



Piano di Qualifica

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

Versione	1.0.0
Redazione	Leonardo Nodari Isacco Maculan Carlo Munarini Mattia Bano
Verifica	Carlo Munarini Mattia Bano Dragos Cristian Lizan
Approvazione	Tommaso Carraro
Uso	Esterno
Distribuzione	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 *progettoG* OpenAPM.

Registro delle modifiche

Descrizione	Autore	Ruolo	Data	Versione
Approvazione del documento	Tommaso Carraro	Responsabile	3-01-2018	1.0.0
Verifica generale del documento	Dragos Cristian Lizan	Verificatore	28-12-2017	0.3.0
Stesura standard di qualità	Mattia Bano	Analista	17-12-2017	0.2.1
Verifica documento	Mattia Bano	Verificatore	14-12-2017	0.2.0
Stesura resoconto delle attività di verifica	Carlo Munarini	Verificatore	12-12-2017	0.1.1
Verifica documento	Mattia Bano	Verificatore	11-12-2017	0.1.0
Stesura e metriche di qualità	Leonardo Nodari	Analista	04-12-2017	0.0.3
Stesura generale delle strategie di gestione della qualità	Leonardo Nodari	Analista	1-12-2017	0.0.2
Stesura introduzione	Isacco Maculan	Analista	29-11-2017	0.0.1
Inserimento template latex	Isacco Maculan	Analista	29-11-2017	0.0.0

Indice

1	Introduzione	7
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
2	Visione generale delle strategie di gestione della qualità	9
2.1	Obiettivi di qualità	9
2.1.1	Qualità di processo	9
2.1.2	Qualità di prodotto	9
2.2	Tabella degli obiettivi	10
2.3	Scadenze temporali	10
3	Misure e metriche di qualità	11
3.1	Misure e metriche per i processi	11
3.1.1	Schedule Variance	11
3.1.2	Cost Variance	12
3.1.3	SPICE	12
3.2	Misure e metriche per i prodotti	12
3.2.1	Misure e metriche per i documenti	12
3.2.1.1	Indice Gulpease	12
3.2.2	Misure e metriche per il software	13
3.2.2.1	Structural Fan-In (SFIN)	13
3.2.2.2	Structural Fan-Out (SFOUT)	13
3.2.2.3	Logical Source Lines of Code	13
3.2.2.4	Code coverage	14
3.2.2.5	Test automation proportion	14
3.2.2.6	Rapporto linee di commento per linee di codice	14
3.2.2.7	Complessità ciclomatica	14
3.2.2.8	Failure avoidance	15
3.2.2.9	Percentuale superamento test	15
3.2.2.10	Requisiti obbligatori soddisfatti	15
3.2.3	Tabella delle metriche	16
A	Resoconto delle attività di verifica	17
A.1	Periodo di analisi	17
A.1.1	Verifica dei processi	17
A.1.1.1	Schedule variance	17
A.1.1.2	Cost variance	17

A.1.1.3	SPICE	17
A.1.2	Verifica dei prodotti	18
A.1.2.1	Indice Gulpease	18
A.1.3	Sommario della verifica nel periodo di analisi	18
B	Standard di qualità	19
B.1	ISO/IEC 15504	19
B.2	PDCA	21
B.3	ISO/IEC 9126	22

Tabelle

1	Tabella degli obiettivi	10
2	Tabella delle metriche	16
3	Esiti della Schedule variance - Periodo di analisi	17
4	Esiti della Cost variance - Periodo di analisi	17
5	Valori SPICE - Periodo di analisi	18
6	Esiti dell'indice Gulpease - Periodo di analisi	18
7	Sommario degli esiti - Periodo di analisi	18

Immagini

1	Schema della capability dimension di SPICE	20
2	Schema del miglioramento continuo tramite PDCA	21
3	Schema del ciclo di qualità del software	22
4	Schema delle caratteristiche definite in ISO/IEC 9126	24

1 Introduzione

1.1 Scopo del documento

Lo scopo del documento è quello di fissare, in modo quantitativo, gli obiettivi di qualità, di processo e di *prodottoG*, 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 uno spreco delle risorse.

1.2 Scopo del prodotto

Lo scopo del prodotto è realizzare un set di funzioni basate su *ElasticsearchG* e *KibanaG* per interpretare i dati raccolti da un *AgentG*. I dati interpretati forniranno a *DevOpsG* statistiche e informazioni utili per comprendere il funzionamento della propria applicazione. In particolare si richiede lo sviluppo di un motore di generazione di *metricheG* da *traceG*, un motore di generazione di *baselineG* basato sulle metriche del punto precedente, e un motore di gestione di *critical eventG*.

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 v1.0.0*. 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 v1.0.0*;
- **Capitolato d'appalto C7:**
<http://www.math.unipd.it/~tullio/IS-1/2017/Progetto/C7.pdf>
(ultima consultazione effettuata in data 11-12-2017).

1.4.2 Riferimenti Informativi

- **Piano di progetto:** *Piano di Progetto v1.0.0*;
- **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 11-12-2017);
- **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 11-12-2017);
- **Sommerville Ian, Software Engineering, 10th ed., Pearson (2015)**
- §24 Quality management
- **Sommerville Ian, Software Engineering, 9th ed., Pearson (2010)**
- §26 Process improvement
- **Standard ISO/IEC 15504:**
https://en.wikipedia.org/wiki/ISO/IEC_15504
(ultima consultazione effettuata in data 11-12-2017);
- **PDCA:**
https://it.wikipedia.org/wiki/Ciclo_di_Deming
(ultima consultazione effettuata in data 11-12-2017);
- **Standard ISO/IEC 9126:**
https://it.wikipedia.org/wiki/ISO/IEC_9126
(ultima consultazione effettuata in data 11-12-2017);
- **Indice di Gulpease:**
http://it.wikipedia.org/wiki/Indice_Gulpease
(ultima consultazione effettuata in data 11-12-2017);
- **Logical SLOC:**
https://en.wikiversity.org/wiki/Software_metrics_and_measurement
(ultima consultazione effettuata in data 11-12-2017).

2 Visione generale delle strategie 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 3.

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. Per approfondimenti vedere l'appendice B. Gli obiettivi fissati per i processi sono:

- rispettare tempi e costi descritti nel *Piano di Progetto v1.0.0*;
- avere prestazioni sempre misurabili;
- perseguire un miglioramento continuo delle prestazioni.

2.1.2 Qualità di prodotto

Basandosi sullo standard ISO/IEC 9126, descritto nell'appendice B, 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 **softwareG** deve:
 - soddisfare tutti i requisiti obbligatori descritti in *Analisi dei Requisiti v1.0.0*;
 - superare gran parte dei test;
 - garantire usabilità e manutenibilità;
 - essere affidabile.

2.2 Tabella degli obiettivi

Viene qui riassunto ogni obiettivo, classificandolo con il suo codice identificativo e indicando la rispettiva metrica per quantificare il raggiungimento dell'obiettivo. Per una descrizione più approfondita delle metriche vedere nella sezione 3. La sintassi del codice obiettivo è spiegata nelle *Norme di Progetto v1.0.0*.

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 ciclica
OPDS4	Affidabilità	MPDS8:Failure avoidance

Table 1: Tabella degli obiettivi

2.3 Scadenze temporali

Il rispetto delle *milestonesG* presenti in *Piano di Progetto v1.0.0* indicano che la realizzazione del prodotto sta procedendo nel migliore dei modi, garantendo la qualità del risultato finale. Eventuali ritardi indicano che sono presenti parti incomplete o che non dispongono di un grado di qualità accettabile. Per prevenire l'insorgenza di errori che potrebbero rimandare la consegna del prodotto, con conseguente impatto nel preventivo fornito in *Piano di Progetto v1.0.0*, MILCTdev attua procedure di verifica descritte in dettaglio nelle *Norme di Progetto v1.0.0*.

3 Misure e metriche di qualità

Ogni processo ed ogni prodotto dovrebbero sempre presentare un set di *KPIG* 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.

All'interno della spiegazione di ogni metrica verrà illustrato quando, come e su cosa viene fatta la misurazione durante il processo di verifica.

Per valutare gli esiti ottenuti dall'utilizzo delle metriche, sono stati individuati tre diversi range di risultati possibili che indicano ciascuno un diverso grado di raggiungimento dell'obiettivo di qualità. Questi sono:

- **Valore negativo:** il valore della misurazione in questo caso viene anche definito inaccettabile, in quanto non soddisfa la qualità minima desiderata;
- **Valore minimo:** valori al di sopra di questa soglia possono essere accettati, ma saranno oggetto di ulteriore analisi in vista di un miglioramento desiderato;
- **Valore ottimale:** rappresenta il valore indice di qualità, da mantenere nel tempo.

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.

3.1 Misure e metriche per i processi

3.1.1 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 il team nell'analisi dell'utilizzo di risorse temporali.

Il calcolo della Schedule Variance avviene in questo modo:

$$SV = \text{data conclusione reale} - \text{data conclusione preventivata}$$

Entrambe le date si riferiscono alla conclusione dell'attività o del processo ed i risultati della misurazione possono quindi essere:

- **Valore negativo:** se il valore è maggiore di 3;
- **Valore minimo:** se il valore è inferiore o uguale a 3;
- **Valore ottimale:** se la Schedule Variance è di 0 giorni o inferiore.

3.1.2 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:

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

I risultati possono quindi essere:

- **Valore negativo:** se il valore è maggiore all'8% del costo preventivato;
- **Valore minimo:** se il valore è inferiore o uguale all'8% del costo preventivato;
- **Valore ottimale:** se il valore è inferiore all'1% del costo preventivato.

3.1.3 SPICE

Al termine di ogni periodo, il *teamG* MILCTdev provvederà alla valutazione della qualità dei processi tramite lo standard ISO/IEC 15504 conosciuto come SPICE. I risultati della misurazione potranno quindi essere:

- **Negativi:** se la maturità di processo è Performed (Livello 1) o Incomplete;
- **Accettabili:** se la maturità di processo è almeno Managed (Livello 2);
- **Ottimali:** se la maturità di processo è Predictable (Livello 4) o Optimizing (Livello 5).

Lo standard SPICE, ed i livelli di maturità appena citati, vengono illustrati in maniera completa ed approfondita nell'Appendice B.

3.2 Misure e metriche per i prodotti

3.2.1 Misure e metriche per i documenti

3.2.1.1 Indice Gulpease

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

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. Per il gruppo MILCTdev questo indice potrà fornire tre tipi di risultato:

- **Valore negativo:** se l'indice Gulpease è inferiore 40;
- **Valore minimo:** se l'indice è superiore o uguale a 40, questa è anche la soglia minima indicata da alcuni esperti per il “Livello di Lettura Indipendente”;
- **Valore ottimale:** se l'indice è superiore a 70.

3.2.2 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.

3.2.2.1 Structtural 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 Structtural 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.

3.2.2.3 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* all'interno del codice.

3.2.2.4 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:

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

3.2.2.5 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:

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

3.2.2.6 Rapporto linee di commento per linee di codice

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

Il valore viene espresso in percentuale e viene così calcolato:

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

Il risultato può quindi essere:

- **Valore negativo:** se la percentuale di commenti è inferiore al 10%;
- **Valore minimo:** se la percentuale è superiore o uguale al 10%;
- **Valore ottimale:** se la percentuale è superiore al 20%.

3.2.2.7 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 *packagesG*, moduli, metodi o classi.

Il calcolo avviene sfruttando il *grafo di controllo di flussoG* 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.

3.2.2.8 Failure avoidance

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

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

3.2.2.9 Percentuale superamento test

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

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

Il risultato può essere:

- **Valore negativo:** se la percentuale è inferiore al 75%;
- **Valore minimo:** se la percentuale è superiore o uguale al 75%;
- **Valore ottimale:** se la percentuale è superiore al 95%.

3.2.2.10 Requisiti obbligatori soddisfatti

Questa metrica aiuta il team a capire in che quantità sono stati soddisfatti i requisiti obbligatori indicati in *Analisi dei Requisiti v1.0.0*.

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

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

Il risultato di questo calcolo potrà poi essere:

- **Valore negativo:** se la percentuale dei requisiti obbligatori soddisfatti è inferiore al 100%;

- **Valore minimo e ottimale:** se la percentuale è del 100%.

3.2.3 Tabella delle metriche

Viene qui riassunto ogni metrica, classificandola con il suo codice identificativo. Il codice identificativo per le metriche è spiegato nelle *Norme di Progetto v1.0.0*.

ID	Metrica
MPC1	Schedule Variance
MPC2	Cost Variance
MPC3	SPICE
MPDD1	Indice Gulpease
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
MPDS8	Failure avoidance
MPDS9	Percentuale superamento test
MPDS10	Requisiti obbligatori soddisfatti

Table 2: Tabella delle metriche

A Resoconto delle attività di verifica

Questa sezione illustra i risultati di verifica, ottenuti utilizzando le metriche descritte nella sezione 3, sui processi ed i prodotti dei diversi periodi descritti nel *Piano di Progetto v1.0.0*.

A.1 Periodo di analisi

A.1.1 Verifica dei processi

A.1.1.1 Schedule variance

Vengono qui illustrati i valori di Schedule variance calcolati sui tempi di stesura dei documenti:

Documento	SV in giorni	Giudizio
Analisi dei Requisiti v1.0.0	0	ottimale
Glossario v1.0.0	0	ottimale
Norme di Progetto v1.0.0	0	ottimale
Piano di Progetto v1.0.0	0	ottimale
Piano di Qualifica v1.0.0	-2	ottimale
Studio di Fattibilità v1.0.0	0	ottimale
Totale	-2	ottimale

Table 3: Esiti della Schedule variance - Periodo di analisi

A.1.1.2 Cost variance

Vengono qui illustrati i valori di Cost variance calcolati sui tempi di stesura dei documenti:

Cost variance	Giudizio
0%	ottimale

Table 4: Esiti della Cost variance - Periodo di analisi

A.1.1.3 SPICE

Vengono qui riportati i valori riguardo i processi del periodo di analisi, secondo lo standard ISO/IEC 15504:

Processo	SPICE inizio	SPICE fine
Fornitura	0	3
Sviluppo	0	2
Documentazione	0	3
Versionamento	0	3
Verifica	0	3

Table 5: Valori SPICE - Periodo di analisi

A.1.2 Verifica dei prodotti

A.1.2.1 Indice Gulpease

Ogni documento viene sottoposto ad un test sulla leggibilità e i risultati sono i seguenti:

Documento	Indice Gulpease	Giudizio
Analisi dei Requisiti v1.0.0	74	ottimale
Glossario v1.0.0	53	accettabile
Norme di Progetto v1.0.0	50	accettabile
Piano di Progetto v1.0.0	48	accettabile
Piano di Qualifica v1.0.0	50	accettabile
Studio di Fattibilità v1.0.0	54	accettabile
Verbale Interno 28-11-2017	57	accettabile
Verbale Esterno 19-12-2017	46	accettabile
Verbale Interno 30-12-2017	53	accettabile

Table 6: Esiti dell'indice Gulpease - Periodo di analisi

A.1.3 Sommario della verifica nel periodo di analisi

Metrica	Giudizio
Schedule variance	ottimale
Cost variance	ottimale
SPICE	accettabile
Indice Gulpease	accettabile

Table 7: Sommario degli esiti - Periodo di analisi

B Standard di qualità

B.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%).

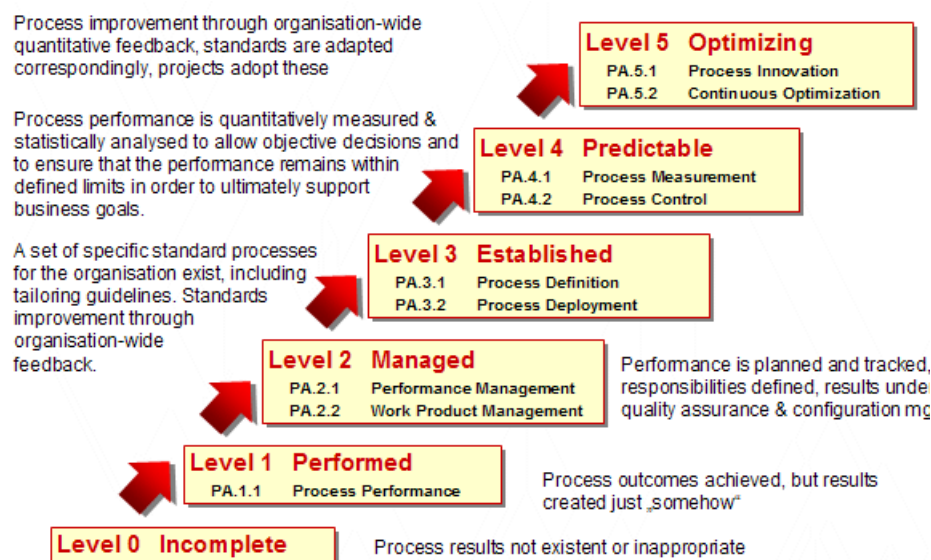


Figure 1: Schema della *capability dimension* di SPiCE (tratta da SPiCE 1-2-1)

B.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.

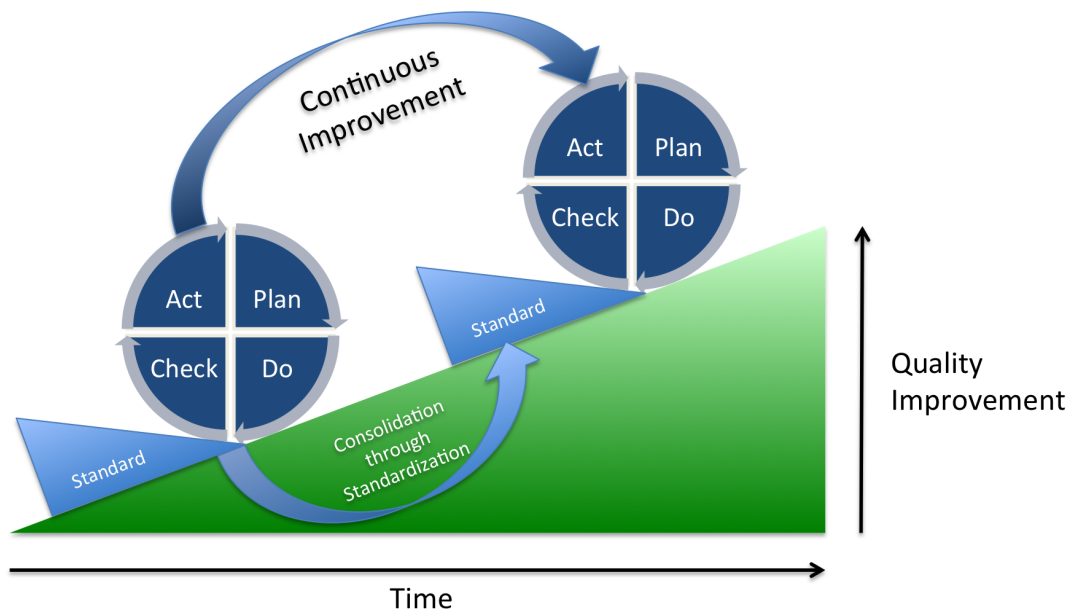


Figure 2: Schema del miglioramento continuo tramite PDCA (creato da Johannes Vietze)

B.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.

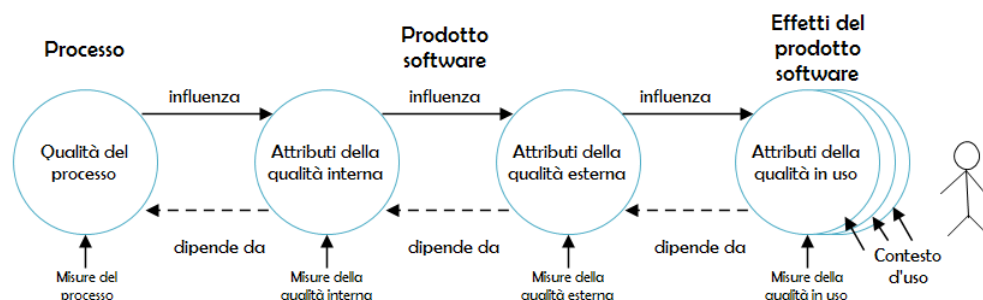


Figure 3: Schema del ciclo di qualità del software (creato da Giuseppe Manuele)

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.

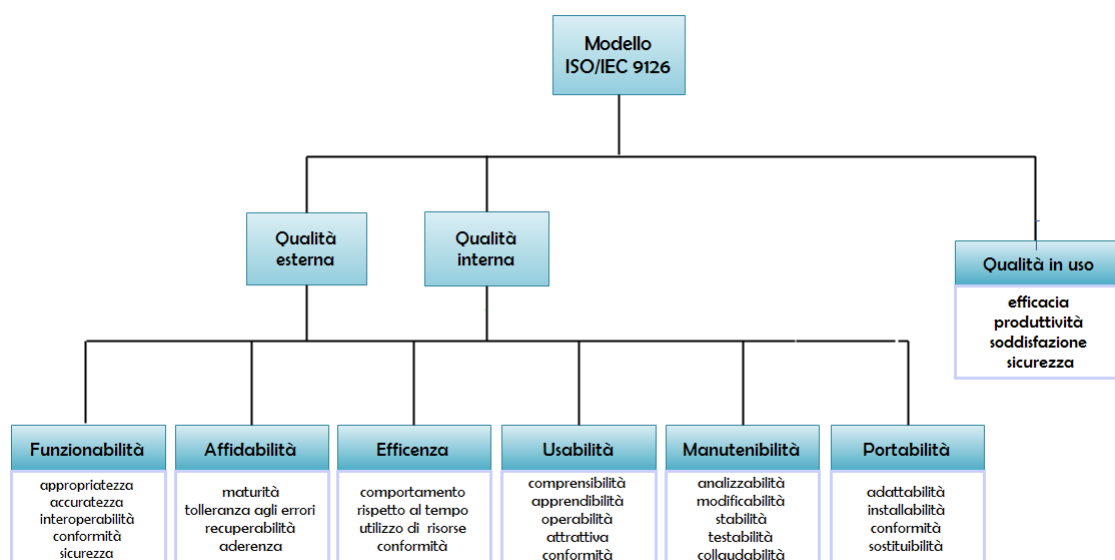


Figure 4: Schema delle caratteristiche definite in ISO/IEC 9126 (creato da Giuseppe Manuele)