



AtAVi

---

# Piano di qualifica v1.0

---

## Sommario

Documento contenente le strategie adottate dal gruppo Co.Code per garantire la qualità del prodotto AtAVi.

<b>Versione</b>	1.0
<b>Data di redazione</b>	2016-12-27
<b>Redazione</b>	Nicola Tintorri Andrea Magnan
<b>Verifica</b>	Simeone Pizzi
<b>Approvazione</b>	Luca Bertolini
<b>Uso</b>	Esterno
<b>Distribuzione</b>	prof. Tullio Vardanega prof. Riccardo Cardin Zero12

## Diario delle modifiche

Versione	Riepilogo	Autore	Ruolo	Data
0.0.9	Aggiunto resoconto fase AR	Andrea Magnan	<i>Analista</i>	2017-01-02
0.0.8	Aggiunta appendice A	Nicola Tintorri	<i>Analista</i>	2017-01-02
0.0.7	Aggiunte appendici B e C	Andrea Magnan	<i>Analista</i>	2016-12-31
0.0.6	Conclusa stesura sezione 3	Nicola Tintorri	<i>Analista</i>	2016-12-30
0.0.5	Inizio stesura sezione 3	Nicola Tintorri	<i>Analista</i>	2016-12-29
0.0.4	Conclusa stesura sezione 2	Andrea Magnan	<i>Analista</i>	2016-12-29
0.0.3	Inizio stesura sezione 2	Andrea Magnan	<i>Analista</i>	2016-12-28
0.0.2	Stesura introduzione	Nicola Tintorri	<i>Analista</i>	2016-12-27
0.0.1	Inizio stesura documento	Nicola Tintorri	<i>Analista</i>	2016-12-27

# Indice

<b>1</b>	<b>Introduzione</b>	<b>5</b>
1.1	Scopo del documento	5
1.2	Scopo del prodotto	5
1.3	Glossario	5
1.4	Riferimenti	5
1.4.1	Normativi	5
1.4.2	Informativi	5
<b>2</b>	<b>Visione generale della strategia di gestione della qualità</b>	<b>6</b>
2.1	Obiettivi qualitativi	6
2.1.1	Qualità di processo	6
2.1.1.1	Miglioramento costante - OPC1	6
2.1.1.2	Rispetto della pianificazione - OPC2	7
2.1.1.3	Rispetto del budget - OPC3	7
2.1.2	Qualità di prodotto	7
2.1.2.1	Qualità dei documenti	7
2.1.2.1.1	Leggibilità e comprensibilità - OPDD1	8
2.1.2.1.2	Correttezza ortografica - OPDD2	8
2.1.2.1.3	Correttezza concettuale - OPDD3	8
2.1.2.2	Qualità del software	9
2.1.2.2.1	Funzionalità obbligatorie - OPDS1	9
2.1.2.2.2	Funzionalità desiderabili - OPDS2	9
2.1.2.2.3	Copertura dei test richiesti - OPDS3	9
2.1.2.2.4	Robustezza - OPDS4	10
2.1.2.2.5	Funzionamento senza interruzioni - OPDS5	10
2.1.2.2.6	Manutenibilità e Comprensibilità del codice - OPDS6	10
2.2	Scadenze temporali	10
<b>3</b>	<b>La strategia di gestione della qualità nel dettaglio</b>	<b>11</b>
3.1	Risorse	11
3.1.1	Risorse necessarie	11
3.1.1.1	Risorse umane	11
3.1.1.2	Risorse hardware	11
3.1.1.3	Risorse software	11
3.1.2	Risorse disponibili	11
3.1.2.1	Risorse umane	11
3.1.2.2	Risorse hardware	11
3.1.2.3	Risorse software	12
3.2	Misure e metriche	12
3.2.1	Misure	12
3.2.2	Metriche per processi	12
3.2.2.1	Capability Maturity Model - MPC1	12
3.2.2.2	Schedule Variance - MPC2	12
3.2.2.3	Cost Variance - MPC3	13
3.2.3	Metriche per i prodotti	13
3.2.3.1	Metriche per i documenti	13
3.2.3.1.1	Indice di leggibilità - MPDD1	13
3.2.3.1.2	Errori ortografici rinvenuti e non corretti - MPDD2	14
3.2.3.1.3	Errori concettuali rinvenuti e non corretti - MPDD3	14
3.2.3.2	Metriche per il software	14
3.2.3.2.1	Copertura requisiti obbligatori - MPDS1	15
3.2.3.2.2	Copertura requisiti desiderabili - MPDS2	15
3.2.3.2.3	Test passati richiesti - MPDS3	15

3.2.3.2.4	Failure Avoidance - MPDS4 . . . . .	15
3.2.3.2.5	Breakdown Avoidance - MPDS5 . . . . .	16
<b>A</b>	<b>Capability Maturity Model</b>	<b>17</b>
<b>B</b>	<b>Standard ISO/IEC 9126</b>	<b>18</b>
B.1	Modello della qualità del software . . . . .	18
B.1.1	Modello della qualità esterna ed interna . . . . .	18
B.1.2	Modello della qualità in uso . . . . .	18
B.2	Metriche per la qualità del software . . . . .	19
B.2.1	Metriche per la qualità esterna . . . . .	19
B.2.2	Metriche per la qualità interna . . . . .	19
B.2.3	Metriche per la qualità in uso . . . . .	19
<b>C</b>	<b>PDCA</b>	<b>20</b>
<b>D</b>	<b>Test</b>	<b>21</b>
<b>E</b>	<b>Resoconto delle attività di verifica - fase AR</b>	<b>22</b>
E.1	Verifica sui processi . . . . .	22
E.1.1	Processo di documentazione . . . . .	22
E.1.1.1	Miglioramento costante . . . . .	22
E.1.1.2	Rispetto della pianificazione . . . . .	22
E.1.1.3	Rispetto del budget . . . . .	22
E.1.2	Processo di verifica . . . . .	23
E.1.2.1	Miglioramento costante . . . . .	23
E.1.2.2	Rispetto della pianificazione . . . . .	23
E.1.2.3	Rispetto del budget . . . . .	23
E.2	Verifica sui prodotti . . . . .	24
E.2.1	Documenti . . . . .	24
E.2.1.1	Leggibilità e comprensibilità . . . . .	24
E.2.1.2	Correttezza ortografica . . . . .	24
E.2.1.3	Correttezza concettuale . . . . .	24

## Elenco delle tabelle

1	Mappa metriche-caratteristiche . . . . .	14
2	Esiti del calcolo della Schedule Variance sul processo di documentazione durante la fase AR . . . . .	22
3	Esiti del calcolo della Cost Variance sul processo di documentazione durante la fase AR . . . . .	23
4	Esiti del calcolo della Schedule Variance sul processo di verifica durante la fase AR . . . . .	23
5	Esiti del calcolo della Cost Variance sul processo di verifica durante la fase AR . . . . .	23
6	Esiti del calcolo dell'indice Gulpease sui documenti della fase AR . . . . .	24
7	Errori ortografici rinvenuti durante la fase AR . . . . .	24
8	Errori concettuali rinvenuti durante la fase AR . . . . .	24

# 1 Introduzione

## 1.1 Scopo del documento

Il documento ha lo scopo di definire gli obiettivi di qualità e le strategie che il gruppo Co.Code adotterà per raggiungerli. Verrà inoltre illustrato come il gruppo affronterà le varie fasi di verifica per poter garantire il miglior risultato qualitativo possibile.

## 1.2 Scopo del prodotto

Si vuole creare un'applicazione web che permetta ad un ospite, in visita all'ufficio di Zero12, di interrogare un assistente virtuale per annunciare la propria presenza, avvisare l'interessato dell'arrivo dell'ospite sul sistema di comunicazione aziendale (Slack) e fornire un'attività di accoglienza.

## 1.3 Glossario

Allo scopo di evitare ogni ambiguità nel linguaggio e rendere più semplice e chiara la comprensione dei documenti, viene allegato il “*Glossario v1.0.0*”. Le parole in esso contenute scritte in corsivo e marcate con una 'g' a pedice (p.es. *Parola<sub>g</sub>*).

## 1.4 Riferimenti

### 1.4.1 Normativi

- Norme di progetto: “*Norme di Progetto v1.0.0*”;
- Standard ISO/IEC 9126:2001:  
[https://en.wikipedia.org/wiki/ISO/IEC\\_9126](https://en.wikipedia.org/wiki/ISO/IEC_9126);
- Capability Maturity Model (CMM):  
[https://en.wikipedia.org/wiki/Capability\\_Maturity\\_Model](https://en.wikipedia.org/wiki/Capability_Maturity_Model);
- Plan-Do-Check-Act (PDCA):  
<https://en.wikipedia.org/wiki/PDCA>.

### 1.4.2 Informativi

- Piano di progetto: “*Piano di Progetto v1.0.0*”;
- 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>;
- Indice Gulpease:  
[https://it.wikipedia.org/wiki/Indice\\_Gulpease](https://it.wikipedia.org/wiki/Indice_Gulpease).

## 2 Visione generale della strategia di gestione della qualità

### 2.1 Obiettivi qualitativi

In questa sezione vengono descritti gli obiettivi di qualità che il gruppo Co.Code decide di perseguire durante l'intero progetto. Ogni obiettivo viene definito in modo quantitativo per permettere al team di valutarne il raggiungimento. Vengono quindi fissati dei valori minimi che è obbligatorio superare per soddisfarlo e dei valori ottimali che ne rappresentano il pieno (ma non obbligatorio) conseguimento. A tale scopo vengono utilizzati modelli, metriche e standard.

Viene assegnato un codice identificativo ad ogni obiettivo, al fine di semplificarne il tracciamento con la metrica ad esso associata.

Il metodo di denominazione degli obiettivi è descritto in dettaglio nel documento “*Norme di Progetto v1.0.0*”.

#### 2.1.1 Qualità di processo

Da processi scadenti derivano prodotti scadenti. Quindi, la qualità di processo è un fattore indispensabile per garantire la qualità dei prodotti. Assicurarla, inoltre, permette di:

- favorire l'ottimizzazione delle risorse;
- migliorare la stima dei rischi;
- ridurre i costi.

Desideriamo che ogni processo possenga le seguenti caratteristiche ottimali:

- dovrebbe essere in grado di migliorarsi continuamente:
  - le sue performance sono costantemente misurabili;
  - deve perseguire sempre gli obiettivi quantitativi di miglioramento.
- dovrebbe rispettare i tempi indicati nel documento “*Piano di Progetto v1.0.0*”;
- dovrebbe rispettare i costi dichiarati nel documento “*Piano di Progetto v1.0.0*”;

Nelle sezioni successive vengono dichiarati gli obiettivi che il gruppo vuole perseguire. Per ognuno di essi, vengono definiti i criteri con cui si effettuano le misurazioni qualitative, specificando valori minimi e valori ottimali.

##### 2.1.1.1 Miglioramento costante - OPC1

Per rendere le performance dei processi costantemente migliorabili e perseguire gli obiettivi quantitativi di miglioramento si è deciso di utilizzare il modello CMM. Si vuole raggiungere come valore minimo il livello 2 di questa scala, mentre, come valore ottimale, si vuole raggiungere il livello 4. Riassumendo:

**Modello utilizzato:** CMM;

**Soglia di accettabilità:** livello 2 previsto da CMM;

**Soglia di ottimalità:** livello 4 previsto da CMM.

Per una più dettagliata descrizione del modello CMM consultare l'appendice A.

Per approfondire la scelta delle soglie di accettabilità e ottimalità consultare la metrica alla sezione (scrivere sezione).

### 2.1.1.2 Rispetto della pianificazione - OPC2

Per capire se l'attività di un processo rispetta i tempi stabiliti dalla pianificazione all'interno del "*Piano di Progetto v1.0.0*" viene utilizzata la metrica Schedule Variance. Si desidera, come soglia minima accettabile, che un processo sia in ritardo non più del 5% rispetto alla pianificazione. Sarebbe ottimale, invece, non avere ritardi rispetto alla pianificazione o, ancora meglio, essere in anticipo.

Riassumendo:

**Metrica utilizzata:** Schedule Variance;

**Soglia di accettabilità:** ritardo al massimo del 5% rispetto alla pianificazione;

**Soglia di ottimalità:** nessun ritardo (0%) o in anticipo rispetto alla pianificazione.

Per approfondire la scelta delle soglie di accettabilità e ottimalità consultare la metrica alla sezione (scrivere sezione).

### 2.1.1.3 Rispetto del budget - OPC3

Per capire se i costi di un processo rientrano nel budget stabilito dalla pianificazione all'interno del "*Piano di Progetto v1.0.0*" viene utilizzata la metrica Cost Variance. Si desidera, come soglia minima accettabile, che un processo non superi il 10% del budget pianificato. Sarebbe ottimale, invece, non superare i costi pianificati o, ancora meglio, spendere meno.

Riassumendo:

**Metrica utilizzata:** Cost Variance;

**Soglia di accettabilità:** costi non superiori al 10% rispetto alla pianificazione;

**Soglia di ottimalità:** costi pianificati (0%) o inferiori.

Per approfondire la scelta delle soglie di accettabilità e ottimalità consultare la metrica alla sezione (scrivere sezione).

## 2.1.2 Qualità di prodotto

Per garantire la migliore qualità del prodotto è necessario che i processi che lo producono abbiano alta qualità. Inoltre, il gruppo Co.Code cercherà di seguire lo standard ISO/IEC 9126:2001 (vedi appendice ...).

È prevista la realizzazione di due tipologie di prodotto: software e documenti. Nelle sezioni successive vengono dichiarati gli obiettivi di qualità di prodotto che il gruppo vuole perseguire, suddivisi per tipo. Per ognuno di essi, vengono definiti i criteri con cui si effettuano le misurazioni qualitative, specificando valori minimi e valori ottimali.

### 2.1.2.1 Qualità dei documenti

Gli obiettivi di qualità riguardanti i documenti prefissati dal gruppo Co.Code sono i seguenti:

- i documenti devono essere corretti a livello ortografico;
- i documenti devono essere corretti a livello concettuale;
- i documenti devono essere comprensibili da individui con licenza superiore.



Verranno ora descritti metriche e criteri utilizzati per garantire le caratteristiche sopra descritte, fissando valori minimi e valori ottimali.

#### **2.1.2.1.1 Leggibilità e comprensibilità - OPDD1**

Per determinare il grado di leggibilità e comprensibilità del documento, il gruppo ha deciso di utilizzare l'indice Gulpease. Si desidera come soglia minima accettabile un indice maggiore o uguale a 40 e, come soglia ottimale, un indice maggiore di 60.

Riassumendo:

**Metrica utilizzata:** indice Gulpease;

**Soglia di accettabilità:** indice maggiore o uguale a 40;

**Soglia di ottimalità:** indice maggiore di 60.

Per approfondire la scelta delle soglie di accettabilità e ottimalità consultare la metrica alla sezione (scrivere sezione).

#### **2.1.2.1.2 Correttezza ortografica - OPDD2**

Per determinare il grado di correttezza ortografica del documento, il gruppo ha deciso di utilizzare la seguente metrica: percentuale di errori ortografici rinvenuti e non corretti. Pertanto, la soglia minima accettabile e la soglia ottimale coincidono e corrispondono a una correzione totale degli errori rinvenuti.

Riassumendo:

**Metrica utilizzata:** percentuale di errori ortografici rinvenuti e non corretti;

**Soglia di accettabilità:** tutti gli errori ortografici rinvenuti sono stati corretti (0%);

**Soglia di ottimalità:** tutti gli errori ortografici rinvenuti sono stati corretti (0%).

Per approfondire la scelta delle soglie di accettabilità e ottimalità consultare la metrica alla sezione (scrivere sezione).

#### **2.1.2.1.3 Correttezza concettuale - OPDD3**

Per determinare il grado di correttezza concettuale del documento, il gruppo ha deciso di utilizzare la seguente metrica: percentuale di errori concettuali rinvenuti e non corretti. Si desidera come soglia minima accettabile che non più del 5% degli errori concettuali rinvenuti non siano stati corretti e, come soglia ottimale, che tutti gli errori concettuali rinvenuti siano stati corretti.

Riassumendo:

**Metrica utilizzata:** percentuale di errori concettuali rinvenuti e non corretti;

**Soglia di accettabilità:** non più del 5% degli errori concettuali rinvenuti non sono stati corretti;

**Soglia di ottimalità:** tutti gli errori concettuali rinvenuti sono stati corretti (0%).

Per approfondire la scelta delle soglie di accettabilità e ottimalità consultare la metrica alla sezione (scrivere sezione).

### 2.1.2.2 Qualità del software

Gli obiettivi di qualità riguardanti il software prefissati dal gruppo Co.Code sono un sottoinsieme di quelli definiti nello standard ISO/IEC 9126:2001:

- il prodotto possiede le funzionalità descritte all'interno dei requisiti obbligatori;
- il prodotto possiede le funzionalità descritte all'interno dei requisiti desiderabili;
- il prodotto è testato in ogni sua parte e in ogni situazione nella quale si può trovare;
- il prodotto è robusto e non interrompe l'esecuzione in seguito a situazioni anomale;
- il prodotto garantisce un funzionamento senza interruzioni;
- il codice risulta manutenibile e facilmente comprensibile.

#### 2.1.2.2.1 Funzionalità obbligatorie - OPDS1

Il prodotto deve possedere tutte le funzionalità descritte nei requisiti obbligatori. Per determinare il numero di requisiti obbligatori soddisfatti viene usata come metrica la percentuale di requisiti obbligatori soddisfatti rispetto al totale.

Riassumendo:

**Metrica utilizzata:** copertura requisiti obbligatori;

**Soglia di accettabilità:** tutti i requisiti obbligatori sono soddisfatti (100%);

**Soglia di ottimalità:** tutti i requisiti obbligatori sono soddisfatti (100%).

Per approfondire la scelta delle soglie di accettabilità e ottimalità consultare la metrica alla sezione (scrivere sezione).

#### 2.1.2.2.2 Funzionalità desiderabili - OPDS2

Per determinare il numero di requisiti desiderabili soddisfatti viene usata come metrica la percentuale di requisiti desiderabili soddisfatti rispetto al totale. Si desidera come soglia minima accettabile che x% dei requisiti desiderabili sia soddisfatto mentre, come soglia ottimale, che tutti i requisiti desiderabili siano soddisfatti.

Riassumendo:

**Metrica utilizzata:** copertura requisiti desiderabili;

**Soglia di accettabilità:** almeno x% dei requisiti desiderabili soddisfatti;

**Soglia di ottimalità:** tutti i requisiti desiderabili sono soddisfatti (100%).

Per approfondire la scelta delle soglie di accettabilità e ottimalità consultare la metrica alla sezione (scrivere sezione).

#### 2.1.2.2.3 Copertura dei test richiesti - OPDS3

Il prodotto deve essere testato in ogni sua parte per garantirne il funzionamento. Vengono considerati solo i test riguardanti le funzionalità descritte nei requisiti. Si desidera come soglia minima accettabile che il numero di test passati sia almeno del 80% mentre, come soglia ottimale, almeno del 90%.

Riassumendo:

**Metrica utilizzata:** percentuale di test passati;

**Soglia di accettabilità:** almeno 80% dei test passati;

**Soglia di ottimalità:** almeno 90% dei test passati.

Per approfondire la scelta delle soglie di accettabilità e ottimalità consultare la metrica alla sezione (scrivere sezione).

#### 2.1.2.2.4 Robustezza - OPDS4

Il prodotto deve essere robusto e non deve interrompere il suo funzionamento in seguito al verificarsi di situazioni anomale. Si desidera come soglia minima accettabile che il numero di situazioni anomale gestite sia almeno del 80% mentre, come soglia ottimale, più del 90%.

Riassumendo:

**Metrica utilizzata:** Failure Avoidance;

**Soglia di accettabilità:** gestite almeno 80% delle situazioni anomale;

**Soglia di ottimalità:** gestite più del 90% delle situazioni anomale.

Per approfondire la scelta delle soglie di accettabilità e ottimalità consultare la metrica alla sezione (scrivere sezione).

#### 2.1.2.2.5 Funzionamento senza interruzioni - OPDS5

Il prodotto deve garantire un funzionamento senza interruzioni. Si desidera come soglia minima accettabile che il numero di interruzioni evitate sia almeno del 80% mentre, come soglia ottimale, più del 90%.

Riassumendo:

**Metrica utilizzata:** Breakdown Avoidance;

**Soglia di accettabilità:** evitate almeno 80% delle interruzioni;

**Soglia di ottimalità:** evitate più del 90% delle interruzioni.

Per approfondire la scelta delle soglie di accettabilità e ottimalità consultare la metrica alla sezione (scrivere sezione).

#### 2.1.2.2.6 Manutenibilità e Comprensibilità del codice - OPDS6

Il grado di manutenibilità e comprensibilità del codice deriva dalla sua complessità e lunghezza. È importante quindi che il prodotto abbia codice manutenibile e privo di incomprensioni al suo interno.

Metriche e soglie verranno definite in dettaglio nelle fasi progettuali successive.

## 2.2 Scadenze temporali

Le scadenze che il gruppo Co.Code 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

Per garantire un buon funzionamento del processo di verifica verranno impiegati 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 necessarie al processo di verifica sono i *Verificatori* e il *Responsabile*. Informazioni più dettagliate sui ruoli sono riportate nelle “*Norme di Progetto v1.0.0*”.

##### 3.1.1.2 Risorse hardware

Per eseguire la verifica, il gruppo dovrà avere a disposizione dei computer con un’adeguata potenza di calcolo in grado di sopportare il carico di lavoro.

##### 3.1.1.3 Risorse software

Le risorse software necessarie alla verifica sono gli strumenti software che eseguono controlli sui documenti e verificano che non violino le “*Norme di Progetto v1.0.0*”. Gli strumenti software devono avere le seguenti caratteristiche:

- rilevare eventuali errori ortografici;
- costruire e visualizzare in tempo reale il documento scritto in L<sup>A</sup>T<sub>E</sub>X (in modo che sia facile accorgersi di errori nell’utilizzo dei comandi).

Inoltre è necessario disporre di una piattaforma che raccolga i vari errori incontrati e li segnali ai componenti del gruppo che dovranno occuparsene.

#### 3.1.2 Risorse disponibili

##### 3.1.2.1 Risorse umane

Tutti i membri del gruppo sono a disposizione per eseguire operazioni di verifica. Ognuno dei componenti, a turno, ricoprirà il ruolo di *Responsabile* o di *Verificatore* come definito nel “*Piano di Progetto v1.0.0*”.

##### 3.1.2.2 Risorse hardware

Le risorse hardware disponibili sono i vari computer dei componenti del gruppo incaricati di svolgere il ruolo di *Responsabile* o *Verificatore*.

### 3.1.2.3 Risorse software

Le risorse software disponibili comprendono editor L<sup>A</sup>T<sub>E</sub>X con controlli integrati e script per controllare la leggibilità e la complessità dei documenti in riferimento all'indice Gulpease. Sarà disponibile anche il sistema di sollevamento delle issue offerto dalla piattaforma GitHub. Per maggiori informazioni sulla procedura di sollevamento e gestione delle issue si veda il documento “*Norme di Progetto v1.0.0*”.

## 3.2 Misure e metriche

### 3.2.1 Misure

Ogni misura effettuata sui processi e sui prodotti deve essere confrontata con una scala. I valori della scala sono:

- **Negativo:** valore non accettabile, bisogna correggere gli errori presenti ed effettuare ulteriori verifiche;
- **Accettabile:** valore accettabile, l'oggetto sottoposto a verifica ha raggiunto una soglia minima.
- **Ottimale:** valore accettabile, l'oggetto sottoposto a verifica ha raggiunto le massime aspettative del team.

### 3.2.2 Metriche per processi

Viene assegnato un codice identificativo ad ogni metrica, al fine di semplificarne il tracciamento con l'obiettivo ad essa associato.

Il metodo di denominazione delle metriche è descritto in dettaglio nel documento “*Norme di Progetto v1.0.0*”.

#### 3.2.2.1 Capability Maturity Model - MPC1

Per controllare e verificare la qualità dei processi, vengono adottate le metriche fornite dal modello CMM. Per ogni fase di lavoro, viene associato un indice che descriverà la qualità della fase. L'indice, relativo ad una scala appartenente al CMM, può assumere i valori tra 1 (il peggiore) e 5 (il migliore). Il gruppo ha stabilito i seguenti intervalli:

- il valore 1 è considerato negativo;
- i valori 2 e 3 sono considerati accettabili;
- i valori 4 e 5 sono considerati ottimali.

Per approfondimenti sul modello CMM consultare l'appendice A.

#### 3.2.2.2 Schedule Variance - MPC2

La presente metrica indica se le attività di progetto sono in anticipo o in ritardo rispetto a quelle pianificate nel “*Piano di Progetto v1.0.0*”.

Viene calcolata come la differenza fra la data pianificata di fine di un'attività e la data di fine reale dell'attività stessa.

Se la Schedule Variance ha un valore minore di 0 allora il processo è in ritardo rispetto alla pianificazione. Viceversa, se il valore è maggiore di 0, allora il processo è in anticipo. Se invece il valore è uguale a 0 allora è in linea con quanto pianificato.

Il gruppo ha stabilito i seguenti intervalli:

- un deficit maggiore del 5% rispetto alla pianificazione è considerato negativo;
- un deficit minore o uguale al 5% rispetto alla pianificazione è considerato accettabile;
- valori maggiori o uguali a 0 sono considerati ottimali.

### 3.2.2.3 Cost Variance - MPC3

La presente metrica indica se i costi alla data corrente sono maggiori o minori rispetto a quanto dichiarato nel “*Piano di Progetto v1.0.0*”.

Viene calcolata come la differenza fra preventivo e consuntivo.

Il gruppo ha stabilito i seguenti intervalli:

- un deficit maggiore del 10% delle risorse preventivate per il processo è considerato negativo;
- un deficit minore o uguale al 10% delle risorse preventivate per il processo è accettabile;
- un valore maggiore o uguale a 0 è considerato ottimale.

### 3.2.3 Metriche per i prodotti

Viene assegnato un codice identificativo ad ogni metrica, al fine di semplificarne il tracciamento con l'obiettivo ad essa associato.

Il metodo di denominazione delle metriche è descritto in dettaglio nel documento “*Norme di Progetto v1.0.0*”.

#### 3.2.3.1 Metriche per i documenti

La qualità di un documento dipende soprattutto dai suoi contenuti. La loro qualità, tuttavia, è difficilmente quantificabile allo stato attuale del progetto a causa dell'esperienza pressoché nulla del gruppo in quest'ambito. Si è deciso dunque di limitarsi a valutare parametri maggiormente oggettivi e soprattutto misurabili automaticamente attraverso strumenti software.

##### 3.2.3.1.1 Indice di leggibilità - MPDD1

Per valutare la qualità del documento si è deciso di utilizzare l'indice di leggibilità. In particolare, è stato considerato l'indice Gulepease, studiato appositamente per la lingua italiana.

Questo particolare indice si basa sulla lunghezza della parola e sulla lunghezza della frase rispetto al numero di lettere. La formula per il suo calcolo è la seguente:

$$\text{Indice Gulepease} = 89 + \frac{300 * \text{numeroFrase} + 10 * \text{numeroLettere}}{\text{numeroParole}} \quad (1)$$

Il risultato è compreso tra 0 e 100, dove valori alti indicano leggibilità elevata e viceversa. In generale, risulta che testi con un indice:

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

Poiché la documentazione è rivolta a persone istruite, il gruppo ha stabilito i seguenti intervalli per l'indice:

- valori minori di 40 sono considerati negativi;

- valori compresi tra 40 e 60 sono considerati accettabili.
- valori maggiori di 60 sono considerati ottimali.

### 3.2.3.1.2 Errori ortografici rinvenuti e non corretti - MPDD2

Tale metrica è necessaria per capire quanto un documento sia corretto dal punto di vista ortografico. Supponendo che gli strumenti automatici siano in grado di trovare tutti gli errori ortografici all'interno di un testo, allora la correttezza ortografica non può che basarsi sul numero di errori rinvenuti ma non successivamente corretti. Notare che per errori corretti si intende un errore revisionato manualmente da parte di un *Verificatore*. Le correzioni automatiche, infatti, non sono molto attendibili. Il gruppo ha stabilito i seguenti intervalli:

- una percentuale di errori non corretti maggiore allo 0% è ritenuta negativa;
- una percentuale di errori non corretti pari allo 0% è ritenuta accettabile;
- una percentuale di errori non corretti pari allo 0% è ritenuta ottimale.

### 3.2.3.1.3 Errori concettuali rinvenuti e non corretti - MPDD3

Tale metrica è necessaria per capire quanto un documento sia corretto dal punto di vista concettuale. Supponendo che, in seguito alle revisioni, siano stati individuati tutti gli errori concettuali all'interno di un testo, allora la correttezza concettuale non può che basarsi sul numero di errori rinvenuti ma non successivamente corretti. Notare che per errori corretti si intende un errore revisionato manualmente da parte di un *Verificatore*. Il gruppo ha stabilito i seguenti intervalli:

- una percentuale di errori non corretti maggiore al 5% è ritenuta negativa;
- una percentuale di errori non corretti minore o uguale al 5% è ritenuta accettabile;
- una percentuale di errori non corretti pari allo 0% è ritenuta ottimale;

### 3.2.3.2 Metriche per il software

Il gruppo Co.Code ha deciso di adottare alcune metriche che hanno il compito di monitorare la qualità interna, la qualità esterna e la qualità in uso del software. Tali metriche sono un sottoinsieme di quelle definite nello standard ISO/IEC 9126:2001.

Ogni metrica scelta viene associata ad una caratteristica di qualità presente all'interno dello standard:

Metriche scelte	Caratteristiche di qualità
MPDS1 - Copertura requisiti obbligatori	Funzionalità
MPDS2 - Copertura requisiti desiderabili	Funzionalità
MPDS3 - Test passati richiesti	Affidabilità
MPDS4 - Failure Avoidance	Affidabilità
MPDS5 - Breakdown Avoidance	Affidabilità

**Tabella 1:** Mappa metriche-caratteristiche

**3.2.3.2.1 Copertura requisiti obbligatori - MPDS1**

La seguente metrica controlla quanti requisiti obbligatori sono stati soddisfatti. Viene calcolata come rapporto, espresso in percentuale, tra i requisiti obbligatori soddisfatti e il numero totale dei requisiti obbligatori.

$$\text{Copertura requisiti obbligatori} = 100 * \frac{\# \text{requisiti obbligatori soddisfatti}}{\# \text{requisiti obbligatori totali}} \quad (2)$$

Il gruppo ha stabilito i seguenti intervalli:

- una percentuale minore del 100% è ritenuta negativa;
- una percentuale uguale al 100% è ritenuta accettabile;
- una percentuale uguale al 100% ottimale.

**3.2.3.2.2 Copertura requisiti desiderabili - MPDS2**

La seguente metrica controlla quanti requisiti desiderabili sono stati soddisfatti. Viene calcolata come rapporto, espresso in percentuale, tra i requisiti desiderabili soddisfatti e il numero totale dei requisiti desiderabili.

$$\text{Copertura requisiti desiderabili} = 100 * \frac{\# \text{requisiti desiderabili soddisfatti}}{\# \text{requisiti desiderabili totali}} \quad (3)$$

Il gruppo ha stabilito i seguenti intervalli:

- una percentuale minore del x% è ritenuta negativa;
- una percentuale maggiore o uguale a x% è ritenuta accettabile;
- una percentuale uguale al 100% ottimale.

**3.2.3.2.3 Test passati richiesti - MPDS3**

La metrica controlla che il rapporto, espresso in percentuale, tra il numero di test passati e il numero di test totali rientri tra i valori definiti. Questo ci permette di valutare se il prodotto supera la maggior parte dei test.

$$\text{Test passati richiesti} = 100 * \frac{\# \text{di test passati}}{\# \text{di test totali}} \quad (4)$$

Il gruppo ha stabilito i seguenti intervalli:

- una percentuale minore del 80% è ritenuta negativa;
- una percentuale maggiore o uguale a 80% è ritenuta accettabile;
- una percentuale maggiore o uguale a 90% è ritenuta ottimale;

**3.2.3.2.4 Failure Avoidance - MPDS4**

La metrica controlla che il rapporto, espresso in percentuale, tra il numero di situazioni anomale gestite e il numero di situazioni anomale presentate rientri tra i valori definiti.

$$\text{Failure Avoidance} = 100 * \frac{\# \text{situazioni anomale gestite}}{\# \text{situazioni anomale presentate}} \quad (5)$$

Il gruppo ha stabilito i seguenti intervalli:

- una percentuale minore del 80% è ritenuta negativa;



- una percentuale compresa tra 80% e 90% è ritenuta accettabile;
- una percentuale maggiore di 90% è ritenuta ottimale.

#### 3.2.3.2.5 Breakdown Avoidance - MPDS5

La metrica controlla che la percentuale di interruzioni evitate dal prodotto rientri tra i valori definiti. Il valore su cui si applicherà la metrica verrà calcolato come il complemento delle interruzioni verificate. Questa metrica ci permette di controllare che il prodotto lavori senza interruzioni.

$$Breakdown\ Avoidance = 100 * \left( 1 - \frac{\# \text{ di interruzioni}}{\# \text{ situazioni anomale presentate}} \right) \quad (6)$$

Il gruppo ha stabilito i seguenti intervalli:

- una percentuale minore del 80% è ritenuta negativa;
- una percentuale compresa tra 80% e 90% è ritenuta accettabile;
- una percentuale maggiore di 90% è ritenuta ottimale.

## A Capability Maturity Model

## B Standard ISO/IEC 9126

Lo Standard ISO/IEC 9126 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).

Lo standard tratta la qualità del software da tre punti di vista:

- **qualità interna:** è la qualità del software non eseguibile e fa quindi riferimento alle caratteristiche implementative del software quali l'architettura e il codice sorgente;
- **qualità esterna:** è la qualità del software nel momento in cui esso viene eseguito e testato in un ambiente di prova;
- **qualità in uso:** è la qualità del software dal punto di vista dell'utente che ne fa uso all'interno di uno specifico sistema e contesto.

### B.1 Modello della qualità del software

Nella prima parte (9126-1) vengono descritti i modelli per la qualità esterna, interna ed in uso.

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

Il modello di qualità esterna ed interna stabilito nella prima parte dello standard è classificato in sei caratteristiche generali:

- **funzionalità:** è la capacità di un software di fornire funzioni in grado di soddisfare specifici bisogni in un determinato contesto;
- **affidabilità:** è la capacità di un software di mantenere un certo livello di prestazioni in condizioni specifiche di utilizzo in un intervallo di tempo fissato;
- **usabilità:** è la capacità di un software di essere capito e utilizzato da utenti di un determinato contesto;
- **efficienza:** è la capacità di un software di fornire un certo livello di prestazioni con il minor numero di risorse possibile;
- **manutenibilità:** è la capacità del software di essere modificato, includendo correzioni, miglioramenti o adattamenti;
- **portabilità:** la capacità del software di essere trasportato da un ambiente di lavoro ad un altro.

Tali caratteristiche sono misurabili attraverso delle metriche.

#### B.1.2 Modello della qualità in uso

Gli attributi presenti nel modello relativo alla qualità del software in uso sono classificati in quattro categorie:

- **efficacia:** è la capacità del software di permettere agli utenti di raggiungere gli obiettivi specificati con accuratezza e completezza;

- **produttività:** la capacità di permettere all'utente di utilizzare un numero di risorse in relazione all'efficienza raggiunta in uno specifico contesto di utilizzo;
- **sicurezza:** è la capacità del software di raggiungere un livello accettabile di rischi per i dati, le persone, il business, la proprietà o gli ambienti in uno specifico contesto di utilizzo;
- **soddisfazione:** è la capacità del software di soddisfare gli utenti in uno specifico contesto di utilizzo.

## B.2 Metriche per la qualità del software

Nelle restanti tre parti (9126-2,9126-3,9126-4) vengono trattate le metriche per la qualità esterna, interna e in uso.

### B.2.1 Metriche per la qualità esterna

Le metriche esterne misurano i comportamenti del software rilevabili dai test, dall'operatività e dall'osservazione durante la sua esecuzione. L'esecuzione del software è fatta in un contesto tecnico rilevante. Le metriche esterne sono scelte in base alle caratteristiche che il prodotto finale dovrà dimostrare durante la sua esecuzione in esercizio.

### B.2.2 Metriche per la qualità interna

Le metriche interne si applicano al software non eseguibile (come 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.2.3 Metriche per la qualità in uso

Le metriche della qualità in uso rappresentano il punto di vista dell'utente e misurano il grado con cui il software permette agli utenti di svolgere le proprie attività con efficacia, produttività, sicurezza e soddisfazione nel contesto operativo previsto.

## C PDCA

Il PDCA, acronimo di Plan-Do-Check-Act, conosciuto anche come "Ciclo di Deming" o "Ciclo di miglioramento continuo", è un modello studiato per il miglioramento continuo della qualità in un'ottica a lungo raggio.

Questo strumento permette di fissare degli obiettivi di miglioramento a partire dagli esiti delle misurazioni effettuate durante le varie attività di verifica. Una volta fissati gli obiettivi che si desiderano raggiungere, si iterano le attività previste dal Ciclo di Deming fino al raggiungimento di tali obiettivi. 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.

Le quattro attività previste dal Ciclo di Deming sono:

- **Plan - Pianificare:** consiste nel definire gli obiettivi di miglioramento e le strategie da utilizzare per raggiungere tali obiettivi. Durante questa attività viene inoltre pianificato il modo in cui attuare queste strategie per raggiungere gli obiettivi di miglioramento fissati;
- **Do - Eseguire:** consiste nell'esecuzione di ciò che è stato pianificato nel punto precedente e nella raccolta dati necessaria all'analisi effettuata nei punti successivi;
- **Check - Verificare:** consiste nel verificare l'esito del processo (per efficienza ed efficacia) in seguito all'attuazione delle strategie di miglioramento. I risultati possono essere di tre tipi:
  - un miglioramento secondo le aspettative;
  - un miglioramento superiore alle aspettative;
  - un miglioramento inferiore alle aspettative.
- **Act - Agire:** consiste nell'attuazione di soluzioni correttive, ovvero di strategie che hanno portato miglioramenti, anche al di fuori dei singoli processi per i quali si erano stati fissati gli obiettivi di miglioramento.

Bisogna tener presente che se l'obiettivo è il miglioramento continuo, le attività devono essere analizzabili, ripetibili e tracciabili. Unendo queste tre caratteristiche è possibile individuare eventuali errori e correggerli.

## **D    Test**

L'analisi dinamica è un argomento non ancora trattato a lezione, pertanto la stesura di questa sezione avverrà in successive versioni del documento.

## E Resoconto delle attività di verifica - fase AR

All'interno di questa sezione sono riportati gli esiti di tutte le attività di verifica effettuate nell'arco della fase AR, come previsto dal documento "*Piano di Progetto v1.0.0*". Ove necessario sono state tratte conclusioni sui risultati e su come essi possano essere migliorati.

### E.1 Verifica sui processi

#### E.1.1 Processo di documentazione

##### E.1.1.1 Miglioramento costante

da fare

##### E.1.1.2 Rispetto della pianificazione

Per capire se l'attività di un processo rispetta i tempi stabiliti dalla pianificazione all'interno del "*Piano di Progetto v1.0.0*" viene utilizzata la metrica Schedule Variance. Si desidera, come soglia minima accettabile, che un processo sia in ritardo non più del 5% rispetto alla pianificazione. Sarebbe ottimale, invece, non avere ritardi rispetto alla pianificazione o, ancora meglio, essere in anticipo.

Di seguito sono riportati i valori ottenuti calcolando la Schedule Variance sui tempi di stesura di ogni documento nella fase AR:

Documento	Schedule Variance	Esito
" <i>Piano di Progetto v1.0.0</i> "	x	x
" <i>Norme di Progetto v1.0.0</i> "	x	x
" <i>Analisi dei Requisiti v1.0.0</i> "	x	x
" <i>Piano di Qualifica v1.0.0</i> "	x	x
" <i>Glossario v1.0.0</i> "	x	x
analisiSDK(macro)	x	x

**Tabella 2:** Esiti del calcolo della Schedule Variance sul processo di documentazione durante la fase AR

##### E.1.1.3 Rispetto del budget

Per capire se i costi di un processo rientrano nel budget stabilito dalla pianificazione all'interno del "*Piano di Progetto v1.0.0*" viene utilizzata la metrica Cost Variance. Si desidera, come soglia minima accettabile, che un processo non superi il 10% del budget pianificato. Sarebbe ottimale, invece, non superare i costi pianificati o, ancora meglio, spendere meno.

Di seguito sono riportati i valori ottenuti calcolando la Cost Variance sui tempi di stesura di ogni documento nella fase AR:

Documento	Cost Variance	Esito
"Piano di Progetto v1.0.0"	x	x
"Norme di Progetto v1.0.0"	x	x
"Analisi dei Requisiti v1.0.0"	x	x
"Piano di Qualifica v1.0.0"	x	x
"Glossario v1.0.0"	x	x
analisiSDK(macro)	x	x

**Tabella 3:** Esiti del calcolo della Cost Variance sul processo di documentazione durante la fase AR

### E.1.2 Processo di verifica

#### E.1.2.1 Miglioramento costante

da fare

#### E.1.2.2 Rispetto della pianificazione

Per capire se l'attività di un processo rispetta i tempi stabiliti dalla pianificazione all'interno del "*Piano di Progetto v1.0.0*" viene utilizzata la metrica Schedule Variance. Si desidera, come soglia minima accettabile, che un processo sia in ritardo non più del 5% rispetto alla pianificazione. Sarebbe ottimale, invece, non avere ritardi rispetto alla pianificazione o, ancora meglio, essere in anticipo.

Di seguito è riportato il valore ottenuto calcolando la Schedule Variance sul processo di verifica nella fase AR:

Processo	Schedule Variance	Esito
Processo di verifica	x	x

**Tabella 4:** Esiti del calcolo della Schedule Variance sul processo di verifica durante la fase AR

#### E.1.2.3 Rispetto del budget

Per capire se i costi di un processo rientrano nel budget stabilito dalla pianificazione all'interno del "*Piano di Progetto v1.0.0*" viene utilizzata la metrica Cost Variance. Si desidera, come soglia minima accettabile, che un processo non superi il 10% del budget pianificato. Sarebbe ottimale, invece, non superare i costi pianificati o, ancora meglio, spendere meno.

Di seguito è riportato il valore ottenuto calcolando la Cost Variance sul processo di verifica nella fase AR:

Processo	Cost Variance	Esito
Processo di verifica	x	x

**Tabella 5:** Esiti del calcolo della Cost Variance sul processo di verifica durante la fase AR



## E.2 Verifica sui prodotti

### E.2.1 Documenti

#### E.2.1.1 Leggibilità e comprensibilità

Per determinare il grado di leggibilità e comprensibilità del documento, il gruppo ha deciso di utilizzare l'indice Gulpease. Si desidera come soglia minima accettabile un indice maggiore o uguale a 40 e, come soglia ottimale, un indice maggiore di 60.

Di seguito sono riportati i valori ottenuti calcolando l'indice Gulpease sui documenti della fase AR:

Documento	Gulpease	Esito
<i>"Piano di Progetto v1.0.0"</i>	x	x
<i>"Norme di Progetto v1.0.0"</i>	x	x
<i>"Analisi dei Requisiti v1.0.0"</i>	x	x
<i>"Piano di Qualifica v1.0.0"</i>	x	x
<i>"Glossario v1.0.0"</i>	x	x
analisiSDK(macro)	x	x

**Tabella 6:** Esiti del calcolo dell'indice Gulpease sui documenti della fase AR

#### E.2.1.2 Correttezza ortografica

Per determinare il grado di correttezza ortografica del documento, il gruppo ha deciso di utilizzare la seguente metrica: percentuale di errori ortografici rinvenuti e non corretti. Pertanto, la soglia minima accettabile e la soglia ottimale coincidono e corrispondono a una correzione totale degli errori rinvenuti.

Di seguito è riportato il numero di errori ortografici trovati:

Errori ortografici	x
--------------------	---

**Tabella 7:** Errori ortografici rinvenuti durante la fase AR

Tutti gli errori ortografici rinvenuti sono stati corretti, quindi è stato raggiunto l'obiettivo **ottimale**.

#### E.2.1.3 Correttezza concettuale

Per determinare il grado di correttezza concettuale del documento, il gruppo ha deciso di utilizzare la seguente metrica: percentuale di errori concettuali rinvenuti e non corretti. Si desidera come soglia minima accettabile che non più del 5% degli errori concettuali rinvenuti non siano stati corretti e, come soglia ottimale, che tutti gli errori concettuali rinvenuti siano stati corretti.

Di seguito è riportato il numero di errori concettuali trovati:

Errori concettuali	x
--------------------	---

**Tabella 8:** Errori concettuali rinvenuti durante la fase AR

Tutti gli errori concettuali rinvenuti sono stati corretti, quindi è stato raggiunto l'obiettivo **ottimale**.