



# Piano di Qualifica

*Gruppo MILCTdev — Progetto OpenAPM*  
milctdev.team@gmail.com

<b>Versione</b>	2.0.0
<b>Redazione</b>	Tommaso Carraro
<b>Verifica</b>	Mattia Bano
<b>Approvazione</b>	Luca Dal Medico
<b>Uso</b>	Esterno
<b>Distribuzione</b>	Kirey Group Prof. Tullio Vardanega Prof. Riccardo Cardin Gruppo MILCTdev

## Descrizione

Questo documento si prefigge di regolamentare le operazioni di verifica del gruppo MILCTdev necessarie ad assicurare i requisiti qualitativi per il *progetto<sub>G</sub>* OpenAPM.

## Registro delle modifiche

Versione	Ruolo	Nominativo	Descrizione	Data
2.0.0	Responsabile	Luca Dal Medico	Approvazione del documento per il rilascio	2018-03-08
1.1.0	Verificatore	Mattia Bano	Verifica documento	2018-03-07
1.0.4	Analista	Tommaso Carraro	Stesura appendice C: Valutazione per il miglioramento	2018-02-23
1.0.3	Analista	Tommaso Carraro	Incremento sezione 2: Visione generale della strategia di gestione della qualità	2018-02-22
1.0.2	Analista	Tommaso Carraro	Incremento sezione 3: La strategia di gestione della qualità nel dettaglio	2018-02-21
1.0.1	Analista	Tommaso Carraro	Modifica alle date del documento	2018-02-21
1.0.0	Responsabile	Tommaso Carraro	Approvazione del documento per il rilascio	2018-01-03
0.3.0	Verificatore	Dragos Cristian Lizan	Verifica documento	2017-12-28
0.2.1	Analista	Mattia Bano	Stesura appendice D: Standard di qualità	2017-12-17
0.2.0	Verificatore	Mattia Bano	Verifica documento	2017-12-14
0.1.1	Verificatore	Carlo Munarini	Stesura appendice A: Resoconto delle attività di verifica	2017-12-12
0.1.0	Verificatore	Mattia Bano	Verifica documento	2017-12-11
0.0.3	Analista	Leonardo Nodari	Stesura sezione 3: Metriche di qualità	2017-12-04
0.0.2	Analista	Leonardo Nodari	Stesura sezione 2: Strategie di gestione qualità	2017-12-01
0.0.1	Analista	Isacco Maculan	Stesura sezione 1: Introduzione	2017-11-29
0.0.0	Analista	Isacco Maculan	Inserimento template documento	2017-11-29

## Indice

<b>1</b>	<b>Introduzione</b>	<b>7</b>
1.1	Scopo del documento . . . . .	7
1.2	Scopo del prodotto . . . . .	7
1.3	Glossario . . . . .	7
1.4	Riferimenti . . . . .	7
1.4.1	Riferimenti normativi . . . . .	7
1.4.2	Riferimenti informativi . . . . .	8
<b>2</b>	<b>Visione generale della strategia di gestione della qualità</b>	<b>9</b>
2.1	Obiettivi di qualità . . . . .	9
2.1.1	Qualità di processo . . . . .	9
2.1.2	Qualità di prodotto . . . . .	9
2.1.3	Tabella degli obiettivi . . . . .	10
2.2	Metriche e misure . . . . .	10
2.2.1	Tabella delle metriche . . . . .	11
2.3	Scadenze temporali . . . . .	12
<b>3</b>	<b>La strategia di gestione della qualità nel dettaglio</b>	<b>13</b>
3.1	Risorse . . . . .	13
3.2	Misure e metriche in dettaglio . . . . .	13
3.2.1	Misure e metriche per i processi . . . . .	13
3.2.1.1	Schedule Variance . . . . .	13
3.2.1.2	Cost Variance . . . . .	13
3.2.1.3	SPICE . . . . .	14
3.2.2	Misure e metriche per i prodotti . . . . .	14
3.2.2.1	Misure e metriche per i documenti . . . . .	14
3.2.2.1.1	Indice Gulpease . . . . .	14
3.2.2.2	Misure e metriche per il software . . . . .	14
3.2.2.2.1	Structural Fan-In (SFIN) . . . . .	14
3.2.2.2.2	Structural Fan-Out (SFOUT) . . . . .	15
3.2.2.2.3	Logical Source Lines of Code . . . . .	15
3.2.2.2.4	Code Coverage . . . . .	15
3.2.2.2.5	Test Automation Proportion . . . . .	15
3.2.2.2.6	Rapporto linee di commento per linee di codice . . . . .	15
3.2.2.2.7	Complessità ciclomatica . . . . .	16
3.2.2.2.8	Failure Avoidance . . . . .	16
3.2.2.2.9	Percentuale superamento test . . . . .	16
3.2.2.2.10	Requisiti obbligatori soddisfatti . . . . .	16
<b>A</b>	<b>Resoconto delle attività di verifica</b>	<b>17</b>
A.1	Verifica dei processi . . . . .	17

A.1.1	Cost Variance . . . . .	17
A.1.2	SPICE . . . . .	18
A.1.3	Schedule Variance . . . . .	19
A.2	Verifica dei prodotti . . . . .	20
A.2.1	Indici Gulpease . . . . .	20
<b>B</b>	<b>Pianificazione test</b>	<b>21</b>
B.1	Test di sistema . . . . .	21
B.1.1	Test di sistema previsti . . . . .	21
B.1.2	Tracciamento test di sistema-requisiti . . . . .	25
<b>C</b>	<b>Valutazione per il miglioramento</b>	<b>27</b>
C.1	Valutazione sui ruoli . . . . .	27
C.2	Valutazione sull'organizzazione . . . . .	27
C.3	Valutazione sugli strumenti . . . . .	28
<b>D</b>	<b>Standard di qualità</b>	<b>29</b>
D.1	ISO/IEC 15504 . . . . .	29
D.2	PDCA . . . . .	31
D.3	ISO/IEC 9126 . . . . .	32

## Tabelle

2	Tabella degli obiettivi . . . . .	10
4	Tabella delle metriche . . . . .	11
5	Schedule Variance - Analisi, analisi in dettaglio . . . . .	19
6	Schedule Variance - Progettazione architetturale . . . . .	19
7	Test di sistema . . . . .	24
8	Tracciamento test di sistema - Requisiti . . . . .	26
9	Valutazione sui ruoli . . . . .	27
10	Valutazione sull'organizzazione . . . . .	27
11	Valutazione sugli strumenti . . . . .	28

## Immagini

1	Variazione della metrica Cost Variance . . . . .	17
2	Variazione dei valori SPICE . . . . .	18
3	Variazione degli indici Gulpease nei documenti . . . . .	20
4	Test di sistema - Stato attuale . . . . .	24
5	Schema della capability dimension di SPICE . . . . .	30
6	Schema del miglioramento continuo tramite PDCA . . . . .	31
7	Schema del ciclo di qualità del software . . . . .	32
8	Schema delle caratteristiche definite in ISO/IEC 9126 . . . . .	34

# 1 Introduzione

## 1.1 Scopo del documento

Lo scopo del documento è fissare, in modo quantitativo, gli obiettivi di qualità, di processo e di *prodotto<sub>G</sub>*, e di illustrare le strategie di verifica e validazione adottate dal gruppo MILCTdev per raggiungerli. A tal fine è necessaria una verifica continua sulle attività svolte, in modo da individuare e correggere eventuali anomalie, evitando così uno spreco di risorse.

Il seguente documento non è da considerarsi completo, contiene infatti le strategie per la realizzazione di un progetto di qualità, relative al periodo di realizzazione corrente. Questo è dovuto alla natura incrementale del progetto che porta, a ogni periodo, all'aggiornamento delle parti che compongono il documento, quali ad esempio:

- la specifica dei test;
- gli esiti delle verifiche.

## 1.2 Scopo del prodotto

Lo scopo del *prodotto<sub>G</sub>* è realizzare un set di funzioni basate su *Elasticsearch<sub>G</sub>* e *Kibana<sub>G</sub>* per interpretare i dati raccolti da un *Agent<sub>G</sub>*. I dati interpretati forniranno a *DevOps<sub>G</sub>* statistiche e informazioni utili per comprendere il funzionamento della propria applicazione. In particolare si richiede lo sviluppo di un motore di generazione di *metriche<sub>G</sub>* da *trace<sub>G</sub>*, un motore di generazione di *baseline<sub>G</sub>* basato sulle metriche del punto precedente, e un motore di gestione di *critical event<sub>G</sub>*.

## 1.3 Glossario

All'interno del documento sono presenti termini che possono assumere significati diversi a seconda del contesto. Per evitare ambiguità, i significati dei termini complessi adottati nella stesura della documentazione sono contenuti nel documento *Glossario v2.0.0*. Per segnalare un termine del testo presente all'interno del Glossario verrà aggiunta una *G* a pedice e il testo sarà in corsivo.

## 1.4 Riferimenti

### 1.4.1 Riferimenti normativi

- **Norme di progetto:** *Norme di Progetto v2.0.0*;

- **Capitolato d'appalto C7:**  
<http://www.math.unipd.it/~tullio/IS-1/2017/Progetto/C7.pdf>  
(ultima consultazione effettuata in data 2018-03-07).

#### 1.4.2 Riferimenti informativi

- **Piano di progetto:** *Piano di Progetto v2.0.0*;
- **Qualità di prodotto - Slide del corso Ingegneria del Software:**  
<http://www.math.unipd.it/~tullio/IS-1/2017/Dispense/L13.pdf>  
(ultima consultazione effettuata in data 2018-03-07);
- **Qualità di processo - Slide del corso Ingegneria del Software:**  
<http://www.math.unipd.it/~tullio/IS-1/2017/Dispense/L15.pdf>  
(ultima consultazione effettuata in data 2018-03-07);
- **Sommerville Ian, Software Engineering, 10<sup>th</sup> ed., Pearson (2015)**  
- §24 Quality management
- **Sommerville Ian, Software Engineering, 9<sup>th</sup> ed., Pearson (2010)**  
- §26 Process improvement
- **Standard ISO/IEC 15504:**  
[https://en.wikipedia.org/wiki/ISO/IEC\\_15504](https://en.wikipedia.org/wiki/ISO/IEC_15504)  
(ultima consultazione effettuata in data 2018-03-07);
- **PDCA:**  
[https://it.wikipedia.org/wiki/Ciclo\\_di\\_Deming](https://it.wikipedia.org/wiki/Ciclo_di_Deming)  
(ultima consultazione effettuata in data 2018-03-07);
- **Standard ISO/IEC 9126:**  
[https://it.wikipedia.org/wiki/ISO/IEC\\_9126](https://it.wikipedia.org/wiki/ISO/IEC_9126)  
(ultima consultazione effettuata in data 2018-03-07);
- **Indice di Gulpease:**  
[http://it.wikipedia.org/wiki/Indice\\_Gulpease](http://it.wikipedia.org/wiki/Indice_Gulpease)  
(ultima consultazione effettuata in data 2018-03-07);
- **Logical SLOC:**  
[https://en.wikiversity.org/wiki/Software\\_metrics\\_and\\_measurement](https://en.wikiversity.org/wiki/Software_metrics_and_measurement)  
(ultima consultazione effettuata in data 2018-03-07).



## 2 Visione generale della strategia di gestione della qualità

### 2.1 Obiettivi di qualità

In questa sezione vengono illustrati gli obiettivi che MILCTdev intende raggiungere per assicurare la qualità di processo e di prodotto per quanto riguarda la realizzazione di OpenAPM. Inoltre, per ognuno di questi obiettivi, vengono fissate metriche per rendere quantificabile il raggiungimento della qualità di processo e di prodotto; queste sono descritte nella sezione 2.2.

#### 2.1.1 Qualità di processo

Per realizzare un prodotto valido, MILCTdev ha deciso di adottare lo standard ISO/IEC 15504 per valutare la qualità di ogni processo necessario allo sviluppo di OpenAPM. Viene inoltre utilizzato il ciclo di Deming per assicurare un miglioramento continuo dei processi, senza eventuali regressioni. Nell'appendice D vengono approfonditi questo metodo e lo standard utilizzato.

Gli obiettivi fissati per i processi sono:

- rispettare tempi e costi descritti nel *Piano di Progetto v2.0.0*;
- avere prestazioni sempre misurabili;
- perseguire un miglioramento continuo delle stesse.

#### 2.1.2 Qualità di prodotto

Basandosi sullo standard ISO/IEC 9126, descritto nell'appendice D, sono stati fissati obiettivi che mirano a garantire la qualità del prodotto finale. Questi sono:

- i **documenti** devono:
  - essere leggibili e comprensibili a chiunque;
  - essere corretti dal punto di vista ortografico, sintattico, semantico e logico.
- il **software** deve:
  - soddisfare tutti i requisiti obbligatori descritti in *Analisi dei Requisiti v2.0.0*;
  - superare gran parte dei test illustrati in appendice B;
  - garantire usabilità e manutenibilità;
  - essere affidabile.

### 2.1.3 Tabella degli obiettivi

Viene qui riassunto ogni obiettivo, classificandolo con il suo codice identificativo e indicando le metriche che ne quantificano il raggiungimento. Per una descrizione delle metriche vedere nella sezione 2.2.

ID	Nome	Metrica
OPC1	Coerenza con Piano di Progetto	MPC1:Schedule Variance MPC2:Cost Variance
OPC2	Miglioramento continuo	MPC3:SPICE
OPDD1	Leggibilità documenti	MPDD1:Indice Gulpease
OPDS1	Implementazione requisiti obbligatori	MPDS10:Requisiti obbligatori soddisfatti
OPDS2	Superamento test	MPDS9:Percentuale superamento test
OPDS3	Manutenibilità e usabilità	MPDS1:Structural Fan-In MPDS2:Structural Fan-Out MPDS3:Logical Source Lines of Code MPDS4:Code coverage MPDS5:Test automation proportion MPDS6:Rapporto linee di commento per linee di codice MPDS7:Complessità ciclomatica
OPDS4	Affidabilità	MPDS8:Failure avoidance

Table 2: Tabella degli obiettivi

Ogni obiettivo si riterrà raggiunto solamente al raggiungimento del valore minimo di ogni metrica che concorre alla quantificazione del suo grado di raggiungimento. La spiegazione di valore minimo si trova in *Norme di Progetto v2.0.0*.

## 2.2 Metriche e misure

Ogni processo ed ogni prodotto dovrebbero sempre presentare un set di  $KPI_G$  che permettano il tracciamento, la comunicazione ed il miglioramento della loro qualità.

In questa sezione pertanto, si provvederà alla presentazione delle metriche che permettano di quantificare e valutare la qualità dei processi e dei prodotti di MILCTdev. Le spiegazioni e le modalità di calcolo di ogni metrica sono definite nella sezione 3.2.

Per alcune metriche, relative al periodo di progettazione e sviluppo, non sono ancora stati

definiti i range di risultati precedentemente definiti. Questo perché, ad oggi, MILCTdev non può indicare con precisione quali questi siano.

### 2.2.1 Tabella delle metriche

Nella seguente tabella vengono indicati, oltre a Identificativo, Nome e Obiettivo a cui si riferisce, anche le soglie di accettazione minime e ottimali di ogni metrica.

ID	Nome	Obiettivo	Soglie di accettazione
MPC1	Schedule Variance	OPC1:Coerenza con Piano di Progetto	<i>Valore minimo:</i> $\leq 3$ <i>Valore ottimale:</i> $\leq 0$
MPC2	Cost Variance	OPC1:Coerenza con Piano di Progetto	<i>Valore minimo:</i> $\leq 8\%$ <i>Valore ottimale:</i> $\leq 1\%$
MPC3	SPICE	OPC2:Miglioramento continuo	<i>Valore minimo:</i> Livello 2 <i>Valore ottimale:</i> $\geq$ Livello 4
MPDD1	Indice Gulpease	OPDD1:Leggibilità documenti	<i>Valore minimo:</i> $\geq 40$ <i>Valore ottimale:</i> $\geq 70$
MPDS1	Structural Fan-In	OPDS3:Manutenibilità e usabilità	
MPDS2	Structural Fan-Out	OPDS3:Manutenibilità e usabilità	
MPDS3	Logical Source Lines of Code	OPDS3:Manutenibilità e usabilità	
MPDS4	Code coverage	OPDS3:Manutenibilità e usabilità	
MPDS5	Test automation proportion	OPDS3:Manutenibilità e usabilità	
MPDS6	Rapporto linee di commento per linee di codice	OPDS3:Manutenibilità e usabilità	<i>Valore minimo:</i> $\geq 10\%$ <i>Valore ottimale:</i> $\geq 10\%$
MPDS7	Complessità ciclomatica	OPDS3:Manutenibilità e usabilità	
MPDS8	Failure avoidance	OPDS4:Affidabilità	
MPDS9	Percentuale superamento test	OPDS2:Superamento test	<i>Valore minimo:</i> $\geq 75\%$ <i>Valore ottimale:</i> $\geq 95\%$
MPDS10	Requisiti obbligatori soddisfatti	OPDS1:Implementazione requisiti obbligatori	<i>Valore minimo:</i> 100% <i>Valore ottimale:</i> 100%

Table 4: Tabella delle metriche

### 2.3 Scadenze temporali

Il rispetto delle *milestones<sub>G</sub>* presenti in *Piano di Progetto v2.0.0* indicano che la realizzazione del prodotto sta procedendo come stabilito; tuttavia la presenza di parti incomplete o che non dispongono di un grado di qualità accettabile, porterebbe il gruppo a far slittare le date definite per la consegna del progetto, con conseguente impatto nel preventivo fornito in *Piano di Progetto v2.0.0*. Per prevenire l'insorgenza di tali errori MILCTdev attua procedure di verifica descritte in dettaglio nelle *Norme di Progetto v2.0.0*.

## 3 La strategia di gestione della qualità nel dettaglio

### 3.1 Risorse

Per la realizzazione di un prodotto di qualità, MILCTdev effettua un controllo su ogni parte del progetto sfruttando risorse umane e tecnologiche. Per quanto riguarda le risorse umane, i ruoli di maggiore impatto sono quello di Responsabile di Progetto, che si occupa della qualità dei processi, e del Verificatore che ha il compito di garantire la qualità dei prodotti. Una descrizione più approfondita di questi e degli altri ruoli è presente nelle *Norme di Progetto v2.0.0*.

Per risorse tecnologiche si intende, invece, l'insieme degli strumenti software ed hardware utilizzati del gruppo durante le attività di verifica, anch'essi presentati nelle *Norme di Progetto v2.0.0*.

### 3.2 Misure e metriche in dettaglio

#### 3.2.1 Misure e metriche per i processi

In questa sezione si provvederà alla descrizione delle metriche che permettono di quantificare e valutare la qualità dei processi e dei prodotti di MILCTdev. All'interno della spiegazione di ogni metrica verrà illustrato quando, come e su cosa viene fatta la misurazione durante il processo di verifica.

##### 3.2.1.1 Schedule Variance

La Schedule Variance è un indice di efficienza che ha come oggetto la durata temporale di un processo o di un'attività. Questa metrica aiuta il Responsabile di Progetto nella creazione dei prospetti orari inseriti nei consuntivi di periodo, e di conseguenza aiuta la *team\_G* nell'analisi dell'utilizzo di risorse temporali.

Il calcolo della Schedule Variance avviene in questo modo:

$$SV = data\ conclusione\ reale - data\ conclusione\ preventivata$$

Entrambe le date si riferiscono alla conclusione dell'attività o del processo.

##### 3.2.1.2 Cost Variance

La Cost Variance, o Variazione di Costo, è una metrica che analizza il costo, nonché le risorse legate ad un processo o ad un attività. Essa può essere influenzata anche dalla metrica sopracitata.

La Variazione di Costo viene così calcolata:

$$CV = \text{costo delle risorse effettivo} - \text{costo delle risorse preventivato}$$

### 3.2.1.3 SPICE

Al termine di ogni periodo, il team MILCTdev provvederà alla valutazione della qualità dei processi tramite lo standard ISO/IEC 15504 conosciuto come SPICE. Lo standard SPICE, ed i livelli di maturità, vengono illustrati in maniera completa ed approfondita nell'Appendice D.

## 3.2.2 Misure e metriche per i prodotti

### 3.2.2.1 Misure e metriche per i documenti

#### 3.2.2.1.1 Indice Gulpease

Per analizzare la leggibilità della documentazione prodotta, il team MILCTdev ha deciso di avvalersi dell'*indice Gulpease<sub>G</sub>*. Questo è stato creato per venire incontro alla complessità della lingua italiana, non contemplata in altri indici, come l'*indice di Flesch<sub>G</sub>*.

L'indice Gulpease viene calcolato tramite questa formula:

$$IG = 89 + \frac{(300 \times \text{numero delle frasi}) - (10 \times \text{numero delle lettere})}{\text{numero delle parole}}$$

Il valore ottenuto indicherà la leggibilità del testo e può variare da 0, indice di bassissima leggibilità, a 100, indice di ottima leggibilità.

### 3.2.2.2 Misure e metriche per il software

Al fine di poter correttamente quantificare e valutare la qualità del prodotto software, il team MILCTdev ha deciso utilizzare diverse metriche. Gran parte delle metriche indicate in questa sezione verranno riviste ed aggiornate nel corso dei successivi periodi.

#### 3.2.2.2.1 Structutal Fan-In (SFIN)

Questa metrica, detta anche grado di accoppiamento afferente, permetterà di avere una visione del numero di moduli che usufruiscono della componente oggetto di analisi.

Il valore di questo indice è semplicemente dato dal conteggio delle componenti indicate poco sopra. Un valore molto basso può indicare una scarsa utilità del modulo analizzato, all'opposto un grado troppo alto potrebbe indicare un pericoloso livello di dipendenza.

#### 3.2.2.2.2 Structutal Fan-Out (SFOUT)

Il grado di accoppiamento efferente, così come l'accoppiamento afferente, ha come oggetto di analisi il numero di moduli che sono legati alla componente in analisi.

Questa volta si prende in considerazione il numero di moduli esterni che vengono utilizzati. Un indice ottimale dovrebbe avere un valore di 0 o 1, questo perché minore è il suo valore, più il modulo è indipendente ai cambiamenti del resto del sistema. Un valore eccessivamente alto è indice di troppa dipendenza rispetto al resto del sistema.

#### 3.2.2.2.3 Logical Source Lines of Code

Questa metrica dà un'idea della grandezza del prodotto software contando il numero di linee di codice. Il team ha scelto di utilizzare la variante definita Logical SLOC, andando quindi a contare solamente il numero di *statements<sub>G</sub>* all'interno del codice.

#### 3.2.2.2.4 Code Coverage

Il code coverage è una metrica che, sfruttando la Logical SLOC, indica la percentuale di *statements* coperti dai test.

Il valore della code coverage è così calcolato:

$$CC = \frac{\text{Logical SLOC}}{\text{Numero di statement coperti da test}} \times 100$$

#### 3.2.2.2.5 Test Automation Proportion

Questa metrica dà un'idea della percentuale di test automatici implementati dal team MILCTdev. La volontà è quella di aumentare sempre più il valore di questa metrica.

Il valore viene così calcolato:

$$TAP = \frac{\text{Numero di test automatici}}{\text{Numero di test manuali}} \times 100$$

#### 3.2.2.2.6 Rapporto linee di commento per linee di codice

Un indice di buona manutenibilità del codice potrebbe essere il rapporto tra *Physical SLOC<sub>G</sub>* e numero di linee di commento all'interno dello stesso.

Il valore viene espresso in percentuale e viene così calcolato:

$$RLCLC = \frac{\text{Numero di linee di codice totali}}{\text{Numero di linee di commento}} \times 100$$

### 3.2.2.2.7 Complessità ciclomatica

L'indice di complessità di un programma aiuta ad identificare il numero di test necessari al raggiungimento di un coverage completo. Questa metrica software può essere applicata anche a *packages<sub>G</sub>*, moduli, metodi o classi.

Il calcolo avviene sfruttando il *grafo di controllo di flusso<sub>G</sub>* e l'indice non è altro che il numero di cammini indipendenti attraverso il codice sorgente. La formula è quindi la seguente:

$$v(G) = e - n + 2p$$

Dove:

- **n**: è il numero di nodi del grafo, nonché il numero di tutti i gruppi indivisibili di istruzioni;
- **e**: rappresenta il numero di archi del grafo, cioè il numero di collegamenti tra due nodi tali che, il nodo seguente possa essere eseguito immediatamente dopo il nodo preso di riferimento;
- **p**: è il numero di componenti connesse.

### 3.2.2.2.8 Failure Avoidance

Indica la robustezza di un prodotto nel far fronte a possibili imprevisti o errori e viene così calcolata:

$$FA = \frac{\text{Numero situazioni anomale evitate}}{\text{Numero totale situazioni anomale occorse}}$$

### 3.2.2.2.9 Percentuale superamento test

Questa metrica indica quanti dei test implementati hanno esito positivo e può essere ottenuta così:

$$PST = \frac{\text{Numero test superati}}{\text{Numero test implementati}} \times 100$$

### 3.2.2.2.10 Requisiti obbligatori soddisfatti

Questa metrica aiuta il team a capire in che quantità sono stati soddisfatti i requisiti obbligatori indicati in *Analisi dei Requisiti v2.0.0*.

Il valore viene espresso in percentuale e viene calcolato come segue:

$$ROS = \frac{\text{Num. Requisiti obbligatori individuati}}{\text{Num. Requisiti obbligatori soddisfatti}} \times 100$$

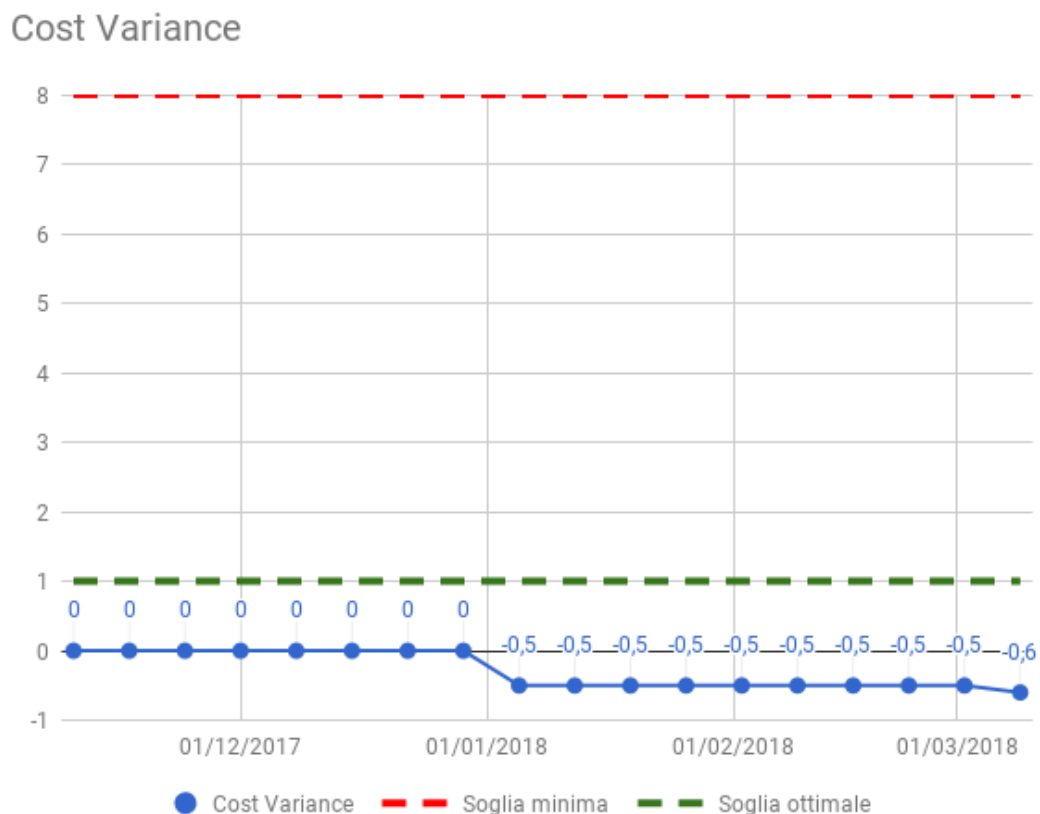


## A Resoconto delle attività di verifica

Questa sezione illustra i risultati di verifica ottenuti utilizzando le metriche descritte nella sezione 2.2 nel corso dello sviluppo del progetto (per la spiegazione dei diversi periodi vedere in *Piano di Progetto v2.0.0*). Le misurazioni sono state fatte a distanza di sette giorni l'una dall'altra e vengono presentate con un diagramma, che ha la funzione di fare da cruscotto, per evidenziare le variazioni nel tempo. È stato scelto il diagramma a cruscotto perché più parlante rispetto alla classica rappresentazione tabellare per gli esiti delle verifiche.

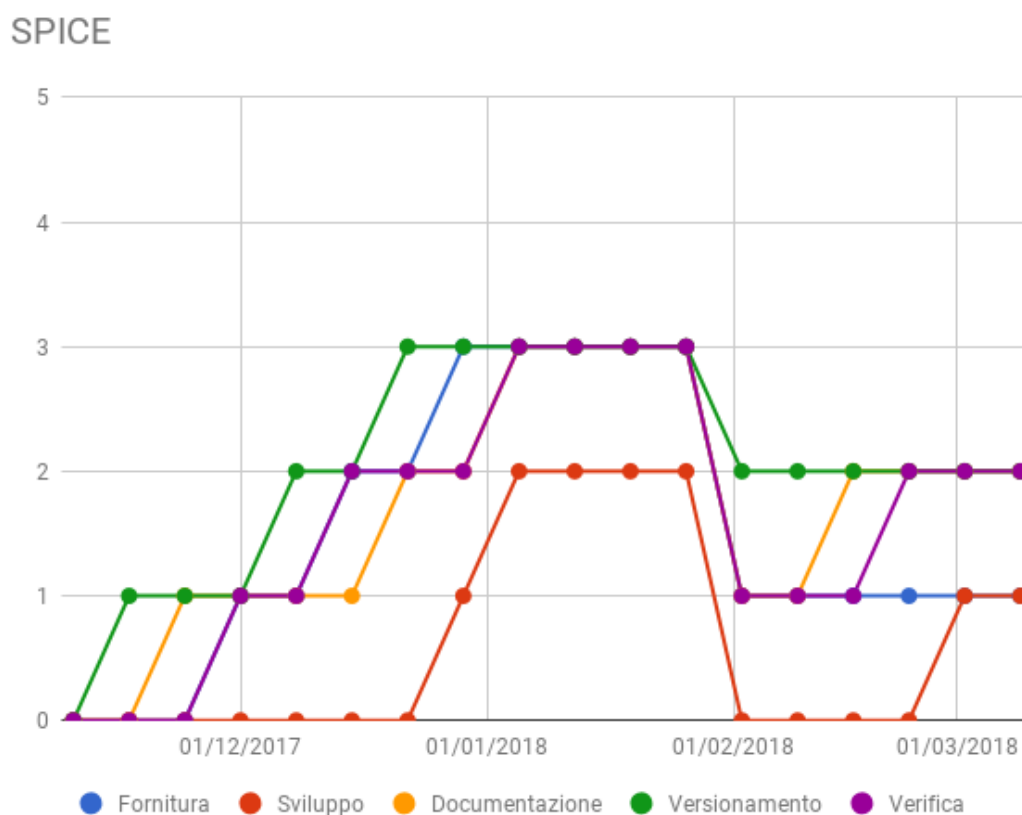
### A.1 Verifica dei processi

#### A.1.1 Cost Variance



Questa metrica è strettamente dipendente con la Schedule Variance infatti, se un'attività termina prima del tempo previsto, la Cost Variance diminuisce perché il monte ore preventivato per quell'attività è superiore alle ore effettive. Quindi, nel grafico precedente, una diminuzione della Cost Variance corrisponde ad attività terminata in anticipo mentre un aumento corrisponde ad un ritardo nei tempi previsti.

### A.1.2 SPICE



Il calo a seguito alla Revisione dei Requisiti(2018-01-26) è causato da una rivalutazione del livello raggiunto e non da una effettiva perdita di maturità dei processi.

Figure 2: Variazione dei valori SPICE

### A.1.3 Schedule Variance

Questi risultati hanno una rappresentazione tabellare perché, per come é definita la Schedule Variance (vedere nella sezione 3.2), i valori sono relativi alle diverse attività presenti nei diversi periodi, descritti nel *Piano di Progetto v2.0.0*, e vengono quindi calcolati solamente a fine del periodo e non durante questo.

Attività	Schedule Variance
Analisi dei Requisiti	0
Glossario	0
Norme di progetto	0
Piano di progetto	0
Piano di qualifica	-2
Studio di fattibilità	0
<b>Totale</b>	<b>-2</b>

Table 5: Schedule Variance nel periodo di analisi e analisi in dettaglio

Attività	Schedule Variance
Incremento documenti precedenti	+1
Technology baseline	-2
<b>Totale</b>	<b>-1</b>

Table 6: Schedule Variance nel periodo di Progettazione architettuale

## A.2 Verifica dei prodotti

### A.2.1 Indici Gulpease

Indice Gulpease per documento

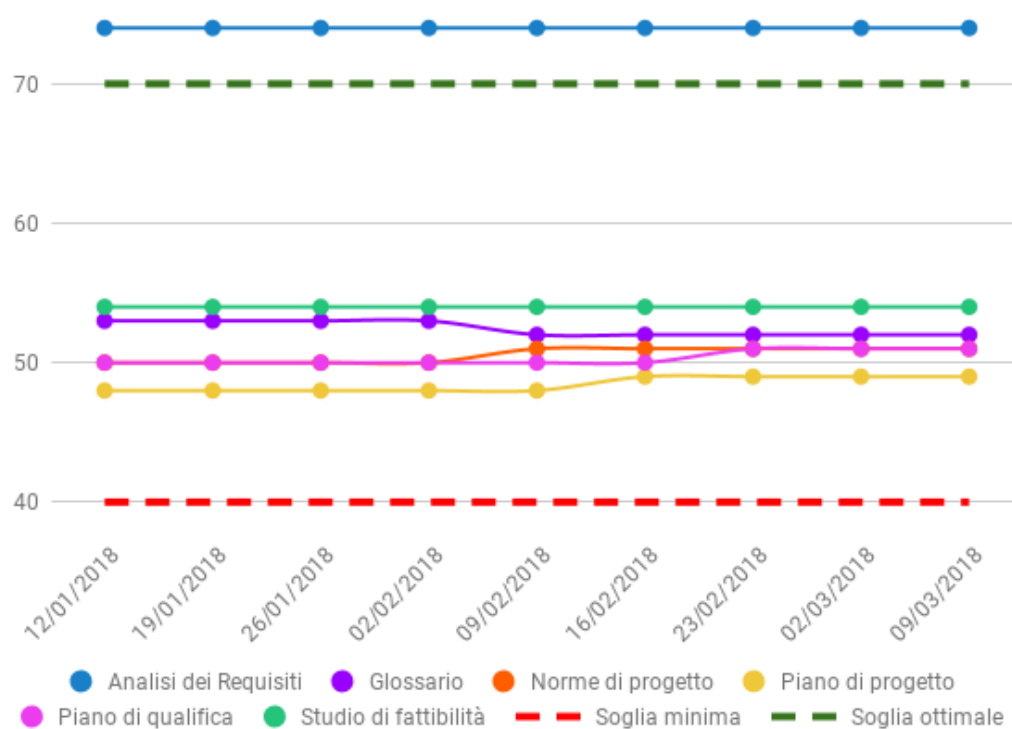


Figure 3: Variazione degli indici Gulpease nei documenti

## B Pianificazione test

### B.1 Test di sistema

Vengono qui presentati i test di sistema necessari a garantire che il prodotto soddisfi i requisiti presenti in *Analisi dei Requisiti v2.0.0*.

Quelli presentati sono relativi ai requisiti che MILCTdev ritiene debbano avere test. La composizione del codice identificativo é presente nelle *Norme di Progetto v2.0.0*.

#### B.1.1 Test di sistema previsti

Id Test	Descrizione	Stato
TSFO1.1	Verifica che la procedura batch sia in grado di leggere trace da un indice ElasticSearch contenente le trace	<i>Non implementato</i>
TSFO1.2	Verifica che la procedura batch sia in grado di filtrare le trace prima di fare dei raggruppamenti su di esse	<i>Non implementato</i>
TSFO1.2.1	Verifica che la procedura batch sia in grado di leggere la modalità di filtraggio delle trace da un indice ElasticSearch	<i>Non implementato</i>
TSFO1.2.2	Verifica che la procedura batch sia in grado di filtrare le trace in base alla configurazione di filtraggio scelta	<i>Non implementato</i>
TSFO1.3	Verifica che la procedura batch sia in grado di raggruppare delle trace in base a dei parametri configurabili	<i>Non implementato</i>
TSFO1.3.1	Verifica che la procedura batch sia in grado di leggere la modalità di raggruppamento delle trace da un indice ElasticSearch	<i>Non implementato</i>
TSFO1.3.2	Verifica che la procedura batch sia in grado di scegliere un valore per configurare la modalità di raggruppamento scelta, prelevandolo da ElasticSearch	<i>Non implementato</i>
TSFO1.3.3	Verifica che la procedura batch sia in grado di raggruppare le trace in base alla modalità e al parametro di raggruppamento scelti	<i>Non implementato</i>
TSFO1.4	Verifica che la procedura batch sia in grado di calcolare una metrica	<i>Non implementato</i>

<b>Id Test</b>	<b>Descrizione</b>	<b>Stato</b>
TSFO1.4.1	Verifica che la procedura batch sia in grado di leggere la tipologia di metrica da calcolare da ElasticSearch	<i>Non implementato</i>
TSFO1.4.2	Verifica che la procedura batch sia in grado di scegliere la granularità di tempo per il calcolo della metrica	<i>Non implementato</i>
TSFO1.4.3	Verifica che la procedura batch sia in grado di calcolare metriche basandosi sullo storico delle metriche	<i>Non implementato</i>
TSFO1.5	Verifica che la procedura batch sia in grado di salvare la metrica calcolata su un indice ElasticSearch	<i>Non implementato</i>
TSFO2	Verifica che l'inserimento di una metrica scateni la generazione di una baseline su tale metrica, da parte della procedura batch	<i>Non implementato</i>
TSFO2.1	Verifica che l'inserimento di una metrica scateni l'aggiornamento della baseline associata a tale tipo di metrica, nel caso in cui la baseline esista già	<i>Non implementato</i>
TSFO2.2	Verifica che la procedura batch sia in grado di scegliere una configurazione temporale, prelevata da un indice ElasticSearch, per il calcolo di una baseline	<i>Non implementato</i>
TSFO2.3	Verifica che la procedura batch sia in grado di leggere le metriche coinvolte dal calcolo della baseline da un indice ElasticSearch	<i>Non implementato</i>
TSFO2.4	Verifica che la procedura batch sia in grado di calcolare una baseline	<i>Non implementato</i>
TSFO2.5	Verifica che la procedura batch sia in grado di salvare la baseline calcolata in un indice ElasticSearch	<i>Non implementato</i>
TSFO3	Verifica che l'inserimento di una nuova metrica scateni un controllo di critical event da parte della procedura batch	<i>Non implementato</i>
TSFO3.1	Verifica che la procedura batch sia in grado di configurare una policy leggendo dati da ElasticSearch	<i>Non implementato</i>
TSFO3.1.1	Verifica che la procedura batch sia in grado di leggere una tipologia di soglia per la policy da un indice ElasticSearch	<i>Non implementato</i>

Id Test	Descrizione	Stato
TSFO3.1.2	Verifica che la procedura batch sia in grado di leggere un valore per la tipologia di soglia scelta, prelevandolo da un indice ElasticSearch	<i>Non implementato</i>
TSFO3.1.3	Verifica che la procedura batch sia in grado di leggere l'azione di rimedio da eseguire, prelevandola da ElasticSearch, nel caso in cui si verifichi un critical event	<i>Non implementato</i>
TSFO3.2	Verifica che la procedura batch sia in grado di verificare la policy configurata	<i>Non implementato</i>
TSFO3.2.1	Verifica che la procedura batch sia in grado di leggere il valore attuale della metrica inserita	<i>Non implementato</i>
TSFO3.2.2	Verifica che la procedura batch controlli che il valore della metrica sia in linea con la soglia selezionata	<i>Non implementato</i>
TSFO3.3	Verifica che la procedura batch lanci un critical event nel caso in cui la soglia viene superata	<i>Non implementato</i>
TSFO3.4	Verifica che la procedura batch, una volta lanciato il critical event, possa eseguire un'azione di rimedio	<i>Non implementato</i>
TSFO3.4.1	Verifica che la procedura batch possa inviare una e-mail di notifica del critical event	<i>Non implementato</i>
TSFO3.4.2	Verifica che la procedura batch possa eseguire una procedura automatica	<i>Non implementato</i>
TSFO3.4.3	Verifica che la procedura batch possa salvare il critical event su un indice ElasticSearch	<i>Non implementato</i>
TSFD4	Verifica che allo scattare di uno critical event, la procedura batch sia in grado di inviare una mail di notifica	<i>Non implementato</i>
TSFD4.1	Verifica che la procedura batch sia in grado di prelevare l'indirizzo e-mail del destinatario da un indice ElasticSearch	<i>Non implementato</i>
TSFD4.2	Verifica che il template della mail venga creato con Spring mail	<i>Non implementato</i>
TSFD4.3	Verifica che la procedura batch configuri la mail leggendo la configurazione da un indice ElasticSearch	<i>Non implementato</i>
TSFD4.4	Verifica che la procedura batch sia in grado di collegarsi al server di invio della mail	<i>Non implementato</i>
TSFD4.5	Verifica che la procedura batch sia in grado di inviare la mail al destinatario scelto e con le configurazioni impostate	<i>Non implementato</i>

Id Test	Descrizione	Stato
TSFD5	Verifica che la procedura batch, al verificarsi di un critical event, possa memorizzarlo in un indice ElasticSearch	<i>Non implementato</i>
TSFD5.1	Verifica che la procedura batch sia in grado di prelevare l'indice ElasticSearch dove memorizzare il critical event	<i>Non implementato</i>
TSFD6	Verifica che la procedura batch, al verificarsi di un critical event, possa eseguire una procedura automatica	<i>Non implementato</i>
TSFD6.1	Verifica che la procedura batch sia in grado di prelevare la procedura automatica da eseguire da un indice ElasticSearch	<i>Non implementato</i>
TSFF7	Verifica che l'amministratore di sistema sia in grado di configurare la schedulazione delle procedure batch da eseguire	<i>Non implementato</i>
TSFF7.1	Verifica che l'amministratore di sistema sia in grado di leggere la configurazione della procedura da un indice ElasticSearch	<i>Non implementato</i>
TSFF7.2	Verifica che l'amministratore di sistema sia in grado di configurare la procedura con i parametri prelevati	<i>Non implementato</i>
TSFF7.3	Verifica che l'amministratore di sistema sia in grado di memorizzare su ElasticSearch la nuova configurazione per la procedura	<i>Non implementato</i>

Table 7: Tabella dei test di sistema

Test di sistema

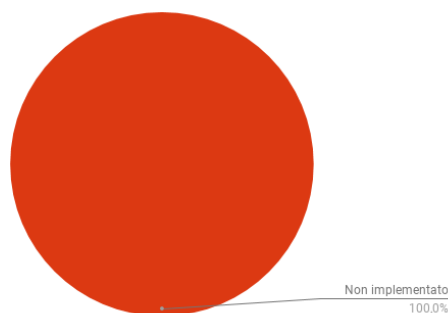


Figure 4: Test di sistema - Stato attuale



### B.1.2 Tracciamento test di sistema-requisiti

Test	Requisito
TSFO1.1	RFO1.1
TSFO1.2	RFO1.2
TSFO1.2.1	RFO1.2.1
TSFO1.2.2	RFO1.2.3
TSFO1.3	RFO1.3
TSFO1.3.1	RFO1.3.1
TSFO1.3.2	RFO1.3.2
TSFO1.3.3	RFO1.3.3
TSFO1.4	RFO1.4
TSFO1.4.1	RFO1.4.1
TSFO1.4.2	RFO1.4.2
TSFO1.4.3	RFO1.4.3
TSFO1.5	RFO1.5
TSFO2	RFO3
TSFO2.1	RFO2.1
TSFO2.2	RFO2.1.1
TSFO2.3	RFO2.1.2
TSFO2.4	RFO2.1.3
TSFO2.5	RFO2.1.5
TSFO3	RFO3
TSFO3.1	RFO3.1
TSFO3.1.1	RFO3.1.1
TSFO3.1.2	RFO3.1.2
TSFO3.1.3	RFO3.1.3
TSFO3.2	RFO3.2
TSFO3.2.1	RFO3.2.1
TSFO3.2.2	RFO3.2.2
TSFO3.3	RFO3.3
TSFO3.4	RFO3.4
TSFO3.4.1	RFO3.4.1
TSFO3.4.2	RFO3.4.2
TSFO3.4	RFO3.4
TSFD4.1	RFD4.1
TSFD4.2	RFD4.2
TSFD4.3	RFD4.3
TSFD4.4	RFD4.4
TSFD4.5	RFD4.5
TSFD5	RFD5
TSFD5.1	RFD5.1

Test	Requisito
TSFD6	RFD6
TSFD6.1	RFD6.1
TSFF7	RFD7
TSFF7.1	RFD7.1
TSFF7.2	RFD7.2
TSFF7.3	RFD7.3

Table 8: Tabella di tracciamento test di sistema-requisiti

## C Valutazione per il miglioramento

### C.1 Valutazione sui ruoli

<b>Ruolo</b>	<b>Problema</b>	<b>Soluzione</b>
<b>Responsabile</b>	Difficoltà nella distribuzione corretta del carico di lavoro	Suddivisione del lavoro in piccole parti
<b>Verificatore</b>	Difficoltà nell'analisi completa e approfondita dei documenti	Aumento del tempo dedicato alle attività di verifica e utilizzo della lista di controllo
<b>Analista</b>	Difficoltà nella classificazione dei requisiti	Gli Analisti collaborano nello svolgimento del compito

Table 9: Valutazione sui ruoli

### C.2 Valutazione sull'organizzazione

<b>Problema</b>	<b>Soluzione</b>
Assegnazione precisa dei compiti	Utilizzo di Ticket, mediante Asana, per l'assegnazione di compiti precisi e con scadenza fissata
Difficoltà nell'organizzare incontri con tutti i membri	Utilizzo di strumenti di videoconferenza

Table 10: Valutazione sull'organizzazione

### C.3 Valutazione sugli strumenti

Strumento	Problema	Soluzione
<b>Strumenti generali</b>	<ol style="list-style-type: none"><li>1. Problemi di compatibilità tra diversi sistemi operativi;</li><li>2. Evidenziare solo la prima occorrenza di un termine presente nel glossario.</li></ol>	<ol style="list-style-type: none"><li>1. Utilizzare software disponibile sia per Linux che per Windows;</li><li>2. Creazione di uno script che aiuti in questo compito.</li></ol>
<b>Texmaker</b>	Problemi con la correzione delle parole italiane.	Impostato in modo corretto il vocabolario italiano.
<b>Github</b>	Problemi con conflitti durante i commit.	Creazione di un branch per ogni documento.

Table 11: Valutazione sugli strumenti

## D Standard di qualità

### D.1 ISO/IEC 15504

Lo standard ISO/IEC 15504, altresì conosciuto come SPICE (Software Process Improvement and Capability Determination), stabilisce un modello di riferimento per la valutazione della maturità (*capability dimension*) dei processi software (*process dimension*).

In particolare, la *process dimension* è definita in riferimento allo standard ISO/IEC 12207 per la gestione del ciclo di vita.

La *capability dimension*, invece, definisce una scala di sei livelli di maturità di processo:

- **0 - Incomplete:** il processo è fallito oppure non è stato implementato;
- **1 - Performed:** il processo è stato implementato ed ha adempito al proprio obiettivo;
- **2 - Managed:** il processo, oltre ad essere semplicemente *performed*, è gestito in maniera organizzata, con responsabilità ben definite, pianificandone e tracciandone l'esecuzione e garantendone la qualità;
- **3 - Established:** il processo, oltre ad essere *managed*, è implementato aderendo ai principi dell'ingegneria del software e agli standard esistenti;
- **4 - Predictable:** il processo, oltre ad essere *established*, è attuato entro limiti prestazionali definiti per il raggiungimento degli obiettivi previsti;
- **5 - Optimizing:** il processo, oltre ad essere *predictable*, è oggetto di miglioramento continuo per il soddisfacimento di obiettivi di business, attuali, previsti e futuri.

Ogni processo è classificabile in base al livello di soddisfacimento dei seguenti nove attributi:

- **1.1 Process performance:** capacità del processo di raggiungere gli obiettivi prefissati;
- **2.1 Performance management:** misura del grado di gestione dell'attuazione del processo in esame;
- **2.2 Work product management:** misura del grado di gestione dei prodotti del processo in esame;
- **3.1 Process definition:** misura dell'adeguatezza del processo rispetto agli standard di riferimento;
- **3.2 Process deployment:** capacità del processo di sfruttare le risorse allocate;
- **4.1 Process measurement:** capacità del processo di produrre misurazioni utili a fini di controllo;

- **4.2 Process control:** capacità del processo di essere corretto o migliorato grazie all'analisi delle misurazioni rilevate;
- **5.1 Process innovation:** misura del grado in cui i cambiamenti strutturali e di esecuzione del processo sono controllati a fini di innovare e migliorare gli standard presenti;
- **5.2 Process optimization:** capacità del processo di implementare le modifiche effettuate in modo da ottenere un miglioramento continuo nella realizzazione degli obiettivi prefissati.

La scala di valutazione degli attributi di processo è la seguente:

- **N:** non posseduto (0 – 15%);
- **P:** parzialmente posseduto (>15% – 50%);
- **L:** largamente posseduto (>50% – 85%);
- **F:** pienamente posseduto (>85% – 100%).

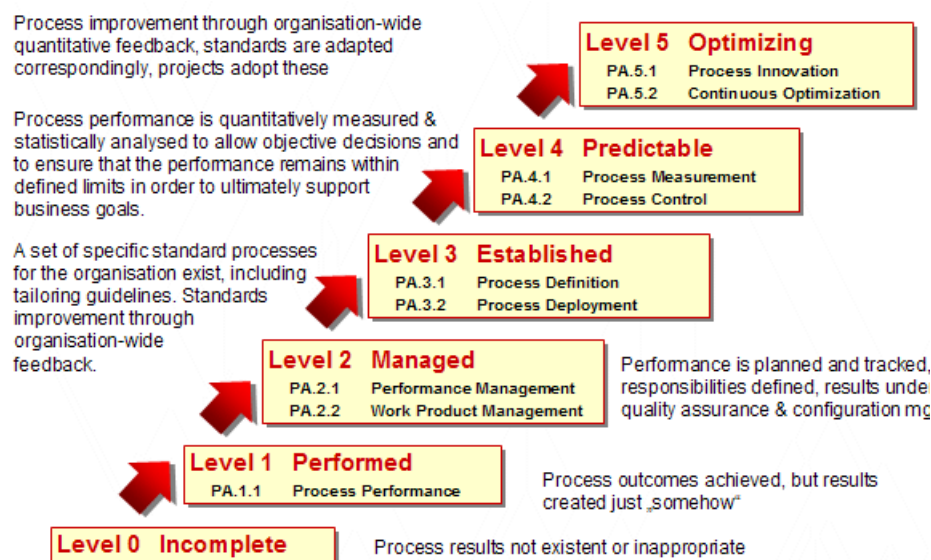


Figure 5: Schema della *capability dimension* di SPICE (tratta da SPiCE 1-2-1)

## D.2 PDCA

Il PDCA, conosciuto anche come *Ciclo di Deming*, è un metodo di gestione dei processi durante il loro ciclo di vita con il fine di controllare il miglioramento continuo della loro qualità e, quindi, anche quella dei loro prodotti. L'approccio che propone è suddiviso in quattro fasi da ripetere iterativamente fino al raggiungimento dell'obiettivo finale:

- **Plan:** fase di pianificazione in cui vengono stabiliti gli obiettivi ed i processi necessari per il raggiungimento dei risultati attesi;
- **Do:** fase di esecuzione di quanto pianificato al punto precedente con rilevamento di dati significativi da poter analizzare nelle fasi successive;
- **Check:** fase di controllo dei dati rilevati nella fase *Do* per confrontare i risultati ottenuti con quelli attesi dalla fase *Plan*. Le differenze riscontrate e le deviazioni nell'attuazione del piano osservate serviranno alla fase successiva;
- **Act:** fase di attuazione del miglioramento della qualità, tramite l'adozione di strategie emerse dallo studio dei risultati della fase di *Check*, eventualmente anche al di fuori del processo in questione.

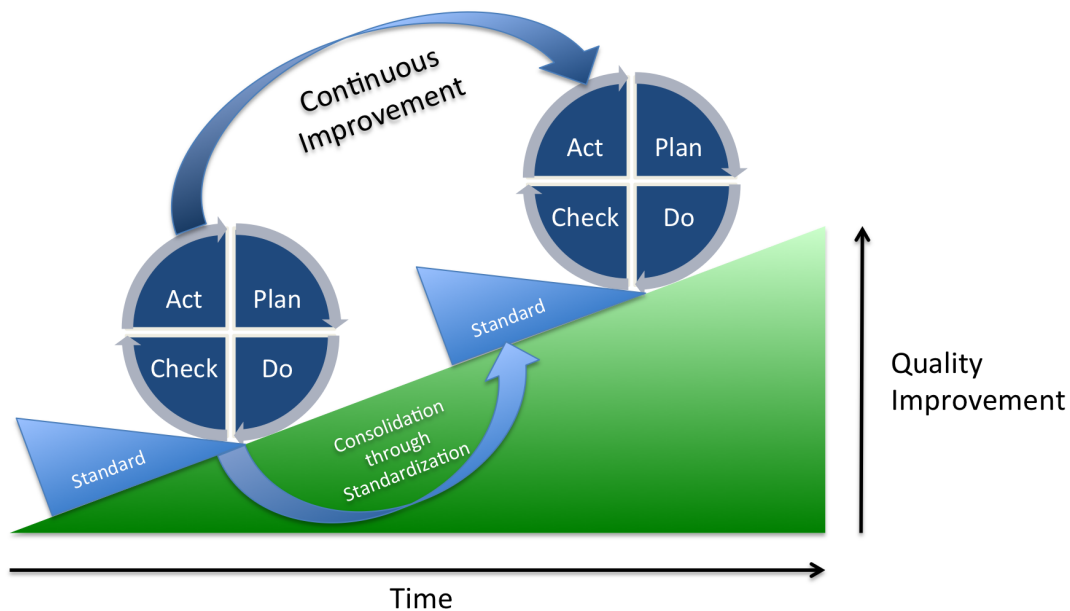


Figure 6: Schema del miglioramento continuo tramite PDCA (creato da Johannes Vietze)

### D.3 ISO/IEC 9126

Lo standard ISO/IEC 9126 fornisce un modello per la definizione della qualità di un software.

In particolare, esso distingue tre punti di vista sul software rispetto ai quali valutarne la qualità:

- **Qualità interna:** relativa al software sorgente non in esecuzione ed alla documentazione correlata. Viene rilevata tramite analisi statica ed è influenzata dalla qualità dei processi del ciclo di vita del prodotto;
- **Qualità esterna:** relativa al software in esecuzione. Viene rilevata tramite test, in funzione degli obiettivi stabiliti, ed è influenzata dalla qualità interna;
- **Qualità in uso:** relativa alla percezione dell'utente del prodotto finito in contesti reali d'uso. È influenzata dalla qualità esterna.

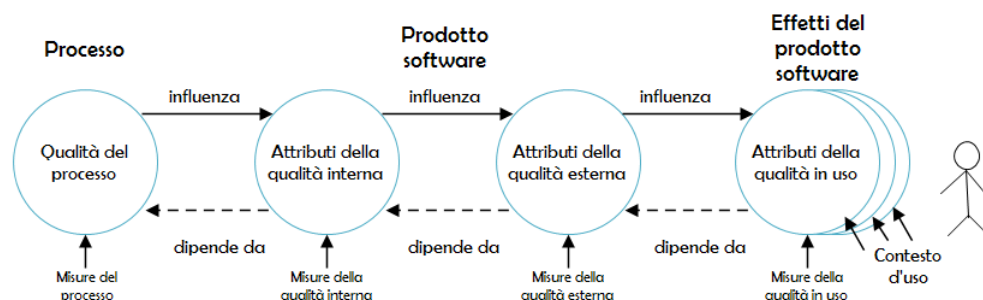


Figure 7: Schema del ciclo di qualità del software (creato da Giuseppe Manuele)

Per ciascuno dei punti di vista vengono inoltre delineate delle caratteristiche e sotto-caratteristiche qualitative, eventualmente misurabili quantitativamente, mediante apposite metriche.

Per la qualità interna ed esterna esse sono:

- **Funzionabilità:** capacità di fornire funzioni che soddisfino le esigenze stabilite, nei relativi contesti di presentazione.
  - **appropriatezza;**
  - **accuratezza;**
  - **interoperabilità;**
  - **conformità;**
  - **sicurezza.**



- **Affidabilità:** capacità di mantenere un determinato livello di prestazioni in date condizioni per un dato periodo.
  - **maturità;**
  - **tolleranza agli errori;**
  - **recuperabilità;**
  - **aderenza.**
- **Efficienza:** capacità di fornire appropriate prestazioni relativamente alle risorse utilizzate.
  - **comportamento rispetto al tempo;**
  - **utilizzo di risorse;**
  - **conformità.**
- **Usabilità:** capacità del prodotto software di essere capito, appreso e usato dall'utente, al verificarsi di determinate condizioni.
  - **comprensibilità;**
  - **apprendibilità;**
  - **operabilità;**
  - **attrattiva;**
  - **conformità.**
- **Manutenibilità:** capacità del prodotto software di essere modificato, corretto o migliorato facilmente nel tempo.
  - **analizzabilità;**
  - **modificabilità;**
  - **stabilità;**
  - **testabilità;**
  - **collaudabilità.**
- **Portabilità:** capacità del prodotto software di essere trasportato da un ambiente di lavoro all'altro.
  - **adattabilità;**
  - **installabilità;**
  - **conformità;**
  - **sostituibilità.**

Le caratteristiche per la qualità in uso sono:

- **Efficacia:** capacità di permettere all'utente di raggiungere gli obiettivi specificati con accuratezza e completezza;
- **Produttività:** capacità di permettere all'utente di spendere una quantità di risorse appropriata all'efficacia ottenuta dall'uso del prodotto;
- **Soddisfacibilità:** capacità di soddisfare l'utente;
- **Sicurezza:** capacità di raggiungere accettabili livelli di rischio nei confronti di persone e dell'ambiente di lavoro.

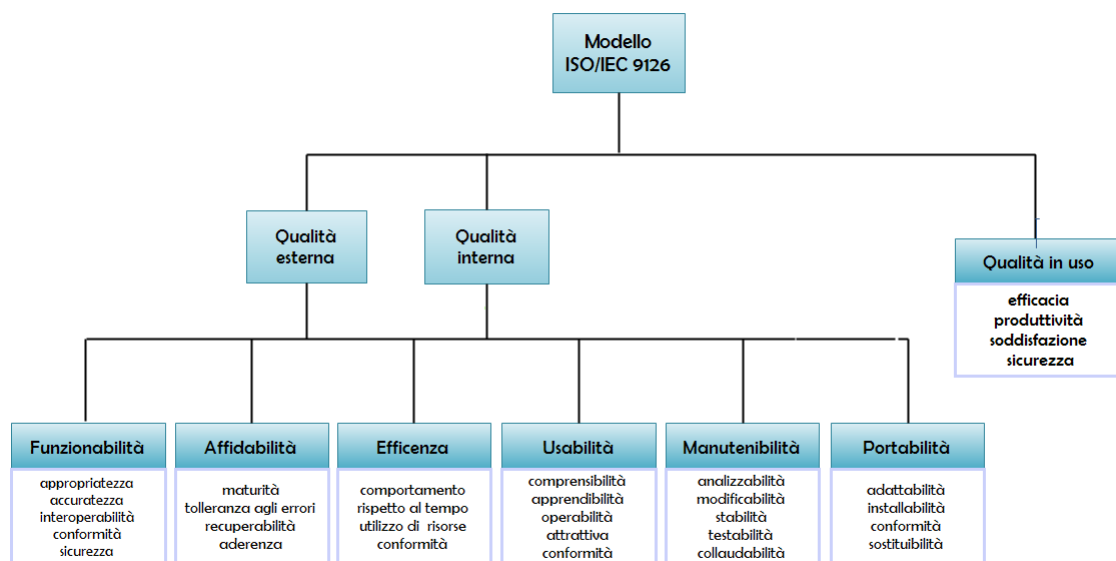


Figure 8: Schema delle caratteristiche definite in ISO/IEC 9126 (creato da Giuseppe Manuele)