Piano di Qualifica

Gruppo SWEight - Progetto Colletta

SWEightGroup@gmail.com

Informazioni sul documento

Versione	1.0.0
Approvatore	Sebastiano Caccaro
Redattori	Alberto Bacco Sebastiano Caccaro Gheorghe Isachi
Verificatori	Gheorghe Isachi Enrico Muraro Alberto Bacco
Uso	Esterno
Distribuzione	MIVOQ Prof. Vardanega Tullio Prof. Cardin Riccardo Gruppo SWEight

Descrizione

Questo documento di occupa di definire le misure attraverso le quali il gruppo *SWEight* intende garantire la qualità del progetto.

Registro delle modifiche

Versione	Data	Descrizione	Nominativo	Ruolo
1.0.0	2019-01-09	Approvazione	Sebastiano Caccaro	*Responsabile*
0.4.1	2019-01-09	Verifica e correzione appendice esito verifica	Isachi Gheorghe	*Verificatore*
0.4.0	2019-12-09	Appendice esito verifica	Alberto Bacco	*Verificatore*
0.3.1	2019-01-03	Correzione Verifica	Sebastiano Caccaro	*Redattore*
0.3.0	2019-01-02	Verifica Documento	Enrico Muraro	*Verificatore*
0.3.0	2018-12-30	Fine metriche software e tabella riassuntiva	Sebastiano Caccaro	*Redattore*
0.2.8	2018-12-27	Aggiunte metriche software	Sebastiano Caccaro	*Redattore*
0.2.7	2018-12-26	Inzio metriche software	Sebastiano Caccaro	*Redattore*
0.2.5	2018-12-23	Metriche processi	Sebastiano Caccaro	*Redattore*
0.2.4	2018-12-20	Verifica	Sebastiano Caccaro	*Redattore*
0.2.4	2018-12-20	Sezione Responsabilità	Alberto Bacco	*Redattore*
0.2.3	2018-12-20	Pianificazione strategica temporale	Alberto Bacco	*Redattore*
0.2.2	2018-12-18	Appendice ISO/IEC 9126	Isachi Gheorghe	*Redattore*
0.2.1	2018-12-16	Appendice SPICE	Alberto Bacco	*Redattore*
0.2.0	2018-12-16	Obiettivi qualità	Gheorghe Isachi	*Redattore*
0.1.0	2018-12-14	Sezione introduzione	Gheorghe Isachi	*Redattore*
0.0.1	2018-12-13	Creazione scheletro	Alberto Bacco	*Redattore*

Indice

Elenco delle figure

Elenco delle tabelle

1 Introduzione

1.1 Scopo del documento

In questo documento è illustrata la strategia$_G$ di verifica$_G$ e validazione$_G$ del gruppo *SWEight*. Tale strategia è fondamentale per dare una misurazione oggettiva e quantificabile del livello di qualità$_G$ di quanto viene prodotto.
Ciò è vantaggioso sia per il gruppo *SWEight*, che può facilmente individuare difetti durante lo svolgimento del progetto, sia per il committente$_G$, che può costantemente monitorare la qualità del prodotto in base a criteri oggettivi e prestabiliti.

1.2 Scopo del prodotto

Il progetto prevede la realizzazione di una piattaforma collaborativa di raccolta dati in cui gli utenti possano predisporre e/o svolgere piccoli esercizi di analisi grammaticale. Lo scopo è raccogliere dati relativi sia agli esercizi predisposti, che al loro svolgimento da parte degli utenti. Sviluppatori e ricercatori utilizzeranno queste informazione per insegnare ad un elaboratore a svolgere i medesimi esercizi, mediante tecniche di apprendimento automatico.

1.3 Glossario

Nel documento è possibile incontrare termini tecnici, i quali potrebbero non essere immediatamente chiari al lettore. Per disambiguarne il significato, essi sono stati marcati con una $_G$ a pedice e la loro definizione è reperibile nel glossario fornito separatamente.

1.4 Riferimenti

1.4.1 Riferimenti normativi

- **Norme di Progetto:** *NormeDiProgetto_v1.0.0*;
- **Capitolato d'appalto C2:** Colletta
  https://www.math.unipd.it/~tullio/IS-1/2018/Progetto/C2.pdf.

1.4.2 Riferimenti informativi

- **Piano di Progetto:** *PianoDiProgetto_v1.0.0*;
- **Slide del corso di Ingegneria del Software:**
  https://www.math.unipd.it/~tullio/IS-1/2018;
- **Ian Sommerville, Software Engineering, Nona edizione:**
  - Capitolo 24: Quality management;
  - Capitolo 26: Process improvement;
- **Standard ISO/IEC 9126:**
  https://it.wikipedia.org/wiki/ISO/IEC_9126
  https://en.wikipedia.org/wiki/ISO/IEC_9126;
- **Standard ISO/IEC 15504:**
  https://en.wikipedia.org/wiki/ISO/IEC_15504
  https://www.researchgate.net/publication/29453909_The_ISOIEC_15504_Measurement_Framework_for_Process_Capability_and_CMMI

- **Metriche software:**
  https://en.wikipedia.org/wiki/Software_metric;
- **Metriche sui processi:**
  https://it.wikipedia.org/wiki/Metriche_di_progetto;
- **Indice Gulpease:**
  https://it.wikipedia.org/wiki/Indice_Gulpease;
- **NOM:** Number of methods
  http://support.objecteering.com/objecteering6.1/help/us/metrics/metrics_in_detail/number_of_methods.htm;
- **Complessità ciclomatica:**
  https://blogs.msdn.microsoft.com/zainnab/2011/05/17/code-metrics-cyclomatic-complexity/;

1.5 Note

Il presente documento presenta delle sezioni che il gruppo *SWEight* si riserva di redigere in un successivo periodo. Questa decisione deriva dal fatto che, allo stato attuale, il gruppo *SWEight* non possiede le informazioni necessarie per redigerle, ma ne prevede la futura necessità.

2 Strategie di verifica

La strategia generale del gruppo *SWEight* mira ad automatizzare il più possibile il lavoro di verifica, facendo gestire tale processo a degli strumenti informatici opportunamente predisposti.
I vantaggi derivanti da tale scelta sono i seguenti:

- **Costo:** l'uso di tool$_G$ automatizzati non impiega risorse$_G$ umane, il che si traduce in un risparmio in ore di lavoro e risorse economiche. È quindi possibile eseguire più spesso attività di verifica.
- **Misurabilità:** se eseguita da strumenti informatici, l'attività di verifica produce dei valori oggettivi e tracciabili nel tempo.
- **Precisione:** la tediosità e la ripetitività della verifica manuale rendono quest'ultima frustrante e prona a contenere imprecisioni. Un computer, invece, esegue tale compito velocemente e senza errori.

2.1 Obiettivi di qualità

2.1.1 Qualità di processo

Al fine di garantire un prodotto di qualità, è fondamentale avere un elevato standard qualitativo anche per i processi. Per garantire tutto ciò, è stato deciso di aderire allo standard SPICE$_G$[1] (ISO/IEC 15504): quest'ultimo permette di valutare il livello di maturità$_G$ e capacità (capability$_G$) dei processi, al fine di apportare modifiche migliorative.
Sono fissati i seguenti obiettivi:

- Rispetto di tempi e costi descritti nel *PianoDiProgetto_v1.0.0*;
- Continuo miglioramento dei processi;
- Misurabilità dello stato dei processi.

2.1.2 Qualità di prodotto

Per garantire la qualità dei prodotti, viene adottato lo standard ISO/IEC 9126[2]. Quest'ultimo permette di monitorare la qualità del software, fornendo delle metriche per misurarla.
Sono fissati i seguenti obiettivi:

- La **documentazione** deve essere:
  - Facilmente leggibile;
  - Scritta in modo corretto, secondo le regole della lingua italiana.
- Il **software** deve:
  - Soddisfare i requisiti stabiliti nell'*AnalisiDeiRequisiti_v1.0.0*;
  - Garantire semplicità di utilizzo;
  - Garantire semplicità di manutenzione;
  - Garantire affidabilità.

[1] SPICE: Vedi appendice §C.1
[2] ISO/IEC 9126: Vedi appendice §C.2

2.2 Responsabilità

Al fine di garantire un maggior controllo della qualità, l'attività di verifica è a carico dei *Verificatori* e del *Responsabile*, che ha compito di approvazione finale su quanto prodotto. Tuttavia, per ovvi motivi, un *Verificatore* non può verificare contenuti da lui stesso prodotti, mentre gli è concessa la verifica di parti scritte da altri membri in file in cui egli stesso è redattore.

2.3 Pianificazione strategica temporale

2.3.1 Strategia

Lo sviluppo del prodotto, come descritto nel *PianoDiProgetto_v1.0.0*, avviene secondo il modello incrementale$_G$. É utile distinguere due tipi di incremento:

- **Programmato:** prefissato nel calendario;
- **Non programmato** insorge in seguito ad attività di verifica, sia manuali che automatiche. Può essere la correzione di un bug$_G$, di un errore errore ortografico, o, in più in generale, di una problematica in un prodotto.

Il mancato svolgimento di quest'ultimo tipo di incremento, può portare alla permanenza di problematiche nel prodotto a scapito della qualità. A tale fine, è necessario focalizzarsi su:

- **Prevenzione:** misure atte ad evitare l'insorgenza di problematiche;
- **Individuazione:** misure atte a individuare tempestivamente possibili problematiche.

Tali misure possono essere efficienti$_G$ ed efficaci$_G$ solo se propriamente automatizzate: le tecniche adottate dal gruppo *SWEight* e la loro evoluzione sono reperibili nelle *NormeDiProgetto_v1.0.0*.

2.3.2 Tempistiche

La verifica avrà luogo nei tempi e nelle scadenze descritte nel *PianoDiProgetto_v1.0.0*, dove, per tenere conto degli incrementi non programmati, la pianificazione aggiunge dello slack$_G$ per ogni attività.

2.4 Misure e metriche

In questa sezione sono riportate le metriche che il gruppo *SWEight* intende utilizzare per rendere misurabile la qualità di:

- Processi;
- Documenti:
- Software.

Ogni metrica elencata conterrà le seguenti voci:

- **Nome**;
- **Descrizione**;
- **Parametri adottati:** range di valori sui confrontare le misure ottenute. Sono definiti i seguenti intervalli:
  - Accettabile;
  - Ottimale.

Non saranno trattati in questo documento gli strumenti per il calcolo delle metriche, che sono reperibili nelle *NormeDiProgetto_v1.0.0*;

2.4.1 Metriche processi

Le metriche presentate in questa sezione monitorano lo stato dei processi del progetto analizzando l'uso che essi fanno di tempo e risorse finanziarie. Sono particolarmente utili per il *Responsabile*, che può quindi decidere di apportare modifiche alla pianificazione quando necessario.

2.4.1.1 Schedule Variance

La Schedule Variance indica se una certa attività o processo è in anticipo, in pari, o in ritardo rispetto alla data di scadenza prevista. Se $SV < 0$ significa che l'attività o il processo è in pari o in anticipo, invece, se $SV \geq 0$ significa che l'attività è in ritardo.

La Schedule Variance è calcolata come segue:

$$SV = \textit{data conclusione effettiva} - \textit{data conclusione pianificata}$$

Parametri adottati:

- Range accettabile: $(-\infty, 3]$;
- Range ottimale: $(-\infty, 0]$.

2.4.1.2 Budget Variance

La Budget Variance misura, ad una determinata data, lo scostamento fra quanto speso e quanto preventivato. Se $BV \geq 0\%$ significa che il progetto sta spendendo le risorse più velocemente di quanto pianificato.

La Budget Variance è calcolata come segue:

$$BV = \frac{\textit{costo effettivo} - \textit{costo preventivato}}{\textit{costo preventivato}}$$

Parametri adottati:

- Range accettabile: $(-\infty, 9\%]$;
- Range ottimale: $(-\infty, 1\%]$.

2.4.2 Metriche documenti

Le metriche presentate in questa sezione hanno come scopo fornire dei parametri per garantire un buon livello di leggibilità dei documenti.

2.4.2.1 Indice Gulpease

L'Indice Gulpease è un indice di leggibilità di un testo tarato sulla lingua italiana. Rispetto ad altri ha il vantaggio di utilizzare la lunghezza delle parole in lettere anziché in sillabe, semplificandone il calcolo automatico. L'indice di Gulpease considera due variabili linguistiche: la lunghezza della parola e la lunghezza della frase rispetto al numero delle lettere.

L'indice di Gulpease può assumere valori fra 0 e 100 e si calcola come segue:

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

Parametri adottati:

- Range accettabile: $[40, 100]$
- Range ottimale: $[55, 100]$

2.4.3 Metriche software

Alcune metriche per il software sono più adatte per alcuni linguaggi di programmazione e meno per altri. Come indicato nel *PianoDiProgetto_v1.0.0*, il gruppo *SWEight* sceglierà quali linguaggi usare nel periodo di Progettazione Architetturale; pertanto, le metriche presenti in questa sezione non sono da considerarsi complete o definitive.
Per alcune metriche, può mancare un'indicazione di valori accettabili e ottimali: ciò significa che il team si riserva di definirli in futuri incrementi.

2.4.3.1 Numero di Metodi

Numero medio di metodi contenuti nelle classi di un package. Un numero di metodi troppo alto può indicare la necessità di scomporre una classe. Un numero di metodi troppo basso, d'altro canto, deve far riflettere sull'effettiva utilità della classe presa in esame.

Parametri adottati:

- Range accettabile: $[3, 9]$;
- Range ottimale: $[3, 7]$.

2.4.3.2 Numero di Parametri

Numero di parametri passati a un metodo. Un eccessivo numero di parametri passati ad un metodo può indicare un'eccessiva complessità dello stesso, che va scomposto o quanto meno ripensato.

Parametri adottati:

- Range accettabile: $[0, 8]$;
- Range ottimale: $[0, 5]$.

2.4.3.3 Funzioni di interfaccia per package

Numero di funzione che un package espone. Un valore troppo elevato potrebbe indicare un errore di progettazione

Parametri adottati:

- Range accettabile: $[0, 20]$;
- Range ottimale: $[0, 10]$.

2.4.3.4 Complessità Ciclomatica

La Complessità Ciclomatica (CC) è una metrica software che misura la complessità di un programma contando il numero di cammini linearmente indipendenti attraverso il grafo di controllo di flusso. In questo grafo, i nodi corrispondono a gruppi indivisibili di istruzioni, mentre gli archi connettono due nodi se il secondo gruppo di istruzioni può essere eseguito immediatamente dopo il primo gruppo: sono quindi responsabili dell'aumento della CC i punti decisionali, *if* e *for*. Tenere bassa la CC può portare a vari vantaggi:

- Minore complessità durante lo sviluppo;
- Maggior facilità nell'aumentare la code coverage in fase di test;
- Maggior coesione del codice;

La Complessità Ciclomatica è calcolata come segue:

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

dove:

- e = numero di archi del grafo;
- n = numero di nodi del grafo;
- p = numero di componenti connesse.

Parametri adottati:

- Range accettabile: $[0, 17]$;
- Range ottimale: $[0, 10]$.

2.4.3.5 Campi dati per classe

Numero di campi dati contenuti da una classe. Una classe con troppi campi dati può essere sintomo di cattiva progettazione e va ripensata.

Parametri adottati:

- Range accettabile: $[0, 15]$;
- Range ottimale: $[0, 10]$.

2.4.3.6 Commenti per linee di codice

Rapporto fra le righe di codice (righe vuote escluse) e le righe di commento. Un codice ben commentato può essere compreso più facilmente e velocemente, facilitando le operazioni di manutenzione.

Parametri adottati:

- Range accettabile: $[10\%, 100\%]$;
- Range ottimale: $[15\%, 100\%]$.

2.4.3.7 Code Coverage

Percentuale delle linee di codice coperte dai test.

Parametri adottati:
Il gruppo *SWEight* si riserva di decidere in futuro i parametri da adottare.

2.4.3.8 Superamento test

Percentuale di test superati. Per avere un prodotto di qualità, è necessario che esso superi i test prestabiliti.

Parametri adottati:

- Range accettabile: $[70\%, 100\%]$;
- Range ottimale: $[95\%, 100\%]$.

2.4.3.9 Requisiti obbligatori soddisfatti

Percentuale di requisiti obbligatori stabiliti dalla proponente soddisfatti. É fondamentale, per la buona riuscita del progetto, soddisfare i requisiti obbligatori individuati nell'*AnalisiDeiRequisiti_v1.0.0*.

Parametri adottati:

- Range accettabile: $[70\%, 100\%]$;
- Range ottimale: $[95\%, 100\%]$.

2.4.4 Riassunto metriche

Nome	Tipo	Accettabile	Ottimale
Indice Gulpease	Documentazione	$[40, 100]$	$[55, 100]$
Schedule Variance	Processo	$(-\infty, 3]$	$(\infty, 0]$
Budget Variance	Processo	$(-\infty, 9\%]$	$(-\infty, 1\%]$
Number di metodi	Software	$[3, 9]$	$[3, 7]$
Numero di parametri	Software	$[0, 8]$	$[0, 5]$
Funzioni di interfaccia per package	Software	$[0, 20]$	$[0, 10]$
Complessità ciclomatica	Software	$[0, 17]$	$[0, 10]$
Campi dati per classe	Software	$[0, 15]$	$[0, 10]$
Commenti per linee di codice	Software	$[10\%, 100\%]$	$[15\%, 100\%]$
Code coverage	Software		
Superamento test	Software	$[70\%, 100\%]$	$[95\%, 100\%]$
Soddisfacimento requisiti obbligatori	Software	$[100\%, 100\%]$	$[100\%, 100\%]$

Tabella 2: Riassunto delle metriche

A Esito Verifica

A.1 Periodo di Analisi

In questa sezione è consultabile l'esito delle attività di verifica diviso per periodo.
Per ogni periodo sono riportate:

- Le **misurazioni** delle metriche descritte in §2.4; Viene stabilita la seguente convenzione cromatica per il valore delle misurazioni:
  - **Rosso**: non accettabile;
  - **Giallo**: accettabile;
  - **Verde**: ottimale;
- Una **retrospettiva** testuale che evidenzia criticità e problemi esposti dalla verifica.

A.1.1 Misurazioni Documenti

A.1.1.1 Indice di Gulpease

Documento	Abbreviazione	Valore Indice	Riscontro
Analisi dei Requisiti	ADR	58,84	Ottimale
Glossario	GLO	51,12	Accettabile
Piano di Progetto	PDQ	54,89	Accettabile
Piano di Qualifica	PDP	56,67	Ottimale
Norme di Progetto	NDP	55,3	Ottimale
Studio di Fattibilità	SDF	56,77	Ottimale

Tabella 3: Indice di Gulpease nel periodo di Analisi

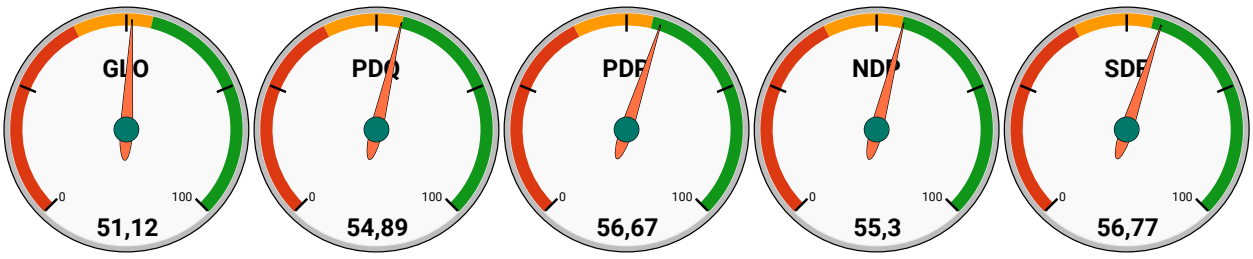

Figura 1: Indice di Gulpease nel periodo di Analisi

A.1.2 Misurazione Processi

A.1.2.1 Schedule Variance

Attività	Abbreviazione	Valore Indice	Riscontro
Stesura Analisi dei Requisiti	ADR	1	Accettabile
Stesura Glossario	GLO	-1	Ottimale
Stesura Piano di Progetto	PDQ	0	Ottimale
Stesura Piano di Qualifica	PDP	-1	Ottimale
Stesura Norme di Progetto	NDP	1	Accettabile
Stesura Studio di Fattibilità	SDF	1	Accettabile

Tabella 4: Schedule Variance nel periodo di Analisi

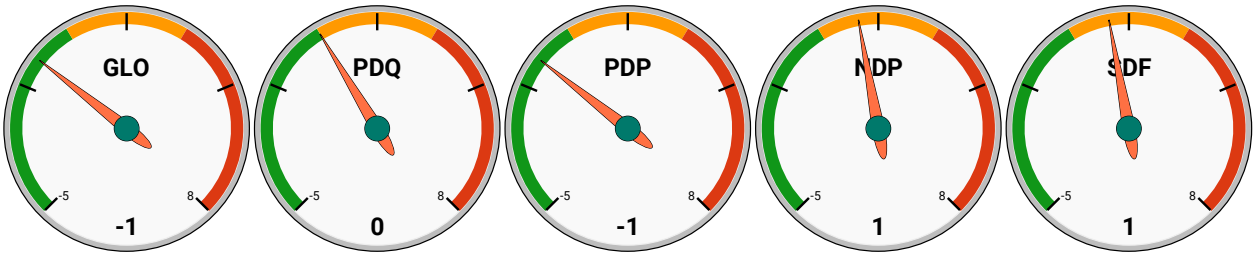

Figura 2: Schedule Variance nel periodo di Analisi

A.1.2.2 Budget Variance

Abbreviazione	Valore Indice	Valore in €	Riscontro
BV	5,71%	200	Accettabile

Tabella 5: Budget Variance nel periodo di Analisi

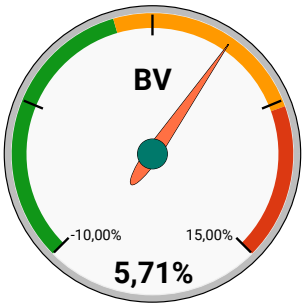

Figura 3: Budget Variance nel periodo di Analisi

A.1.3 Retrospettiva

Le misurazioni effettuate non hanno trovato valori non accettabili.
É però pericolosamente alta la Budget Variance: ciò è dovuto principalmente a dei difetti nell'organizzazione che sono in parte stati risolti, e che verranno del tutto corretti nel prossimo periodo.

B Pianificazione Test

Per assicurarsi che i requisiti individuati nell'*AnalisiDeiRequisiti_v1.0.0* vengano soddisfatti, è necessario implementare dei test che ne verifichino l'effettiva implementazione.
Tali test verranno inseriti in questa sezione una volta individuati.

C Qualità

C.1 SPICE

Il modello ISO/IEC 1554, meglio conosciuto come SPICE (Software Process Improvement and Capability Determination), è lo standard di riferimento per valutare in modo oggettivo la qualità dei processi nello sviluppo del software.
Sono definiti:

- Una serie di sei livelli utilizzati per classificare la capacità$_G$ e la maturità$_G$ del processo software. Ogni livello è caratterizzato dal soddisfacimento degli attributi associati:
  1. **Incompleto**: il processo non è stato implementato o non ha raggiunto il successo desiderato;
  2. **Eseguito**: il processo è implementato e ha realizzato il suo obiettivo (conformità); Attributi:
     - Process performance;
  3. **Gestito**: il processo è gestito e il prodotto finale è verificato, controllato e manutenuto (affidabilità); Attributi:
     - Performance management;
     - Work product management;
  4. **Stabilito**: il processo è basato sullo standard di processo (standardizzazione); Attributi:
     - Process definition;
     - Process deployment;
  5. **Predicibile**: il processo è consistente e rispetta limiti definiti (strategico); Attributi:
     - Process measurement;
     - Process control;
  6. **Ottimizzato**: il processo segue un miglioramento continuo per rispettare tutti gli obiettivi di progetto; Attributi:
     - Process innovation;
     - Process optimization;

  Ogni attributo riceve una valutazione nella seguente scala, andando a definire il rispettivo livello di capacità del processo:
  - **N:** non raggiunto (0 - 15%);
  - **P:** parzialmente raggiunto (>15% - 50%);
  - **L:** largamente raggiunto (>50%- 85%);
  - **F:** pienamente raggiunto (>85% - 100%);
- Delle linee guida per effettuare delle **stime**, eseguite tramite:
  - **Processi di misurazione**, descritti nel *PianoDiProgetto_v1.0.0*;
  - **Modello di misurazione**, descritto in questo documento;
  - **Strumenti di misurazione**, descritti nelle *NormeDiProgetto_v1.0.0*;
- Una serie di **competenze** che chi effettua misurazioni deve possedere. La mancanza di esperienza degli elementi del gruppo *SWEight*, fa sì che nessun membro possieda queste skill, rendendo così impossibile la piena adesione allo standard. Tuttavia, ogni componente è chiamato a studiare SPICE e a applicare al meglio le indicazioni descritte in questo documento e nelle *NormeDiProgetto_v1.0.0*, al fine di perseguire un livello di qualità accettabile.

C.2 ISO/IEC 9126

Lo standard ISO/IEC 9126 stabilisce una serie di linee guida mirate al miglioramento delle qualità del software sviluppato.

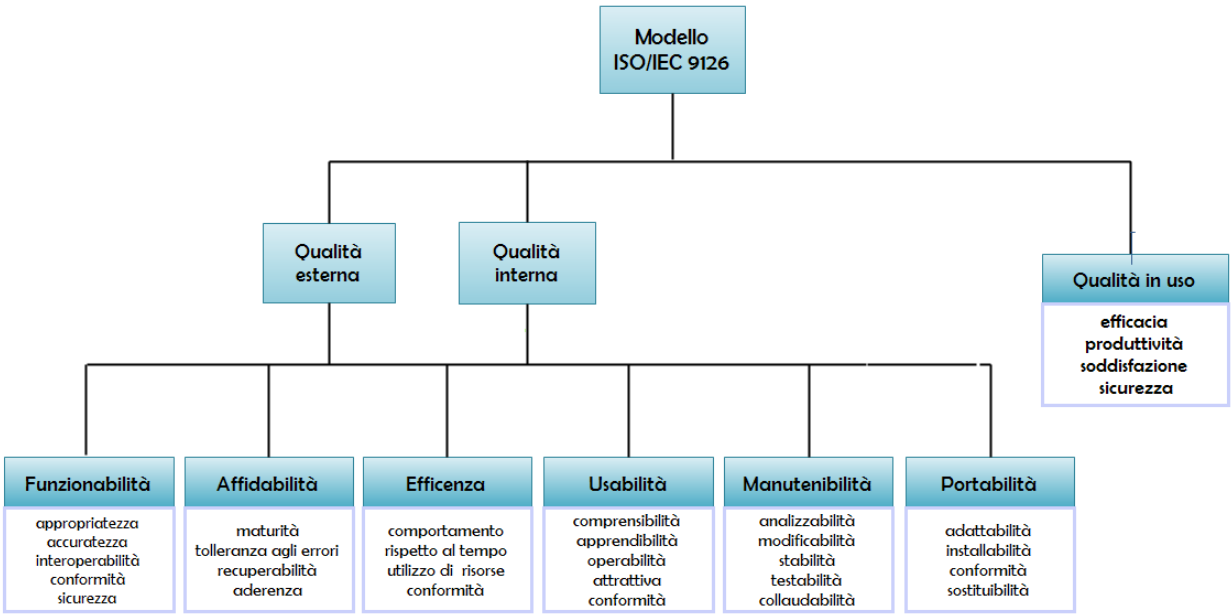

Figura 4: Rappresentazione grafica di ISO/IEC 9126 [Wikipedia]

Come presentato in Figura 4, ISO/IEC 9126 prescrive indicazioni su:

- **Qualità interna:** è misurata sul software non eseguibile, come, ad esempio il codice sorgente. Le misure effettuate (appendice §A) permettono di avere una buona previsione della qualità esterna. Le metriche usate dal gruppo *SWEight* sono reperibili in §2.4.
- **Qualità esterna:** è misurata tramite l'analisi dinamica su software eseguibile. Le misure effettuate (appendice §A) permettono di avere una buona previsione della qualità in uso prodotto. Le metriche usate dal gruppo *SWEight* sono le tecniche di analisi dinamica reperibili in §2.4.
- **Qualità in uso:** definita in base all'esperienza utente. Sono da perseguire i seguenti obiettivi:
  - Efficacia;
  - Produttività;
  - Soddisfazione;
  - Sicurezza.

ISO/IEC 9126 definisce inoltre una serie di requisiti da soddisfare per produrre software di qualità:

- **Funzionalità:** capacità di un prodotto software di soddisfare le esigenze stabilite (vedi *AnalisiDeiRequisiti_v1.0.0*);
- **Affidabilità:** capacità di un prodotto di mantenere un determinato livello di prestazioni in date condizioni d'uso per un certo periodo;
- **Efficienza:** capacità di un prodotto software di eseguire il proprio compito minimizzando il numero di risorse usate;

- **Usabilità:** capacità del prodotto software di essere utilizzato, capito e studiato dall'utente a cui è rivolto;
- **Manutenibilità:** capacità del prodotto software di evolvere mediante modifiche, correzioni e miglioramenti;
- **Portabilità:** capacità del prodotto software di funzionare ed essere installato su più ambienti hardware e software.