente di lavoro.

### 3.6.1 Obiettivi

- Adattabilità: Capacità del prodotto di essere adattato per diversi ambienti operativi;
- Installabilità: Capacità del prodotto di essere installato in uno specificato ambiente operativo;
- Conformità: Capacità del software di aderire a standard relativi alla portabilità;
- Sostituibilità: Capacità del software di sostituire un altro prodotto con le stesse funzionalità.

### 3.6.2 Metriche

IL software dovrà eseguire solamente su ambiente Docker, quindi non sono necessarie varie metriche.



## **4 Test**

### **4.1 Specifica dei test**

### **4.2 Test di accettazione**

### **4.3 Test di sistema**

### **4.4 Test di integrazione**

### **4.5 Test d'unità**

### **4.6 Resoconto attività di verifica**

#### **4.6.1 Strategia per l'analisi statica dei documenti**

#### **4.6.2 Esiti dell'indice di Gulpease**

#### **4.6.3 Conclusioni del periodo di analisi**

### **4.7 Lista di controllo**

### **4.8 Valutazioni per il miglioramento**