



## Piano di Qualifica

### Informazioni sul documento

<b>Versione</b>	1.0.0
<b>Data di Creazione</b>	2016-12-04
<b>Data ultima modifica</b>	2017-01-07
<b>Stato</b>	Approvato
<b>Redazione</b>	Jordan Gottardo Marco Pasqualini
<b>Verifica</b>	Leonardo Brutesco
<b>Approvazione</b>	Giulia Petenazzi
<b>Uso</b>	Esterno
<b>Lista di distribuzione</b>	Professor Tullio Vardanega Professor Riccardo Cardin Zephyrus RiskApp
<b>Email di riferimento</b>	zephyrus.swe@gmail.com

## Registro delle modifiche

Versione	Data	Autore	Ruolo	Descrizione
1.0.0	2017-01-07	Giulia Petenazzi	<i>Responsabile</i>	Approvazione
0.2.0	2017-01-06	Leonardo Brutesco	<i>Verificatore</i>	Verifica documento
0.1.1	2017-01-06	Jordan Gottardo	<i>Analista</i>	Correzione errori
0.1.0	2017-01-06	Leonardo Brutesco	<i>Verificatore</i>	Verifica documento
0.0.7	2017-01-04	Marco Pasqualini	<i>Verificatore</i>	Aggiunta sezione resoconto delle attività di verifica - fase analisi
0.0.6	2016-12-13	Marco Pasqualini	<i>Analista</i>	Aggiunte sezioni CMM, ISO/IEC-9126, Test, PDCA
0.0.5	2016-12-12	Jordan Gottardo	<i>Analista</i>	Aggiunta sezione la strategia di gestione della qualità nel dettaglio
0.0.3	2016-12-05	Jordan Gottardo	<i>Analista</i>	Aggiunta sezione visione generale della strategia di gestione
0.0.2	2016-12-04	Jordan Gottardo	<i>Analista</i>	Aggiunta introduzione
0.0.1	2016-12-04	Jordan Gottardo	<i>Analista</i>	Creazione template e indice

## Indice

<b>1</b>	<b>Introduzione</b>	<b>1</b>
1.1	Scopo del documento . . . . .	1
1.2	Scopo del prodotto . . . . .	1
1.3	Glossario . . . . .	1
1.4	Riferimenti . . . . .	1
1.4.1	Riferimenti normativi . . . . .	1
1.4.2	Riferimenti informativi . . . . .	1
<b>2</b>	<b>Visione generale della strategia di gestione della qualità</b>	<b>3</b>
2.1	Obiettivi di qualità . . . . .	3
2.1.1	Qualità di processo . . . . .	3
2.1.1.1	Miglioramento costante . . . . .	3
2.1.1.2	Rispetto della pianificazione . . . . .	4
2.1.1.3	Rispetto del budget . . . . .	4
2.1.1.4	Completezza dell'analisi dei rischi . . . . .	4
2.1.2	Qualità di prodotto . . . . .	4
2.1.2.1	Qualità dei documenti . . . . .	5
2.1.2.1.1	Leggibilità e comprensibilità . . . . .	5
2.1.2.2	Adesione alle norme interne . . . . .	5
2.1.2.2.1	Correttezza ortografica . . . . .	5
2.1.2.2.2	Correttezza concettuale . . . . .	5
2.1.2.3	Qualità del software . . . . .	6
2.1.2.3.1	Implementazione delle funzionalità obbligatorie . . . . .	6
2.1.2.3.2	Implementazione delle funzionalità desiderabili . . . . .	6
2.1.2.3.3	Manutenibilità e comprensibilità del codice . . . . .	6
2.1.2.3.4	Documentazione del codice . . . . .	7
2.1.2.3.5	Validazione web . . . . .	8
2.1.2.3.6	Copertura dei test richiesti . . . . .	8
2.1.2.3.7	Robustezza . . . . .	8
2.1.2.3.8	Correzione delle situazioni di fallimento . . . . .	8
2.1.3	Scadenze temporali . . . . .	8
<b>3</b>	<b>La strategia di gestione della qualità nel dettaglio</b>	<b>9</b>
3.1	Risorse . . . . .	9
3.1.1	Risorse necessarie . . . . .	9
3.1.1.1	Risorse umane . . . . .	9
3.1.1.2	Risorse hardware . . . . .	9
3.1.1.3	Risorse software . . . . .	9
3.1.2	Risorse disponibili . . . . .	9
3.1.2.1	Risorse umane . . . . .	9
3.1.2.2	Risorse hardware . . . . .	9
3.1.2.3	Risorse software . . . . .	9
3.2	Misure e metriche . . . . .	10
3.2.1	Misure . . . . .	10
3.2.2	Metriche per i processi . . . . .	10
3.2.2.1	Miglioramento costante . . . . .	10
3.2.2.2	Rispetto della pianificazione . . . . .	10
3.2.2.3	Rispetto del budget . . . . .	11

3.2.2.4	Completezza dell'analisi dei rischi . . . . .	11
3.2.3	Metriche per i prodotti . . . . .	11
3.2.3.1	Metriche per i documenti . . . . .	11
3.2.3.1.1	Leggibilità e comprensibilità . . . . .	11
3.2.3.1.2	Adesione alle norme interne . . . . .	12
3.2.3.1.3	Correttezza ortografica . . . . .	12
3.2.3.1.4	Correttezza concettuale . . . . .	12
3.2.3.2	Metriche per il software . . . . .	13
3.2.3.2.1	Implementazione delle funzionalità obbligatorie . . . . .	13
3.2.3.2.2	Implementazione delle funzionalità desiderabili . . . . .	14
3.2.3.2.3	Numero di statement per metodo . . . . .	14
3.2.3.2.4	Numero di parametri per metodo . . . . .	14
3.2.3.2.5	Numero di campi dati per classe . . . . .	14
3.2.3.2.6	Grado di accoppiamento . . . . .	15
3.2.3.2.7	Complessità ciclomatica . . . . .	15
3.2.3.2.8	Numero di variabili dichiarate e non utilizzate . . . . .	15
3.2.3.2.9	Documentazione del codice . . . . .	15
3.2.3.2.10	Validazione web . . . . .	16
3.2.3.2.11	Copertura dei test richiesti . . . . .	16
3.2.3.2.12	Robustezza . . . . .	16
3.2.3.2.13	Correzione delle situazioni di fallimento . . . . .	17
<b>A</b>	<b>CMM</b>	<b>18</b>
A.1	Struttura . . . . .	18
A.2	Livelli . . . . .	18
<b>B</b>	<b>ISO/IEC 9126</b>	<b>20</b>
B.1	Modello della qualità del software . . . . .	20
B.1.1	Modello della qualità esterna e interna . . . . .	20
B.1.2	Modello della qualità in uso . . . . .	21
B.2	Qualità esterna e relative metriche . . . . .	21
B.3	Qualità interna e relative metriche . . . . .	21
B.4	Qualità in uso e relative metriche . . . . .	21
<b>C</b>	<b>PDCA</b>	<b>22</b>
C.1	Fasi . . . . .	22
<b>D</b>	<b>Test</b>	<b>23</b>
D.1	Test di accettazione . . . . .	23
D.2	Test di sistema . . . . .	23
D.3	Test di integrazione . . . . .	23
D.4	Test di unità . . . . .	23
<b>E</b>	<b>Resoconto delle attività di verifica -Fase Analisi</b>	<b>24</b>
E.1	Verifica dei processi . . . . .	24
E.1.1	Considerazioni finali . . . . .	24
E.1.1.1	Miglioramento costante . . . . .	24
E.1.1.2	Rispetto della pianificazione . . . . .	24
E.1.1.3	Rispetto del budget . . . . .	24
E.1.1.4	Completezza dell'analisi dei rischi . . . . .	25

E.2	Verifica dei prodotti . . . . .	25
E.2.1	Verifica dei documenti . . . . .	25
E.2.1.1	Leggibilità e comprensibilità . . . . .	25
E.2.1.1.1	Considerazioni finali . . . . .	25
E.2.1.2	Adesione alle norme interne . . . . .	26
E.2.1.2.1	Considerazioni finali . . . . .	26
E.2.1.3	Correttezza ortografica . . . . .	26
E.2.1.3.1	Considerazioni finali . . . . .	26
E.2.1.4	Correttezza concettuale . . . . .	27
E.2.1.4.1	Considerazioni finali . . . . .	27

# 1 Introduzione

## 1.1 Scopo del documento

Il presente documento ha lo scopo di fissare in modo quantitativo gli obiettivi di qualità di processo e di prodotto che il *team<sub>G</sub>* si è prefissato. Inoltre verranno illustrate le strategie di verifica e validazione utilizzate per raggiungere tali obiettivi e la loro effettiva applicazione nelle varie *fasi<sub>G</sub>* di sviluppo.

## 1.2 Scopo del prodotto

Lo scopo del prodotto consiste nella creazione di un'interfaccia web contenente una mappa geografica su cui potranno essere rappresentati:

- il processo produttivo aziendale;
- gli scenari di danno;
- i risultati dell'analisi dei rischi.

Il prodotto verrà utilizzato da agenti assicuratori per l'inserimento delle informazioni utili allo svolgimento dell'analisi dei rischi dell'assicurando.

L'interfaccia dovrà essere in grado di connettersi ai sistemi preesistenti di *RiskApp* per la memorizzazione e gestione dei dati inseriti.

## 1.3 Glossario

Allo scopo di rendere più semplice e chiara la comprensione dei documenti viene allegato il *Glossario v1.0.0*, nel quale verranno raccolte le spiegazioni di terminologia tecnica o ambigua, abbreviazioni ed acronimi. Per evidenziare un termine presente in tale documento, esso verrà marcato con il pedice *G*.

Tutti i termini del glossario evidenziati sono link ipertestuali al glossario stesso; affinché funzionino correttamente è necessario che la posizione delle directory e dei file forniti non venga alterata.

## 1.4 Riferimenti

### 1.4.1 Riferimenti normativi

- *Norme di progetto v1.0.0*;
- *capitolato<sub>G</sub> d'appalto C3*: <http://www.math.unipd.it/tullio/IS-1/2016/Progetto/C3.pdf>;
- standard *ISO<sub>G</sub>/IEC<sub>G</sub> 9126:2001*: [https://en.wikipedia.org/wiki/ISO/IEC\\_9126](https://en.wikipedia.org/wiki/ISO/IEC_9126);
- *Capability Maturity Model(CMM<sub>G</sub>)*: [https://en.wikipedia.org/wiki/Capability\\_Maturity\\_Model](https://en.wikipedia.org/wiki/Capability_Maturity_Model).

### 1.4.2 Riferimenti informativi

- *Piano di progetto v1.0.0*;
- *Plan-Do-Check-Act<sub>G</sub>*: <https://en.wikipedia.org/wiki/PDCA>;
- *indice di Gulpease<sub>G</sub>*: [http://it.wikipedia.org/wiki/Indice\\_Gulpease](http://it.wikipedia.org/wiki/Indice_Gulpease);

- slide del corso di Ingegneria del Software - Qualità del software:  
<http://www.math.unipd.it/tullio/IS-1/2016/Dispense/L10.pdf>;
- slide del corso di Ingegneria del Software - Qualità di processo:  
<http://www.math.unipd.it/tullio/IS-1/2016/Dispense/L11.pdf>;
- metriche di progetto: [https://it.wikipedia.org/wiki/Metriche\\_di\\_progetto](https://it.wikipedia.org/wiki/Metriche_di_progetto).

## 2 Visione generale della strategia di gestione della qualità

### 2.1 Obiettivi di qualità

In questa sezione vengono espressi gli obiettivi di qualità che il *team<sub>G</sub>* si è prefissato. Data la difficoltà (e in alcuni casi l'impossibilità) nel misurare direttamente la qualità, sono stati scelti standard, modelli e metriche. Ognuno di questi fa uso di scale differenti e fissate a priori. Per ogni criterio, il *team<sub>G</sub>* ha fissato dei range di valori accettabili e ottimi. A prescindere dal livello raggiunto in ogni misurazione, l'obiettivo da perseguire è il miglioramento continuo della qualità, realizzata attraverso il ciclo *PDCA<sub>G</sub>*. Nel caso in cui non si raggiungesse l'obiettivo minimo, dovranno essere attuate misure correttive come previsto dalle *Norme di progetto v1.0.0*. Siccome la qualità non è una proprietà intrinseca dei processi, è fondamentale dotarsi di buoni strumenti per effettuare le misurazioni. È possibile trovare una descrizione di tali strumenti nelle *Norme di progetto v1.0.0*.

#### 2.1.1 Qualità di processo

Garantire la qualità dei processi è fondamentale se si vogliono ottenere prodotti di qualità. L'unico modo di garantire *quality assurance<sub>G</sub>* è far sì che i processi siano normati e misurati. Inoltre, è possibile anche ottenere maggiore efficienza, efficacia e ripetibilità dei risultati. Gli obiettivi relativi a questi ambiti sono illustrati nel *Piano di progetto v1.0.0*. Le caratteristiche che i processi dovrebbero avere sono le seguenti:

- un processo dovrebbe essere in grado di migliorare continuamente le proprie performance:
  - le performance di un processo dovrebbero essere misurabili;
  - un processo dovrebbe perseguire obiettivi quantitativi di miglioramento.
- i processi e le loro attività dovrebbero rispettare i tempi e i costi stabiliti dal *Piano di progetto v1.0.0*.

Seguono gli obiettivi e le metriche riguardanti la qualità di processo che il *team<sub>G</sub>* ha stabilito.

##### 2.1.1.1 Miglioramento costante

Per quantificare il livello di performance raggiunto dai processi, si è deciso di adottare il modello *CMM<sub>G</sub>*. L'obiettivo è migliorare costantemente tale livello, secondo quanto definito dal ciclo *PDCA<sub>G</sub>*.

- **metrica utilizzata:** Livello *CMM<sub>G</sub>* (*LCMM*). Fa riferimento alla scala stabilita dal modello *CMM<sub>G</sub>*;
- **soglia di accettabilità:**  $LCMM \geq 2$ ;
- **soglia di ottimalità:**  $LCMM \geq 4$ .

Per una descrizione più dettagliata del modello *CMM<sub>G</sub>*, si faccia riferimento all'[appendice A](#). Per una descrizione più dettagliata della metrica consultare la sezione [3.2.2.1](#).



### 2.1.1.2 Rispetto della pianificazione

Rispettare la pianificazione del lavoro stabilita nel *Piano di progetto v1.0.0* è fondamentale per evitare ritardi e garantire la qualità del processo. Qualora non la si rispettasse, è molto probabile che il processo non abbia le caratteristiche di qualità desiderate.

- **metrica utilizzata:** Schedule Variance (*SV*);
- **soglia di accettabilità:**  $SV \leq 4$  giorni rispetto a quanto pianificato;
- **soglia di ottimalità:**  $SV \leq 0$  giorni rispetto a quanto pianificato, ovvero essere allineato alla pianificazione o in anticipo.

Per una descrizione più dettagliata della metrica consultare la sezione [3.2.2.2](#)

### 2.1.1.3 Rispetto del budget

Rispettare il budget stabilito nel *Piano di progetto v1.0.0* è un obiettivo importante per evitare inefficienze nell'utilizzo delle risorse.

- **metrica utilizzata:** Cost Variance (*CV*);
- **soglia di accettabilità:**  $CV \leq 10\%$  rispetto a quanto preventivato;
- **soglia di ottimalità:**  $CV \leq 0\%$  rispetto a quanto preventivato.

Per una descrizione più dettagliata della metrica consultare la sezione [3.2.2.3](#)

### 2.1.1.4 Completezza dell'analisi dei rischi

Il *team<sub>G</sub>* desidera che il processo di analisi dei rischi sia il più completo possibile, così da ridurre la probabilità di subire danni da rischi non preventivati.

- **metrica utilizzata:** contatore Rischi Non Preventivati (*RNP*) che aumenta di 1 ogni volta che si presenta un rischio non previsto nel *Piano di progetto v1.0.0*;
- **soglia di accettabilità:**  $RNP \leq 2$ ;
- **soglia di ottimalità:**  $RNP = 0$ .

Per una descrizione più dettagliata della metrica consultare la sezione [3.2.2.4](#)

## 2.1.2 Qualità di prodotto

Oltre alla qualità dei processi, il *team<sub>G</sub>* desidera anche garantire determinate caratteristiche di qualità dei prodotti. Per raggiungere questo obiettivo, è necessario che il processo con cui tale prodotto viene realizzato sia controllato e vincolato. A tal fine, è stato scelto di seguire lo standard *ISO<sub>G</sub>/IEC<sub>G</sub> 9126:2001*.

Le tipologie di prodotti che verranno realizzati sono due:

- documenti;
- software.

### 2.1.2.1 Qualità dei documenti

Il *team<sub>G</sub>* si pone come obiettivo la produzione di documenti di qualità. Essi sono infatti fondamentali per la comprensione del prodotto software fin dal concepimento, sia da parte di soggetti interni che esterni. Seguono gli obiettivi e le metriche riguardanti la qualità dei documenti che il *team<sub>G</sub>* si è prefissato.

#### 2.1.2.1.1 Leggibilità e comprensibilità

La leggibilità e comprensibilità dei documenti sono caratteristiche fondamentali affinché essi siano utili a coloro che li leggono.

- **metrica utilizzata:** *Indice Gulpase<sub>G</sub>* (*IG*);
- **soglia di accettabilità:**  $IG \geq 40$ ;
- **soglia di ottimalità:**  $IG \geq 60$ .

Per una descrizione più dettagliata della metrica consultare la sezione [3.2.3.1.1](#).

#### 2.1.2.2 Adesione alle norme interne

Aderire alle regole di stesura dei documenti definite nelle *Norme di progetto v1.0.0* è fondamentale per assicurare l'omogeneità del testo e della terminologia. Alcuni esempi di norme riguardanti i documenti sono quelle relative ai loro nomi (e relativa versione), agli elenchi puntati, ai ruoli dei membri, ecc.

- **metrica utilizzata:** numero di Errori riguardanti le Norme interne rinvenuti e Non Corretti (*ENNC*);
- **soglia di accettabilità:**  $ENNC = 0$ ;
- **soglia di ottimalità:**  $ENNC = 0$ .

Per una descrizione più dettagliata della metrica consultare la sezione [3.2.3.1.2](#).

##### 2.1.2.2.1 Correttezza ortografica

Il *team<sub>G</sub>* desidera che i documenti prodotti siano completamente esenti da errori ortografici rilevati e non corretti.

- **metrica utilizzata:** numero di Errori Ortografici rinvenuti e Non Corretti (*EONC*);
- **soglia di accettabilità:**  $EONC = 0$ ;
- **soglia di ottimalità:**  $EONC = 0$ .

Per una descrizione più dettagliata della metrica consultare la sezione [3.2.3.1.3](#).

##### 2.1.2.2.2 Correttezza concettuale

L'obiettivo è ridurre il più possibile il numero di errori concettuali rinvenuti e non corretti.

- **metrica utilizzata:** percentuale di Errori Concettuali rinvenuti e Non Corretti (*ECNC*);
- **soglia di accettabilità:**  $ECNC \leq 5\%$ ;
- **soglia di ottimalità:**  $ECNC = 0\%$ .

Per una descrizione più dettagliata della metrica consultare la sezione [3.2.3.1.4](#).

### 2.1.2.3 Qualità del software

Il *team<sub>G</sub>* desidera che il software prodotto sia di qualità. Seguono le caratteristiche e le metriche riguardanti la qualità del software che il *team<sub>G</sub>* si è prefissato.

#### 2.1.2.3.1 Implementazione delle funzionalità obbligatorie

Il software deve implementare completamente le funzionalità descritte nei requisiti obbligatori.

- **metrica utilizzata:** numero di requisiti obbligatori soddisfatti (*IFO*);
- **soglia di accettabilità:** 100% dei requisiti obbligatori soddisfatti;
- **soglia di ottimalità:** 100% dei requisiti obbligatori soddisfatti.

Per una descrizione più dettagliata della metrica consultare la sezione [3.2.3.2.1](#).

#### 2.1.2.3.2 Implementazione delle funzionalità desiderabili

Il software deve implementare completamente le funzionalità descritte nei requisiti desiderabili.

- **metrica utilizzata:** numero di requisiti desiderabili soddisfatti;
- **soglia di accettabilità:** 100% dei requisiti desiderabili soddisfatti;
- **soglia di ottimalità:** 100% dei requisiti desiderabili soddisfatti.

Per una descrizione più dettagliata della metrica consultare la sezione [3.2.3.2.2](#).

#### 2.1.2.3.3 Manutenibilità e comprensibilità del codice

L'obiettivo è fornire codice privo di incomprensioni e manutenibile nel tempo. Per cercare di soddisfare questo obiettivo, sono state selezionate varie metriche.

- **metrica utilizzata:** Numero di *Statement<sub>G</sub>* per Metodo (*NSM*);
- **soglia di accettabilità:**  $30 < NSM \leq 60$ ;
- **soglia di ottimalità:**  $NSM \leq 30$ .
  
- **metrica utilizzata:** Numero di Parametri per Metodo (*NPM*);
- **soglia di accettabilità:**  $5 < NPM \leq 10$ ;
- **soglia di ottimalità:**  $NPM \leq 5$ .
  
- **metrica utilizzata:** Numero di Campi Dati Per Classe (*NCDPC*);
- **soglia di accettabilità:**  $10 < NCDPC \leq 15$ ;
- **soglia di ottimalità:**  $NCDPC \leq 10$ .
  
- **metrica utilizzata:** Grado di Accoppiamento (*GA*);

- **soglia di accettabilità:**  $3 < GA \leq 10$ ;
- **soglia di ottimalità:**  $GA \leq 3$ .

$GA$ =numero di dipendenze tra classi in un [package](#).

- **metrica utilizzata:** Numero Ciclomatico ( $NC$ );
- **soglia di accettabilità:**  $10 < NC \leq 20$ ;
- **soglia di ottimalità:**  $NC \leq 10$ .

$NC$ =numero di cammini linearmente indipendenti presenti all'interno del codice.

- **metrica utilizzata:** Numero di Variabili dichiarate e Non Utilizzate ( $NVNU$ );
- **soglia di accettabilità:**  $NVNU = 0$ ;
- **soglia di ottimalità:**  $NVNU = 0$ .

Per una descrizione più dettagliata della metriche relative al codice, consultare le sezioni:

- [Numero di statement per metodo](#);
- [Numero di parametri per metodo](#);
- [Numero di campi dati per classe](#);
- [Grado di accoppiamento](#);
- [Complessità ciclomatica](#);
- [Numero di variabili dichiarate e non utilizzate](#);
- [Documentazione del codice](#).

#### 2.1.2.3.4 Documentazione del codice

Avere codice documentato è importante per garantire manutenibilità e comprensibilità dello stesso. Il mezzo con cui si intende raggiungere tale obiettivo è commentare il codice. Verrà posta particolare attenzione nello scrivere commenti comprensibili anche a eventuali manutentori, che potranno anche essere soggetti esterni.

- **metrica utilizzata:** Rapporto linee di Commento e linee di Codice ( $RCC$ );
- **soglia di accettabilità:**  $RCC \geq 10\%$ ;
- **soglia di ottimalità:**  $RCC \geq 30\%$ .

Per una descrizione più dettagliata della metrica consultare la sezione [3.2.3.2.9](#).

### 2.1.2.3.5 Validazione web

Il *team<sub>G</sub>* desidera che il codice *HTML<sub>G</sub>* risulti valido secondo gli strumenti offerti dal *W3C<sub>G</sub>*. Nonostante codice validato non implichi direttamente la sua qualità, è un buon punto di partenza. Inoltre, serve a limitare i casi in cui l'interpretazione del codice *HTML<sub>G</sub>* viene affidata al browser, ad esempio a causa di *tag<sub>G</sub>* non chiusi nell'ordine corretto.

- **metrica utilizzata:** Numero di Errori di Validazione (*NEV*) rilevati dal validatore online del *W3C<sub>G</sub>*;
- **soglia di accettabilità:**  $NEV \leq 10$ ;
- **soglia di ottimalità:**  $NEV=0$ .

Per una descrizione più dettagliata della metrica consultare la sezione [3.2.3.2.10](#).

### 2.1.2.3.6 Copertura dei test richiesti

Assicurare la copertura dei test è fondamentale per poter verificare la corretta implementazione delle funzionalità previste dai requisiti.

- **metrica utilizzata:** Copertura dei Test Richiesti (*CTR*);
- **soglia di accettabilità:**  $80\% \leq CTR < 90\%$ ;
- **soglia di ottimalità:**  $90\% \leq CTR \leq 100\%$ .

*CTR*=percentuale di test passati.

Per una descrizione più dettagliata della metrica consultare la sezione [3.2.3.2.11](#).

### 2.1.2.3.7 Robustezza

Il prodotto non deve interrompere il suo funzionamento al verificarsi di situazioni anomale e di errore. È preferibile la segnalazione dell'errore all'arresto improvviso.

- **metrica utilizzata:** Breakdown Avoidance (*BA*);
- **soglia di accettabilità:**  $80\% \leq BA < 95\%$ ;
- **soglia di ottimalità:**  $BA \geq 95\%$ .

Per una descrizione più dettagliata della metrica consultare la sezione [3.2.3.2.12](#).

### 2.1.2.3.8 Correzione delle situazioni di fallimento

Il prodotto deve superare la maggior parte dei test che provino a compromettere la sua stabilità.

- **metrica utilizzata:** Failure Avoidance (*FA*);
- **soglia di accettabilità:**  $80\% \leq FA < 95\%$ ;
- **soglia di ottimalità:**  $FA \geq 95\%$ .

*FA*=percentuale di situazioni anomale evitate su situazioni anomale prese in considerazione.

Per una descrizione più dettagliata della metrica consultare la sezione [3.2.3.2.13](#).

## 2.1.3 Scadenze temporali

Le scadenze che il *team<sub>G</sub>* ha deciso di rispettare sono riportate nel *Piano di progetto v1.0.0*.

## 3 La strategia di gestione della qualità nel dettaglio

### 3.1 Risorse

Il processo di verifica necessita di risorse per poter ottenere gli obiettivi prefissati. Il *team<sub>G</sub>* ha identificato i seguenti tipi di risorse:

- risorse umane;
- risorse hardware;
- risorse software.

#### 3.1.1 Risorse necessarie

##### 3.1.1.1 Risorse umane

Le risorse umane comprendono il *Responsabile di progetto* e i *Verificatori*.

##### 3.1.1.2 Risorse hardware

Le risorse hardware comprendono i computer con una potenza di calcolo sufficiente a sopportare il carico di lavoro.

##### 3.1.1.3 Risorse software

Le risorse software comprendono gli strumenti software, sia installabili localmente che disponibili online, che permettano di eseguire controlli su documenti e codice. Inoltre è necessario anche un tool per il tracciamento dei requisiti e il controllo dei test.

### 3.1.2 Risorse disponibili

#### 3.1.2.1 Risorse umane

Le risorse umane disponibili sono tutti i membri del *team<sub>G</sub>*, che ricopriranno a turno i ruoli di *Responsabile di progetto* e di *Verificatore* come previsto dal *Piano di progetto v1.0.0*.

#### 3.1.2.2 Risorse hardware

Le risorse hardware disponibili sono i personal computer dei membri del *team<sub>G</sub>*. In caso tali non fossero disponibili, si potranno sfruttare i computer del Servizio Calcolo dell'Università di Padova.

#### 3.1.2.3 Risorse software

Le risorse software disponibili sono:

- editor  $\text{\LaTeX}$  con pacchetto lingua italiana installato e correzioni ortografiche abilitate;
- script per calcolare l'*indice Gulpease<sub>G</sub>*;
- *issue<sub>G</sub>* offerte da *GitHub<sub>G</sub>*.

## 3.2 Misure e metriche

### 3.2.1 Misure

Ogni volta che viene effettuata una misurazione si ottiene un valore, ovvero una misura. Per poter comparare due valori è necessario rapportarli su una scala: facendo ciò si crea una metrica.

Tenendo conto degli obiettivi precedentemente stabiliti, il valore può avere giudizio:

- **negativo:** obiettivo non raggiunto. È necessario fare ulteriori verifiche o correzioni. Per alcune metriche, le azioni da intraprendere in caso di valori negativi sono descritte nelle *Norme di progetto v1.0.0*;
- **accettabile:** obiettivo raggiunto, soglia di accettabilità superata; Se il criterio a cui il valore si riferisce è un obiettivo importante in ottica *PDCA<sub>G</sub>*, è necessario attuare azioni per migliorare la qualità;
- **ottimale:** obiettivo raggiunto, soglia di ottimalità superata. Anche in questo caso vale quanto detto nel punto precedente riguardo il miglioramento continuo.

### 3.2.2 Metriche per i processi

#### 3.2.2.1 Miglioramento costante

Per misurare l'efficacia del miglioramento costante, il *team<sub>G</sub>* ha deciso di utilizzare la metrica fornita dal modello *CMM<sub>G</sub>*. Per ogni *fase<sub>G</sub>*, verrà misurata la qualità dei processi. La scala assume valori da 1 (peggiore) a 5 (migliore).

- **metrica utilizzata:** Livello *CMM<sub>G</sub>* (*LCMM*);
- **valore negativo:** 1;
- **valore accettabile:** 2 e 3;
- **valore ottimale:** 4 e 5.

Questa metrica fa riferimento all'obiettivo riportato alla sezione [2.1.1.1](#).

#### 3.2.2.2 Rispetto della pianificazione

La metrica utilizzata è la Schedule Variance. È implementata come differenza tra la pianificazione dei costi del lavoro eseguito e del lavoro pianificato. Entrambi questi valori sono intesi nella loro accezione temporale (giorni) e non monetaria.

- **metrica utilizzata:**

$$SV = BCWP - BCWS$$

dove *BCWP*=Budgeted Cost of Work Performed e *BCWS*=Budgeted Cost of Work Scheduled;

- **valore negativo:**  $SV > 5$  giorni;
- **valore accettabile:**  $0 < SV \leq 4$  giorni;
- **valore ottimale:**  $SV \leq 0$  giorni.

Questa metrica fa riferimento all'obiettivo riportato alla sezione [2.1.1.2](#).

### 3.2.2.3 Rispetto del budget

La metrica utilizzata è la Cost Variance. È implementata come differenza tra costo pianificato e costo effettivo del lavoro eseguito.

- **metrica utilizzata:**

$$CV = BCWP - ACWP$$

dove  $BCWP$ =Budgeted Cost of Work Performed e  $ACWP$ =Actual Cost of Work Performed;

- **valore negativo:**  $CV > 10\%$ ;
- **valore accettabile:**  $0\% < CV \leq 10\%$ ;
- **valore ottimale:**  $CV \leq 0\%$ .

Questa metrica fa riferimento all'obiettivo riportato alla sezione [2.1.1.3](#).

### 3.2.2.4 Completezza dell'analisi dei rischi

- **metrica utilizzata:** contatore  $RNP$  (Rischi Non Preventivati) che aumenta di 1 ogni volta che si presenta un rischio non preventivato nel *Piano di progetto v1.0.0*. Il contatore non si resetta al cambiamento di *fase*: è globale di progetto;
- **valore negativo:**  $RNP > 2$ ;
- **valore accettabile:**  $1 \leq RNP \leq 2$ ;
- **valore ottimale:**  $RNP = 0$ .

Questa metrica fa riferimento all'obiettivo riportato alla sezione [2.1.1.4](#).

## 3.2.3 Metriche per i prodotti

In questa sezione sono spiegate nel dettaglio le metriche scelte per la valutazione dei prodotti.

### 3.2.3.1 Metriche per i documenti

Le metriche che il *teamG* ha scelto per i documenti sono il più oggettive possibili. Da sole non garantiscono la qualità generale del documento, quindi è necessario un'ulteriore e accurata verifica, soprattutto per evitare gli errori concettuali e di forma.

#### 3.2.3.1.1 Leggibilità e comprensibilità

La metrica utilizzata è l'*Indice GulpeaseG* ( $IG$ ), un indice di leggibilità di un testo tarato sulla lingua italiana. La scala va da 0 a 100, dove "0" indica un documento di bassa leggibilità e "100" uno di alta. Risulta che i 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.

Il *teamG* desidera che i testi siano comprensibili da persone con almeno un diploma superiore.



- **metrica utilizzata:**

$$IG = 89 + \frac{300 \cdot NF - 10 \cdot NL}{NP}$$

dove  $NF$  è il numero di frasi,  $NL$  è il numero di lettere e  $NP$  è il numero di parole presenti nel testo.

- **valore negativo:**  $IG < 40$ ;
- **valore accettabile:**  $40 \leq IG < 60$ ;
- **valore ottimale:**  $IG \geq 60$ .

Questa metrica fa riferimento all'obiettivo riportato alla sezione [2.1.2.1.1](#).

#### 3.2.3.1.2 Adesione alle norme interne

La metrica utilizzata è il numero di Errori riguardanti le Norme interne rinvenuti e Non Corretti ( $ENNC$ ). È implementata con un contatore che aumenta di 1 ogni volta che un errore riguardante le norme interne rilevato da un *Verificatore* non viene corretto. Il contatore fa riferimento ad una specifica *fase<sub>G</sub>* e viene azzerato successivamente.

- **metrica utilizzata:**  $ENNC$ ;
- **valore negativo:**  $ENNC > 0$ ;
- **valore accettabile:**  $ENNC = 0$ ;
- **valore ottimale:**  $ENNC = 0$ .

Questa metrica fa riferimento all'obiettivo riportato alla sezione [2.1.2.2](#).

#### 3.2.3.1.3 Correttezza ortografica

La metrica utilizzata è un contatore di Errori Ortografici Non Corretti ( $EONC$ ), che aumenta di 1 ogni volta che un errore ortografico rilevato da un *Verificatore* non viene corretto. Il contatore fa riferimento ad una specifica *fase<sub>G</sub>* e viene azzerato successivamente.

- **metrica utilizzata:**  $EONC$ ;
- **valore negativo:**  $EONC > 0$ ;
- **valore accettabile:**  $EONC = 0$ ;
- **valore ottimale:**  $EONC = 0$ .

Questa metrica fa riferimento all'obiettivo riportato alla sezione [2.1.2.2.1](#).

#### 3.2.3.1.4 Correttezza concettuale

La metrica utilizzata è chiamata Errori Concettuali Non Corretti ( $ECNC$ ). La formula è data dal complemento a 1 del rapporto tra errori concettuali corretti e rilevati. Tali errori possono essere rilevati dai Verificatori o dal committente.

- **metrica utilizzata:**

$$ECNC = \left(1 - \frac{ECC}{ECR}\right) \cdot 100$$

dove  $ECC$ =Errori Concettuali Corretti e  $ECR$ =Errori Concettuali Rinvenuti;

- **valore negativo:**  $ECNC > 5\%$ ;
- **valore accettabile:**  $ECNC \leq 5\%$ ;
- **valore ottimale:**  $ECNC = 0\%$ .

Questa metrica fa riferimento all'obiettivo riportato alla sezione [2.1.2.2.2](#).

### 3.2.3.2 Metriche per il software

Le metriche che il *team<sub>G</sub>* ha scelto per misurare la qualità del software si possono riferire a:

- qualità interna;
- qualità esterna;
- qualità in uso.

Come descritto nello standard *ISO<sub>G</sub>/IEC<sub>G</sub> 9126:2001*, ogni metrica corrisponde a determinate caratteristiche di qualità. Segue una tabella che associa ogni metrica scelta alla relativa caratteristica. Inoltre è specificato a quale tipologia di qualità essa fa riferimento.

Obiettivo	Metrica	Tipo	Caratteristica
Implementazione funzionalità obbligatorie	IFO	Esterna	Funzionalità
Implementazione funzionalità desiderabili	IFD	Esterna	Funzionalità
Numero di statement per metodo	NSM	Interna	Manutenibilità
Numero di parametri per metodo	NPM	Interna	Manutenibilità
Numero di campi dati per classe	NCD	Interna	Manutenibilità
Grado di accoppiamento	GA	Interna	Manutenibilità
Complessità ciclomatica	NC	Interna	Manutenibilità
Variabili dichiarate e non utilizzate	NVNU	Interna	Manutenibilità
Documentazione del codice	LCC	Interna	Manutenibilità
Validazione web	NEV	Interna	Manutenibilità
Copertura dei test richiesti	CTR	Interna	Affidabilità
Robustezza	BA	Esterna	Affidabilità
Correzione delle situazioni di fallimento	FA	Esterna	Affidabilità

Tabella 1: Mappa Metriche-Caratteristiche

#### 3.2.3.2.1 Implementazione delle funzionalità obbligatorie

La metrica utilizzata è chiamata Implementazione delle Funzionalità Obbligatorie (*IFO*). Consiste in un rapporto tra il numero di requisiti obbligatori soddisfatti e identificati

- **metrica utilizzata:**

$$IFO = \frac{ROS}{ROI} \cdot 100$$

dove *ROS*=numero di Requisiti Obbligatori Soddisfatti e *ROI*=numero di Requisiti Obbligatori Identificati;

- **valore negativo:**  $IFO < 100\%$ ;
- **valore accettabile:**  $IFO = 100\%$ ;
- **valore ottimale:**  $IFO = 100\%$ .

Questa metrica fa riferimento all'obiettivo riportato alla sezione [2.1.2.3.1](#).

### 3.2.3.2.2 Implementazione delle funzionalità desiderabili

La metrica utilizzata è chiamata Implementazione delle Funzionalità Desiderabili (*IFD*). Consiste in un rapporto tra il numero di requisiti desiderabili soddisfatti e identificati

- **metrica utilizzata:**

$$IFD = \frac{RDS}{RDI} \cdot 100$$

dove *RDS*=numero di Requisiti Desiderabili Soddisfatti e *RDI*=numero di Requisiti Desiderabili Identificati;

- **valore negativo:**  $IFD < 100\%$ ;
- **valore accettabile:**  $IFD = 100\%$ ;
- **valore ottimale:**  $IFD = 100\%$ .

Questa metrica fa riferimento all'obiettivo riportato alla sezione [2.1.2.3.2](#).

Le seguenti metriche fanno riferimento all'obiettivo di [Manutenibilità e comprensibilità del codice](#) riportato alla sezione [2.1.2.3.3](#).

### 3.2.3.2.3 Numero di statement per metodo

- **metrica utilizzata:** Numero di *Statement<sub>G</sub>* per Metodo (*NSM*);
- **valore negativo:**  $NSM > 60$ ;
- **valore accettabile:**  $30 < NSM \leq 60$ ;
- **valore ottimale:**  $NSM \leq 30$ .

### 3.2.3.2.4 Numero di parametri per metodo

- **metrica utilizzata:** Numero di Parametri per Metodo (*NPM*);
- **valore negativo:**  $NPM > 12$ ;
- **valore accettabile:**  $5 < NPM \leq 12$ ;
- **valore ottimale:**  $NPM \leq 5$ .

### 3.2.3.2.5 Numero di campi dati per classe

- **metrica utilizzata:** Numero Campi Dati Per Classe (*NCDPC*);
- **valore negativo:**  $NCDPC > 15$ ;
- **valore accettabile:**  $10 < NCDPC \leq 15$ ;
- **valore ottimale:**  $NCDPC \leq 10$ .

### 3.2.3.2.6 Grado di accoppiamento

Il grado di accoppiamento ( $GA$ ) è calcolato in base al numero di dipendenze tra classi in un [package<sub>G</sub>](#).

- **metrica utilizzata:** Grado di Accoppiamento ( $GA$ );
- **valore negativo:**  $GA > 10$ ;
- **valore accettabile:**  $3 < GA \leq 10$ ;
- **valore ottimale:**  $GA \leq 3$ .

### 3.2.3.2.7 Complessità ciclomatica

La metrica è basata sul numero ciclomatico ( $NC$ ), che rappresenta il numero di cammini linearmente indipendenti presenti all'interno del codice.

- **metrica utilizzata:**

$$NC = e - n + 2p$$

con  $e$ =numero di archi,  $n$ =numero di nodi,  $p$ =numero di componenti connesse;

- **valore negativo:**  $NC > 20$ ;
- **valore accettabile:**  $10 < NC \leq 20$ ;
- **valore ottimale:**  $NC \leq 10$ .

### 3.2.3.2.8 Numero di variabili dichiarate e non utilizzate

- **metrica utilizzata:** Numero di Variabili dichiarate e Non Utilizzate ( $NVNU$ );
- **valore negativo:**  $NVNU > 0$ ;
- **valore accettabile:**  $NVNU = 0$ ;
- **valore ottimale:**  $NVNU = 0$ .

### 3.2.3.2.9 Documentazione del codice

La metrica è implementata come Rapporto tra le linee di Commento e le linee di Codice ( $RCC$ ).

- **metrica utilizzata:**

$$RCC = \left( \frac{LDCM}{LDCC} \right) \cdot 100$$

dove  $LDCC$ =Linee Di Codice e  $LDCM$ =Linee Di Commento.

- **valore negativo:**  $RCC < 10\%$ ;
- **valore accettabile:**  $RCC \geq 10\%$ ;
- **valore ottimale:**  $RCC \geq 30\%$ .

Questa metrica fa riferimento all'obiettivo riportato alla sezione [2.1.2.3.4](#).

### 3.2.3.2.10 Validazione web

La metrica è basata sul Numero di Errori di Validazione (*NEV*) rilevati dal validatore online del [W3C](#). Eventuali errori causati dall'utilizzo di librerie esterne non verranno presi in considerazione.

- **metrica utilizzata:** *NEV*;
- **valore negativo:**  $NEV > 10$ ;
- **valore accettabile:**  $0 < NEV \leq 10$ ;
- **valore ottimale:**  $NEV = 0$ .

Questa metrica fa riferimento all'obiettivo riportato alla sezione [2.1.2.3.5](#).

### 3.2.3.2.11 Copertura dei test richiesti

La metrica utilizzata è la copertura dei test richiesti (*CTR*). I test presi in considerazione sono quelli necessari a verificare l'implementazione delle funzionalità previste dai requisiti.

- **metrica utilizzata:**

$$CTR = \frac{TS}{TR} \cdot 100$$

dove *TS*=numero di Test Superati e *TR*=numero di Test Richiesti;

- **valore negativo:**  $CTR < 80\%$ ;
- **valore accettabile:**  $80 \leq CTR < 90\%$ ;
- **valore ottimale:**  $90\% \leq CTR \leq 100\%$ .

Questa metrica fa riferimento all'obiettivo riportato alla sezione [2.1.2.3.6](#).

### 3.2.3.2.12 Robustezza

La metrica utilizzata è la Breakdown Avoidance (*BA*).

- **metrica utilizzata:**

$$BA = \left(1 - \frac{NI}{NSA}\right) \cdot 100$$

dove *NI*=Numero di Interruzioni e *NSA*=Numero di Situazioni Anomale;

- **valore negativo:**  $BA < 80\%$ ;
- **valore accettabile:**  $80 \leq BA < 95\%$ ;
- **valore ottimale:**  $BA \geq 95\%$ .

Questa metrica fa riferimento all'obiettivo riportato alla sezione [2.1.2.3.7](#).

### 3.2.3.2.13 Correzione delle situazioni di fallimento

La metrica utilizzata è la Failure Avoidance ( $FA$ ).

- **metrica utilizzata:**

$$FA = \left( \frac{SAE}{SAT} \right) \cdot 100$$

con  $SAE$ =Situazioni Anomale Evitate e  $SAT$ =Situazioni Anomale Testate;

- **valore negativo:**  $FA < 80\%$ ;
- **valore accettabile:**  $80 \leq FA < 95\%$ ;
- **valore ottimale:**  $FA \geq 95\%$ .

Questa metrica fa riferimento all'obiettivo riportato alla sezione [2.1.2.3.8](#).

## A CMM

Il **CMM<sub>G</sub>** (Capability Maturity Model) è un modello che mira a migliorare e rifinire i processi software di un'organizzazione. Il modello descrive un percorso evolutivo a cinque livelli riguardante processi sempre più maturi e organizzati.

Il **CMM<sub>G</sub>** è stato sviluppato dal **SEI<sub>G</sub>** e promosso e finanziato dal Dipartimento della Difesa statunitense per valutare la qualità dei processi software delle organizzazioni che collaboravano con esso.

Il modello fornisce:

- una base concettuale a cui appoggiarsi per valutare il livello dei processi;
- un insieme di best practices consolidate negli anni da esperti e utilizzatori;
- un linguaggio comune e una visione condivisa;
- un metodo per definire un miglioramento in ambito organizzativo.

### A.1 Struttura

Il modello è costituito da cinque aspetti:

- **livelli di maturità:** il **CMM<sub>G</sub>** identifica un processo continuo di maturazione a cinque livelli (in cui il maggiore è uno stato ideale dove i processi sono sistematicamente gestiti da una combinazione di ottimizzazione e miglioramento del processo);
- **area chiave di processo:** identifica un insieme di attività correlate che, quando eseguite assieme, raggiungono un insieme di obiettivi considerati importanti;
- **obiettivi:** gli obiettivi di un'area chiave di processo riassumono gli stati che devono sussistere affinché tale area sia implementata in modo efficace e duraturo. La quantità di obiettivi soddisfatti indica il livello di capability raggiunto dall'organizzazione in un dato livello di maturità. Gli obiettivi denotano l'ambito, i limiti e lo scopo di ogni area chiave di processo;
- **caratteristiche comuni:** includono le pratiche che implementano e regolamentano le aree chiave di processo. Esistono cinque tipi di caratteristiche comuni:
  - impegno nell'esecuzione;
  - abilità nell'esecuzione;
  - attività eseguite;
  - misurazioni e analisi;
  - verifica e implementazione.
- **pratiche chiave:** descrivono gli elementi di infrastruttura e prassi che contribuiscono all'implementazione e regolamentazione dell'area.

### A.2 Livelli

Sono presenti cinque livelli:

- **livello 1 - Iniziale:** i processi in questo livello hanno la tendenza ad essere non documentati e in uno stato di continuo cambiamento. Date queste caratteristiche, l'esito molto spesso dipende dallo sforzo dei singoli e non si considera essere ripetibile. I processi vengono riadattati di volta in volta, risultando caotici e scarsamente controllabili;
- **livello 2 - Ripetibile:** I processi di questo livello sono generalmente ripetibili, eventualmente con buoni risultati. La disciplina, se presente, pur non essendo rigorosa, aiuta a sostenere i processi durante i periodi di elevato carico di lavoro;
- **livello 3 - Definito:** i processi cominciano ad essere standardizzati, in quanto la disciplina è più rigorosa e la documentazione più completa. Inoltre sono soggetti ad certo livello di miglioramento nel lungo periodo;
- **livello 4 - Gestito:** i processi sono controllati quantitativamente in accordo alle metriche di processo prestabilite. L'amministrazione aziendale può adeguare e adattare i processi a particolari progetti senza perdite sostanziali di qualità o deviazioni dalle specifiche;
- **livello 5 - Ottimizzato:** i processi in questo livello hanno come obiettivo il miglioramento continuo delle loro performance attraverso miglioramenti tecnologici sia incrementali che innovativi.



## B ISO/IEC 9126

Lo standard *ISO/IEC* 9126 prevede una serie di normative e linee guida nate dalla collaborazione tra *ISO* e *IEC* per descrivere un modello della qualità del software.

Esso si suddivide in quattro parti:

- modello della qualità del software (9126-1);
- metriche per la qualità esterna (9126-2);
- metriche per la qualità interna (9126-3);
- metriche per la qualità in uso (9126-4).

### B.1 Modello della qualità del software

Il modello della qualità del software viene definito nella prima parte dello standard e viene suddiviso in:

- modello della qualità esterna e interna;
- modello della qualità in uso.

Le caratteristiche contenute in tali modelli sono misurabili attraverso l'utilizzo di metriche.

#### B.1.1 Modello della qualità esterna e interna

Il modello della qualità esterna e interna classifica la qualità del software con sei caratteristiche generali:

- **funzionalità:** rappresenta la capacità del prodotto software di fornire le funzioni necessarie per operare in determinate condizioni, cioè in un determinato contesto;
- **affidabilità:** rappresenta la capacità del prodotto software di mantenere un certo livello di prestazioni quando viene usato in condizioni specifiche e per un intervallo di tempo fissato;
- **usabilità:** rappresenta la capacità del prodotto software di essere comprensibile. Un software è considerato usabile in proporzione alla facilità con cui gli utenti operano per sfruttare a pieno le funzionalità che il software realizza;
- **efficienza:** rappresenta la capacità del prodotto software di realizzare le funzioni richieste nel minor tempo possibile, utilizzando le risorse a disposizione nel miglior modo possibile;
- **manutenibilità:** rappresenta la capacità del prodotto software di essere modificato (a costi accessibili e in tempi rapidi). Le modifiche possono includere correzioni, adattamenti o migliorie del software; Le ultime possono essere richieste in seguito a cambiamenti nell'ambiente, nei requisiti o nelle specifiche funzionali;
- **portabilità:** rappresenta la capacità del prodotto software di poter essere trasportato da un ambiente all'altro (in modo sufficientemente veloce). L'ambiente include aspetti hardware e software.

### B.1.2 Modello della qualità in uso

Gli attributi del modello della qualità in uso vengono suddivisi nelle seguenti quattro categorie:

- **efficacia:** rappresenta la capacità del prodotto software di permettere all'utente di raggiungere obiettivi specifici con accuratezza e completezza in uno specifico contesto d'utilizzo;
- **produttività:** rappresenta la capacità del prodotto software di permettere all'utente di impiegare un numero definito di risorse, in relazione all'efficienza raggiunta in uno specifico contesto di utilizzo;
- **sicurezza fisica:** rappresenta la capacità del prodotto software di raggiungere un livello accettabile di rischio per i dati, le persone, il business, la proprietà o gli ambienti in uno specifico contesto di utilizzo;
- **soddisfazione:** rappresenta la capacità del prodotto software di soddisfare gli utenti in uno specifico contesto di utilizzo.

### B.2 Qualità esterna e relative metriche

È la qualità del prodotto software vista dall'esterno nel momento in cui esso viene eseguito e testato in un ambiente di prova. Le metriche associate ne misurano i comportamenti rilevabili:

- dai test;
- dall'operabilità;
- dall'osservazione in un contesto specifico.

Tali metriche vengono selezionate sulla base delle caratteristiche che il prodotto finale dovrà dimostrare durante la sua esecuzione.

### B.3 Qualità interna e relative metriche

È la qualità del prodotto software vista dall'interno e fa riferimento alle caratteristiche implementative quali la sua architettura e il codice che ne deriva.

Le metriche associate si applicano al software non eseguibile (es: il codice sorgente) e alla documentazione. Le misure effettuate permettono di prevedere il livello di qualità esterna ed in uso del prodotto finale poiché gli attributi interni influenzano le caratteristiche esterne e quelle in uso.

### B.4 Qualità in uso e relative metriche

È la qualità del prodotto software dal punto di vista dell'utilizzatore che ne fa uso all'interno di uno specifico sistema e contesto. Le metriche associate misurano il grado con cui il prodotto software permette agli utenti di svolgere, in un contesto operativo specifico, le proprie attività in modo:

- efficace;
- produttivo;
- sicuro;
- soddisfacente.

## C PDCA

Il **PDCA<sub>G</sub>** (conosciuto anche come "Ciclo di Deming" o "Ciclo di Shewhart" ) è un metodo per la gestione delle attività di processo ripetibili e misurabili e per la manutenibilità dei processi stessi. È un metodo iterativo suddiviso in quattro **fasi<sub>G</sub>** (Plan-Do-Check-Act, da cui l'acronimo) e assicura un non decremento della qualità ad ogni ciclo. Fissati degli obiettivi di miglioramento desiderati si iterano le attività previste dal **PDCA<sub>G</sub>** fino al raggiungimento degli stessi. I miglioramenti ai quali si fa riferimento sono legati all'efficienza e all'efficacia. Migliorare l'efficienza significa usare meno risorse per fare lo stesso lavoro. Migliorare l'efficacia significa divenire più conformi alle aspettative.

### C.1 Fasi

Sono presenti quattro **fasi<sub>G</sub>**:

- **Plan:** vengono definiti gli obiettivi di miglioramento, le strategie da utilizzare per perseguire tali obiettivi e il modo in cui queste verranno utilizzate.  
Per far ciò si svolgono i seguenti passi:
  1. si svolge una prima **fase<sub>G</sub>** di identificazione del problema (ad esempio un processo da migliorare) nella quale saranno raccolti dei dati in seguito a delle misurazioni;
  2. viene analizzato il problema e vengono individuati gli aspetti negativi, decidendone la loro importanza e le priorità di intervento;
  3. vengono definiti gli obiettivi di massima in modo chiaro e quantitativo, indicando i benefici ottenibili con il loro raggiungimento. Vengono inoltre specificati i tempi necessari per la loro attuazione, gli indicatori e gli strumenti di controllo necessari.
- **Do:** viene attuato ciò che è stato pianificato per risolvere il problema. Nello stesso tempo si devono anche raccogliere i dati necessari all'analisi che verrà svolta in seguito;
- **Check:** consiste nel verificare i risultati ottenuti (per efficienza ed efficacia) in seguito all'attuazione delle strategie di miglioramento. Essi saranno analizzati e studiati (anche attraverso grafici e tabelle riassuntive) in modo tale da avere una visione chiara di quanto rilevato. Se gli obiettivi sono stati raggiunti, ovvero se è avvenuto un miglioramento, si può passare alla **fase<sub>G</sub>** successiva; in caso contrario è necessario ripetere il ciclo **PDCA<sub>G</sub>** sullo stesso problema, analizzando gli stadi del ciclo precedente e individuando le cause del mancato raggiungimento degli obiettivi stabiliti.

L'esito del processo può essere di tre tipi:

- miglioramento secondo le aspettative;
  - miglioramento superiore alle aspettative;
  - miglioramento inferiore alle aspettative.
- **Act:** i miglioramenti individuati vengono regolamentati e integrati nello standard dell'organizzazione e tutti i membri del **gruppo<sub>G</sub>** vengono informati e conseguentemente formati. Verrà quindi eseguita una nuova iterazione dell'intero ciclo.

## D Test

I test, eseguiti tramite analisi dinamica, sono attività che servono a verificare che il software prodotto implementi le funzionalità richieste. Una caratteristica fondamentale dei test è la ripetibilità: i risultati che essi forniscono devono essere deterministici, in modo da eseguire azioni correttive in caso gli esiti non siano quelli attesi. Per tracciare i test eseguiti e i risultati ottenuti sarà necessario produrre dei log di facile consultazione.

I test verranno specificati quando sarà cominciata la *fase<sub>G</sub>* di progettazione.

### D.1 Test di accettazione

I test di accettazione, eseguiti durante il collaudo finale, servono a verificare che il software soddisfi le richieste del proponente.

### D.2 Test di sistema

I test di sistema servono a verificare il corretto funzionamento delle componenti dell'intero sistema.

### D.3 Test di integrazione

I test di integrazione servono a verificare il corretto funzionamento di più unità. Più precisamente, l'obiettivo è quello di testare i vari *package<sub>G</sub>*, sia singolarmente che nel loro insieme.

### D.4 Test di unità

I test di unità servono a verificare il corretto funzionamento della singola unità, ovvero della più piccola parte di lavoro realizzabile dal singolo programmatore.

## E Resoconto delle attività di verifica -Fase Analisi

### E.1 Verifica dei processi

Segue una tabella riassuntiva riguardante le metriche di processo. Le righe che riportano una serie di trattini orizzontali sotto la voce "Processo" sono relative all'intera *fase<sub>G</sub>* e non al singolo processo. Per avere informazioni dettagliate sugli scopi dei processi e sulle attività che li compongono, consultare le *Norme di progetto v1.0.0*.

Per avere un resoconto testuale degli obiettivi in tabella, fare click sul nome dell'obiettivo.

Per una descrizione delle metriche in tabella, fare click sul nome della metrica.

Processo	Obiettivo	Metrica	Valore	Giudizio
Fornitura	Miglioramento costante	LCMM	2	Accettabile
Sviluppo	Miglioramento costante	LCMM	2	Accettabile
Documentazione	Miglioramento costante	LCMM	2	Accettabile
Verifica	Miglioramento costante	LCMM	2	Accettabile
Gestione delle infrastrutture	Miglioramento costante	LCMM	2	Accettabile
Gestione dei processi	Miglioramento costante	LCMM	2	Accettabile
Apprendimento	Miglioramento costante	LCMM	2	Accettabile
- - - - -	Rispetto della pianificazione	SV	3 giorni	Accettabile
- - - - -	Rispetto del budget	CV	0%	Ottimale
- - - - -	Completezza dell'analisi dei rischi	RNP	0 rischi	Ottimale

Tabella 2: Resoconto metriche di processo

#### E.1.1 Considerazioni finali

##### E.1.1.1 Miglioramento costante

Il livello *CMM<sub>G</sub>* dei processi di Fornitura, Sviluppo, Documentazione, Verifica, Gestione delle infrastrutture, Gestione di Processo e Apprendimento in questa *fase<sub>G</sub>* è pari a 2. Dopo lo stato iniziale, durato quasi fino a metà della *fase<sub>G</sub>*, in cui i processi si trovavano in uno stato caotico, il rispetto delle *Norme di progetto v1.0.0* e l'adozione di strumenti automatici ha portato ad un guadagno di ripetibilità. Alcuni esempi di tali strumenti sono i correttori ortografici e lo script per il calcolo dell'indice di leggibilità per quanto riguarda i processi di Documentazione e Verifica e l'utilizzo di *Trender<sub>G</sub>* per quanto riguarda l'attività di analisi dei requisiti del processo di Sviluppo.

Tutti i processi non sono standardizzati ad un livello tale da raggiungere il livello 3 della scala. Inoltre, la disciplina non è ancora molto rigorosa. L'obiettivo per le prossime *fasi<sub>G</sub>* è migliorare tale livello.

##### E.1.1.2 Rispetto della pianificazione

Il ritardo riscontrato nella *fase<sub>G</sub>* di Analisi è pari a 3 giorni. Dato che il ritardo è all'interno della soglia di accettabilità, il *team<sub>G</sub>* è ancora in grado di rispettare la scadenza. L'obiettivo è cercare di evitare ritardi nelle *fasi<sub>G</sub>* successive.

##### E.1.1.3 Rispetto del budget

Non sono state riscontrate spese aggiuntive. La metrica assume quindi un valore ottimale.

#### E.1.1.4 Completezza dell'analisi dei rischi

Dall'inizio del progetto non sono sorti rischi non preventivati, pertanto la metrica assume un valore ottimale.

### E.2 Verifica dei prodotti

#### E.2.1 Verifica dei documenti

I documenti sono stati analizzati principalmente tramite *walkthrough<sub>G</sub>* data la scarsa esperienza dei verificatori. Gli errori più ricorrenti sono stati annotati e serviranno a creare una lista per la successiva attività di verifica, da effettuare utilizzando *inspection<sub>G</sub>*.

Seguono tabelle riassuntive riguardante le metriche relative ai documenti.

Per una descrizione delle metriche in tabella, fare click sul nome della metrica.

##### E.2.1.1 Leggibilità e comprensibilità

Documento	Metrica	Valore	Giudizio
<i>Piano di progetto v1.0.0</i>	IG	58	Accettabile
<i>Piano di qualifica v1.0.0</i>	IG	58	Accettabile
<i>Norme di progetto v1.0.0</i>	IG	61	Ottimale
<i>Studio di fattibilità v1.0.0</i>	IG	52	Accettabile
<i>Analisi dei requisiti v1.0.0</i>	IG	45	Accettabile
<i>Glossario v1.0.0</i>	IG	56	Accettabile
<i>VerbaleInterno_1_20161202</i>	IG	79	Ottimale
<i>VerbaleInterno_2_20161220</i>	IG	79	Ottimale
<i>VerbaleInterno_3_20161227</i>	IG	79	Ottimale
<i>VerbaleInterno_4_20170103</i>	IG	79	Ottimale
<i>VerbaleEsterno_1_20161203</i>	IG	73	Ottimale
<i>VerbaleEsterno_2_20161227</i>	IG	68	Ottimale

Tabella 3: Resoconto leggibilità e comprensibilità

##### E.2.1.1.1 Considerazioni finali

Tutti i documenti presentano un *indice Gulpease<sub>G</sub>* ad un livello almeno accettabile; ciò dovrebbe garantire una lettura non particolarmente difficoltosa da parte di soggetti con almeno licenza superiore. Il documento che assume il valore più basso è l'*Analisi dei requisiti v1.0.0*. Questo è dovuto al fatto che esso è un documento particolarmente tecnico e i contenuti sono esposti sotto forma di tabelle.

### E.2.1.2 Adesione alle norme interne

Documento	Metrica	Valore	Giudizio
<i>Piano di progetto v1.0.0</i>	ENNC	0	Ottimale
<i>Piano di qualifica v1.0.0</i>	ENNC	0	Ottimale
<i>Norme di progetto v1.0.0</i>	ENNC	0	Ottimale
<i>Studio di fattibilità v1.0.0</i>	ENNC	0	Ottimale
<i>Analisi dei requisiti v1.0.0</i>	ENNC	0	Ottimale
<i>Glossario v1.0.0</i>	ENNC	0	Ottimale
<i>VerbaleInterno_1_20161202</i>	ENNC	0	Ottimale
<i>VerbaleInterno_2_20161220</i>	ENNC	0	Ottimale
<i>VerbaleInterno_3_20161227</i>	ENNC	0	Ottimale
<i>VerbaleInterno_4_20170103</i>	ENNC	0	Ottimale
<i>VerbaleEsterno_1_20161203</i>	ENNC	0	Ottimale
<i>VerbaleEsterno_2_20161227</i>	ENNC	0	Ottimale

Tabella 4: Resoconto adesione alle norme interne

#### E.2.1.2.1 Considerazioni finali

Per tutti i documenti non risultano errori residui che violino le norme interne, pertanto le metriche hanno un valore ottimale.

### E.2.1.3 Correttezza ortografica

Documento	Metrica	Valore	Giudizio
<i>Piano di progetto v1.0.0</i>	EONC	0	Ottimale
<i>Piano di qualifica v1.0.0</i>	EONC	0	Ottimale
<i>Norme di progetto v1.0.0</i>	EONC	0	Ottimale
<i>Studio di fattibilità v1.0.0</i>	EONC	0	Ottimale
<i>Analisi dei requisiti v1.0.0</i>	EONC	0	Ottimale
<i>Glossario v1.0.0</i>	EONC	0	Ottimale
<i>VerbaleInterno_1_20161202</i>	EONC	0	Ottimale
<i>VerbaleInterno_2_20161220</i>	EONC	0	Ottimale
<i>VerbaleInterno_3_20161227</i>	EONC	0	Ottimale
<i>VerbaleInterno_4_20170103</i>	EONC	0	Ottimale
<i>VerbaleEsterno_1_20161203</i>	EONC	0	Ottimale
<i>VerbaleEsterno_2_20161227</i>	EONC	0	Ottimale

Tabella 5: Resoconto correttezza ortografica

#### E.2.1.3.1 Considerazioni finali

Dopo l'analisi automatica dei correttori ortografici e quella mediante [walkthrough<sub>G</sub>](#) da parte dei *Verificatori* non sono stati rilevati ulteriori errori che violano le norme interne, pertanto le metriche assumono un valore ottimale.

**E.2.1.4 Correttezza concettuale**

Documento	Metrica	Valore	Giudizio
<i>Piano di progetto v1.0.0</i>	ECNC	0	Ottimale
<i>Piano di qualifica v1.0.0</i>	ECNC	0	Ottimale
<i>Norme di progetto v1.0.0</i>	ECNC	0	Ottimale
<i>Studio di fattibilità v1.0.0</i>	ECNC	0	Ottimale
<i>Analisi dei requisiti v1.0.0</i>	ECNC	0	Ottimale
<i>Glossario v1.0.0</i>	ECNC	0	Ottimale
<i>VerbaleInterno_1_20161202</i>	ECNC	0	Ottimale
<i>VerbaleInterno_2_20161220</i>	ECNC	0	Ottimale
<i>VerbaleInterno_3_20161227</i>	ECNC	0	Ottimale
<i>VerbaleInterno_4_20170103</i>	ECNC	0	Ottimale
<i>VerbaleEsterno_1_20161203</i>	ECNC	0	Ottimale
<i>VerbaleEsterno_2_20161227</i>	ECNC	0	Ottimale

Tabella 6: Resoconto correttezza concettuale

**E.2.1.4.1 Considerazioni finali**

Per tutti i documenti non sono stati rilevati errori concettuali non corretti, pertanto le metriche assumono un valore ottimale.