



Piano di Qualifica

Gruppo JurassicSWE · Progetto IronWorks

JurassicSWE@gmail.com

Informazioni sul documento

Versione	1.0.0
Redazione	Francesco Minna, Gianluca Travasci
Verifica	Leo Moz, Lidia Alecci
Approvazione	Marco Masiero
Uso	Esterno
Distribuzione	Prof. Tullio Vardanega Prof. Riccardo Cardin gruppo JurassicSWE

Sommario

Tale documento descrive le operazioni di *verifica_g* e *validazione_g* seguite dal gruppo JurassicSWE relative al progetto IronWorks.

Registro delle modifiche

Versione	Data	Ruolo	Nominativo	Descrizione
1.0.0	2018-04-12	Responsabile	Marco Masiero	Approvazione del documento
0.2.0	2018-04-12	Verificatore	Lidia Alecci	Verifica del documento
0.1.1	2018-04-11	Verificatore	Francesco Minna	Stesura sezione C.2.1
0.1.0	2018-03-26	Verificatore	Leo Moz	Verifica del documento
0.0.7	2018-03-25	Analista	Francesco Minna	Stesura Appendice C
0.0.6	2018-03-24	Verificatore	Gianluca Travasci	Stesura sezione 3
0.0.5	2018-03-23	Verificatore	Francesco Minna	Stesura Appendici A e B
0.0.4	2018-03-20	Analista	Gianluca Travasci	Stesura sezione 2.3
0.0.3	2018-03-19	Verificatore	Gianluca Travasci	Stesura sezioni 2.1 e 2.2
0.0.2	2018-03-18	Analista	Francesco Minna	Stesura sezione Introduzione
0.0.1	2018-03-16	Responsabile	Francesco Minna	Creazione template e stesura dell'indice.



Indice

1	Introduzione	5
1.1	Scopo del documento	5
1.2	Scopo del prodotto	5
1.3	Glossario	5
1.4	Riferimenti	5
1.4.1	Riferimenti normativi	5
1.4.2	Riferimenti informativi	6
2	Strategie di gestione della qualità	7
2.1	Definizione degli obiettivi	7
2.1.1	Qualità di processo	7
2.1.2	Qualità di prodotto	7
2.1.3	Tabella degli obiettivi	8
2.2	Metriche e misure	9
2.2.1	Tabella delle metriche	10
2.3	Scadenze temporali	12
3	Gestione amministrativa della revisione	13
3.1	Misure e metriche in dettaglio	13
3.1.1	Metriche per i processi	13
3.1.2	Metriche per la documentazione	13
3.1.3	Metriche per il software	14
3.2	Comunicazione e risoluzione delle anomalie	17
A	Standard di qualità	18
A.1	ISO/IEC 15504	18
A.2	ISO/IEC 9126	20
A.3	Ciclo di Deming	23
B	Valutazioni per il miglioramento	24
B.1	Valutazioni sull'organizzazione	25
B.2	Valutazioni sui ruoli	26
B.3	Valutazioni sugli strumenti	27
C	Resoconto delle attività di verifica	28
C.1	Riassunto delle attività di verifica per le revisioni	28
C.2	Dettaglio delle verifiche tramite analisi	29
C.2.1	Periodo di Analisi dei requisiti di massima	29

Elenco delle figure

1	Standard ISO/IEC 15504	19
2	Modello ISO/IEC 9126	20
3	Fasi del Ciclo di Deming	23
4	Esempio ciclo di Deming	23

Elenco delle tabelle

1	Tabella degli obiettivi e delle metriche.	9
2	Tabella delle metriche utilizzate	11
3	Problemi e relative soluzioni riscontrati nell' <i>organizzazione</i>	25
4	Problemi e relative soluzioni riscontrati nei <i>ruoli</i>	26
5	Problemi e relative soluzioni riscontrati nell'uso degli <i>strumenti</i>	27
6	Valore indice di Gulpease nel periodo di <i>Analisi dei requisiti di massima</i>	29
7	Valore della schedule variance nel periodo di <i>Analisi dei requisiti di massima</i> . .	30
8	Valore del Cost variance nel periodo di <i>Analisi dei requisiti di massima</i>	30

1 Introduzione

1.1 Scopo del documento

Tale documento ha lo scopo di illustrare le strategie di verifica e validazione adottate dal gruppo JurassicSWE al fine di garantire la qualità di *prodotto_g* e di *processo_g*.

Per raggiungere tale obiettivo viene applicato un sistema di verifica continua sui processi e sulle attività svolte; in questo modo è possibile rilevare e correggere all'istante eventuali anomalie, riducendo al minimo lo spreco di risorse.

1.2 Scopo del prodotto

Lo scopo del prodotto è quello di realizzare un software, in particolare un'*applicazione web_g*, per disegnare diagrammi *UML_g* di robustezza, e di un generatore di codice che, a partire dalle definizioni contenute in un diagramma, produca il codice di classi *Java_g* per ospitare i dati delle entità persistenti, ed i metodi per leggere e scrivere questi dati in un *database relazionale_g*.

1.3 Glossario

Al fine di evitare ambiguità, i termini che possono essere interpretati in modi diversi a seconda del contesto, o che necessitano di una descrizione approfondita, sono scritti in *corsivo* con una "g" pedice, solo alla loro prima occorrenza.

La definizione di tali termini è contenuta nel documento *Glossario v1.0.0*.

1.4 Riferimenti

1.4.1 Riferimenti normativi

- **Norme di Progetto:** *Norme di Progetto v1.0.0*;
- *Capitolato_g*
<http://www.math.unipd.it/~tullio/IS-1/2017/Progetto/C5.pdf> (consultato il 2018-04-06);
- *Standard_g ISO/IEC 15504*
http://en.wikipedia.org/wiki/ISO/IEC_15504 (consultato il 2018-04-06);
- *Standard ISO/IEC 9126*
http://en.wikipedia.org/wiki/ISO/IEC_9126 (consultato il 2018-04-06);

- **Standard IEEE 610.12-90**
https://cow.ceng.metu.edu.tr/Courses/download_courseFile.php?id=2677 (consultato il 2018-04-06).

1.4.2 Riferimenti informativi

- **Piano di Progetto:** *Piano di Progetto v1.0.0*;
- **Qualità del software** - Slide del corso di Ingegneria del Software
<http://www.math.unipd.it/~tullio/IS-1/2017/Dispense/L13.pdf> (consultato il 2018-04-06);
- **SWEBOK 2004 Version - capitolo 11**
<http://www.computer.org/portal/web/swebok/htmlformat> (consultato il 2018-04-06);
- **Qualità di processo** - Slide del corso di Ingegneria del Software
<http://www.math.unipd.it/~tullio/IS-1/2017/Dispense/L15.pdf> (consultato il 2018-04-06);
- **Ciclo di Deming (PDCA)**
https://it.wikipedia.org/wiki/Ciclo_di_Deming (consultato il 2018-04-06);
- **Indice Gulpease**
https://it.wikipedia.org/wiki/Indice_Gulpease (consultato il 2018-04-06).

2 Strategie di gestione della qualità

2.1 Definizione degli obiettivi

In questo paragrafo verranno illustrati gli obiettivi fissati dal gruppo JurassicSWE al fine di garantire la qualità di processo e di prodotto nella realizzazione di "IronWorks: utilità per la costruzione di software robusto".

Con l'obiettivo di controllare periodicamente lo stato e il raggiungimento degli obiettivi sono stati adottati *standard_g* e *metriche_g* consone.

Sia gli obiettivi che le metriche sono individuati da un codice alfanumerico che li rende facilmente tracciabili e quindi controllabili. La classificazione di obiettivi e metriche è descritta in dettaglio nelle *Norme di Progetto v1.0.0 _ 3.1.5.6* - "Notazione per obiettivi e metriche di qualità".

2.1.1 Qualità di processo

Per rispettare le scadenze prefissate e per ottenere un prodotto valido il gruppo JurassicSWE ha optato per l'utilizzo dello standard ISO/IEC 15504 che consente la definizione degli obiettivi di processo. Tale standard, chiamato anche *SPICE_g*, consente di valutare la qualità di processo ed è dettagliatamente descritto nell'appendice A.1.

Per poter applicare correttamente SPICE si farà uso del *ciclo di Deming_g*, conosciuto anche come ciclo PDCA: tale ciclo definisce un metodo di controllo mirato al miglioramento continuo del livello di qualità di processo evitando possibili regressioni.

L'uso di SPICE e del ciclo PDCA permettono ai processi di:

- avere performance costantemente misurabili;
- perseguire un miglioramento continuo delle performance;
- rispettare tempi e costi definiti nel *Piano di Progetto v1.0.0*.

2.1.2 Qualità di prodotto

Per delineare e definire tali obiettivi, il gruppo JurassicSWE ha deciso di aderire allo standard ISO/IEC 9126, descritto nell'appendice A.2.

Questo standard definisce i criteri di applicazione delle metriche utilizzate per la valutazione del livello di raggiungimento degli obiettivi riportati nella tabella sottostante.

I prodotti realizzati si dividono in due tipologie:

- documenti, che devono essere leggibili, comprensibili e corretti sotto tutti i punti di vista;
- software, che dovrà essere:
 - comprensibile, ben commentato e manutenibile;

- ampiamente testato;
- robusto, cioè in grado di far fronte a situazioni critiche senza arrestarsi.

2.1.3 Tabella degli obiettivi

ID	Nome	Descrizione	Metriche
OPC1	Coerenza con il <i>Piano di Progetto v1.0.0</i>	Il lavoro del team deve rispettare quanto stabilito nel documento del <i>Piano di Progetto v1.0.0</i>	MPC1: Schedule Variance MPC2: Cost Variance
OPC2	Miglioramento continuo	Capacità del processo di misurare e migliorare le proprie performance	MPC3: SPICE
OPDD1	Leggibilità dei documenti	I documenti devono essere chiari, leggibili e comprensibili	MPDD1: Indice Gulpase
OPDD2	Correttezza ortografica	I documenti non devono presentare errori grammaticali e ortografici	MPDD2: Correzione errori ortografici
PODS1	Implementazione requisiti obbligatori	Il prodotto finale dovrà implementare tutti i requisiti obbligatori descritti nella <i>Analisi dei Requisiti v1.0.0</i>	MPDS1: Requisiti obbligatori soddisfatti
PODS2	Implementazione requisiti accettati	Il prodotto finale dovrà implementare parte dei requisiti accettati descritti nella <i>Analisi dei Requisiti v1.0.0</i>	MPDS2: Requisiti accettati soddisfatti
OPDS3	Superamento test	Il prodotto finale dovrà avere una percentuale di superamento dei test almeno del 80% del totale	MPDS3: percentuale superamento test

ID	Nome	Descrizione	Metriche
OPDS4	Manutenibilità e usabilità	Il codice deve essere il più possibile comprensibili e manutenibile	MPDS4: Numero parametri per metodo MPDS5: Numero attributi per metodo MPDS6: Grado di accoppiamento MPDS7: Complessità ciclomatica MPDS8: Numero metodi per classe MPDS9: Numero di classi per <i>package_g</i> MPDS10: Test Automatici MPDS11: Rapporto linee di commento/linee di codice
OPDS5	Affidabilità	Il prodotto finale dovrà poter gestire situazioni anomale senza arrestarsi	MPDS12: Failure avoidance

Tabella 1: Tabella degli obiettivi e delle metriche.

2.2 Metriche e misure

Verranno di seguito stabilite le metriche ed i valori minimi necessari che indicano se i livelli qualitativi di processo e di prodotto sono in linea con gli obiettivi prefissati.

Ogni metrica rappresentata nella tabella sottostante consisterà di quattro elementi principali:

- **ID:** valore alfanumerico che identifica la metrica;
- **Nome:** nome che rappresenta la metrica;
- **Soglie di accettabilità:** verranno esposti due valori, quello minimo e quello ottimale;
- **Obiettivo:** obiettivo al quale la metrica si riferisce.

2.2.1 Tabella delle metriche

ID	Nome	Soglie di accettabilità	Obbiettivi
MPC1	Schedule Variance	Valore minimo: ≤ 3 Valore ottimale: ≤ 0	OPC1: Coerenza con il <i>Piano di Progetto v1.0.0</i> .
MPC2	Cost Variance	Valore minimo: $\leq 8\%$ Valore ottimale: $\leq 1\%$	OPC1: Coerenza con il <i>Piano di Progetto v1.0.0</i> .
MPC3	SPICE	Valore minimo: livello 2 Valore ottimale: \geq livello 4	OPC1: Miglioramento continuo
MPDD1	Indice Gulpease	Valore minimo: ≥ 40 Valore ottimale: ≥ 50	OPDD1: Leggibilità dei documenti
MPDD2	Correzione errori ortografici	Valore minimo: 100% corretti Valore ottimale: 100% corretti	OPDD2: Correttezza ortografica
MPDS1	Requisiti obbligatori soddisfatti	Valore minimo: 100% Valore ottimale: 100%	PODS1: Implementazione requisiti obbligatori
MPDS2	Requisiti accettati soddisfatti	Valore minimo: 40% Valore ottimale: $\geq 60\%$	PODS2: Implementazione requisiti accettati
MPDS3	Percentuale superamento test	Valore minimo: $\geq 80\%$ Valore ottimale: $\geq 100\%$	OPDS3: Superamento test
MPDS4	Numero parametri per metodo	Valore minimo: ≤ 5 Valore ottimale: ≤ 3	OPDS4: Manutenibilità e usabilità

ID	Nome	Soglie di accettabilità	Obbiettivi
MPDS5	Numero attributi per metodo	Valore minimo: tra 0 e 15 Valore ottimale: tra 0 e 8	OPDS4: Manutenibilità e usabilità
MPDS6	Grado di accoppiamento	Valore minimo: ≤ 10 Valore ottimale: ≤ 3	OPDS4: Manutenibilità e usabilità
MPDS7	Complessità ciclomatica	Valore minimo: ≤ 20 Valore ottimale: ≤ 10	OPDS4: Manutenibilità e usabilità
MPDS8	Numero metodi per classe	Valore minimo: tra 0 e 15 Valore ottimale: tra 0 e 5	OPDS4: Manutenibilità e usabilità
MPDS9	Numero di classi per package	Valore minimo: ≤ 6 Valore ottimale: ≤ 4	OPDS4: Manutenibilità e usabilità
MPDS10	Test automatici	Valore minimo: $\geq 60\%$ Valore ottimale: $\geq 80\%$	OPDS4: Manutenibilità e usabilità
MPDS11	Rapporto linee di commento/linee di codice	Valore minimo: ≥ 0.25 Valore ottimale: ≥ 0.30	OPDS4: Manutenibilità e usabilità
MPDS12	Failure avoidance	Valore minimo: $\geq 80\%$ Valore ottimale: $\geq 90\%$	OPDS5: Affidabilità

Tabella 2: Tabella delle metriche utilizzate

2.3 Scadenze temporali

Il rispetto delle *milestone_g* presenti nel *Piano di Progetto v1.0.0* indica 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 ritardare la consegna del prodotto, con conseguente impatto nel *preventivo_g* fornito nel *Piano di Progetto v1.0.0*, il gruppo JurassicSWE attua procedure di verifica descritte in dettaglio nelle *Norme di Progetto v1.0.0*.

3 Gestione amministrativa della revisione

3.1 Misure e metriche in dettaglio

Di seguito viene illustrata ogni metrica attraverso tre caratteristiche principali:

- **Nome e descrizione:** viene riportato il nome ed una breve descrizione, che indica cosa la metrica rappresenta;
- **Modalità di calcolo:** descrive come calcolare i valori della relativa metrica, in modo da poterli confrontare con i valori nella tabella della sezione 2.2.1.

Per ogni metrica, le soglie di accettabilità sono riportate nella tabella della sezione 2.2.1: nel caso in cui una metrica avesse un valore inferiore ai limiti accettabili, il prodotto o processo dovrà essere sottoposto ad ulteriori indagini e verifiche.

3.1.1 Metriche per i processi

- **MPC1 - Schedule Variance:** è un indice di efficienza che ha come oggetto la durata temporale di un processo o di un'attività. Questo indice aiuta il team nell'analisi dell'utilizzo di risorse temporali.

La formula per calcolarlo è:

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

- **MPC2 - Cost Variance:** è una metrica che analizza il costo e le risorse devolute ad un processo o ad una attività.

La formula per calcolarla è:

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

- **MPC3 - SPICE:** tale standard viene illustrato nel dettaglio nell'Appendice A.1; in generale viene utilizzato alla fine di ogni periodo per monitorare e valutare la qualità dei processi impiegati.

3.1.2 Metriche per la documentazione

- **MPDD1 - Indice Gulpease:** l'indice di Gulpease è un indice di leggibilità di un testo, tarato sulla lingua italiana; è stato scelto questo indice poiché tutti i documenti scritti dal gruppo JurassicSWE sono in lingua italiana.

L'indice di Gulpease considera due variabili linguistiche: la lunghezza della parola e la lunghezza della frase rispetto al numero di lettere.

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

Il risultato della precedente formula è un valore compreso tra 0 e 100, dove 100 indica la leggibilità più alta mentre 0 la leggibilità più bassa.

In generale risulta che i testi con un 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.

Naturalmente, oltre al valore dell'indice di un documento, bisogna tenere anche presente la notorietà dei singoli termini utilizzati nel documento stesso.

- **MPDD2 - Correzione errori ortografici:** indica il numero di errori ortografici rilevati manualmente da un Verificatore o automaticamente dal controllo ortografico dell'*editor_g* in cui vengono redatti i documenti.

Poiché i documenti non devono contenere errori ortografici, questi dovranno essere immediatamente corretti dal Verificatore stesso.

3.1.3 Metriche per il software

- **MPDS1 - Requisiti obbligatori soddisfatti:** possibilità di monitorare in ogni istante la percentuale di requisiti obbligatori soddisfatti, con la seguente formula:

$$\frac{\text{numero requisiti obbligatori soddisfatti}}{\text{numero totale requisiti obbligatori}}$$

- **MPDS2 - Requisiti accettati soddisfatti:** possibilità di monitorare in ogni istante la percentuale di requisiti accettati soddisfatti, con la seguente formula:

$$\frac{\text{numero requisiti accettati soddisfatti}}{\text{numero totale requisiti accettati}}$$

- **MPDS3 - percentuale superamento test:** questa metrica indica quanti dei test implementati hanno esito positivo e può essere ottenuta così:

$$\frac{\text{numero requisiti obbligatori individuati}}{\text{numero requisiti obbligatori soddisfatti}} * 100$$

- **MPDS4 - Numero parametri per metodo:** un numero troppo elevato di parametri in un metodo potrebbe indicare un grado di complessità troppo elevato del metodo.
- **MPDS5 - Numero attributi per metodo:** un valore elevato di attributi può indicare che una classe si fa carico di una quantità eccessiva di responsabilità. Una soluzione potrebbe quindi essere quella di scomporre una parte di tale classe in una seconda classe collegata tramite incapsulamento, ridefinendone le interfacce.
- **MPDS6 - Grado di accoppiamento:** il grado di accoppiamento viene calcolato per capire quanto le classi siano o meno dipendenti rispetto al resto del sistema. I valori risultanti sono utilizzati per il calcolo dell'instabilità.

- **MPDS7 - Complessità ciclomatica:** l'indice di complessità di un programma aiuta ad identificare il numero di test necessari al raggiungimento di un *coverage_g* completo. Questa metrica software può essere applicata anche a *packages_g*, moduli, metodi o classi. Il calcolo avviene sfruttando il grafo di controllo di flusso e l'indice non è altro che il numero di cammini indipendenti attraverso il *codice sorgente_g*.

La formula utilizzata è:

$$e - n + 2p$$

Dove:

- **e:** è il numero di nodi del grafo, nonché il numero di tutti i gruppi indivisibili di istruzioni;
 - **n:** 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.
- **MPDS8 - Numero metodi per classe:** qualora una classe dovesse avere un numero elevato di metodi, probabilmente essa viola i principi della programmazione *SOLID_g* soprattutto quello della *Single Responsibility_g*, per il quale ogni classe deve assolvere ad un solo compito. In caso la classe presenti un elevato numero di metodi, sarà preferibile scomporla in più classi, se possibile.
 - **MPDS9 - Numero di classi per *package_g*:** un numero troppo elevato di classi all'interno di un unico package incrementerebbe le responsabilità dello stesso, andando così contro i principi della programmazione SOLID.
 - **MPDS10 - Test automatici:** questa metrica dà un'idea della percentuale di test automatici implementati dal gruppo JurassicSWE. La volontà è quella di aumentare sempre più il valore di questa metrica.

Tale metrica viene calcolata nel seguente modo:

$$\frac{\text{numero test automatici}}{\text{numero test manuali}} * 100$$

- **MPDS11 - Rapporto linee di commento/linee di codice:** avere un buon rapporto linee di codice su linee di commento all'interno dello stesso aumenta la manutenibilità.

Tale rapporto si calcola nel seguente modo:

$$\frac{\text{numero di linee di codice totali}}{\text{numero di linee di commento}} * 100$$



- **MPDS12 - Failure Avoidance:** questa metrica viene applicata per monitorare l'affidabilità del prodotto nel far fronte a situazioni erronee e/o impreviste.

La formula per calcolarla è:

$$\frac{\text{numero situazioni anomale evitate}}{\text{numero totale situazioni anomale occorse}}$$

3.2 Comunicazione e risoluzione delle anomalie

Il processo *Software Quality Management_g* è finalizzato alla ricerca delle anomalie, ovvero alle deviazioni dal piano prefissato.

Rappresentano un'anomalia:

- errori di codifica;
- violazioni delle norme tipografiche prefissate;
- presenza di contenuti non inerenti all'argomento trattato;
- mancato rispetto dei valori di accettazione fissati in questo documento;
- incongruenze tra il prodotto finale e le funzionalità determinate nell'*Analisi dei Requisiti v1.0.0*.

L'identificazione delle anomalie ne permette la correzione e informa il Responsabile di progetto sullo stato del prodotto.

Di seguito vengono elencate le definizioni di anomalie (*IEEE 610.12-90_g*) adottate dal gruppo:

- **Error**: differenza tra il risultato di una computazione ed il valore teorico atteso;
- **Fault**: corrisponde a quanto definito come *bug_g*, ovvero un passo, processo o dato definito in modo erraneo;
- **Failure**: risultato di un *Fault*;
- **Mistake**: azione umana che produce un risultato errato.

Se viene rilevata una nuova anomalia durante l'attività di verifica, questa dovrà essere immediatamente segnalata tramite il sistema di ticketing, come descritto nelle *Norme di Progetto v1.0.0*. Questo permetterà di segnalare con rapidità l'anomalia ed informare il Responsabile della sua presenza, in modo che possa gestirla correttamente.

A Standard di qualità

A.1 ISO/IEC 15504

Lo standard ISO/IEC 15504, anche conosciuto come *SPICE_g*, acronimo di *Software Process Improvement and Capability Determination*, è lo standard di riferimento per una valutazione oggettiva della qualità dei processi software con l'obiettivo di migliorarli.

Questo standard permette di misurare indipendentemente la capacità di ogni processo tramite degli attributi, studiando il range di risultati che si ottengono eseguendolo. Tali risultati devono essere ripetibili, oggettivi e comparabili affinché possano contribuire al miglioramento dei processi.

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

- **Process performance:** capacità del processo di adempiere agli obiettivi prefissati;
- **Performance management:** misura del grado di organizzazione con cui sono raggiunti gli obiettivi prefissati;
- **Work product management:** misura del grado di gestione dei prodotti del processo in esame;
- **Process definition:** misura dell'adeguatezza del processo rispetto agli standard di riferimento;
- **Process deployment:** capacità di sfruttare le risorse allocate;
- **Process measurement:** capacità del processo di realizzare misurazioni utili per garantire che esso raggiunga i suoi obiettivi;
- **Process control:** capacità del processo di essere corretto o migliorato grazie all'analisi delle misurazione effettuate;
- **Process innovation:** misura del grado in cui le modifiche da apportare al processo sono identificate grazie ad una fase di analisi delle performance e allo studio di approcci innovativi;
- **Process optimization:** capacità del processo di implementare le modifiche effettuate in modo da ottenere un miglioramento continuo nella realizzazione degli obiettivi prefissati.

La maturità di ogni attributo viene misurata e classificata con uno dei seguenti livelli:

- **0 - Incomplete:** fallimento del processo oppure il processo non è stato implementato;
- **1 - Performed:** il processo è stato implementato e ha completato il suo obiettivo;
- **2 - Managed:** il processo, oltre ad essere già *Performed*, è gestito in maniera organizzata con pianificazione, controllo e correzioni. Il tutto genererà dei prodotti considerati sicuri;
- **3 - Established:** il processo, oltre ad essere *Managed*, è stato implementato aderendo agli standard esistenti;

- **4 - Predictable:** il processo, oltre ad essere *Established*, è stato implementato entro limiti prestazionali definiti per raggiungere i propri obiettivi;
- **5 - Optimizing:** il processo, oltre ad essere *Predictable*, è caratterizzato da miglioramento continuo per raggiungere gli obiettivi di business.

Ogni attributo di processo viene misurato e classificato in uno dei seguenti livelli:

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

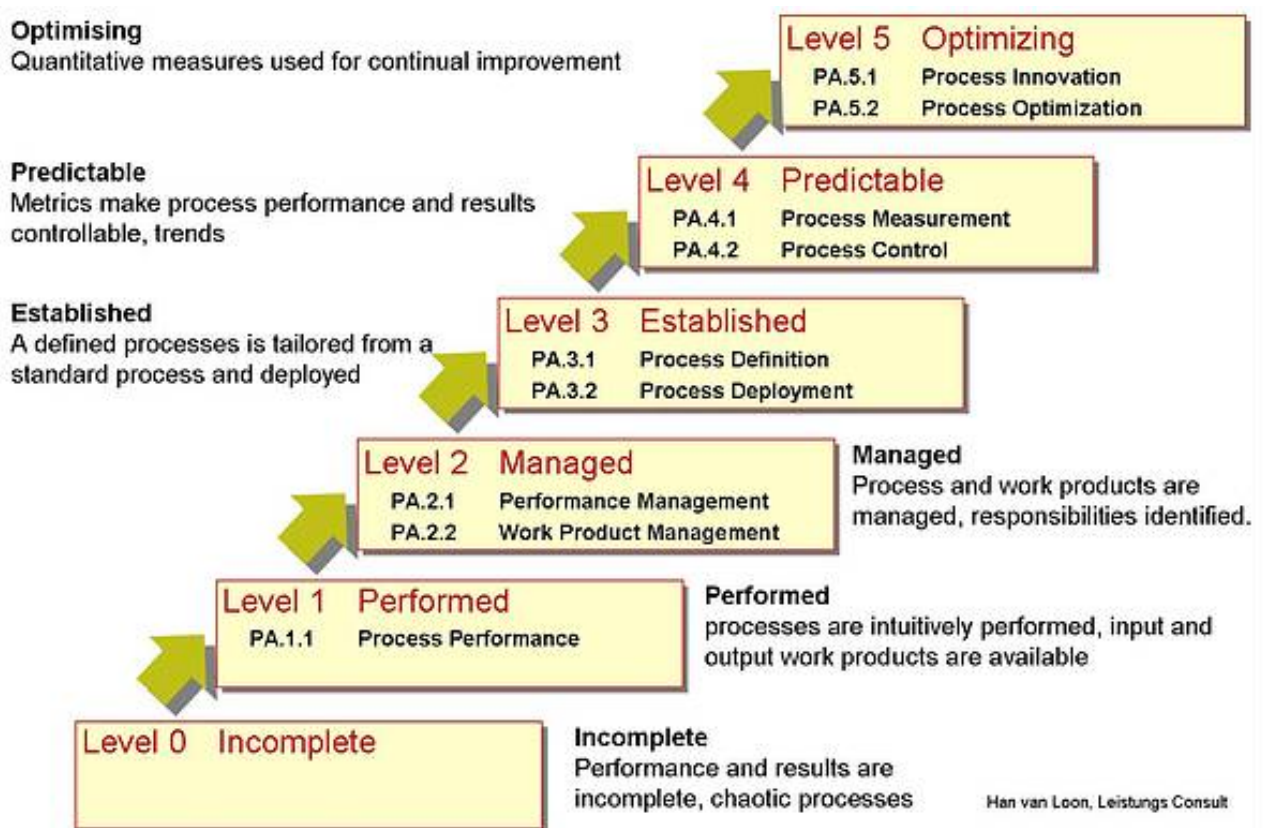


Figura 1: Standard ISO/IEC 15504

A.2 ISO/IEC 9126

ISO/IEC 9126 *Software engineering - Product quality* è uno standard internazionale per valutare la qualità del software.

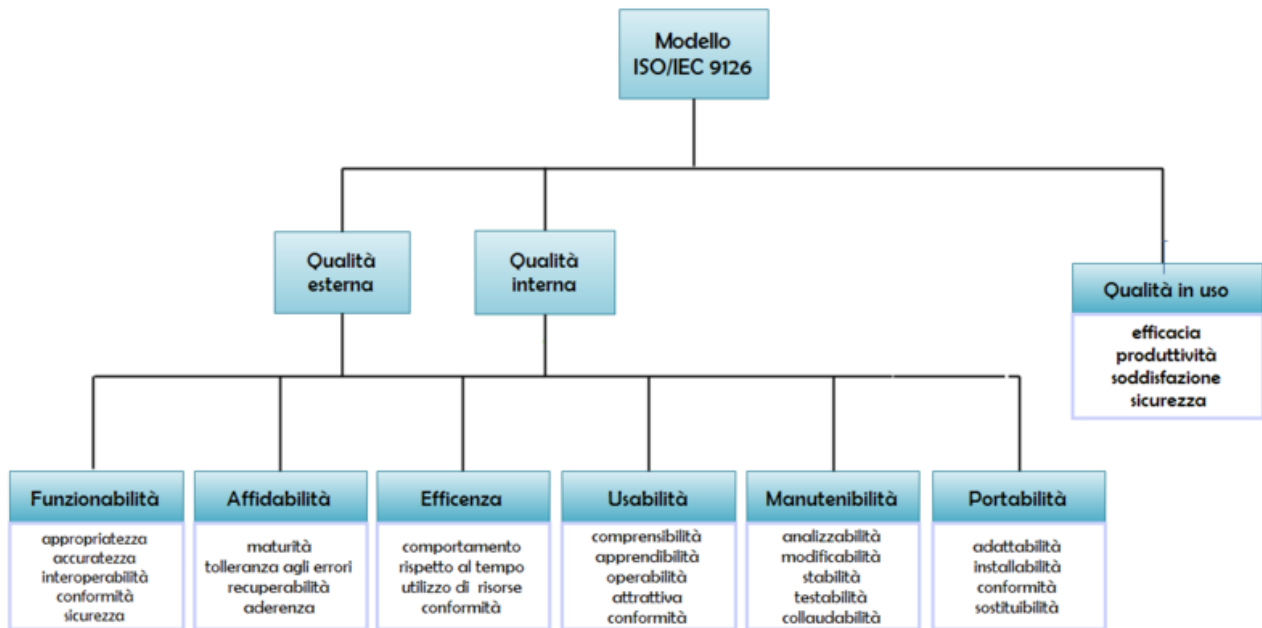


Figura 2: Modello ISO/IEC 9126

Questo standard è diviso in 4 parti:

- **Modello della qualità del software** (descritto dopo le successive 3 parti)
- **Metriche per la qualità interna:** metriche che si applicano al software non eseguibile, per esempio al *codice sorgente_g*, durante le fasi di progettazione e codifica. Permettono di individuare eventuali problemi che potrebbero influire sulla qualità finale del prodotto prima che venga realizzato un *eseguibile_g*. Grazie alle misure effettuate tramite le metriche interne è possibile prevedere il livello di qualità esterna e di qualità in uso del prodotto finale, poiché entrambe vengono influenzate dalla qualità interna. Viene rilevata tramite analisi statica. Idealmente la qualità interna determina la qualità esterna;
- **Metriche per la qualità esterna:** metriche applicabili al software in esecuzione che ne misurano il comportamento attraverso dei test, in funzione degli obiettivi stabiliti. Viene rilevata tramite analisi dinamica. Idealmente la qualità esterna determina la qualità in uso;

- **Metriche per la qualità in uso:** metriche applicabili solo al prodotto finito ed in uso in condizioni reali.

La qualità in uso viene raggiunta solo se è stato raggiunto il livello di qualità interna e di qualità esterna.

Il modello di qualità del software, presentato nella prima parte dello standard, suddivide la qualità in 6 caratteristiche generali e varie sotto caratteristiche, misurabili attraverso delle metriche, utilizzate per fornire una scala ed un metodo per la misurazione.

Elenco delle caratteristiche:

1. **Funzionalità:** capacità del software di soddisfare i requisiti, descritti nell'*Analisi dei Requisiti*, in un determinato contesto.

Nello specifico il software deve soddisfare le seguenti caratteristiche:

- **Appropriatezza:** capacità di fornire funzioni appropriate per attività specifiche, che permettano di raggiungere gli obiettivi prefissati;
- **Accuratezza:** capacità di fornire i risultati concordati o la precisione richiesta;
- **Interoperabilità:** capacità di interagire ed operare con uno o più sistemi specificati;
- **Conformità:** capacità di aderire a standard;
- **Sicurezza:** capacità di proteggere informazioni e dati.

2. **Affidabilità:** capacità del software di mantenere uno specifico livello di prestazioni quando usato in condizioni specificate.

Nello specifico il software deve soddisfare le seguenti caratteristiche:

- **Maturità:** capacità di evitare il verificarsi di errori, malfunzionamenti o risultati non corretti;
- **Tolleranza agli errori:** capacità di mantenere livelli prefissati di prestazioni anche in presenza di malfunzionamenti o usi scorretti del prodotto finale;
- **Recuperabilità:** capacità di ripristinare un livello appropriato di prestazioni o di recupero di informazioni rilevanti a seguito di un *malfunzionamento*_g;
- **Aderenza:** capacità di aderire a standard, regole e convenzioni che riguardano l'affidabilità.

3. **Efficienza:** capacità del prodotto software di eseguire le proprie funzioni minimizzando il tempo necessario e sfruttando al meglio le risorse che necessita.

Nello specifico il software deve soddisfare le seguenti caratteristiche:

- **Nel tempo:** capacità di fornire adeguati tempi di risposta, elaborazione e velocità di attraversamento in determinate condizioni;
- **Nello spazio:** capacità di utilizzo di quantità e tipo di risorse in maniera adeguata;

4. **Usabilità:** capacità del prodotto software di essere compreso, appreso, usato e accettato dall'utente, quando usato sotto determinate condizioni.

Nello specifico il software deve soddisfare le seguenti caratteristiche:

- **Comprensibilità:** capacità di essere chiaro riguardo le proprie funzionalità e il proprio utilizzo;
- **Apprendibilità:** capacità di essere facilmente apprendibile dagli utenti;
- **Operabilità:** capacità di permettere all'utente di eseguire i suoi scopi e controllarne l'uso;
- **Attrattività:** capacità di essere piacevole all'utente che l'utilizza.

5. **Manutenibilità:** capacità del software di essere modificato, al fine di aggiungere correzioni, miglioramenti o adattamenti.

Nello specifico il software deve soddisfare le seguenti caratteristiche:

- **Analizzabilità:** capacità di essere facilmente analizzato al fine di localizzare un errore;
- **Modificabilità:** capacità di poter essere agevolmente modificato nel codice, nella progettazione o nella documentazione;
- **Stabilità:** capacità di evitare effetti indesiderati a seguito di una modifica;
- **Testabilità:** capacità di essere facilmente testato per validare le modifiche apportate.

6. **Portabilità:** capacità del software di essere trasportato da un ambiente di lavoro ad un altro, sia esso hardware che software.

Nello specifico il software deve soddisfare le seguenti caratteristiche:

- **Adattabilità:** capacità di essere facilmente adattato a differenti ambienti operativi, senza applicare modifiche;
- **Installabilità:** capacità di poter essere installato in un determinato ambiente;
- **Conformità:** capacità di coesistere con altre applicazioni e di condividere risorse;
- **Sostituibilità:** capacità di essere utilizzato al posto di un altro software per svolgere gli stessi compiti, nello stesso ambiente.

A.3 Ciclo di Deming

Il *ciclo di Deming*_g (o ciclo di PDCA, acronimo di *Plan-Do-Check-Act*) è un metodo iterativo in quattro fasi utilizzato per il controllo e il miglioramento continuo della qualità dei processi e, quindi, della qualità dei prodotti.

Ogni iterazione del ciclo di Deming consiste di quattro fasi:

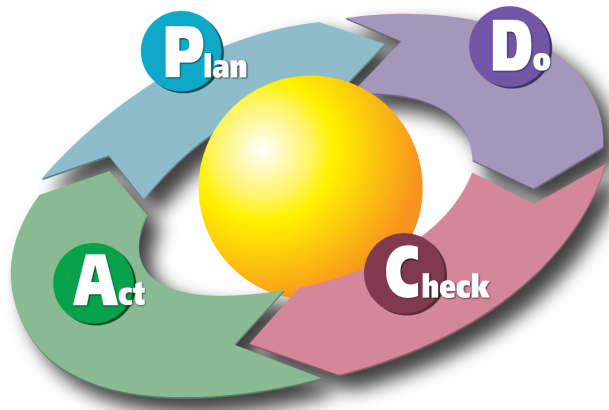


Figura 3: Fasi del Ciclo di Deming

1. **Plan**: fase di pianificazione degli obiettivi di miglioramento. A tale scopo vengono definite le attività da svolgere, le risorse necessarie e le scadenze;
2. **Do**: fase di esecuzione del programma, attuando ciò che è stato pianificato;
3. **Check**: fase di test, controllo, studio e raccolta dei risultati dell'esecuzione. Vengono studiati e misurati i risultati ottenuti per confrontarli con i risultati attesi, cioè gli obiettivi della prima fase (*Plan*), al fine di verificare le eventuali differenze;
4. **Act**: fase di attuazione, al fine di rendere standard i processi che hanno beneficiato delle correzioni e delle modifiche eseguite.

È un modello studiato per il miglioramento della qualità a lungo termine e per l'utilizzo ottimale delle risorse disponibili. Quindi, per migliorare la qualità e soddisfare il cliente, è necessario attraversare tutte le quattro fasi, tenendo come criterio principale la qualità.

Il ciclo di Deming viene infatti ripetuto finché non viene raggiunto l'obiettivo finale:

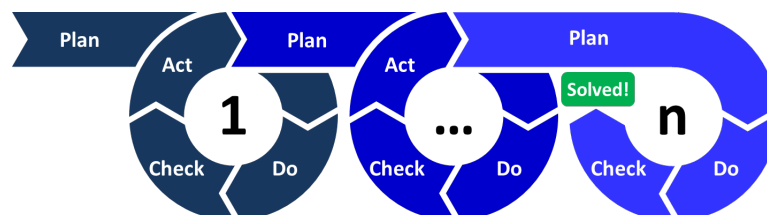


Figura 4: Esempio ciclo di Deming

B Valutazioni per il miglioramento

L'obiettivo di tale appendice è la valutazione atta a migliorare l'intero processo produttivo legato al progetto corrente.

Per questo risulta necessario trovare un modo per affrontare i problemi organizzativi che possono sorgere durante il lavoro, al fine di proporre soluzioni efficienti alla risoluzione degli stessi ed evitare che si ripresentino.

Verranno dunque tracciati problemi riguardanti i seguenti ambiti:

- **Organizzazione:** problemi inerenti l'organizzazione e la comunicazione all'interno del gruppo;
- **Ruoli:** problemi inerenti il corretto svolgimento di un ruolo di progetto;
- **Strumenti:** problemi inerenti l'uso degli strumenti che sono stati scelti.

Per evitare di dover riaffrontare lo stesso problema più volte, si è deciso di tenere un'appendice in cui verranno registrati tutti i problemi riscontrati e le soluzioni proposte.

Ogni problema viene sollevato sulla base dell'autovalutazione dei membri del gruppo, a causa dell'assenza di una persona esterna che dia una valutazione oggettiva.

Nonostante questo sistema sia meno efficace, può comunque contribuire a:

- risolvere problemi difficili da affrontare singolarmente;
- condividere con i componenti del gruppo le pratiche migliori;
- aiutare nell'individuare gli errori;
- evitare di ripetere lo stesso errore o una pratica poco efficace;
- migliorare progressivamente la qualità e l'efficacia del lavoro.

Naturalmente, affinché questo sistema funzioni, i membri del gruppo dovranno essere onesti e sinceri nell'espone i propri problemi o errori, riguardanti le attività svolte o i ruoli ricoperti.

Questa sezione verrà aggiornata man mano che sorgeranno nuovi problemi, con le relative soluzioni, nell'arco di periodo dell'intero progetto.

B.1 Valutazioni sull'organizzazione

Indice	Problema	Soluzione
1	Difficoltà di comunicazione ed incontro tra i membri del gruppo: durante il primo periodo di lavoro, è risultata complicata l'organizzazione tra i membri del gruppo, causando difficoltà ad organizzare gli incontri in modo che tutti i membri fossero presenti.	Si è cambiato lo stile di incontro, passando ad incontri più frequenti pur con alcuni membri assenti.

Tabella 3: Problemi e relative soluzioni riscontrati nell'*organizzazione*

B.2 Valutazioni sui ruoli

Ruolo	Problema	Soluzione
Responsabile	Difficoltà nella suddivisione corretta del carico di lavoro: a causa della inesperienza nel ruolo, si è inizialmente assegnata una quantità di lavoro troppo elevata, per poi accorgersi che non era bilanciata.	Per evitare che si ripresentasse lo stesso problema, si è scelto di ridurre la quantità di lavoro assegnata.
Analista	Difficoltà nell'individuare, suddividere e relazionare i requisiti provenienti dalle varie fonti: non avendo esperienza in tale ruolo e data la varietà delle fonti alcuni requisiti non sono stati inizialmente compresi a fondo o ben interpretati.	Per risolvere il problema, si è scelto di approfondire le conoscenze del dominio e chiedere chiarimenti al Proponente.
Verificatore	Difficoltà nel verificare approfonditamente i documenti per correttezza e completezza: dovuta all'inesperienza nel ricoprire tale ruolo e dal tempo necessario a studiare il dominio del relativo documento.	Per porre rimedio si è scelto di assegnare maggiori risorse all'attività di verifica.

Tabella 4: Problemi e relative soluzioni riscontrati nei *ruoli*

B.3 Valutazioni sugli strumenti

Strumento	Problema	Soluzione
\LaTeX	Difficoltà nel creare un template generale per tutti i documenti che il gruppo ha dovuto redigere e nel far comprendere i comandi utilizzati. Questo perché alcuni membri del gruppo non avevano mai usato \LaTeX .	Si è cercato sempre di affiancare un membro inesperto ad un membro esperto di \LaTeX nella stesura dei vari documenti e di fornire guide ed esempi utili dei vari comandi utilizzati.

Tabella 5: Problemi e relative soluzioni riscontrati nell'uso degli *strumenti*

C Resoconto delle attività di verifica

C.1 Riassunto delle attività di verifica per le revisioni

Precedentemente alla consegna del materiale per la *Revisione dei Requisiti* sono stati verificati i documenti redatti ed i processi eseguiti.

Ogni documento è stato verificato dai Verificatori come pianificato nel *Piano di Progetto v1.0.0* e secondo i criteri per l'analisi statica definiti nel documento *Norme di Progetto v1.0.0*, applicando il sistema *Walkthrough_g* e *Inspection_g*.

Analisi statica:

1. **Walkthrough:** prima si è verificata l'interezza del documento, cercando eventuali errori presenti e, in caso vi fossero, trattandoli nel seguente modo:
 - (a) correzione di errori ed eventuali violazioni delle norme tipografiche;
 - (b) segnalazione ed aggiunta alla lista di controllo degli errori più comuni;
 - (c) applicazione del ciclo di Deming per migliorare e velocizzare le verifiche future.
2. **Inspection:** successivamente si è passato al metodo Inspection, usando la lista di controllo degli errori precedentemente stilata, per controllare i documenti già verificati, ma ponendo maggiore attenzione agli errori più frequenti.

Il tracciamento dei requisiti è stato effettuato tramite *PragmaDB_g*, successivamente ricontrollato manualmente per accertarne la correttezza.

Infine sono stati controllati i documenti ed il processo stesso, al fine di assicurare che rispettassero le metriche proposte in questo documento.

I risultati vengono riportati nella sezione successiva 'Dettaglio delle verifiche tramite analisi'.

C.2 Dettaglio delle verifiche tramite analisi

C.2.1 Periodo di Analisi dei requisiti di massima

- **Indice di Gulpease**

Di seguito viene riportata una tabella contenente il *valore Gulpease_g*, descritto nella sezione 3.1.2, relativo a ciascun documento.

Per calcolare tale indice sono state escluse le tabelle, le pagine di frontespizio e il diario delle modifiche, poiché la loro inclusione avrebbe generato valori errati.

L'esito della misurazione è:

- *Positivo*: se l'indice è maggiore o uguale a 40;
- *Negativo*: viceversa, ovvero se l'indice è minore di 40.

Nome Documento	Valore Indice	Esito
<i>Analisi dei Requisiti v1.0.0</i>	70	Positivo
<i>Glossario v1.0.0</i>	67	Positivo
<i>Piano di Progetto v1.0.0</i>	87	Positivo
<i>Piano di Qualifica v1.0.0</i>	84	Positivo
<i>Norme di Progetto v1.0.0</i>	75	Positivo
<i>Studio di Fattibilità v1.0.0</i>	81	Positivo

Tabella 6: Valore indice di Gulpease nel periodo di *Analisi dei requisiti di massima*

- **Schedule variance**

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

Nome Documento	Valore SV	Esito
<i>Analisi dei Requisiti v1.0.0</i>	-2	Ottimale
<i>Glossario v1.0.0</i>	0	Ottimale
<i>Piano di Progetto v1.0.0</i>	0	Ottimale
<i>Piano di Qualifica v1.0.0</i>	0	Ottimale
<i>Norme di Progetto v1.0.0</i>	0	Ottimale
<i>Studio di Fattibilità v1.0.0</i>	0	Ottimale

Tabella 7: Valore della schedule variance nel periodo di *Analisi dei requisiti di massima*

- **Cost variance**

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

Cost variance	Esito
0%	Ottimale

Tabella 8: Valore del Cost variance nel periodo di *Analisi dei requisiti di massima*