



# Piano di Qualifica

7DOS - 17 Dicembre 2018

## Informazioni sul documento

<b>Versione</b>	1.0.0
<b>Responsabile</b>	Marco Costantino
<b>Verifica</b>	Lorenzo Busin Michele Roverato
<b>Redazione</b>	Giacomo Barzon Giovanni Sorice Nicolò Tartaggia Andrea Trevisin
<b>Stato</b>	Approvato
<b>Uso</b>	Esterno
<b>Destinato a</b>	Prof.Tullio Vardanega Prof.Riccardo Cardin Zucchetti 7DOS
<b>Email</b>	<a href="mailto:7dos.swe@gmail.com">7dos.swe@gmail.com</a>

## Descrizione

Questo documento descrive le operazioni di verifica e validazione relative al progetto *G&B*.

---

## Diario delle modifiche

Modifica	Autore	Ruolo	Data	Versione
<i>Approvazione del documento</i>	Marco Costantino	Responsabile	2018-12-17	1.0.0
<i>Verifica del documento</i>	Nicolò Tartaggia	Verificatore	2018-12-3	0.5.1
<i>Stesura capitolato C4</i>	Giovanni Sorice	Analista	2018-11-30	0.5.0
<i>Stesura capitolati C1 e C2</i>	Giacomo Barzon	Analista	2018-11-29	0.4.0
<i>Verifica capitolati C3, C5, C6</i>	Lorenzo Busin	Verificatore	2018-11-28	0.3.1
<i>Stesura capitolato C3</i>	Giovanni Sorice	Analista	2018-11-28	0.3.0
<i>Stesura capitolato C6</i>	Michele Roverato	Analista	2018-11-27	0.2.0
<i>Stesura capitolato C5</i>	Marco Costantino	Analista	2018-11-26	0.1.0
<i>Stesura della sezione Introduzione</i>	Giovanni Sorice	Analista	2018-11-25	0.0.2
<i>Stesura dello scheletro del documento</i>	Giovanni Sorice	Analista	2018-11-25	0.0.1

---

# Indice

<b>1</b>	<b>Introduzione</b>	<b>4</b>
1.1	Scopo del documento	4
1.2	Scopo del prodotto	4
1.3	Glossario	4
1.4	Riferimenti	4
1.4.1	Normativi	5
1.4.2	Informativi	5
<b>2</b>	<b>Qualità di processo</b>	<b>6</b>
2.1	Scopo	6
2.2	Gestione	7
2.3	Processi	7
2.3.1		7
2.3.2		7
2.3.3		7
<b>3</b>	<b>Qualità di prodotto</b>	<b>8</b>
3.1	Definizione delle metriche	8
3.2	Functional Suitability	8
3.2.1	Sotto-caratteristiche notevoli	8
3.2.2	Metriche adottate	9
3.2.2.1	Functional Implementation Completeness	9
3.2.2.2	Average Functional Implementation Correctness	9
3.3	Performance Efficiency	10
3.3.1	Sotto-caratteristiche notevoli	10
3.3.2	Metriche adottate	10
3.3.2.1	Average Response Time	10
3.3.2.2	Average CPU Usage	10
3.3.2.3	Average Memory Usage	11
3.4	Usability	11
3.4.1	Sotto-caratteristiche notevoli	11
3.4.2	Metriche adottate	11
3.4.2.1	Average Learning Time	11
3.5	Reliability	12
3.5.1	Sotto-caratteristiche notevoli	12
3.5.2	Metriche adottate	12
3.5.2.1	Failure Density	12
3.6	Maintainability	12
3.6.1	Sotto-caratteristiche notevoli	13
3.6.2	Metriche Adottate	13
3.6.2.1	Average Time Spent per Feature Development	13
3.6.2.2	Average time spent per error correction	13
3.6.2.3	Average time spent per unit testing	14
<b>4</b>	<b>Specifica dei test</b>	<b>15</b>

<b>5</b>	<b>Metriche dei test</b>	<b>16</b>
5.1	Metriche per i processi	16
5.2	Metriche per i prodotti	17
5.2.1	Metriche relative al codice software	17
5.2.1.1	Percentuale di requisiti soddisfatti	17
5.2.1.2	Percentuale di test case passati	17
5.2.1.3	Average Response Time	18
5.2.1.4	Average time spent per feature development	18
5.2.1.5	Average time spent per error correction	18
5.2.1.6	Average time spent per unit testing	19
5.2.2	Metriche relative alla documentazione	19
5.2.2.1	Numero di errori grammaticali	19
5.2.2.2	Gunning fog index	19
5.2.2.3	Indice di Gulpease	20
5.2.2.4	Indice di Flesh	20
<b>6</b>	<b>Resoconto delle attività di verifica</b>	<b>21</b>

# 1 Introduzione

## 1.1 Scopo del documento

Il presente documento ha lo scopo di esporre dettagliatamente le norme, le metodologie e gli standard che il gruppo 7DOS intende adottare per assicurare che ogni *prodotto<sub>g</sub>*, di natura documentale o applicativa, aderisca ai vincoli di *qualità<sub>g</sub>* stabiliti dal *proponente<sub>g</sub>*. Per garantire il rispetto di tali vincoli si prevede un continuo *processo<sub>g</sub>* di verifica delle attività svolte dal gruppo, al fine di individuare eventuali problematiche nel minor tempo possibile permettendo immediati interventi risolutivi.

## 1.2 Scopo del prodotto

Il prodotto da realizzare consiste in un *plugin<sub>g</sub>* per il software di monitoraggio *Grafana<sub>g</sub>*, da sviluppare in linguaggio *JavaScript<sub>g</sub>*. Il prodotto dovrà svolgere almeno le seguenti funzioni:

- Leggere la definizione di una *rete Bayesiana<sub>g</sub>*, memorizzata in formato *JSON<sub>g</sub>*;
- Associare dei nodi della rete Bayesiana ad un flusso di dati presente nel sistema di Grafana;
- Ricalcolare i valori delle probabilità della rete secondo regole temporali prestabilite;
- Derivare nuovi dati dai nodi della rete non collegati al flusso di dati, e fornirli al sistema di Grafana;
- Visualizzare i dati mediante il sistema di creazione di grafici e *dashboard<sub>g</sub>* a disposizione.

Viene previsto un utilizzo del prodotto da parte dell'azienda proponente per il monitoraggio di sistemi gestionali in Cloud; tuttavia, dato l'obiettivo di rendere il prodotto open-source, esso dovrà essere utilizzabile indipendentemente dal particolare sistema che si desidera monitorare.

## 1.3 Glossario

Per rendere la lettura del documento più semplice, chiara e comprensibile viene allegato il *Glossario v1.0.0* nel quale sono contenute le definizioni dei termini tecnici, dei vocaboli ambigui, degli acronimi e delle abbreviazioni. La presenza di un termine all'interno del Glossario è segnalata con una "g" posta come pedice (esempio: *Glossario<sub>g</sub>*).

## 1.4 Riferimenti

TODO in fase di redazione

#### 1.4.1 Normativi

- **ISO/IEC 12207:** [https://www.math.unipd.it/~tullio/IS-1/2009/Approfondimenti/ISO\\_12207-1995.pdf](https://www.math.unipd.it/~tullio/IS-1/2009/Approfondimenti/ISO_12207-1995.pdf) (Ultima consultazione effettuata: TODO da inserire);
- Da verbali .

#### 1.4.2 Informativi

- **Norme di Progetto:** *Norme di Progetto v1.0.0.*
- **Piano di Progetto:** *Piano di Progetto v1.0.0.*
- **Capitolato C6:** Soldino: piattaforma *Ethereum<sub>g</sub>* per pagamenti IVA  
<https://www.math.unipd.it/~tullio/IS-1/2018/Progetto/C6.pdf>;

## 2 Qualità di processo

### 2.1 Scopo

Per poter raggiungere gli obiettivi prefissati è necessario che i processi che portano al loro compimento garantiscano un buon livello di qualità. Il gruppo 7DOS, per questo motivo, ha deciso di adottare la normativa ISO/IEC 15504 (chiamata anche SPICE) e di seguire il principio di miglioramento continuo ( $PDCA_g$ ).

Per ogni  $processo_g$  lo standard definisce una scala di maturità a cinque livelli (più il livello base, detto "livello 0"), riportati di seguito:

- **Livello 0 - Incomplete process:** il processo riporta  $performance_g$  e risultati incompleti, inoltre è gestito in modo caotico.
- **Livello 1 - Performed process:** il processo raggiunge i risultati attesi ma viene eseguito in modo non controllato. Gli attributi di tale processo sono:
  - **1.1 - Process performance:**
- **Livello 2 - Managed process:** il processo è pianificato e tracciato secondo standard prefissati, dunque il suo prodotto è controllato, mantenuto e soddisfa determinati criteri di qualità. Gli attributi di tale processo sono:
  - **2.1 - Performance management:**
  - **2.2 - Work product management:**
- **Livello 3 - Established process:** il processo possiede specifici standard organizzativi che includono linee guida personalizzate, il tutto è consolidato tramite una politica di feedback del prodotto. Gli attributi di tale processo sono:
  - **3.1 - Process definition:**
  - **3.2 - Process deployment:**
- **Livello 4 - Predictable process:** il processo è quantitativamente misurato e statisticamente analizzato per permettere di prendere decisioni oggettive e per assicurare che le prestazioni rimangano all'interno di limiti definiti. Gli obiettivi sono, di conseguenza, supportati in maniera consistente. Gli attributi di tale processo sono:
  - **4.1 - Process measurement:**
  - **4.2 - Process control:**
- **Livello 5 - Optimizing process:** il processo è in continuo miglioramento per raggiungere adeguatamente gli obiettivi prefissati. Gli attributi di tale processo sono:
  - **5.1 - Process innovation:**
  - **5.2 - Process optimization:**

Lo standard SPICE offre una scala di valutazione per ogni processo, in modo da misurare il livello di raggiungimento degli stessi:

- **N - Not achieved:** 0 - 15%;
- **P - Partially achieved:** >15% - 50%;
- **L - Largely achieved:** >50% - 85%;
- **F - Fully Achieved:** >85% - 100%;

Il ciclo di miglioramento continuo (PDCA, *Plan-Do-Check-Act*) prevede quattro fasi iterative che permettono di controllare costantemente lo sviluppo di un processo, in modo da poter perseguire la miglior qualità di quest'ultimo:

- **Plan:** in questa fase vengono definiti elementi estremamente importanti che riguardano il ciclo di miglioramento continuo. In particolare vengono fissati obiettivi, processi da utilizzare, risultati da ottenere, personale incaricato per i vari processi e scadenze da rispettare;
- **Do:** in questa fase vengono avviate tutte le attività previste da completare entro la data stabilita;
- **Check:** in questa fase vengono confrontati i risultati ottenuti dalle varie attività con quelli ipotizzati durante la fase Plan;
- **Act:** in questa fase vengono individuate le possibili problematiche che hanno prodotto risultati differenti da quelli attesi. Di conseguenza, verranno determinate tutte le attività da revisionare per migliorare la qualità del processo.

## 2.2 Gestione

Il gruppo 7dos ha deciso di seguire l'approccio a maturità di processo per i seguenti motivi:

- predisposizione alla cura della qualità di prodotto e di processi;
- previsto nelle buone pratiche di *management<sub>g</sub>*;
- adatto per i neofiti della programmazione progettuale.

## 2.3 Processi

Per garantire qualità di processo verranno seguiti macro-processi.

### 2.3.1

### 2.3.2

### 2.3.3



## 3 Qualità di prodotto

Per poter garantire che il prodotto realizzato sia di alta qualità, è necessario definire un modello per la valutazione di quest'ultima; il team 7DOS, per questo motivo, ha scelto di adottare il modello di qualità delineato nello standard ISO/IEC 25010, anche noto come *SQuaRE<sub>g</sub>*.

Tale modello comprende 8 caratteristiche (ciascuna divisa in sotto-caratteristiche, per un totale di 31) che vanno prese in considerazione durante lo sviluppo del progetto per garantire un'elevata qualità complessiva del prodotto finale.

Per praticità e rilevanza ai fini del prodotto, sono state selezionate 5 caratteristiche da considerare e per ciascuna sono state individuate le sotto caratteristiche più rilevanti al progetto. In particolare, sono state scartate: *Compatibility<sub>g</sub>*, in quanto andando a realizzare un plugin (di natura integrato in un sistema preesistente) è stata giudicata superflua; *Security<sub>g</sub>*, in quanto il plugin non dovrà gestire autenticazione o raccolta di dati; ed infine *Portability<sub>g</sub>* in quanto essendo il prodotto un plugin per un determinato sistema, non è rilevante la sua portabilità ad altri ambienti.

### 3.1 Definizione delle metriche

Il processo di verifica, per essere informativo, deve esse quantificabile. Sono state quindi stabilite delle metriche di natura numerica, in modo da misurare accuratamente le varie caratteristiche di qualità del prodotto. Per ogni metrica sono presenti spiegazione, motivazione, formula usata per il calcolo e tre intervalli di valori. Questi ultimi sono distinti come segue:

- **Intervallo possibile:** tutti i valori che la metrica può assumere;
- **Intervallo accettabile:** i valori che vengono considerati oltre la soglia di accettabilità per la metrica;
- **Intervallo obiettivo:** i valori che vengono considerati l'obiettivo ottimale da raggiungere.

### 3.2 Functional Suitability

Questa caratteristica esprime il grado di soddisfacimento dei requisiti espliciti ed impliciti da parte di un prodotto o servizio, quando utilizzato sotto determinate condizioni.

#### 3.2.1 Sotto-caratteristiche notevoli

In riguardo al prodotto che si intende realizzare, sono state individuate le seguenti sotto-caratteristiche da perseguire come obiettivi prioritari:

- **Functional Completeness:** esprime il grado in cui l'insieme di funzioni copre i compiti specificati e gli obiettivi dell'utente;

- **Functional Correctness:** esprime il grado in cui il prodotto restituisce risultati corretti, entro il livello di precisione desiderato.

### 3.2.2 Metriche adottate

#### 3.2.2.1 Functional Implementation Completeness

Misurazione in percentuale del grado in cui le funzionalità offerte dalla corrente implementazione del software coprono l'insieme di funzioni specificate nei requisiti.

Questa metrica è stata scelta per valutare il grado di completezza del prodotto; l'obiettivo è implementare tutte le funzionalità richieste.

Viene utilizzata la seguente formula:

$$FI_{Comp} = \frac{NF_i}{NF_r} * 100$$

dove  $FI_{Comp}$  è il valore della metrica,  $NF_i$  è numero di funzioni attualmente implementate e  $NF_r$  è numero di funzioni specificate dai requisiti.

- **Intervallo possibile:** 0-100;
- **Intervallo accettabile:** 75-100;
- **Intervallo obiettivo:** 100.

#### 3.2.2.2 Average Functional Implementation Correctness

Misurazione in percentuale del grado in cui le funzionalità offerte dalla corrente implementazione del software, in media, rispettano il livello di precisione indicato nei requisiti.

Questa metrica è stata scelta per valutare il grado di accuratezza e garantire la qualità dei risultati restituiti dal prodotto, in quanto andrà a fare previsioni sulla verosimiglianza di alcuni eventi in base ai dati forniti, ed è necessario che tali previsioni siano sufficientemente accurate.

Viene utilizzata la seguente formula:

$$aFI_{Corr} = \frac{\sum_{i=1}^N \frac{iPF_i}{rPF_i}}{N} * 100$$

dove  $aFI_{Corr}$  è il valore della metrica,  $iPF_i$  è il livello di precisione della i-esima funzione implementata,  $rPF_i$  è il livello di precisione della i-esima funzione secondo i requisiti, e N è il numero totale di funzioni considerate.

- **Intervallo possibile:** 0-100;
- **Intervallo accettabile:** 80-100;
- **Intervallo obiettivo:** 95-100.

### 3.3 Performance Efficiency

Definisce le prestazioni relative al sistema come, quantità di risorse utilizzate per eseguire una determinata funzionalità del sistema sotto delle specifiche condizioni.

#### 3.3.1 Sotto-caratteristiche notevoli

- **Time Behaviour:** esprime il grado in cui i tempi di risposta ed elaborazione e i volumi di produzione di un prodotto o sistema, durante l'esecuzione delle sue funzionalità, rispettano i requisiti.
- **Resource Utilization:** esprime il grado in cui il numero e tipo di risorse utilizzate da un prodotto o sistema, durante l'esecuzione delle sue funzionalità, rispetta i requisiti.
- **Capacity:** esprime il grado in cui i limiti massimi di un prodotto o sistema rispettano quelli definiti dai requisiti.

#### 3.3.2 Metriche adottate

##### 3.3.2.1 Average Response Time

Tempo medio richiesto per completare l'esecuzione di una funzionalità richiesta ed esporre il risultato

$$ARM = \frac{\sum_{i=1}^N TF_i}{NF}$$

Dove:

- **ARM** :Average Response Time
- **TF<sub>i</sub>**: Tempo trascorso dal momento in cui viene richiesta l'esecuzione dell'i-esima funzionalità fino al momento in cui la funzionalità viene effettivamente completata ed il risultato viene esposto
- **NF**: Numero di funzionalità eseguite durante una singola esecuzione del prodotto software

**Range Ottimale:**0.3s-0.5s

**Range Accettabile:**0.5s-1.0s

##### 3.3.2.2 Average CPU Usage

Percentuale media di utilizzo del processore durante una singola esecuzione del prodotto SW(Non so come rappresentarla sotto forma di formula)

### 3.3.2.3 Average Memory Usage

Percentuale media di utilizzo della Memoria RAM durante una singola esecuzione del prodotto SW(Non so come rappresentarla sotto forma di formula)

## 3.4 Usability

Questa caratteristica esprime il grado in cui un prodotto o sistema può essere usato da un determinato utente per raggiungere determinati scopi con efficacia, efficienza e soddisfazione in uno specifico contesto d'uso.

### 3.4.1 Sotto-caratteristiche notevoli

In riguardo al prodotto che si intende realizzare, sono state individuate le seguenti sotto-caratteristiche da perseguire come obiettivi prioritari.

- **Learnability:** esprime il grado in cui un prodotto o sistema può essere usato da determinati utenti per raggiungere specifici obiettivi di imparare ad usare il prodotto o sistema con efficacia, efficienza, sicurezza da rischi e soddisfazione in un dato contesto d'uso.
- **Operability:** esprime il grado in cui un prodotto o sistema ha attributi che lo rendono facile da operare e controllare.
- **User Error Protection:** esprime il grado in cui un sistema protegge gli utenti dal commettere errori.

### 3.4.2 Metriche adottate

#### 3.4.2.1 Average Learning Time

Misurazione in minuti del tempo medio impiegato da un utente per imparare ad utilizzare una singola funzionalità del prodotto.

Questa metrica è stata scelta poiché, trattandosi di un prodotto che verrà reso disponibile pubblicamente, è stato ritenuto importante renderlo semplice da imparare per permetterne l'uso ad una vasta gamma di utenti.

Viene utilizzata la seguente formula:

$$aLT = \frac{\sum_{i=1}^N LT_i}{N}$$

dove aLT è il valore della metrica,  $LT_i$  è il tempo necessario ad imparare ad utilizzare la i-esima funzione implementata, espresso in minuti, e N è il numero totale di funzioni considerate.

- **Intervallo possibile:** 0- $\infty$ ;

- **Intervallo accettabile:** 0-30;
- **Intervallo obiettivo:** 0-15.

## 3.5 Reliability

Questa caratteristica esprime il grado in cui un sistema, prodotto o componente esegue determinate funzioni sotto specifiche condizioni per un dato periodo di tempo.

### 3.5.1 Sotto-caratteristiche notevoli

In riguardo al prodotto che si intende realizzare, sono state individuate le seguenti sotto-caratteristiche da perseguire come obiettivi prioritari:

- **Maturity:** esprime il grado in cui un sistema, prodotto o componente raggiunge i requisiti di affidabilità in normali condizioni operative;
- **Fault Tolerance:** esprime il grado in cui un sistema, prodotto o componente opera come previsto nonostante la presenza di malfunzionamenti hardware o software.

### 3.5.2 Metriche adottate

#### 3.5.2.1 Failure Density

Misurazione in percentuale della quantità di test falliti rispetto alla quantità di test eseguiti.

Questa metrica è stata scelta per garantire che il prodotto sia generalmente stabile e non risulti poco utilizzabile o inutilizzabile a causa di eccessive failure. Valutazioni più precise saranno effettuate in base ai singoli risultati dei test.

Viene utilizzata la seguente formula:

$$FD = \frac{T_f}{T_c} * 100$$

dove FD è il valore della metrica,  $T_f$  è il numero di test falliti e  $T_e$  è il numero di test eseguiti.

- **Intervallo possibile:** 0-100;
- **Intervallo accettabile:** 0-10;
- **Intervallo obiettivo:** 0.

## 3.6 Maintainability

Definisce il grado di efficacia ed efficienza con cui un prodotto o un sistema può essere modificato per migliorarlo, correggerlo o adattarlo a dei cambiamenti all'ambiente.

### 3.6.1 Sotto-caratteristiche notevoli

- **Modularity**: Grado di scomposizione del sistema in parti minimali tali che un cambiamento ad una specifica componente ha il minimo impatto su tutte le altre componenti.
- **Reusability**: Grado con cui una determinata componente del sistema può essere adattata ed utilizzata in altri sistemi o per realizzare altre componenti.
- **Analizability**: Grado di efficacia ed efficienza con cui è possibile analizzare l'impatto nel sistema di uno specifico cambiamento ad una o più delle sue parti, ai fini di rilevare eventuali casi di fallimento.
- **Modificability**: Grado con il quale un prodotto o sistema può essere modificato efficacemente ed efficientemente senza introdurre difetti che ne possano intaccare la qualità complessiva.
- **Testability**: Grado di efficacia ed efficienza con cui è possibile stabilire ed eseguire dei test per valutare la qualità del sistema

### 3.6.2 Metriche Adottate

#### 3.6.2.1 Average Time Spent per Feature Development

Tempo medio speso dall'intero team di sviluppo per lo sviluppo di nuove funzionalità'.

$$ATF = \frac{\sum_{i=1}^N TFi}{NF}$$

Dove

- **ATF**: Average time spent per feature development
- **TFi**: Tempo uomo speso dall'intero team di sviluppo per lo sviluppo dell'i-esima funzionalità'.
- **NU**: Numero di funzionalità sviluppate

**Range Ottimale**: 3h-4h 30m

**Range Accettabile**: 4h 30m-5h 30m

#### 3.6.2.2 Average time spent per error correction

Tempo medio speso dall'intero team di sviluppo per la risoluzione di una singola funzionalità.

$$ATE = \frac{\sum_{i=1}^N TEi}{NE}$$

Dove

- **ATE**: Average time spent per error correction

- **TE<sub>i</sub>**:Tempo uomo speso per risolvere l'i-esimo errore rilevato
- **NE**:Numero complessivo di errori rilevati

**Range Ottimale:**45m-1h 30m

**Range Accettabile:**1h 30m-2h 30m

### 3.6.2.3 Average time spent per unit testing

Tempo medio speso dall'intero team di sviluppo per la scrittura di test corretti e completi per una singola unita'.

$$ATU = \frac{\sum_{i=1}^N TU_i}{NU}$$

Dove

- **ATU**:Average time spent per unit testing
- **TU<sub>i</sub>**:Tempo uomo speso dall'intero team di sviluppo per la scrittura di test corretti e completi per l'i-esima unita'.
- **NU**:Numero unita' testate

**Range Ottimale:**1h 30m-2h

**Range Accettabile:**2h-2h 30m

## 4 Specifica dei test



## 5 Metriche dei test

### 5.1 Metriche per i processi

Per la valutazione della qualità dei processi il team 7DOS ha deciso di basarsi sullo standard ISO \IEC 15504 anche conosciuto come SPICE (Software Process improvement and Capability Determination).

Lo standard definisce per ogni processo una scala di capability a cinque livelli, più un livello base, chiamato livello 0.

- **Optimizing process(Level 5):** Il processo persegue i principi del miglioramento continuo.
- **Predictable process(Level 4):** Sono stati definiti dei limiti entro cui il processo può operare.
- **Established process(Level 3):** Il processo è basato su standard ben definiti.
- **Managed process(Level 2):** L'esecuzione del processo è pianificata, monitorata ed eventualmente corretta se necessario.
- **Performed process(Level 1):** Il processo è stato implementato e raggiunge gli obiettivi prefissati.
- **Incomplete process(Level 0):** Il processo non è stato ancora implementato o non raggiunge gli obiettivi prefissati.

La capability di un processo è misurata tramite nove attributi di processo definiti a livello internazionale:

- **Livello 0:**
- **Livello 1:**
  - **Process performance:** Il processo è in grado di produrre in output un prodotto identificabile
- **Livello 2:**
  - **Performance management:** Il processo è in grado di produrre un prodotto coerente con gli obiettivi fissati.
  - **Work product management:** Il prodotto del processo possiede dei requisiti ben definiti ed una documentazione esaustiva. Inoltre vengono continuamente effettuate verifiche sul prodotto ed eventuali correzioni se necessario.
- **Livello 3:**
  - **Process definition:** L'esecuzione del processo è basata su standard ben definiti.
  - **Process deployment:** Sono stati definiti ed assegnati dei ruoli a ciascun membro del team, ogni risorsa necessaria per l'esecuzione del processo è disponibile ed utilizzabile.

- **Livello 4:**

- **Process measurement:** Vengono utilizzate delle metriche per garantire che il prodotto del processo rispetti gli standard qualitativi aziendali.
- **Process control:** Il processo è facilmente gestibile, produce risultati facilmente prevedibili, ed opera all'interno di limiti ben definiti.

- **Livello 5:**

- **Process innovation:** Vengono definite delle proposte di miglioramento del processo sulla base dei dati raccolti.
- **Process Optimization:** L'impatto di tutti i miglioramenti proposti viene attentamente analizzato in modo tale da assicurarsi che essi apportino degli effettivi benefici al processo.

Ciascuno degli attributi elencati qui sopra è valutato secondo una scala a quattro valori N-P-L-F:

- **Not achieved**(0% - 15%)
- **Partially achieved**(>15% - 50%)
- **Largely achieved**(>50% - 85%)
- **Fully achieved**(85% - 100%)

## 5.2 Metriche per i prodotti

### 5.2.1 Metriche relative al codice software

#### 5.2.1.1 Percentuale di requisiti soddisfatti

Fondamentale per comprendere lo stato di avanzamento del progetto.  
Calcolata attraverso la seguente formula:

$$\left(\frac{n^{\circ} \text{ requisiti soddisfatti}}{n^{\circ} \text{ totale di requisiti}}\right) * 100$$

**Range Ottimale:**100%-100%

**Range Accettabile:**100%-100%

#### 5.2.1.2 Percentuale di test case passati

Fondamentale per determinare l'efficienza del prodotto software.  
Calcolata tramite la seguente formula:

$$\left(\frac{n^{\circ} \text{ test superati}}{n^{\circ} \text{ test eseguiti}}\right) * 100$$

**Range Ottimale:**100%-100%

**Range Accettabile:**100%-100%

### 5.2.1.3 Average Response Time

Tempo medio richiesto per completare l'esecuzione di una funzionalità richiesta ed esporne il risultato.

Calcolato tramite la seguente formula:

$$\frac{\sum_{i=1}^N TFE_i}{NFE}$$

Dove

- **TFE<sub>i</sub>**:Tempo trascorso dal momento in cui viene richiesta l'esecuzione dell'i-esima funzionalità fino al momento in cui la funzionalità viene effettivamente completata ed il risultato viene esposto.
- **NFE**:Numero di funzionalità eseguite durante una singola esecuzione del prodotto software.

**Range Ottimale:**0.3s-0.5s

**Range Accettabile:**0.5s-1.0s

### 5.2.1.4 Average time spent per feature development

Tempo medio speso dall'intero team di sviluppo per lo sviluppo di nuove funzionalità.

Calcolato attraverso la seguente formula:

$$\frac{\sum_{i=1}^N TFi}{NF}$$

Dove

- **TF<sub>i</sub>**:Tempo uomo speso dall'intero team di sviluppo per lo sviluppo dell'iesima funzionalità'.
- **NF**:Numero di funzionalità sviluppate

**Range Ottimale:**3h-4h 30m

**Range Accettabile:**4h 30m-5h 30m

### 5.2.1.5 Average time spent per error correction

Tempo medio speso dall'intero team di sviluppo per la risoluzione di un singolo errore.

Calcolato tramite la seguente formula

$$A \frac{\sum_{i=1}^N TE_i}{NE}$$

Dove

- **TE<sub>i</sub>**:Tempo uomo speso per la risoluzione dell'i-esimo errore rilevato
- **NE**:Numero complessivo di errori rilevati

**Range Ottimale:**45m-1h 30m

**Range Accettabile:**1h 30m-2h 30m

#### 5.2.1.6 Average time spent per unit testing

Tempo medio speso dall'intero team di sviluppo per la scrittura di test corretti e completi per una singola unita'. Calcolato attraverso la seguente formula

$$\frac{\sum_{i=1}^N TU_i}{NU}$$

Dove

- **TU<sub>i</sub>**:Tempo uomo speso dall'intero team di sviluppo per la scrittura di test corretti e completi per l'i-esima unita'.
- **NU**:Numero unita' testate.

**Range Ottimale:**1h 30m-2h

**Range Accettabile:**2h-2h 30m

### 5.2.2 Metriche relative alla documentazione

#### 5.2.2.1 Numero di errori grammaticali

Tutti i documenti verranno analizzati da un apposito strumento di analisi grammaticale. Per poter essere accettati non potranno avere un numero di errori grammaticali superiore a zero.

#### 5.2.2.2 Gunning fog index

Utilizzato per misurare la facilita' di lettura e di comprensione di un testo. Il numero risultante è un indicatore del numero di anni di educazione formale della quale una persona necessita al fine di leggere il testo con facilità.

L'indice di Gunning fog è calcolabile tramite la seguente formula:

$$0.4 * ((\frac{n^{\circ} \text{ parole}}{n^{\circ} \text{ frasi}}) + 100 * (\frac{n^{\circ} \text{ parole complesse}}{n^{\circ} \text{ parole}}))$$

**Range Ottimale:**<=12

**Range Accettabile:**>12-15

### 5.2.2.3 Indice di Gulpease

Utilizzato per misurare la leggibilità di un testo in lingua italiana.

L'indice di Gulpease è calcolabile tramite la seguente formula:

$$89 + \frac{(\text{numero delle frasi}) - 10 * (\text{numero delle lettere})}{\text{numero delle parole}}$$

I risultati sono compresi tra 0 e 100 dove il valore 0 indica la leggibilità più bassa e 100 indica la leggibilità più alta. In generale risulta che testi con indice:

- **Inferiore a 80:** Sono difficili da leggere per chi ha la licenza elementare
- **Inferiore a 60:** Sono difficili da leggere per chi ha la licenza media
- **Inferiore a 40:** Sono difficili da leggere per chi ha la licenza superiore

**Range Ottimale:**60-100

**Range Accettabile:**40-100

### 5.2.2.4 Indice di Flesh

Utilizzato per misurare la leggibilità di un testo in lingua inglese.

L'indice di Flesh è calcolabile tramite la seguente formula:

$$206,835 - (0,846 * S) - (1,015 * P)$$

Dove

- **S:**Numero di sillabe calcolato su un campione di 100 parole.
- **P:**Numero medio di parole per frase.

La leggibilità è considerata come:

- **Alta:**se l'indice e' superiore a 60.
- **Media:**se l'indice e' compresa tra 50 e 60.
- **Bassa:**se l'indice e' inferiore a 50.

**Range Ottimale:**60-100

**Range Accettabile:**50-60

---

## 6 Resoconto delle attività di verifica