



Piano di qualifica

Informazioni Documento

Versione	1.0.0
Data approvazione	10 gennaio 2018
Responsabile	Matteo Rizzo
Redattori	Kevin Silvestri, Samuele Modena, Matteo Rizzo
Verificatori	Manfredi Smaniotto, Giulio Rossetti
Distribuzione	Prof. Tullio Vardanega Prof. Riccardo Cardin Gruppo Graphite
Uso	Esterno
Recapito	graphite.swe@gmail.com



Registro delle modifiche

Versione	Data	Autore	Ruolo	Modifiche
----------	------	--------	-------	-----------



Indice

1	Introduzione	4
1.1	Scopo del documento	4
1.2	Scopo del prodotto	4
1.3	Glossario	4
1.4	Riferimenti normativi	4
1.5	Riferimenti informativi	5
2	Visione generale delle strategie di gestione della qualità	6
2.1	Introduzione	6
2.2	Definizione degli obiettivi di qualità	6
2.2.1	Qualità di processo	6
2.2.1.1	Standard di riferimento	6
2.2.1.2	Obiettivi	7
2.2.2	Qualità di prodotto	7
2.2.2.1	Standard di riferimento	7
2.2.2.2	Obiettivi	8
2.3	Gestione dei processi di verifica e validazione	9
2.4	Scadenze temporali	9
2.5	Risorse	10
3	Misure e metriche	11
3.1	Introduzione	11
3.2	Metriche per i processi	11
3.2.1	ISO/IEC 15504 (SPICE)	11
3.3	Metriche per i documenti	12
3.3.1	Errori ortografici corretti	12
3.3.2	Gulpease	12
3.4	Metriche per il software	13
3.4.1	Copertura requisiti obbligatori	13
3.4.2	Copertura del codice	14
3.4.3	Percentuale superamento test	14



3.4.4	Failure avoidance	15
3.4.5	Numero di parametri per metodo	15
3.4.6	Numero di attributi per classe	15
3.4.7	Numero di metodi per classe	16
3.4.8	Complessità ciclomatica	16
3.4.9	Grado di instabilità	17
3.4.10	Altezza albero della gerarchia	18
3.4.11	Rapporto tra linee di codice e linee di commento	18
4	Gestione amministrativa della revisione	19
4.1	Definizione delle anomalie	19
4.2	Scadenze temporali	19
4.3	Risorse	20
A	Standard di qualità	21
A.1	Qualità di processo: ISO/IEC 15504	21
A.1.1	Introduzione allo standard	21
A.1.2	Classificazione dei processi	22
A.2	Qualità di prodotto: ISO/IEC 9126	23
A.2.1	Introduzione allo standard	23
A.2.2	Modello della qualità interna e esterna del software	24
A.2.3	Modello della qualità in uso del software	25
A.3	Ciclo di Deming	26
B	Resoconto delle attività di verifica	27
B.1	Riassunto delle attività di verifica per le revisioni	27
B.2	Dettaglio delle verifiche tramite analisi	27



1. Introduzione

1.1 Scopo del documento

Lo scopo del documento è quello di esporre le strategie, le tecnologie e le metriche che il gruppo Graphite adotta al fine di garantire le qualità di prodotto e di processo. Il documento ha dunque l'intento di chiarificare il *Sistema Qualità*_G instaurato e accettato dal gruppo in relazione al progetto corrente. Con l'obiettivo di rivelare e correggere in maniera efficace ed economica ogni errore, viene costantemente applicato un sistema di *verifica*_G e *validazione*_G ai processi e alle attività svolte.

1.2 Scopo del prodotto

Lo scopo del progetto è la realizzazione di un'interfaccia grafica per *Speect*_G [Meraka Institute(2008-2013)], una libreria per la creazione di sistemi di sintesi vocale, che agevoli l'ispezione del suo stato interno durante il funzionamento e la scrittura di test per le sue funzionalità.

1.3 Glossario

Al fine di evitare ogni ambiguità relativa al linguaggio utilizzato nei documenti, viene fornito il *Glossario v1.0.0*, contenente la definizione dei termini in corsivo marcati con il pedice "G".

1.4 Riferimenti normativi

- **Norme di progetto:** documento *Norme di progetto v1.0.0*.
- **Capitolato d'appalto C3:** DeSpeect: interfaccia grafica per Speect
<http://www.math.unipd.it/~tullio/IS-1/2017/Progetto/C3.pdf>



1.5 Riferimenti informativi

- **Piano di Progetto:** documento *Piano di Progetto v 1.0.0*;
- **Qualità di prodotto - Slide del corso:**
<http://www.math.unipd.it/~tullio/IS-1/2017/Dispense/L13.pdf>;
- **Qualità di processo - Slide del corso:**
<http://www.math.unipd.it/~tullio/IS-1/2017/Dispense/L15.pdf>;
- **Libro del corso:** Software Engineering - Ian Sommerville - 9 th Edition (2010);
- **Standard ISO/IEC 15504:**
https://en.wikipedia.org/wiki/ISO/IEC_15504;
- **HM&S - SPICE Process Assessment Model:**
<http://www.spice121.com/cms/en/about-spice-1-2-1.html>;
- **Standard ISO/IEC 9126:**
https://en.wikipedia.org/wiki/ISO/IEC_9126;
- **Ciclo di Deming (PDCA):**
<https://en.wikipedia.org/wiki/PDCA>;
- **Complessità ciclomatica:**
https://it.wikipedia.org/wiki/Complessit%e0_ciclomatica.



2. Visione generale delle strategie di gestione della qualità

2.1 Introduzione

Vengono di seguito illustrati gli obiettivi fissati dal gruppo al fine di garantire la qualità di processo e di prodotto. Con lo scopo di monitorare costantemente lo stato dei processi e il raggiungimento degli obiettivi definiti, il gruppo adotta standard e metriche specializzate. Segue dunque l'esposizione delle metodologie adottate, delle metriche applicate e delle rispettive scale di riferimento. Obiettivi e metriche sono univocamente identificati da un codice che ne agevola il tracciamento. La classificazione di obiettivi e metriche è descritta in dettaglio nelle *Norme di progetto v 1.0.0*.

2.2 Definizione degli obiettivi di qualità

2.2.1 Qualità di processo

2.2.1.1 Standard di riferimento

ISO/IEC 15504_G, anche noto come *SPICE_G*, è lo standard scelto per la definizione degli obiettivi di processo. Si rimanda all'appendice A.1 per un'approfondita descrizione di tale standard. La scelta di SPICE è motivata dal fatto che esso fornisce gli strumenti utili a valutare la qualità di processo, parametro da tenere in grande considerazione per il conseguimento di un prodotto qualitativamente valido entro tempi prestabiliti. Per poter applicare correttamente SPICE, viene utilizzato il *ciclo di Deming* o *ciclo di PDCA_G*. Si rimanda all'appendice A.3 per un'approfondita descrizione del ciclo di Deming. Tale ciclo definisce un metodo di controllo orientato al miglioramento continuo del livello qualitativo dei processi, evitando nel contempo regressioni. Il ciclo di Deming si applica solo conoscendo lo stato di maturità



attuale dei processi di interesse, definendo specifici obiettivi di miglioramento, e studiando i risultati delle azioni migliorative sperimentate. Esistono dunque stringenti precondizioni alla sua applicabilità, ovvero l'attuazione di processi ripetibili e misurabili, qualità di processo che il gruppo ha intenzione di perseguire. L'affiancamento dello standard ISO al ciclo PDCA permette di:

- Misurare costantemente le performance di processo;
- Perseguire un miglioramento continuo di tali performance;
- Rispettare tempi e costi indicati nel PP.

2.2.1.2 Obiettivi

- **ID:** OQP001;
- **Nome:** Miglioramento continuo;
- **Descrizione:** Capacità del processo di misurare e migliorare le proprie performance;
- **Metriche associate:** MP001: SPICE.

2.2.2 Qualità di prodotto

2.2.2.1 Standard di riferimento

*ISO/IEC 9126*_G è lo standard scelto per la definizione degli obiettivi di prodotto. Si rimanda all'appendice A.2 per un'approfondita descrizione di tale standard. La scelta di ISO/IEC 9126 è motivata dal fatto che esso definisce criteri di applicazioni nell'ambito di metriche per la qualità interna esterna e in uso del software, qui approfondite nella sezione *Misure e metriche*, utili a valutare il grado di raggiungimento degli obiettivi prefissati. I prodotti realizzati durante lo svolgimento del progetto sono di due tipologie:

- **Documentazione:** deve essere leggibile, comprensibile e corretta dal punto di vista ortografico, sintattico e semantico.
- **Software:** deve avere le seguenti caratteristiche:
 - Possedere funzionalità che soddisfino i requisiti fissati;
 - *Manutenibilità*_G ;
 - Essere ampiamente testato;
 - *Robustezza*_G ;



2.2.2.2 Obiettivi

- **ID:** OQPPD001;
 - **Nome:** Correttezza ortografica dei documenti;
 - **Descrizione:** I documenti non devono riportare errori ortografici o grammaticali;
 - **Metriche associate:** MPPD002: Errori ortografici corretti.
- **ID:** OQPPD002;
 - **Nome:** Leggibilità dei documenti;
 - **Descrizione:** I documenti devono essere leggibili e comprensibili da persone con licenza media o superiore;
 - **Metriche associate:** MPPD001: Gulpease.
- **ID:** OQPPS001;
 - **Nome:** Implementazione requisiti obbligatori;
 - **Descrizione:** Il prodotto richiesto dalla Proponente deve implementare tutti i requisiti obbligatori descritti nella AR;
 - **Metriche associate:** MPPS001: Copertura Requisiti Obbligatori.
- **ID:** OQPPS002;
 - **Nome:** Copertura del codice;
 - **Descrizione:** Il prodotto richiesto dalla Proponente deve essere testato in ogni sua parte per garantire le funzionalità relative ai requisiti;
 - **Metriche associate:** MPPS011: Linee di codice coperte dai test.
- **ID:** OQPPS003;
 - **Nome:** Superamento Test;
 - **Descrizione:** La percentuale di superamento dei test del prodotto software dovrà essere almeno l'80% del totale;
 - **Metriche associate:** MPPS013: Percentuale superamento test.
- **ID:** OQPPS004;



- **Nome:** Robustezza;
 - **Descrizione:** Il prodotto richiesto dalla proponente deve affrontare situazioni anomale senza arrestare la sua esecuzione;
 - **Metriche associate:** MPPS010: Failure Avoidance.
- **ID:** OQPPS006;
 - **Nome:** Manutenzione e comprensione del codice;
 - **Descrizione:** Il codice del prodotto richiesto dalla proponente deve essere quanto più possibile comprensibile e manutenibile;
 - **Metriche associate:**
 - * **MPPS003:** Numero di parametri per metodo;
 - * **MPPS004:** Numero di attributi per classe;
 - * **MPPS005:** Numero di metodi per classe;
 - * **MPPS006:** Complessità ciclomatica;
 - * **MPPS007:** Grado di instabilità;
 - * **MPPS008:** Altezza albero della gerarchia;
 - * **MPPS009:** Rapporto linee di commento / linee di codice.

2.3 Gestione dei processi di verifica e validazione

Il processo di verifica viene istanziato per ogni processo in esecuzione quando questo raggiunge un livello di maturità significativo, e/o in seguito a modifiche notevoli del suo stato. Per ogni processo viene verificata la qualità dello stesso e del suo esito. Ognuno dei periodi descritti nel *Piano di Progetto v1.0.0* produce degli esiti diversi, pertanto le procedure di verifica saranno specializzate e i loro risultati riportati in un'apposita appendice al termine di questo documento. Al processo di verifica segue quello di approvazione, nel quale il *Responsabile* si accerta che i risultati prodotti siano conformi con quanto atteso e accettabili dal punto di vista qualitativo.

2.4 Scadenze temporali

Le procedure di controllo che vengono attuate allo scopo di garantire il rispetto delle scadenze temporali imposte dal progetto sono descritte nelle *Norme di Progetto v1.0.0*. Ad ogni processo vengono assegnate precise date di inizio



e di fine, precedute da un ben definito lasso temporale dedicato allo studio dell'oggetto del processo stesso.

2.5 Risorse

Il controllo eseguito per garantire il livello qualitativo di processi e prodotti necessita di risorse umane e tecnologiche. In relazione a questa attività, i ruoli di maggior rilievo sono il *Responsabile* e il *Verificatore*, che rispettivamente si occupano del controllo di qualità del processo e del prodotto risultante. Una descrizione dettagliata di tali ruoli è reperibile nel documento *Norme di Progetto v1.0.0*. Le risorse tecnologiche comprendono tutti gli strumenti software e hardware che vengono utilizzati per attuare le procedure di verifica, automatizzate e non. Una descrizione dettagliata di tali risorse è reperibile nel documento *Norme di Progetto v1.0.0*.



3. Misure e metriche

3.1 Introduzione

Allo scopo di conseguire e monitorare gli obiettivi di qualità definiti, è necessario che il processo di verifica produca risultati quantificabili che sia possibile confrontare con delle costanti di riferimento. Vengono di seguito stabilite le metriche ed i valori di riferimento indicanti se i livelli qualitativi di processo e di prodotto sono in linea con gli obiettivi prefissati o meno. La presentazione di ogni metrica si articolerà nelle seguenti sottosezioni:

- **Nome:** nome descrittivo della metrica;
- **Codice:** codice univocamente identificativo della metrica;
- **Descrizione:** illustrazione della metrica e del suo significato;
- **Modalità di calcolo:** dove definito, viene riportato il modo in cui la metrica viene calcolata;
- **Range o valore minimo accettabile:** indica il range minimo (o la soglia minima), rispetto alla metrica di riferimento, in cui l'esito della verifica del processo o del prodotto deve ricadere perché si consideri garantito il livello qualitativo minimo richiesto;
- **Range o valore ottimale:** indica il range di valori (o il valore) desiderabile, rispetto alla metrica di riferimento, in cui l'esito della verifica del processo o del prodotto deve ricadere perché si consideri ampiamente garantito il livello qualitativo richiesto.

3.2 Metriche per i processi

3.2.1 ISO/IEC 15504 (SPICE)

- **Codice:** MP001;



- **Descrizione:** La metrica definita dallo standard ISO/IEC 15504 viene utilizzata alla fine di ogni periodo per monitorare e valutare la qualità dei processi impiegati. Tale standard è illustrato dettagliatamente in Appendice A.1;
- **Range di accettazione:** [Managed - Optimizing] per ogni processo;
- **Range ottimale:** [Established - Optimizing] per ogni processo.

3.3 Metriche per i documenti

3.3.1 Errori ortografici corretti

- **Codice:** OQPPD010;
- **Descrizione:** La documentazione prodotta deve essere corretta dal punto di vista ortografico e grammaticale;
- **Modalità di calcolo:** Controllo ortografico di TexStudio e del *Verificatore*;
- **Valore di accettazione:** 100% degli errori corretti;
- **Valore ottimale:** 100% degli errori corretti.

3.3.2 Gulpease

- **Codice:** OQPPD002;
- **Descrizione:** L' *indice Gulpease* G 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. Tale indice considera due variabili linguistiche: la lunghezza della parola e la lunghezza della frase rispetto al numero delle lettere;
- **Modalità di calcolo:** L'indice Gulpease si calcola con la seguente formula:

$$I_{\text{Gulpease}} = 89 + \frac{(300 * (\text{numero delle frasi}) - 10 * (\text{numero delle lettere}))}{(\text{numero delle parole})}$$



Il risultato è un valore compreso nell'intervallo tra 0 e 100, dove il valore 100 indica la più alta leggibilità, mentre 0 la più bassa. In generale risulta che testi con indice:

- inferiore a 80 sono difficili da leggere per chi ha la licenza elementare;
- inferiore a 60 sono difficili da leggere per chi ha la licenza media;
- inferiore a 40 sono difficili da leggere per chi ha un diploma superiore.

Nonostante l'importanza riconosciuta all'indice Gulpease, il gruppo tiene conto delle seguenti considerazioni:

- l'indice non rileva la comprensibilità del testo, il cui contenuto potrebbe essere totalmente incomprensibile ma ottenere ugualmente un buon valore;
- ai fini dei documenti vengono necessariamente usati termini tecnici possibilmente ostici ma insostituibili;
- il gruppo decide di prediligere la chiarezza e la precisione del contenuto dei documenti anche qualora questa dovesse minarne (in modo superficiale) la leggibilità.

- **Range di accettazione:** [55, 100];
- **Range ottimale:** [65, 100]

3.4 Metriche per il software

3.4.1 Copertura requisiti obbligatori

- **Codice:** MPSP001;
- **Descrizione:** La copertura dei requisiti obbligatori permette di monitorare la percentuale di requisiti obbligatori soddisfatti;
- **Modalità di calcolo:** La copertura dei requisiti obbligatori si calcola con la seguente formula:

$$CRO = \frac{\text{numero requisiti obbligatori soddisfatti}}{\text{totale dei requisiti obbligatori}}$$

- **Range di accettazione:** [65% - 100%];



- **Range ottimale:** [80% - 100%].

3.4.2 Copertura del codice

- **Codice:** OQPPS002;
- **Descrizione:** La copertura del codice (o code coverage) indica la percentuale di istruzioni, rispetto al totale, che vengono eseguite durante i test. Maggiore è il numero delle istruzioni testate e maggiore è la possibilità di individuare e risolvere errori di codifica. Un valore troppo basso indica che molte istruzioni non vengono testate, divenendo possibile causa di anomalie.
- **Modalità di calcolo:** La copertura del codice si calcola con la seguente formula:

$$CC = \frac{\text{numero di righe di codice testate}}{\text{totale delle righe di codice}}$$

- **Range di accettazione:** [65% - 100%];
- **Range ottimale:** [85% - 100%].

3.4.3 Percentuale superamento test

- **Codice:** MPPS013;
- **Descrizione:** La metrica indica la percentuale di test implementati superati dal oggetto del test;
- **Modalità di calcolo:** Percentuale superamento test si calcola con la seguente formula:

$$PST = \frac{\text{numero test superati}}{\text{totale dei test implementati}}$$

- **Range di accettazione:** [85% - 100%];
- **Range ottimale:** [100% - 100%].



3.4.4 Failure avoidance

- **Codice:** MPPS010;
- **Descrizione:** La Failure Avoidance è una metrica applicata per monitorare la robustezza del prodotto quando deve affrontare a situazioni critiche;
- **Modalità di calcolo:** La Failure Avoidance si calcola con la seguente formula:

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

- **Range di accettazione:** [80% - 100%];
- **Range ottimale:** [90% - 100%].

3.4.5 Numero di parametri per metodo

- **Codice:** MPPS003;
- **Descrizione:** Il numero di parametri di un metodo incide sulla sua complessità. Un valore elevato può indicare che esso si fa carico di una quantità eccessiva di responsabilità e che potrebbe essere decomposto in metodi più semplici;
- **Modalità di calcolo:** controllo del *Verificatore*;
- **Range di accettazione:** [0, 5];
- **Range ottimale:** [0, 3].

3.4.6 Numero di attributi per classe

- **Codice:** MPPS004;
- **Descrizione:** Il numero di attributi di una classe incide sulla sua complessità. Un valore elevato può indicare che essa si fa carico di una quantità eccessiva di responsabilità e che potrebbe essere decomposta in classi più semplici;
- **Modalità di calcolo:** controllo del *Verificatore*;
- **Range di accettazione:** [0, 15];
- **Range ottimale:** [0, 8].



3.4.7 Numero di metodi per classe

- **Codice:** MPPS005;
- **Descrizione:** Il numero di metodi di una classe incide sulla sua complessità. Un valore elevato può indicare che essa si fa carico di una quantità eccessiva di responsabilità e che potrebbe essere decomposta in classi più semplici;
- **Modalità di calcolo:** controllo del *Verificatore*;
- **Range di accettazione:** $[0, 15]$;
- **Range ottimale:** $[0, 5]$.

3.4.8 Complessità ciclomatica

- **Codice:** MPPS006;
- **Descrizione:** La *Complessità ciclomatica* G (o complessità condizionale) è una metrica software utilizzata per misurare la complessità di un programma. Nello specifico, essa stima la complessità di funzioni, moduli, metodi o classi di un programma. Il valore della Complessità Ciclomatica rappresenta quanto un metodo è complesso, tramite la misura del numero di cammini linearmente indipendenti che attraversano il *grafo* G di flusso di controllo. In tale grafo, i nodi rappresentano gruppi indivisibili di istruzioni. Un arco connette due nodi se le istruzioni di uno dei nodi possono essere eseguite direttamente dopo l'esecuzione delle istruzioni dell'altro nodo. Durante l'identificazione dei test, la Complessità Ciclomatica è inoltre utile per determinare il numero di casi di test necessari: l'indice di Complessità Ciclomatica è infatti un limite superiore al numero di test necessari per raggiungere la completa *copertura del codice* G (o *code coverage*) del modulo testato. Un valore troppo elevato indica un'eccessiva complessità del codice, cui consegue una complessa manutenzione, al contrario un valore ridotto potrebbe indicare una scarsa efficienza dei metodi;
- **Modalità di calcolo:** Rappresentando un programma con il grafo di controllo del flusso, un modo di calcolare il numero ciclomatico $V(G)$ è il seguente:

$$V(G) = E - N + 2P$$

Dove:



- **N:** indica il numero di nodi del grafo;
- **E:** indica il numero di archi del grafo;
- **P:** indica il numero di componenti connesse del grafo;
- **Range di accettazione:** $[0, 15]$;
- **Range ottimale:** $[0, 10]$.

3.4.9 Grado di instabilità

- **Codice:** MPPS007;
- **Descrizione:** Il livello di stabilità di un software indica il rischio con una modifica apportata ad un *package* G influenzi il funzionamento dell'intero sistema;
- **Modalità di calcolo:** Il valore di instabilità è calcolato usando il grado di *accoppiamento* G tramite la seguente formula:

$$Instabilità = \frac{Accoppiamento\ efferente}{Accoppiamento\ efferente + Accoppiamento\ afferente}$$

Dove:

- **Accoppiamento efferente:** indica il grado di dipendenza delle classi di un package verso classi di package esterni. Un valore basso garantisce la stabilità del package indipendentemente dalle possibili modifiche al resto del sistema;
- **Accoppiamento afferente:** indica il grado di utilità delle classi di un package verso classi esterne ad esso. Un valore troppo basso potrebbe essere indice di scarsa utilità della classe, dato che poche delle sue funzionalità sono usate al suo esterno. Al contrario, un valore troppo elevato può indicare un livello di dipendenza pericoloso del package in esame, che può portare ad effetti indesiderati nelle classi esterne in caso di modifiche. Un valore elevato, d'altro canto, non è necessariamente indice di un errore di progettazione bensì potrebbe indicare semplicemente la criticità del package in esame.
- **Range di accettazione:** $[0.0 - 0.8]$;
- **Range ottimale:** $[0.0 - 0.4]$.



3.4.10 Altezza albero della gerarchia

- **Codice:** MPPS008;
- **Descrizione:** L'altezza degli alberi di gerarchia delle classi va limitata così da limitare l'accoppiamento. Preferibilmente, le classi dovranno dipendere solo da classi astratte e potranno implementare una o più interfacce. Viene invece proibito l'uso di ereditarietà multipla.
- **Modalità di calcolo:** controllo del *Verificatore*;
- **Range di accettazione:** $\text{rapporto} \geq 0.3\%$;
- **Range ottimale:** $\text{rapporto} \geq 0.5\%$.

3.4.11 Rapporto tra linee di codice e linee di commento

- **Codice:** MPPS009;
- **Descrizione:** Il rapporto tra linee di codice (escluse le righe vuote) e linee di commento rappresenta un indice utile a stimare la manutenibilità del codice sorgente. Un rapporto troppo basso indica una scarsa documentazione del codice scritto, cui consegue una possibile elevata complessità nel mantenerlo.
- **Modalità di calcolo:**

$$\frac{\text{numero linee di commento}}{\text{numero linee di codice (non vuote)}}$$

- **Range di accettazione:** $\text{rapporto} \geq 0.3\%$;
- **Range ottimale:** $\text{rapporto} \geq 0.5\%$.



4. Gestione amministrativa della revisione

4.1 Definizione delle anomalie

L'identificazione delle anomalie è finalizzata alla loro risoluzione e rappresenta un importante dato per il monitoraggio dello stato del prodotto. Distinguere e catalogare le anomalie permette di organizzare (in particolar modo di priorizzare) e affinare le correzioni da attuare per eliminarle. Di seguito vengono quindi elencate le definizioni di anomalie (secondo glossario IEEE 610.12-90) adottate dal gruppo:

- **Error:** differenza riscontrata tra risultato di una computazione e valore teorico atteso;
- **Fault:** un passo, un processo o un dato definito in modo erroneo che corrisponde a quanto viene definito come bug;
- **Failure:** il risultato di un fault;
- **Mistake:** azione umana che produce un risultato errato.

4.2 Scadenze temporali

Vista la presenza delle scadenze temporali definite nel PP, si necessita di un sistema di controllo efficiente dei tempi. Le procedure di controllo che verranno attuate per individuare e correggere eventuali errori sono descritte nelle NP. Nel tentativo di prevenire l'insorgenza di errori stessi, ogni attività svolta detiene un periodo iniziale di studio sull'argomento, che riduce la quantità di interventi correttivi a posteriori.



4.3 Risorse

Il controllo effettuato per garantire il livello di qualità di processi e prodotti necessita dell'uso di risorse umane e tecnologiche. I ruoli di maggiore impatto per questa attività sono il ruolo di *Responsabile* e di *Verificatore*, rispettivamente adibiti al controllo di qualità di processo e al controllo di qualità di prodotto. Una descrizione dettagliata di questi e altri ruoli si trova nelle NP. Le risorse tecnologiche comprendono tutti gli strumenti software e hardware che vengono utilizzati per attuare le procedure di verifica. Una descrizione dettagliata di queste e altre tecnologie si trova nelle NP.



A. Standard di qualità

A.1 Qualità di processo: ISO/IEC 15504

A.1.1 Introduzione allo standard

Il modello ISO/IEC 15504, anche noto come SPICE (acronimo di Software Process Improvement and Capability Determination, dove per *capability* si intende la capacità intesa come abilità di un processo nel raggiungere un obiettivo) è lo standard di riferimento per la valutazione oggettiva della qualità dei processi software e permette la misurazione indipendente della capacità di ogni processo tramite la classificazione di alcuni attributi, eseguita previo studio del range di risultati che la sua esecuzione restituisce. Perché possano contribuire al miglioramento dei processi, le singole valutazioni devono essere ripetibili, oggettive e fornire esiti comparabili. Gli attributi associati alle capacità di ogni processo sono:

- **Process performance:** indica in quale misura sono raggiunti gli obiettivi fissati;
- **Performance management:** indica il grado di organizzazione con cui sono raggiunti gli obiettivi fissati;
- **Work product management:** indica in quale misura i prodotti sono gestiti correttamente per quanto riguarda documentazione, controllo e verifica;
- **Process definition:** indica in quale misura il processo si appoggia agli standard;
- **Process distribution:** indica in quale misura il processo standard viene effettivamente rilasciato e distribuito come un processo definito in grado di raggiungere sempre gli stessi risultati;



- **Process measurement:** indica il grado in cui i risultati delle misure sono utilizzati per garantire che il processo raggiunga i suoi obiettivi;
- **Process control:** indica in quale misura il processo risulta stabile, capace e predicibile (entro certi limiti);
- **Process change:** indica in quale misura le modifiche da apportare al processo sono identificate grazie ad una fase di analisi delle performance e allo studio di approcci innovativi;
- **Process improvement:** indica in quale misura i cambiamenti all'organizzazione, alle performance e alla definizione del processo hanno un impatto effettivo che porta a raggiungere importanti obiettivi di miglioramento al processo.

A.1.2 Classificazione dei processi

Gli attributi vengono misurati e classificati secondo uno dei seguenti livelli:

- **N - not implemented:** il processo non possiede l'attributo o dimostra gravi carenze in merito;
- **P - partially implemented:** esiste un approccio sistematico volto al possesso di un attributo già parzialmente ottenuto, ma alcuni aspetti non sono ancora prevedibili;
- **L - largely implemented:** esiste un approccio sistematico volto al possesso di un attributo già significativamente ottenuto, ma l'attuazione varia nelle diverse unità;
- **F - fully implemented:** l'attributo è stato completamente conseguito grazie ad un approccio sistematico e l'attuazione è uguale in tutte le unità.

Secondo la classificazione degli attributi, ad un processo viene assegnato uno dei seguenti livelli di capacità:

- **Incomplete:** il processo è incompleto in quanto non è stato implementato, o fallisce nel raggiungere il proprio obiettivo. Questo livello non ha alcun attributo associato;
- **Performed:** il processo è stato implementato e ha successo nel raggiungere il proprio obiettivo. L'attributo associato a questo livello è *process performance*;



- **Managed:** il processo, che già apparteneva al livello *performed*, è implementato in maniera organizzata tramite pianificazione, controllo e correzione; i suoi prodotti sono sicuri. Gli attributi associati a questo livello sono *performance management* e *work product management*;
- **Established:** il processo, che già apparteneva al livello *managed*, è stato implementato come processo definito in grado di raggiungere sempre gli stessi risultati. Gli attributi associati a questo livello sono *process definition* e *process distribution*;
- **Predictable:** il processo, che già apparteneva al livello *established*, opera entro limiti definiti per raggiungere i propri risultati. Gli attributi associati a questo livello sono *process control* e *process measurement*;
- **Optimizing:** il processo, che già apparteneva al livello *predictable*, è oggetto di miglioramento continuo per raggiungere gli obiettivi di progetto. Gli attributi associati a questo livello sono *process change* e *process improvement*.

A.2 Qualità di prodotto: ISO/IEC 9126

A.2.1 Introduzione allo standard

La sigla ISO/IEC 9126 individua una serie di normative e linee guida preposte a descrivere un modello di qualità del software. Nello specifico, esso definisce un modello (costituito da metriche qualitative che possono essere misurate in termini quantitativi) per:

- **Qualità interna:** la qualità interna definisce metriche applicabili al codice sorgente utili a rilevarvi problemi che ne possano inficiare la qualità prima che il software venga eseguito. Essa viene rilevata tramite analisi statica e, idealmente, determina la qualità esterna;
- **Qualità esterna:** la qualità esterna definisce metriche applicabili al software in esecuzione utili a valutarne i comportamenti tramite test, rispetto agli obiettivi stabiliti. Essa viene rilevata tramite analisi dinamica e, idealmente, determina la qualità in uso;
- **Qualità in uso** la qualità in uso definisce metriche applicabili al solo prodotto finito e calato in reali condizioni di utilizzo.



A.2.2 Modello della qualità interna e esterna del software

- **Funzionalità:** il software è tenuto a fornire funzionalità atte a soddisfare i bisogni evidenziati nell'*Analisi dei Requisiti*, e che permettano di operare nel *dominio applicativo* G desiderato. Nello specifico, esso deve avere le seguenti caratteristiche:
 - **Appropriatezza:** ovvero la capacità di fornire funzionalità appropriate in relazione ad attività specifiche, e che permettano di raggiungere gli obiettivi fissati;
 - **Accuratezza:** ovvero la capacità di fornire risultati corretti con la precisione richiesta;
 - **Interoperabilità:** ovvero la capacità di interagire con dati sistemi;
 - **Sicurezza:** ovvero la capacità di proteggere informazioni e dati.
- **Affidabilità:** il software è tenuto a mantenere un livello di prestazioni quando utilizzato in condizioni date situazioni critiche. Nello specifico, esso deve avere le seguenti caratteristiche:
 - **Maturità:** ovvero la capacità di evitare errori durante l'esecuzione;
 - **Robustezza:** ovvero la capacità di mantenere uno stato funzionante anche in caso di errori;
 - **Recuperabilità:** ovvero la capacità di ripristinare prestazioni e dati in caso di errori o malfunzionamenti.
- **Efficienza:** il software è tenuto a eseguire le proprie funzionalità minimizzando tempo, spazio e tutte le altre risorse di cui necessita per il suo corretto funzionamento;
- **Usabilità:** il software è tenuto ad essere comprensibile, studiabile e pienamente utilizzabile dal suo *target* G di utenza. Nello specifico, esso deve avere le seguenti caratteristiche:
 - **Comprensibilità:** ovvero la capacità di essere inequivocabilmente chiaro rispetto alle proprie funzionalità e modalità di utilizzo;
 - **Apprendibilità:** ovvero la capacità di rendere palesi, studiabili e dunque apprendili le proprie applicazioni;



- **Operabilità:** ovvero la capacità di essere pienamente utilizzabile e sotto il controllo dell'utente;
- **Attrattiva:** ovvero la capacità di risultare interessante, utile e attraente nei confronti dell'utente.
- **Manutenibilità:** il software deve essere in grado di evolvere sulla base di modifiche, correzioni e adattamenti. Nello specifico, esso deve avere le seguenti caratteristiche:
 - **Analizzabilità:** ovvero la capacità di essere analizzato agevolmente al fine di individuarne errori;
 - **Modificabilità:** ovvero la capacità di essere modificato agevolmente a livello di codice, progettazione o documentazione;
 - **Stabilità:** ovvero la capacità di evitare effetti indesiderati in seguito ad una modifica;
 - **Testabilità:** ovvero la capacità di poter essere agevolmente verificato e validato.
- **Portabilità:** il software deve poter essere trasportato da un ambiente hardware o software ad un altro, seguendo le evoluzioni tecnologiche. Nello specifico, esso deve avere le seguenti caratteristiche:
 - **Adattabilità:** ovvero la capacità di adattarsi a differenti ambienti senza la necessità di azioni specifiche;
 - **Installabilità:** ovvero la capacità di essere installato in un dato ambiente;
 - **Conformità:** ovvero la capacità di coesistere con altre applicazioni e di condividere efficientemente le risorse;
 - **Sostituibilità:** ovvero la capacità di sostituire un altro software, che abbia lo stesso scopo, nello stesso ambiente.

A.2.3 Modello della qualità in uso del software

Il software è tenuto a permettere agli utenti di conseguire obiettivi specifici con:

- **Efficacia:** il software deve effettivamente permettere agli utenti di raggiungere l'obiettivo fissato;



- **Produttività:** il software deve utilizzare in maniera efficiente le risorse a lui necessarie;
- **Soddisfazione:** il software deve soddisfare i bisogni degli utenti;
- **Sicurezza:** il software deve detenere livelli di rischio accettabili rispetto a danni nei confronti di persone, apparecchiature e ambiente operativo.

A.3 Ciclo di Deming

Il ciclo di Deming (anche conosciuto come ciclo PDCA, l'acronimo di Plan-Do-Check-Act) è un metodo iterativo utilizzato per il controllo dei processi finalizzato al miglioramento continuo della loro qualità e, conseguentemente, della qualità dei prodotti. Ogni iterazione del ciclo consiste di quattro fasi:

1. **Plan:** la fase di pianificazione degli obiettivi di miglioramento. Qui vengono definite le attività da svolgere, le risorse da assegnarvi e le scadenze utili allo scopo di raggiungere tali obiettivi;
2. **Do:** la fase in cui ciò che è stato precedentemente pianificato viene messo in atto;
3. **Check:** la fase di verifica in cui si accerta che la fase *Do* sia stata eseguita rispettando la fase *Plan* e che abbia ottenuto esiti positivi secondo date metriche;
4. **Act:** la fase di attuazione, in cui i processi che hanno beneficiato delle correzioni e delle modifiche eseguite vengono resi standard.



B. Resoconto delle attività di verifica

B.1 Riassunto delle attività di verifica per le revisioni

Nel periodo precedente alla consegna per la Revisione dei Requisiti vengono verificati i documenti redatti ed i processi eseguiti. I documenti sono verificati dai *Verificatori* secondo i criteri per l'analisi statica definiti nel documento *Norme di Progetto v1.0.0*, applicando il sistema *Walkthrough_G* ed *Inspection_G*. In primo luogo, viene verificato il documento nella sua interezza, cercando eventuali errori presenti e trattandoli nel modo seguente:

1. Correzione di errori grammaticali o di eventuali violazioni delle norme tipografiche definite nelle *Norme di Progetto v 1.0.0*;
2. Segnalazione ed aggiunta alla lista di controllo degli errori più frequenti;
3. Applicazione del ciclo PDCA allo scopo di migliorare e velocizzare le future verifiche.

In secondo luogo, viene applicato il metodo *Inspection* mediante l'uso della lista di controllo stilata sulla base dei documenti già sottoposti a verifica, con particolare enfasi sugli errori più comuni.

Il tracciamento dei requisiti viene effettuato tramite il software *SWEgo_G* e successivamente controllato manualmente per assicurarne la correttezza. Vengono infine controllati i documenti ed il processo stesso ponendo attenzione sul rispetto delle metriche proposte in questo documento, come approfondito nella sezione seguente.

B.2 Dettaglio delle verifiche tramite analisi