



>Three Way Milkshake_

Piano di progetto

Three Way Milkshake - Progetto "PORTACS"

threewaymilkshake@gmail.com

| | |
|---------------------|---|
| Versione | 0.0.1 |
| Stato | Non approvato |
| Uso | Esterno |
| Approvazione | Nome approvatore |
| Redazione | Tessari Andrea |
| Verifica | Nome verificatore |
| Destinatari | 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



Registro delle modifiche

| Versione | Descrizione | Data | Nominativo | Ruolo |
|----------|-------------------------|------------|--------------------|------------------|
| 0.0.9 | Redazione sezione §4 | 2021-01-06 | Crivellari Alberto | <i>Redattore</i> |
| 0.0.8 | Modifiche sezione §1 | 2021-01-06 | Crivellari Alberto | <i>Redattore</i> |
| 0.0.7 | Modifiche sezione §3 | 2021-01-05 | Crivellari Alberto | <i>Redattore</i> |
| 0.0.5 | Tabelle sezione §2 e §3 | 2021-01-04 | Crivellari Alberto | <i>Redattore</i> |
| 0.0.4 | Redazione sezione §2 | 2021-01-04 | Crivellari Alberto | <i>Redattore</i> |
| 0.0.3 | Modifiche sezione §1 | 2021-01-01 | Crivellari Alberto | <i>Redattore</i> |
| 0.0.3 | Redazione sezione §3 | 2021-01-01 | Tessari Andrea | <i>Redattore</i> |
| 0.0.2 | Redazione sezione §1 | 2020-12-30 | Crivellari Alberto | <i>Redattore</i> |
| 0.0.1 | Impostazione iniziale | 2020-12-27 | Tessari Andrea | <i>Redattore</i> |



Indice

| | | |
|----------|---------------------------------|-----------|
| 1 | Introduzione | 5 |
| 1.1 | Scopo del documento | 5 |
| 1.2 | Scopo del prodotto | 5 |
| 1.3 | Riferimenti | 5 |
| 1.3.1 | Normativi | 5 |
| 1.3.2 | Informativi | 5 |
| 2 | Qualità del processo | 7 |
| 2.1 | Scopo | 7 |
| 2.2 | Processi di Sviluppo | 7 |
| 2.2.1 | Analisi dei Requisiti | 7 |
| 2.2.1.1 | Metriche | 7 |
| 2.2.2 | Progettazione | 7 |
| 2.2.2.1 | Metriche | 7 |
| 2.2.3 | Codifica | 7 |
| 2.2.3.1 | Metriche | 7 |
| 2.3 | Processi di Supporto | 8 |
| 2.3.1 | Pianificazione | 8 |
| 2.3.1.1 | Metriche | 8 |
| 2.3.2 | Verifica | 9 |
| 2.3.2.1 | Metriche | 9 |
| 2.3.3 | Documentazione | 9 |
| 2.3.3.1 | Metriche | 9 |
| 2.3.4 | Tabella riassuntiva | 10 |
| 3 | Qualità del prodotto | 11 |
| 3.1 | Funzionalità | 11 |
| 3.1.1 | Obiettivi | 11 |
| 3.1.2 | Metriche | 11 |
| 3.2 | Affidabilità | 12 |
| 3.2.1 | Obiettivi | 12 |
| 3.2.2 | Metriche | 12 |
| 3.3 | Efficienza | 12 |
| 3.3.1 | Obiettivi | 12 |
| 3.3.2 | Metriche | 12 |
| 3.4 | Usabilità | 12 |
| 3.4.1 | Obiettivi | 13 |
| 3.4.2 | Metriche | 13 |
| 3.5 | Manutenibilità | 13 |
| 3.5.1 | Obiettivi | 14 |
| 3.5.2 | Metriche | 14 |
| 3.6 | Portabilità | 14 |
| 3.6.1 | Obiettivi | 14 |
| 3.6.2 | Metriche | 14 |
| 3.7 | Tabella Riassuntiva | 15 |



| | | |
|----------|--|-----------|
| 4 | Test | 16 |
| 4.1 | Specifica dei test | 16 |
| 4.2 | Test di Accettazione | 17 |
| 4.3 | Test di Sistema | 17 |
| 4.4 | Test di Integrazione | 17 |
| 4.5 | Test di Unità | 17 |
| 4.6 | Resoconto attività di verifica | 17 |
| 4.6.1 | Esiti dell'indice di Gulpease | 17 |



Elenco delle figure

| | |
|--|----|
| 4.1.1 Figura esplicativa del Modello a V_G | 16 |
|--|----|



Elenco delle tabelle

| | |
|--|----|
| 2.3.1 Tabella riassuntiva metriche di processo | 10 |
| 3.7.1 Tabella riassuntiva metriche di processo | 15 |

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

Il capitolato C5 propone un progetto in cui viene richiesto lo sviluppo di un software per il monitoraggio in tempo reale di unità che si muovono in uno spazio definito. All'interno di questo spazio, creato dall'utente per riprodurre le caratteristiche di un ambiente reale, le unità dovranno essere in grado di circolare in autonomia, o sotto il controllo dell'utente, per raggiungere dei punti di interesse posti nella mappa. La circolazione è sottoposta a vincoli di viabilità e ad ostacoli propri della topologia dell'ambiente, deve evitare le collisioni con le altre unità e prevedere la gestione di situazioni critiche nel traffico.

Il progetto PORTACS si pone come obiettivo finale di dimostrare la fattibilità di sviluppare un software che permetta il monitoraggio in tempo reale di unità che si muovono in uno spazio per raggiungere una lista ordinata di punti d'interesse. Per facilitare lo sviluppo del progetto e dopo accordo con l'azienda, si è deciso di contestualizzare lo sviluppo ad un magazzino in cui il sistema centrale pilota i vari muletti verso le destinazioni.

1.3 Riferimenti

1.3.1 Normativi

- NORME DI PROGETTO: per qualsiasi convenzione sulla nomenclatura degli elementi presenti all'interno del documento;
- 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>

1.3.2 Informativi

- GLOSSARIO: per la definizione dei termini e degli acronimi evidenziati nel documento;
- Capitolato d'appalto C5-PORTACS:
<https://www.math.unipd.it/~tullio/IS-1/2020/Progetto/C5.pdf>
- Software Engineering - Iam Sommerville - 10th Edition
- 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 Scopo

Per valutare la qualità del prodotto, il gruppo Three Way Milkshake ha deciso di avvalersi degli standard ISO/IEC_G 12207:1995 e ISO/IEC_G 15504, semplificandoli e riadattandoli in base alle esigenze del gruppo di lavoro.

I processi individuati sono presentati di seguito.

2.2 Processi di Sviluppo

2.2.1 Analisi dei Requisiti

2.2.1.1 Metriche

Requisiti obbligatori soddisfatti (PROS)

Indica la quantità di requisiti obbligatori soddisfatti rispetto al totale.

- **Misurazione:** percentuale; $\frac{\text{requisiti_obbligatori_soddisfatti}}{\text{requisiti_obbligatori_totali}}$;
- **Valore preferibile:** 100%;
- **Valore accettabile:** 100%.

2.2.2 Progettazione

2.2.2.1 Metriche

Accoppiamento tra classi ed oggetti (CBO)

Indica l'accoppiamento tra classi e oggetti; due classi si dicono accoppiate se una utilizza metodi o variabili dell'altra.

- **Misurazione:** valore intero;
- **Valore preferibile:** $0 \leq \text{CBO} \leq 1$;
- **Valore accettabile:** $0 \leq \text{CBO} \leq 6$.

2.2.3 Codifica

2.2.3.1 Metriche

Profondità delle gerarchie (DEP)

Indica la profondità delle gerarchie nel codice sviluppato.

- **Misurazione:** valore intero;
- **Valore preferibile:** $\text{DEP} \leq 2$;
- **Valore accettabile:** $\text{DEP} \leq 3$.



Livello di Annidamento (LEV)

Indica il livello di annidamento nei vari metodi presenti nel codice prodotto.

- **Misurazione:** valore intero;
- **Valore preferibile:** $1 \leq \text{LEV} \leq 3$;
- **Valore accettabile:** $1 \leq \text{LEV} \leq 6$.

Parametri per metodo (PAR)

Indica il numero di parametri presenti nei metodi sviluppati nel codice.

- **Misurazione:** valore intero;
- **Valore preferibile:** $\text{PAR} \leq 4$;
- **Valore accettabile:** $\text{PAR} \leq 6$.

Rapporto Codice-Commenti (RCC)

Indica il rapporto tra le linee di codice e le linee di commento all'interno del file.

- **Misurazione:** valore decimale; $\frac{\text{linee_codice}}{\text{linee_commento}}$;
- **Valore preferibile:** $\text{RCC} \geq 0.4$;
- **Valore accettabile:** $\text{RCC} \geq 0.2$.

2.3 Processi di Supporto

2.3.1 Pianificazione

2.3.1.1 Metriche

Budget at Completion (BAC)

Indica il budget totale allocato per il progetto

- **Misurazione:** valore intero;
- **Valore preferibile:** *preventivo*;
- **Valore accettabile:** $\text{preventivo} - 5\% \leq \text{BAC} \leq \text{preventivo} + 5\%$.

Earned Value (EV)

Indica la quantità di guadagno ottenuta dal lavoro effettuato fino al momento del calcolo.

- **Misurazione:** $\text{preventivo} \cdot \%_{\text{lavoro_pianificato}}$;
- **Valore preferibile:** $\text{EV} \geq 0$;
- **Valore accettabile:** $\text{EV} \geq 0$.

Planned Value (PV)

Indica la quantità di guadagno stimata sul lavoro pianificato al momento del calcolo.

- **Misurazione:** $\text{preventivo} \cdot \%_{\text{lavoro_pianificato}}$;
- **Valore preferibile:** $\text{PV} \geq 0$;
- **Valore accettabile:** $\text{PV} \geq 0$;



Schedule Variance (SV)

Indica l'anticipo o il ritardo del lavoro effettuato rispetto alla pianificazione.

- **Misurazione:** EV-PV;
- **Valore preferibile:** $SV \geq 0$;
- **Valore accettabile:** $SV = 0$.

Actual Cost (AC)

Il denaro speso fino al momento del calcolo.

- **Misurazione:** valore intero;
- **Valore preferibile:** $0 \leq AC \leq PV$;
- **Valore accettabile:** $0 \leq AC \leq budget$.

Cost Variance (CV)

Il discostamento tra il costo del lavoro ad ora effettuato e il costo preventivato.

- **Misurazione:** EV-AC;
- **Valore preferibile:** $CV \geq 0$;
- **Valore accettabile:** $CV \geq 0$.

2.3.2 Verifica

2.3.2.1 Metriche

Code Coverage (CC)

Indica la quantità di codice attraversato durante l'esecuzione dei test; aiuta a valutare la completezza dei test.

- **Misurazione:** percentuale; $\frac{linee_codice_verificate}{linee_codice_totali}$;
- **Valore preferibile:** 100%;
- **Valore accettabile:** 75%.

2.3.3 Documentazione

2.3.3.1 Metriche

Indice di Gulpease(IG)

Esprime una valutazione della qualità prodotta, stimandone la leggibilità.

- **Misurazione:** $\left[89 + \frac{(300 - num_frasi - 10 \cdot num_lettere)}{num_parole} \right]$;
- **Valore preferibile:** $80 \leq IG \leq 100$;
- **Valore accettabile:** $50 \leq IG \leq 100$.

2.3.4 Tabella riassuntiva

Tabella 2.3.1: Tabella riassuntiva metriche di processo

| Codice | Tipo Processo | Valori Preferibili | Valori Accettabili |
|--------|-----------------------|-----------------------|--|
| PROS | Analisi dei Requisiti | 100% | 100% |
| CBO | Progettazione | $0 \leq CBO \leq 1$ | $0 \leq CBO \leq 6$ |
| DEP | Codifica | $DEP \leq 2$ | $DEP \leq 3$ |
| LEV | Codifica | $1 \leq LEV \leq 3$ | $1 \leq LEV \leq 6$ |
| PAR | Codifica | $PAR \leq 4$ | $PAR \leq 6$ |
| RCC | Codifica | $RCC \geq 0.4$ | $RCC \geq 0.2$ |
| BAC | Pianificazione | <i>preventivo</i> | <i>preventivo $\pm 5\%$</i> |
| EV | Pianificazione | $EV \geq 0$ | $EV \geq 0$ |
| PV | Pianificazione | $PV \geq 0$ | $PV \geq 0$ |
| SV | Pianificazione | $SV \geq 0$ | $SV = 0$ |
| AC | Pianificazione | $0 \leq AC \leq PV$ | $0 \leq AC \leq budget$ |
| CV | Pianificazione | $CV \geq 0$ | $CV \geq 0$ |
| CC | Verifica | 100% | 75% |
| IG | Documentazione | $80 \leq IG \leq 100$ | $50 \leq IG \leq 100$ |

3 Qualità del prodotto

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

Completezza del Software(Cs)

Viene specificata la completezza del software.

- **Misurazione:** $C = (1 - (\text{Funzionalità non implementate} / \text{Funzionalità implementate}))$;
- **Valore preferibile:** Cs = 1;
- **Valore accettabile:** Cs = 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

Affidabilità del Software A

Viene specificata l'abilità del software di resistere a malfunzionamenti.

- **Misurazione:** $A = \text{Numero di errori} / \text{Numero di test eseguiti}$;
- **Valore preferibile:** $A = 0$;
- **Valore accettabile:** $A < 0.15$.

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

Numero di tocchi/click necessari C)

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.

- **Misurazione:** L = Numero di tocchi o click necessari per il raggiungimento dell'obiettivo;
- **Valore preferibile:** $C < 4$;
- **Valore accettabile:** $C < 6$.

Numero di secondi necessari S 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.

- **Misurazione:** L = Numero di secondi necessari per il raggiungimento dell'obiettivo;
- **Valore preferibile:** $S < 15$;
- **Valore accettabile:** $S < 40$.

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

- **Misurazione:** P = Profondità gerarchica massima dei collegamenti e delle funzionalità presenti all'interno del software;
- **Valore preferibile:** $P < 4$;
- **Valore accettabile:** $P < 6$.

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

Leggibilità del Software L

Viene specificata l'abilità del software di resistere a malfunzionamenti.

- **Misurazione:** $L = \text{Numero di linee di codice commentate} / \text{Numero di linee di codice}$;
- **Valore preferibile:** $L > 0.15$;
- **Valore accettabile:** $L > 0.10$.

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.

3.7 Tabella Riassuntiva

Tabella 3.7.1: Tabella riassuntiva metriche di processo

| Nome Metrica | Descrizione | Tipo Capacità | Val. Pref. | Val. Accett. |
|------------------------------------|---|----------------|------------|--------------|
| Completezza del Software Cs | Funzionalità non implementate rispetto alle funzionalità implementate | Funzionalità | Cs = 1 | Cs = 1 |
| Affidabilità del Software A | Errori rispetto al numero di test eseguiti | Affidabilità | A = 0 | A < 0.15 |
| Numero di tocchi/click necessari C | Numero di tocchi o click necessari per visualizzare la propria lista di task | Usabilità | C < 4 | C < 6 |
| Numero di secondi necessari S | Numero di secondi necessari per visualizzare la propria lista di task | Usabilità | S < 15 | S < 40 |
| Profondità gerarchica P | Profondità gerarchica massima dei collegamenti e funzionalità presenti all'interno del software | Usabilità | P < 4 | P < 6 |
| Leggibilità software L | Numero di linee di codice commentate rispetto al totale di linee di codice | Manutenibilità | L > 0.15 | L > 0.10 |

4 Test

4.1 Specifica dei test

Per garantire la qualità del prodotto, *Three Way Milkshake* adotta il Modello a V_G per verificare tramite test ogni passo della produzione software.

Qui vedremo un'immagine rappresentativa del Modello a V_G (o V-Model), quest'ultimo si può schematizzare posizionando il tempo nell'asse delle ascisse e il livello di astrazione nell'asse delle ordinate.

Il modello idealmente si divide in 2 rami.

Il ramo sinistro contiene le fasi di progettazione e ideazione; il ramo destro contiene le fasi di testing e integrazione.

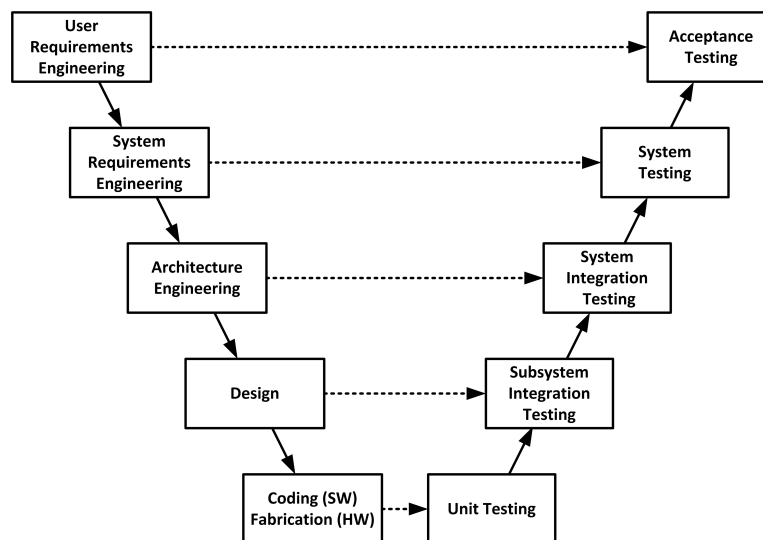


Figura 4.1.1: Figura esplicativa del Modello a V_G

Per definire lo stato dei test viene utilizzato un valore da 0 a 2:

- **0:** - Il test non è stato implementato;
- **1:** - Il test è stato implementato, ma fallisce;
- **2:** - Il test è stato implementato ed è stato superato.

Vi sono quattro tipi di test:

- Test di Accettazione;
- Test di Sistema;
- Test di Integrazione;
- Test di Unità.



4.2 Test di Accettazione

I Test di Accettazione verificano che il software nel suo complesso soddisfi i criteri di accettazione decisi con il cliente.

I test di accettazione verranno indicati nel seguente modo:

TA[Importanza][Tipo][Codice]

ove:

- **Importanza:** indica l'importanza del requisito tra i seguenti livelli d'importanza:
 - **O** per i requisiti obbligatori;
 - **D** per i requisiti desiderabili;
 - **F** per i requisiti facoltativi.
- **Tipo:** indica il tipo dei requisiti tra i seguenti tipi:
 - **F** per i requisiti funzionali;
 - **V** per i requisiti di vincolo;
 - **Q** per i requisiti di qualità;
 - **P** per i requisiti prestazionali.
- **Codice:** rappresenta il codice identificativo crescente del componente da verificare.

4.3 Test di Sistema

I Test di Sistema verificano la conformità dell'intero sistema con i requisiti specificati.

I Test di Sistema verranno sviluppati quando verrà raggiunta la fase appropriata, secondo il Modello a V_G.

4.4 Test di Integrazione

I Test di Integrazione verificano l'integrazione di più componenti software o hardware.

I Test di Integrazione verranno sviluppati quando verrà raggiunta la fase appropriata, secondo il Modello a V_G.

4.5 Test di Unità

I Test di Unità verificano le parti atomiche del software (per esempio funzioni o procedure). Vengono utilizzati per assicurarsi che la logica interna del codice sia rispettata.

I Test di Unità verranno sviluppati quando verrà raggiunta la fase appropriata, secondo il Modello a V_G.

4.6 Resoconto attività di verifica

4.6.1 Esiti dell'indice di Gulpease



Glossario dei Termini

Modello a V Modello di sviluppo software esteso dal modello a cascata. Pone la scrittura del test del software nelle fasi iniziali dello sviluppo e prima della codifica.. 16, 17