



Carbon12 – Predire in Grafana

# Piano di Qualifica

## Informazioni sul documento

<b>Versione</b>	1.0.0
<b>Stato</b>	Approvato
<b>Data di creazione</b>	2019/11/29
<b>Data di approvazione</b>	2020/01/13
<b>Redazione</b>	Nicolò Fassina Francesco Gobbo Alessandro Lovo
<b>Verifica</b>	Manuel De Franceschi Andrea Longo
<b>Approvazione</b>	Giacomo Callegari
<b>Uso</b>	Esterno
<b>Destinatari</b>	Carbon12 Zucchetti SPA Prof. Tullio Vardanega Prof. Riccardo Cardin
<b>E-mail di riferimento</b>	carbon.dodici@gmail.com

## Scopo del documento

Definizione delle operazioni di test, verifica e validazione eseguite dal gruppo Carbon12 durante la realizzazione del progetto Predire in Grafana.

# Indice

<b>1 INTRODUZIONE.....</b>	<b>1</b>
<b>1.1 SCOPO DEL DOCUMENTO .....</b>	<b>1</b>
<b>1.2 SCOPO DEL PRODOTTO .....</b>	<b>1</b>
<b>1.3 DOCUMENTI COMPLEMENTARI .....</b>	<b>1</b>
<b>1.4 RIFERIMENTI .....</b>	<b>1</b>
1.4.1 NORMATIVI .....	1
1.4.2 INFORMATIVI.....	1
<b>2 QUALITÀ DEL SOFTWARE.....</b>	<b>3</b>
<b>2.1 FUNZIONALITÀ.....</b>	<b>3</b>
2.1.1 CARATTERISTICHE .....	3
2.1.2 MISURAZIONE E OBIETTIVI DELLE METRICHE .....	3
<b>2.2 AFFIDABILITÀ .....</b>	<b>4</b>
2.2.1 CARATTERISTICHE .....	4
2.2.2 METRICHE DI VALUTAZIONE .....	4
<b>2.3 EFFICIENZA.....</b>	<b>4</b>
2.3.1 CARATTERISTICHE .....	4
2.3.2 METRICHE DI VALUTAZIONE .....	5
<b>2.4 USABILITÀ.....</b>	<b>5</b>
2.4.1 CARATTERISTICHE .....	5
2.4.2 METRICHE DI VALUTAZIONE .....	6
<b>2.5 MANUTENIBILITÀ .....</b>	<b>7</b>
2.5.1 CARATTERISTICHE .....	7
2.5.2 METRICHE DI VALUTAZIONE .....	7
<b>2.6 PORTABILITÀ .....</b>	<b>8</b>
2.6.1 CARATTERISTICHE .....	8
2.6.2 METRICHE DI VALUTAZIONE .....	8
<b>3 QUALITÀ DI PROCESSO.....</b>	<b>9</b>
<b>3.1 PROCESSI PRIMARI .....</b>	<b>9</b>
3.1.1 ANALISI DEI REQUISITI .....	9
3.1.1.1 Finalità .....	9
3.1.1.2 Strategie .....	9
3.1.1.3 Obiettivi delle metriche.....	9
3.1.2 PROGETTAZIONE ARCHITETTURALE .....	10
3.1.2.1 Finalità .....	10
3.1.2.2 Strategie .....	10
3.1.2.3 Obiettivi delle metriche.....	10
3.1.3 PROGETTAZIONE DI DETTAGLIO .....	10
3.1.3.1 Finalità .....	10
3.1.3.2 Strategie .....	10
3.1.3.3 Obiettivi delle metriche.....	11
<b>3.2 PROCESSI DI SUPPORTO .....</b>	<b>11</b>

3.2.1 PIANIFICAZIONE .....	11
3.2.1.1 Finalità .....	11
3.2.1.2 Strategie .....	11
3.2.1.3 Obiettivi delle metriche.....	11
3.2.2 DOCUMENTAZIONE.....	12
3.2.2.1 Finalità .....	12
3.2.2.2 Strategie .....	12
3.2.2.3 Obiettivi delle metriche.....	12
3.2.3 VERIFICA.....	12
3.2.3.1 Finalità .....	13
3.2.3.2 Strategie .....	13
3.2.3.3 Obiettivi delle metriche.....	13
<b>3.3 PROCESSI ORGANIZZATIVI .....</b>	<b>13</b>
3.3.1 GESTIONE DELLA QUALITÀ.....	13
3.3.1.1 Finalità .....	13
3.3.1.2 Strategie .....	13
3.3.1.3 Obiettivi delle metriche.....	14
<b>A STANDARD DI QUALITÀ .....</b>	<b>15</b>
<b>A.1 ISO/IEC 9126.....</b>	<b>15</b>
A.1.1 MODELLO DELLA QUALITÀ.....	15
A.1.1.1 Funzionalità.....	15
A.1.1.2 Affidabilità .....	15
A.1.1.3 Efficienza .....	15
A.1.1.4 Usabilità .....	16
A.1.1.5 Manutenibilità .....	16
A.1.1.6 Portabilità .....	16
A.1.2 METRICHE DI QUALITÀ ESTERNE .....	16
A.1.3 METRICHE DI QUALITÀ INTERNE.....	17
A.1.4 METRICHE DI QUALITÀ IN USO.....	17
<b>A.2 ISO/IEC 15504.....</b>	<b>17</b>
<b>A.3 CICLO DI DEMING.....</b>	<b>19</b>
<b>B RESOCONTO ATTIVITÀ DI VERIFICA .....</b>	<b>20</b>
<b>B.1 REVISIONE DEI REQUISITI .....</b>	<b>20</b>
B.1.1 ANALISI STATICA DEI DOCUMENTI.....	20
B.1.2 ESITI VERIFICHE AUTOMATIZZATE .....	20
<b>C VALUTAZIONI PER IL MIGLIORAMENTO .....</b>	<b>21</b>
<b>C.1 VALUTAZIONE SULL'ORGANIZZAZIONE.....</b>	<b>21</b>
<b>C.2 VALUTAZIONE SUI RUOLI.....</b>	<b>22</b>
C.2.1 AMMINISTRATORE.....	22
C.2.2 RESPONSABILE .....	22
C.2.3 ANALISTA.....	22
C.2.4 VERIFICATORE.....	22
<b>C.3 VALUTAZIONE SULL'USO DEGLI STRUMENTI.....</b>	<b>23</b>

**REGISTRO DELLE MODIFICHE.....24**

# 1 Introduzione

Il *Piano di Qualifica* è un documento che prevede una lavorazione continua per l'intero sviluppo del progetto, poiché molti dei contenuti trattati sono considerati di natura instabile.

Viene prodotto per costruzione e non per correzione, limitando così lo spreco di risorse. La costruzione applicata è di tipo incrementale e i contenuti iniziali sono da considerarsi incompleti, in quanto le parti non presenti verranno aggiornate successivamente.

## 1.1 Scopo del documento

Il *Piano di Qualifica* ha l'obiettivo di definire le strategie adottate per la verifica e la validazione, preservando la qualità di prodotto e di processo. Durante le attività viene applicato un sistema di verifica continua al fine di conseguire lo scopo prefissato. Operando in questo modo è possibile rilevare e attivarsi per la correzione delle irregolarità riscontrate, mantenendo l'EFFICIENZA e l'EFFICACIA.

## 1.2 Scopo del prodotto

Lo scopo del prodotto è l'ottenimento di un PLUG-IN applicabile in GRAFANA, che monitori il flusso di dati fornendo un'analisi predittiva su di essi basata sull'ADDESTRAMENTO di modelli di SUPPORT-VECTOR MACHINE e REGRESSIONE LINEARE. I dati monitorati e le predizioni misurate saranno visualizzati su una Dashboard contenente i grafici prodotti dal sistema di creazione di grafici di Grafana.

## 1.3 Documenti complementari

Onde evitare possibili situazioni di ambiguità nella lettura del documento, viene fornito il *Glossario Esterno v.1.0.0*. I termini o concetti che si sono ritenuti rilevanti e che necessitano di una ulteriore specifica sono contrassegnati con la dicitura GE a pedice e sono evidenziati in maiuscolo.

## 1.4 Riferimenti

### 1.4.1 Normativi

- Capitolato d'Appalto C4 – *Predire in Grafana*  
<https://www.math.unipd.it/~tullio/IS-1/2019/Progetto/C4.pdf>
- Norme di Progetto: *Norme di progetto v.1.0.0*

### 1.4.2 Informativi

- Piano di Progetto: *Piano di Progetto v.1.0.0*
- ISO/IEC 9126  
[https://en.wikipedia.org/wiki/ISO/IEC\\_9126](https://en.wikipedia.org/wiki/ISO/IEC_9126)
- ISO/IEC 12207  
[https://www.math.unipd.it/~tullio/IS-1/2009/Approfondimenti/ISO\\_12207-1995.pdf](https://www.math.unipd.it/~tullio/IS-1/2009/Approfondimenti/ISO_12207-1995.pdf)
- ISO/IEC 15504  
[https://en.wikipedia.org/wiki/ISO/IEC\\_15504](https://en.wikipedia.org/wiki/ISO/IEC_15504)
- Ciclo di Deming (PDCA)  
[https://it.wikipedia.org/wiki/Ciclo\\_di\\_Deming](https://it.wikipedia.org/wiki/Ciclo_di_Deming)
- HM&S - SPICE Process Assessment Model

<http://www.spice121.com/cms/en/about-spice-1-2-1.html>

- Indice di Gulpease  
[https://it.wikipedia.org/wiki/Indice\\_Gulpease](https://it.wikipedia.org/wiki/Indice_Gulpease)
- Schedule Variance e metriche correlate:  
<https://www.smartsheet.com/hacking-pmp-how-calculate-schedule-variance>
- Slide corso “Ingegneria del Software” - UNIPD: Qualità di Processo e di Prodotto  
<https://www.math.unipd.it/~tullio/IS-1/2019/Dispense/L12.pdf>  
<https://www.math.unipd.it/~tullio/IS-1/2019/Dispense/L13.pdf>

## 2 Qualità del software

Per garantire la qualità del prodotto software, il gruppo Carbon12 ha deciso di far riferimento al modello di qualità descritto nello standard *ISO/IEC 9126*. In questa sezione verranno trattate le caratteristiche che incidono sulla capacità del prodotto software di soddisfare i requisiti impliciti ed espliciti. Sono inoltre riportate le metriche utilizzate per valutare la qualità ricercata.

### 2.1 Funzionalità

È la capacità del prodotto software di fornire funzioni che soddisfino le esigenze stabilite dagli Analisti nell'*Analisi dei Requisiti*.

#### 2.1.1 Caratteristiche

- **Adeguatezza:** capacità del prodotto software di fornire funzioni appropriate all'utente per svolgere determinati compiti;
- **Accuratezza:** capacità del prodotto software di fornire risultati o effetti attesi con il livello di precisione richiesto dal proponente.

#### 2.1.2 Misurazione e obiettivi delle metriche

- **Funzionalità sviluppate (FS):**
  - **Scopo:** misurare quanti requisiti sono stati implementati;
  - **Risultato:** percentuale di requisiti implementati sul totale di requisiti;
  - **Calcolo:**  $FS = \frac{N_{RS}}{N_{RO} + N_{RF}} * 100$ , dove:
    - $N_{RS}$  = numero di requisiti soddisfatti;
    - $N_{RO}$  = numero di requisiti obbligatori;
    - $N_{RF}$  = numero di requisiti facoltativi;
  - **Intervallo accettabile:** [60%, 100%]
  - **Intervallo desiderabile:** [70%, 100%]
- **Correttezza funzionale (CF):**
  - **Scopo:** misurare la correttezza dei risultati, o effetti, prodotti dalle funzionalità implementate;
  - **Risultato:** percentuale dei risultati, o effetti, corretti ottenuti utilizzando le funzionalità del prodotto software;
  - **Calcolo:**  $CF = \frac{N_{RC}}{N_{PE}} * 100$ , dove:
    - $N_{RC}$  = numero di risultati o effetti corretti;
    - $N_{PE}$  = numero di prove effettuate;
  - **Intervallo accettabile:** [80%, 100%]
  - **Intervallo desiderabile:** [90%, 100%]

## 2.2 Affidabilità

È la capacità del prodotto software di operare correttamente anche in caso di situazioni anomale.

### 2.2.1 Caratteristiche

- **Maturità:** capacità del prodotto software di evitare errori o risultati anomali durante l'esecuzione;
- **Tolleranza all'errore:** capacità di mantenere determinati livelli di prestazione in caso di anomalie software o utilizzo scorretto dell'applicativo;
- **Recuperabilità:** capacità di ripristinare determinati livelli di prestazioni e di recuperare i dati a seguito di errori o malfunzionamenti.

### 2.2.2 Metriche di valutazione

- **Densità degli errori (DE):**
  - **Scopo:** misurare l'attitudine del prodotto software a evitare errori o risultati anomali;
  - **Risultato:** percentuale dei test falliti sul totale di test eseguiti;
  - **Calcolo:**  $DE = \frac{N_{TF}}{N_{TE}} * 100$ , dove:
    - $N_{TF}$  = numero di test falliti;
    - $N_{TE}$  = numero dei test effettuati;
  - **Intervallo accettabile:** [0%, 15%]
  - **Intervallo desiderabile:** [0%, 5%]
- **Capacità di recupero (CR):**
  - **Scopo:** misurare l'attitudine del prodotto software a mantenere i dati dopo errori o malfunzionamenti;
  - **Risultato:** percentuale di quante volte ha avuto successo il recupero totale delle informazioni dopo errori o malfunzionamenti;
  - **Calcolo:**  $CR = \frac{N_{RS}}{N_{RT}} * 100$ , dove:
    - $N_{RS}$  = numero di recuperi che hanno avuto successo;
    - $N_{RT}$  = numero di recuperi tentati;
  - **Intervallo accettabile:** [70%, 100%]
  - **Intervallo desiderabile:** [90%, 100%]

## 2.3 Efficienza

È la capacità del prodotto software di fornire le funzioni richieste nel minor tempo possibile, sfruttando al meglio le risorse a disposizione.

### 2.3.1 Caratteristiche

- **Comportamento rispetto al tempo:** capacità del prodotto software di fornire appropriati tempi di risposta e di elaborazione nell'eseguire le sue funzioni;
- **Uso delle risorse:** capacità del prodotto software di usare un appropriato numero e tipo di risorse quando esegue le sue funzioni.



### 2.3.2 Metriche di valutazione

- **Tempo medio di risposta (TMR):**
  - **Scopo:** misurare i tempi delle elaborazioni;
  - **Risultato:** tempo medio impiegato dal prodotto software per rispondere a una richiesta dell'utente o per svolgere un'attività di sistema;
  - **Calcolo:**  $TMR = \frac{\sum_{i=1}^n T_{Ri}}{n}$ , dove:
    - $T_{Ri}$  = tempo di risposta della funzionalità i;
    - $n$  = numero di funzionalità disponibili;
  - **Intervallo accettabile:** [0s, 5s]
  - **Intervallo desiderabile:** [0s, 3s]
- **Consumo medio di potenza elaborativa (CMPE):**
  - **Scopo:** misurare l'utilizzo di CPU per compiere una elaborazione;
  - **Risultato:** tempo medio di CPU utilizzato per compiere una elaborazione;
  - **Calcolo:**  $CMPE = \frac{\sum_{i=1}^n T_{CPUi}}{n}$ , dove:
    - $T_{CPUi}$  = tempo di utilizzo della CPU per la funzionalità i;
    - $n$  = numero di funzionalità disponibili;
  - **Intervallo accettabile:** non ancora definito;
  - **Intervallo desiderabile:** non ancora definito.
- **Consumo medio di memoria (CMM):**
  - **Scopo:** misurare l'utilizzo di memoria per compiere una elaborazione;
  - **Risultato:** quantità media di memoria utilizzata per compiere una elaborazione;
  - **Calcolo:**  $CMM = \frac{\sum_{i=1}^n Q_{CMI}}{n}$ , dove:
    - $Q_{CMI}$  = quantità di memoria [MB] usata dalla funzionalità i;
    - $n$  = numero di funzionalità disponibili;
  - **Intervallo accettabile:** non ancora definito;
  - **Intervallo desiderabile:** non ancora definito.

## 2.4 Usabilità

È la capacità del prodotto software di essere comprensibile, facile da apprendere e da usare e di risultare attraente all'utente.

### 2.4.1 Caratteristiche

- **Comprensibilità:** capacità del prodotto software di permettere all'utente di capire le sue funzionalità e come utilizzarle per svolgere determinati compiti;
- **Apprendibilità:** capacità del prodotto software di permettere all'utente di imparare ad utilizzare l'applicativo;
- **Operabilità:** capacità del prodotto software di permettere all'utente di interagire con l'applicativo;
- **Attrattività:** capacità del prodotto software di risultare piacevole all'utente.

### 2.4.2 Metriche di valutazione

- **Ambiguità delle funzioni (AF):**
  - **Scopo:** misurare la qualità espositiva delle descrizioni delle funzioni presenti nella documentazione del prodotto software;
  - **Risultato:** numero di funzioni non comprese dall'utente dopo aver letto la documentazione;
  - **Calcolo:**  $AF = \frac{N_{FA}}{N_F} * 100$ , dove:
    - $N_{FA}$  = numero di funzioni ambigue;
    - $N_F$  = numero di funzioni presenti;
  - **Intervallo accettabile:** [0%, 5%]
  - **Intervallo desiderabile:** [0%, 2%]
- **Ambiguità degli errori (AE):**
  - **Scopo:** misurare la qualità espositiva delle descrizioni degli errori stampati a video;
  - **Risultato:** numero di errori stampati a video non compresi dall'utente;
  - **Calcolo:**  $AE = \frac{N_{MEA}}{N_{ME}} * 100$ , dove:
    - $N_{MEA}$  = numero di messaggi di errore ambigui;
    - $N_{ME}$  = numero di messaggi d'errore stampati a video;
  - **Intervallo accettabile:** [0%, 10%]
  - **Intervallo desiderabile:** [0%, 5%]
- **Descrizione delle funzioni (DF):**
  - **Scopo:** misurare il livello di comprensibilità delle funzionalità offerte dal prodotto software;
  - **Misurazione:** percentuale di funzioni descritte nella documentazione del prodotto software sul totale delle funzioni presenti;
  - **Calcolo:**  $DF = \frac{N_{FD}}{N_F} * 100$ , dove:
    - $N_{FD}$  = numero di funzioni descritte nella documentazione del prodotto software;
    - $N_F$  = numero di funzioni presenti;
  - **Intervallo accettabile:** [80%, 100%]
  - **Intervallo desiderabile:** [90%, 100%]
- **Tempo medio di apprendimento (TMA):**
  - **Scopo:** misurare il tempo mediamente necessario all'utente per imparare ad usare il prodotto software;
  - **Risultato:** tempo impiegato dall'utente per prendere familiarità con le funzioni presenti nel prodotto software;
  - **Intervallo accettabile:** non ancora definito;
  - **Intervallo desiderabile:** non ancora definito.
- **Estetica dell'interfaccia (EI):**
  - **Scopo:** misura del grado di attrattività dell'interfaccia del prodotto software;
  - **Risultato:** valutazione in decimi dell'estetica dell'interfaccia del prodotto software;
  - **Intervallo accettabile:** [7, 10]
  - **Intervallo desiderabile:** [8, 10]

## 2.5 Manutenibilità

È la capacità del prodotto software di essere corretto, migliorato o adattato per cambiamenti nei requisiti, nelle specifiche funzionali o nell'ambiente operativo.

### 2.5.1 Caratteristiche

- **Analizzabilità:** capacità del prodotto software ad essere diagnosticato per individuare le cause di errori o malfunzionamenti;
- **Modificabilità:** capacità del prodotto software di permettere l'attuazione di una o più modifiche;
- **Stabilità:** capacità del prodotto software di evitare effetti indesiderati dovuti alle modifiche apportate;
- **Testabilità:** capacità del prodotto software di permettere la verifica e la validazione delle modifiche apportate tramite test;

### 2.5.2 Metriche di valutazione

- **Presenza di commenti (PC):**
  - **Scopo:** misurare la facilità di comprensione del codice al fine di semplificarne l'analisi;
  - **Risultato:** percentuale di righe di commento sul totale di righe di codice;
  - **Calcolo:**  $PC = \frac{N_{RCC}}{N_{RC}} * 100$ , dove:
    - $N_{RCC}$  = numero di righe di commento;
    - $N_{RC}$  = numero di righe di codice;
  - **Intervallo accettabile:** [10%, 25%]
  - **Intervallo desiderabile:** [10%, 20%]
- **Complessità ciclomatica (CC):**
  - **Scopo:** misurare la complessità strutturale del codice al fine di semplificare la modifica. Viene definita in riferimento ad un grafo contenente i blocchi base del programma, con un arco tra due blocchi se il controllo può passare dal primo al secondo;
  - **Risultato:** numero massimo di cammini linearmente indipendenti attraverso il codice sorgente;
  - **Calcolo:**  $CC = e - n + p$ , dove:
    - $e$  = numero di archi del grafo;
    - $n$  = numero di nodi del grafo;
    - $p$  = numero delle componenti connesse da ogni arco;
  - **Intervallo accettabile:** [1, 10]
  - **Intervallo desiderabile:** [1, 7]
- **Impatto negativo delle modifiche (INM):**
  - **Scopo:** misurare il comportamento del prodotto software dopo le modifiche effettuate;
  - **Risultato:** percentuale delle modifiche che hanno introdotto errori sul totale di modifiche effettuate;
  - **Calcolo:**  $INM = \frac{N_{ME}}{N_M} * 100$ , dove:
    - $N_{ME}$  = numero di modifiche che hanno apportato errori;

- $N_M$  = numero di modifiche apportate;
  - **Valore accettabile:** 0%
  - **Valore desiderabile:** 0%
- **Copertura dei test (CT):**
  - **Scopo:** misurare la completezza dei test implementati;
  - **Risultato:** percentuale dei test che sono stati implementati rispetto a quelli necessari per una copertura completa del codice. Questa metrica viene ulteriormente analizzata nella [sezione 3.2.3.3](#);
  - **Calcolo:**  $CT = \frac{N_{TE}}{N_{TCC}} * 100$ , dove:
    - $N_{TE}$  = numero di test implementati;
    - $N_{TCC}$  = numero dei test necessari per una copertura completa del codice;
  - **Intervallo accettabile:** [90%, 100%]
  - **Valore desiderabile:** 100%

## 2.6 Portabilità

È la capacità del prodotto software di essere trasferito da un ambiente ad un altro.

### 2.6.1 Caratteristiche

- **Adattabilità:** capacità del prodotto software di essere adattato ad altri ambienti di lavoro senza apportare particolari modifiche per farlo.

### 2.6.2 Metriche di valutazione

- **Browser supportati (BS):**
  - **Scopo:** misurare il livello di adattamento dell'applicativo web di ADDESTRAMENTOGE andando ad individuare i browser e le relative versioni che ne supportano l'esecuzione senza perdita di funzionalità;
  - **Risultato:** lista di coppie (nome, versione minima) dei browser supportati;
  - **Lista minima accettabile:** {(Chrome, 32), (Firefox, 27)}
  - **Lista minima desiderabile:** {(Chrome, 32), (Firefox, 27), (Opera, 19), (Internet Explorer, 10)}

### 3 Qualità di processo

I processi di sviluppo utilizzati durante la creazione di un prodotto software hanno una significativa influenza sulla qualità del prodotto stesso. Una buona gestione e il miglioramento della qualità dei processi può infatti portare più facilmente a un software di buona qualità. Per raggiungere questo obiettivo si è scelto di attenersi allo standard *ISO/IEC 15504*, chiamato anche SPICE, e di seguire lo schema del CICLO DI DEMINGGE (o PDCA<sub>GE</sub>) che fornisce un approccio per il controllo e il miglioramento continuo di processi e prodotti.

La descrizione generale delle metriche di Qualità di Processo è contenuta nelle *Norme di Progetto* nella sezione 3.3.5.

Nei paragrafi seguenti vengono analizzate finalità e strategie per ogni processo e vengono specificati i valori relativi alle metriche specificate nelle *Norme di Progetto*.

#### 3.1 Processi primari

##### 3.1.1 Analisi dei requisiti

Grazie all'analisi le informazioni raccolte e a disposizione contribuiscono a definire CASI D'USO<sub>GE</sub> e requisiti. Viene definito che cosa il sistema dovrebbe fare e quali sono i vincoli che incidono sulla sua implementazione e sulle operazioni che esegue, in ogni sua parte.

###### 3.1.1.1 Finalità

Gli obiettivi sono quelli di definire i CASI D'USO<sub>GE</sub>, scoprire, classificare, organizzare e documentare i requisiti ed ottenere la loro approvazione. Inoltre, è necessario gestire e controllare le loro modifiche nel tempo dovute a qualsiasi tipo di cambiamento.

###### 3.1.1.2 Strategie

- Comprendere lo scopo del progetto e individuare i requisiti anche attraverso l'interazione con gli stakeholder;
- Classificare e organizzare i requisiti in gruppi coerenti (opzionali, obbligatori, ...);
- Assegnare delle priorità ai requisiti, individuando possibili RISCHIG<sub>E</sub> e negoziando eventuali conflitti;
- Documentare i requisiti;
- Revisione dei requisiti;
- Approvazione dei requisiti da parte del proponente.

###### 3.1.1.3 Obiettivi delle metriche

- **Percentuale requisiti obbligatori soddisfatti (PROS):**
  - **Risultato:** percentuale di requisiti obbligatori soddisfatti;
  - **Calcolo:**  $PROS = \frac{N_{ROS}}{N_{ROT}} * 100$ , dove:
    - $N_{ROS}$  = numero di requisiti obbligatori soddisfatti;
    - $N_{ROT}$  = numero di requisiti obbligatori totali;
  - **Valore accettabile:** 100%
  - **Valore desiderabile:** 100%

### 3.1.2 Progettazione architetturale

La progettazione architetturale è un processo in cui si progetta un'organizzazione di un sistema software che soddisfa i suoi requisiti funzionali e non funzionali. Identifica perciò i principali componenti strutturali e le loro interazioni e produce un modello architetturale ad alto livello di dettaglio che descrive come il sistema è organizzato. Le decisioni prese durante il processo di progettazione architetturale influiscono sulle proprietà del sistema e sulla sua realizzazione.

#### 3.1.2.1 Finalità

Gli obiettivi sono di suddividere il sistema in parti di complessità trattabile per meglio comprenderlo e per individuare le interazioni tra questi sottosistemi. Si vuole poi fornire una soluzione architetturale in grado di soddisfare i requisiti e che consenta di agevolare la realizzazione del sistema.

#### 3.1.2.2 Strategie

- Valutare i principali requisiti e CASI D'USO da soddisfare;
- Determinare i principali componenti del sistema;
- Scegliere un adeguato pattern architetturale;
- Identificare i modi in cui i componenti interagiscono tra loro;
- Decidere come saranno distribuite le funzionalità tra i vari componenti;
- Considerare ogni caso d'uso e modificare l'architettura per renderlo realizzabile.

#### 3.1.2.3 Obiettivi delle metriche

- **Structural Fan-in (SFIN):**
  - **Risultato:** valore intero dato dal conteggio delle componenti;
  - **Soglia minima accettabile:** 1
  - **Soglia minima desiderabile:** 2
- **Structural Fan-out (SFOUT):**
  - **Risultato:** valore intero dato dal conteggio delle componenti;
  - **Intervallo accettabile:** [0, 3]
  - **Valore desiderabile:** 0

### 3.1.3 Progettazione di dettaglio

La progettazione di dettaglio segue la progettazione architetturale e prevede di scomporre ulteriormente le macro-componenti precedentemente individuate in moduli che siano trattabili dal singolo sviluppatore e che corrispondano a funzionalità ben definite.

#### 3.1.3.1 Finalità

Le finalità della progettazione di dettaglio sono di definire, verificare e documentare le unità realizzative (moduli) che costituiscono le componenti individuate nell'architettura logica per organizzare meglio il lavoro di programmazione e assicurare conformità con l'architettura stessa, tracciando requisiti e componenti.

#### 3.1.3.2 Strategie

- Scomporre le componenti architetture nei più piccoli moduli che sia utile rappresentare;
- Documentare, verificare e realizzare le singole unità individuate.

### 3.1.3.3 Obiettivi delle metriche

- **Coupling between objects (CBO):**
  - **Risultato:** valore intero dato dall'accoppiamento delle classi;
  - **Intervallo accettabile:** [0, 7]
  - **Intervallo desiderabile:** [0, 5]

## 3.2 Processi di supporto

### 3.2.1 Pianificazione

La pianificazione di progetto è un processo iterativo che inizia quando viene creato il primo piano di progettazione e che termina quando il progetto è completato. Un piano di progettazione definisce le risorse disponibili, l'organizzazione dei ruoli, l'analisi dei possibili RISCHIE del progetto e le strategie per limitarli, la suddivisione e le tempistiche dei lavori da svolgere.

#### 3.2.1.1 Finalità

Gli obiettivi del processo di pianificazione prevedono di definire i vincoli in termini di budget e tempo che influiscono sulla gestione del progetto e descrivere in che modo sono organizzati i ruoli del team di sviluppo. Comporta inoltre di stabilire delle BASELINEGE durante il progetto per evitare scostamenti dal budget preventivato e cerca di definire standard di processo quando possibile, al fine di facilitare il lavoro di gruppo e di seguire una metodologia di lavoro volta ad un miglioramento continuo.

#### 3.2.1.2 Strategie

- Definire un piano di progettazione;
- Aggiornare periodicamente il piano di sviluppo in base alle nuove informazioni a disposizione sul software e sul suo avanzamento.

### 3.2.1.3 Obiettivi delle metriche

- **Schedule Variance (SV):**
  - **Risultato:** valore intero dato dal conteggio delle componenti;
  - **Calcolo:**  $SV = BCWP - BCWS$ , dove:
    - **Budgeted Cost of Work Performed (BCWP):** valore delle attività realizzate fino al momento della misurazione;
    - **Budgeted Cost of Work Scheduled (BCWS):** costo pianificato per realizzare le attività di progetto fino alla rilevazione;
  - **Intervallo accettabile:** [-2, 2] (il numero di attività svolte sono considerate accettabili se in confronto alle attività pianificate se ne sono svolte o 2 in meno o 2 in più. Si sono limitate le attività in anticipo per rispettare la qualità di ciò che si è prodotto)
  - **Intervallo desiderabile:** [0, 2] (il numero desiderabile di attività svolte in confronto alle attività pianificate può variare tra 0 e 2. Si sono limitate le attività in anticipo per rispettare la qualità di ciò che si è prodotto)
- **Budget Variance (BV):**
  - **Risultato:** valore percentuale di tolleranza dei costi sostenuti;
  - **Calcolo:**  $BV = \frac{BCWS - ACWP}{BCWS} * 100$ , dove:
    - **Budgeted Cost of Work Scheduled (BCWS):** costo pianificato per realizzare le attività di progetto fino alla rilevazione;

- **Actual Cost of Work Performed (ACWP):** costo effettivamente sostenuto alla rilevazione corrente;
- **Intervallo accettabile:** [-10%, 10%] (margine di tolleranza dei costi sostenuti)
- **Intervallo desiderabile:** [0%, 10%] (margine di tolleranza dei costi sostenuti)

### 3.2.2 Documentazione

Questo processo ha il compito di rilasciare documenti, leggibili e comprensibili, che riportino le azioni intraprese, gli strumenti utilizzati e le modifiche operate, rimanendo pertanto attivo per tutta la durata dello sviluppo del prodotto.

#### 3.2.2.1 Finalità

L'obiettivo è quello di produrre un'adeguata documentazione per il progetto che sia completa coprendo ogni aspetto dello sviluppo, non presenti ambiguità o contraddizioni, sia di facile comprensione e consultazione, oltre che ortograficamente e sintatticamente corretta.

#### 3.2.2.2 Strategie

- Redazione dei documenti in modo collaborativo durante le attività di sviluppo;
- Creazione e aggiornamento del glossario interno ed esterno;
- Verifica della documentazione prodotta;
- Approvazione della documentazione.

#### 3.2.2.3 Obiettivi delle metriche

- **Indice di Gulepease<sub>GE</sub> (IG):**
  - **Risultato:** valore intero compreso tra 0 e 100 dove il valore "100" indica la leggibilità più alta e "0" la leggibilità più bassa. In generale i testi con indice inferiore a 80 sono difficili da leggere per chi ha la licenza elementare, quelli con indice inferiore a 60 sono difficili da leggere per chi ha la licenza media e se l'indice è inferiore a 40 risultano difficili da leggere per chi ha un diploma superiore;
  - **Calcolo:**  $IG = 89 + \frac{300 \cdot N_F - 10 \cdot N_L}{N_P}$ 
    - $N_F$  = numero delle frasi;
    - $N_L$  = numero delle lettere;
    - $N_P$  = numero delle parole;
  - **Intervallo accettabile:** [40, 100]
  - **Intervallo desiderabile:** [60, 100]
- **Correttezza ortografica:**
  - **Risultato:** valore intero che indica il numero di errori grammaticali o ortografici presenti nel documento;
  - **Valore accettabile:** 0
  - **Valore desiderabile:** 0

### 3.2.3 Verifica

Il processo di verifica si occupa di valutare la correttezza dei prodotti e dei processi stabilendo se questi presentino o meno errori o difetti e se è stato garantito un adeguato livello di qualità. Tale processo rimane attivo per tutta la durata del progetto.



### 3.2.3.1 Finalità

La verifica mira ad individuare eventuali errori e CRITICITÀ del progetto ponendovi rimedio e garantendo così il soddisfacimento dei requisiti e delle *Norme di Progetto*.

### 3.2.3.2 Strategie

- Operare mediante tecniche predefinite e, quando possibile, automatiche;
- Migliorare tali tecniche attraverso l'esperienza e la collaborazione;
- Effettuare una verifica costante.

### 3.2.3.3 Obiettivi delle metriche

- **Copertura dei test (CT):**
  - **Risultato:** valore percentuale che esprime la copertura del codice testato. La copertura dei test si suddivide in:
    - **Function coverage (CT-FC):**
      - Intervallo accettabile: [96%, 100%]
      - Intervallo desiderabile: [98%, 100%]
    - **Statement coverage (CT-SC):**
      - Intervallo accettabile: [95%, 100%]
      - Intervallo desiderabile: [96%, 100%]
    - **Branch coverage (CT-BC):**
      - Intervallo accettabile: [92%, 100%]
      - Intervallo desiderabile: [95%, 100%]
    - **Condition coverage (CT-CC):**
      - Intervallo accettabile: [96%, 100%]
      - Intervallo desiderabile: [98%, 100%]

## 3.3 Processi organizzativi

### 3.3.1 Gestione della qualità

Questo processo consiste di attività atte a garantire il raggiungimento degli obiettivi di qualità.

#### 3.3.1.1 Finalità

Lo scopo del processo di gestione della qualità è quello di assicurare che il prodotto e i processi rispettino gli standard di qualità richiesti dal proponente.

#### 3.3.1.2 Strategie

Questo processo prevede:

- Pianificazione delle attività da svolgere per raggiungere gli obiettivi di qualità prefissate:
  - Verifiche ispettive;
  - Azioni preventive;
  - Azioni correttive;
- Utilizzo di procedure e metriche come metodo di valutazione;
- Valutazione dei risultati con conseguente aggiornamento di obiettivi, procedure e metriche.

### 3.3.1.3 Obiettivi delle metriche

- **Metriche soddisfatte (MS):**
  - **Risultato:** percentuale di metriche soddisfatte sul totale di metriche proposte;
  - **Calcolo:**  $MS = \frac{N_{MS}}{N_M} * 100$ , dove:
    - $N_{MS}$  = numero di metriche che rientrano nell'intervallo accettabile;
    - $N_M$  = numero di metriche totali;
  - **Intervallo accettabile:** [70%, 100%]
  - **Intervallo desiderabile:** [80%, 100%]
- **Metriche pienamente soddisfatte (MPS):**
  - **Risultato:** percentuale di metriche pienamente soddisfatte sul totale di metriche proposte;
  - **Calcolo:**  $MPS = \frac{N_{MPS}}{N_M} * 100$ , dove:
    - $N_{MS}$  = numero di metriche che rientrano nell'intervallo desiderabile;
    - $N_M$  = numero di metriche totali;
  - **Intervallo accettabile:** [60%, 100%]
  - **Intervallo desiderabile:** [70%, 100%]

## A Standard di Qualità

### A.1 ISO/IEC 9126

*ISO/IEC 9126 (1991): Software product evaluation - Quality characteristics and guidelines for their use* - era lo standard internazionale sviluppato per definire le caratteristiche di qualità appropriate tenendo conto dell'uso finale del prodotto.

Questo standard ha definito sei caratteristiche di qualità che descrivono il modello del processo di valutazione del prodotto software.

Lo standard è suddiviso in 4 parti riportate di seguito.

#### A.1.1 Modello della qualità

Il modello di qualità del software definisce le sei caratteristiche principali, suddivise a loro volta in sotto-caratteristiche, misurabili attraverso specifiche metriche, al fine di fornire una scala e un metodo per la misurazione. Vengono riportate le caratteristiche alla base del modello.

##### A.1.1.1 Funzionalità

Caratteristica che descrive la capacità del software di soddisfare i requisiti, approfonditi *nell'Analisi dei Requisiti*, in un determinato contesto.

Affinché il software abbia questa caratteristica deve essere:

- **Appropriato:** deve fornire appropriate funzioni per specifiche attività, che permettano il raggiungimento degli obiettivi prefissati;
- **Accurato:** deve fornire i risultati concordati o la precisione ricercata;
- **Interoperabile:** l'interazione e l'operabilità tra uno o più sistemi;
- **Sicuro:** la capacità di fornire la protezione per le informazioni e i dati;
- **Conforme alla funzionalità:** l'adesione agli standard.

##### A.1.1.2 Affidabilità

Caratteristica del software di mantenere un determinato livello di prestazioni, in condizioni specifiche per un limitato periodo di tempo.

- **Maturità:** l'abilità di evitare il verificarsi di errori, malfunzionamenti o risultati non desiderati;
- **Robustezza:** il mantenimento del livello delle prestazioni stabilite anche al verificarsi di malfunzionamenti o uno scorretto uso del prodotto;
- **Recuperabilità:** la ricostituzione del livello appropriato delle prestazioni o il recupero di informazioni rilevanti anche successive ad un malfunzionamento;
- **Conforme alla affidabilità:** l'adesione agli standard in merito all'affidabilità.

##### A.1.1.3 Efficienza

Il prodotto ottiene questa capacità se esegue le proprie funzioni minimizzando il tempo d'esecuzione e sfruttando le risorse al meglio.

- **Tempo:** la capacità di reagire velocemente con opportuni tempi di risposta tendenzialmente brevi;
- **Spazio:** l'uso corretto e efficiente delle risorse disponibili;
- **Conforme all'efficienza:** l'adesione agli standard in merito all'EFFICIENZA.

#### A.1.1.4 Usabilità

Caratteristica per la quale l'uso del prodotto software permette una comprensione e una facilità nell'utilizzo da parte dell'utente, quando usato sotto determinate condizioni.

Per il raggiungimento di tale risultato, il prodotto deve essere:

- **Comprensibile:** l'essere chiaro in merito alle funzionalità e l'utilizzo;
- **Apprendibile:** l'essere facile da apprendere dagli utenti;
- **Operabile:** il permettere all'utente di poter eseguire le attività che vuole eseguire controllandone l'uso;
- **Attrattivo:** la capacità di attrarre l'utente per l'utilizzo;
- **Conforme all'usabilità:** l'adesione agli standard in merito all'usabilità.

#### A.1.1.5 Manutenibilità

Capacità del software nel poter essere modificato, ottenendo aggiornamenti per correzioni, miglioramenti o adattamenti.

Per il raggiungimento di tale caratteristica il prodotto deve poter essere:

- **Analizzabile:** facilità nell'analisi per la localizzazione di errori;
- **Modificabile:** avere il codice facilmente modificabile ed avere una progettazione e documentazione semplicemente rettificabili;
- **Stabile:** l'evitare effetti non desiderati a seguito di modifiche;
- **Testabile:** poter essere facilmente testato, per la validazione delle modifiche apportate;
- **Conforme alla manutenibilità:** l'adesione agli standard in merito alla manutenibilità.

#### A.1.1.6 Portabilità

Caratteristica che permette al prodotto di poter essere utilizzato anche dopo essere stato spostato in un ambiente di lavoro differente che sia di tipologia hardware o software.

- **Adattabile:** il potersi adattare a differenti ambienti operativi, senza la necessità di modifiche;
- **Installabile:** la possibilità di essere installato in un determinato ambiente;
- **Coesistente:** lo stare in un ambiente condiviso con altre applicazioni e risorse comuni;
- **Sostituibile:** la capacità di poter sostituire un altro software svolgendo gli stessi compiti nello stesso ambiente;
- **Conforme alla portabilità:** l'adesione agli standard in merito alla portabilità.

#### A.1.2 Metriche di qualità esterne

La qualità esterna è l'insieme di tutte le caratteristiche del prodotto software da un punto di vista esterno; viene generalmente misurata e valutata eseguendo il software ed effettuando i test in un ambiente simulato con dati simulati utilizzando metriche esterne. Durante i test, la maggior parte degli errori dovrebbero essere scoperti e corretti, tuttavia non sempre tali errori vengono evidenziati. Poiché, è difficile correggere l'architettura del software o altri aspetti fondamentali della progettazione del software, la progettazione fondamentale di solito rimane invariata durante i test. Viene rilevata tramite analisi dinamica. Idealmente la qualità esterna determina la qualità in uso.

### A.1.3 Metriche di qualità interne

La qualità interna è l'insieme delle caratteristiche del prodotto software da un punto di vista interno. La qualità interna viene misurata e valutata rispetto ai requisiti di qualità interna. I dettagli sulla qualità del prodotto software possono essere migliorati durante l'implementazione, la revisione del codice e i test, ma la natura fondamentale della qualità del prodotto software rappresentata dalla qualità interna rimane invariata se non ri-progettata.

### A.1.4 Metriche di qualità in uso

Qualità in uso è la visione dell'utente riguardo la qualità del prodotto software quando viene utilizzato in un ambiente e in un contesto specifico di utilizzo. Misura la metrica con cui gli utenti<sup>1</sup> possono raggiungere i propri obiettivi in un determinato ambiente, piuttosto che misurare le proprietà del software stesso.

Il livello di qualità nell'ambiente degli utenti può essere diverso da quello nell'ambiente degli sviluppatori, a causa delle differenze tra le esigenze e le capacità dei diversi utenti e le differenze tra i diversi ambienti hardware e di supporto. L'utente valuta solo quegli attributi del software, che vengono utilizzati per i suoi compiti. A volte, gli attributi software specificati da un utente finale durante la FASEGE di analisi dei requisiti, non soddisfano più i requisiti dell'utente quando il prodotto è in uso, a causa della modifica dei requisiti dell'utente e della difficoltà di specificare le esigenze implicite.

Le metriche vengono applicate solo al prodotto finito ed in uso in condizioni reali. Tale qualità viene raggiunta solo se vengono raggiunti i livelli di qualità interna e qualità esterna.

## A.2 ISO/IEC 15504

Il modello *ISO/IEC 15504* conosciuto come SPICE, acronimo di Software Process Improvement and Capability Determination è il modello standard per la valutazione oggettiva della qualità dei processi di sviluppo del software. Ogni valutazione permette di determinare il livello di maturità raggiunto e la capacità di ogni progetto tramite degli attributi, inoltre lo studio dei range risultanti permette di definire il margine di miglioramento.

I risultati delle singole valutazioni devono essere ripetibili, oggettivi e comparabili, perché possano contribuire al miglioramento dei processi.

Gli attributi delle capacità di processo sono:

- **Process performance:** misura del raggiungimento degli obiettivi fissati;
- **Performance management:** misura del grado di organizzazione con cui sono raggiunti gli obiettivi fissati;
- **Work product management:** misura della gestione corretta dei prodotti in merito alla documentazione, controllo e verifica;
- **Process definition:** misura dell'applicazione dello standard;
- **Process deployment:** misura del rilascio e distribuzione di un processo standard definito in modo tale da ottenere sempre gli stessi risultati;

<sup>1</sup> Utenti si riferisce a qualsiasi tipo di utente previsto, inclusi operatori e manutentori, e i loro requisiti possono essere diversi

- **Process measurement:** indica il grado in cui i risultati delle misure sono utilizzati per garantire che il processo raggiunga i suoi obiettivi;
- **Process control:** misura se il processo è stabile, capace e predicibile entro certi limiti;
- **Process innovation:** misura delle modifiche identificate da apportare al processo in FASEGE di analisi delle performance e in studio di approcci innovativi;
- **Process optimization:** misura dell'impatto dei cambiamenti dell'organizzazione, delle performance e della definizione del processo in funzione del raggiungimento degli obiettivi di miglioramento dei processi.

I livelli di misurazione degli attributi sono:

- **N - not implemented** (0-15%): il processo non possiede l'attributo o dimostra gravi carenze;
- **P - partially implemented** (15-50%): presenza di un approccio SISTEMATICOGE volto al possesso di un attributo già significativamente ottenuto, ma possiede aspetti non ancora prevedibili;
- **L - largely implemented** (50-85%): presenza di un approccio sistematico volto al possesso di un attributo già significativamente ottenuto, ma l'attuazione varia tra attività diverse;
- **F - fully implemented** (85-100%): attributo completamente ottenuto, seguendo un approccio sistematico e un'attuazione corretta per tutte le unità.

Basandosi sulla classificazione degli attributi, i livelli di capacità di un processo sono:

1. **Incomplete:** il processo è incompleto in quanto non è stato implementato o fallisce nel raggiungimento dell'obiettivo. Nessun attributo associato.
2. **Performed:** il processo è stato implementato e ha successo nel raggiungere l'obiettivo. L'attributo associato a questo livello è "process performance".
3. **Managed:** il processo che aveva già raggiunto il livello *performed*, è stato implementato ulteriormente con più organizzazione, tramite pianificazione, controllo e correzioni sui prodotti sicuri. Gli attributi associati a questo livello sono: "performance management" e "work product management".
4. **Established:** il processo che aveva già raggiunto il livello *managed*, è stato implementato ulteriormente come processo che è in grado di raggiungere sempre gli stessi risultati. Gli attributi associati a questo livello sono "process definition" e "process distribution".
5. **Predictable:** il processo che aveva già raggiunto il livello *established*, è stato implementato ulteriormente operando entro limiti definiti per raggiungere i risultati ambiti. Gli attributi associati a questo livello sono "process control" e "process measurement".
6. **Optimizing:** il processo che aveva già raggiunto il livello *predictable*, è stato implementato ulteriormente applicando il miglioramento continuo per raggiungere gli obiettivi di progetto/aziendali.  
Gli attributi associati a questo livello sono "process change" e "process improvement".

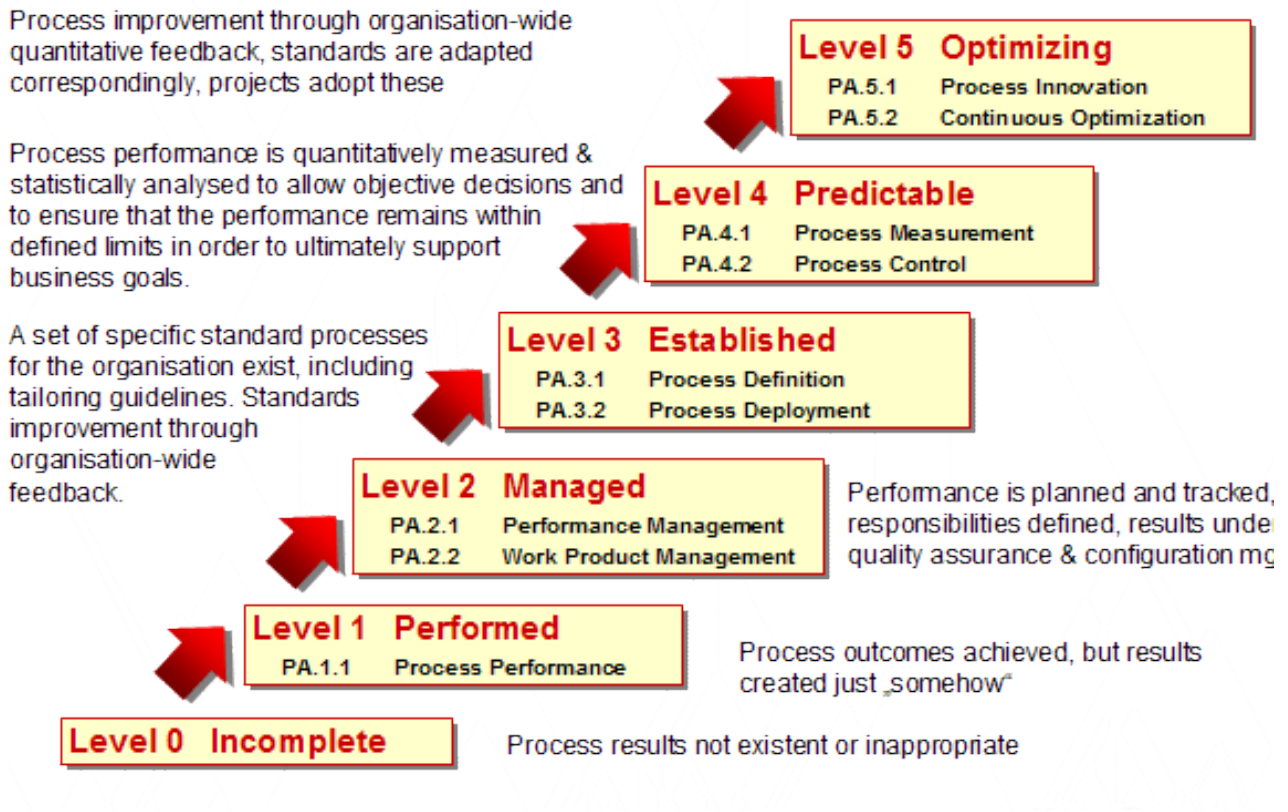


FIGURA 1 - SPICE CAPABILITY (FONTE: HM&amp;S)

### A.3 Ciclo di Deming

Il CICLO DI DEMINGGE (o PDCA, acronimo di Plan-Do-Check-Act) è un metodo di gestione iterativo, suddiviso in quattro fasi finalizzate al controllo dei processi e del miglioramento continuo della qualità dei processi e dei prodotti.

Come specificato dall'acronimo le quattro fasi di cui è costituito sono:

- **Plan:** FASEGE dedicata alla pianificazione degli obiettivi di miglioramento. In questa fase vengono definite le attività da svolgere, le risorse da assegnarvi e le scadenze;
- **Do:** fase di attuazione delle attività pianificate;
- **Check:** fase dedicata alla verifica per l'accertamento delle attività svolte nella fase *Do* rispetto alle attività pianificate nella fase *Plan* e all'esito ottenuto, se positivo o negativo.
- **Act:** fase di attuazione per standardizzare i processi che hanno subito modifiche, che si sono ritenute positive.

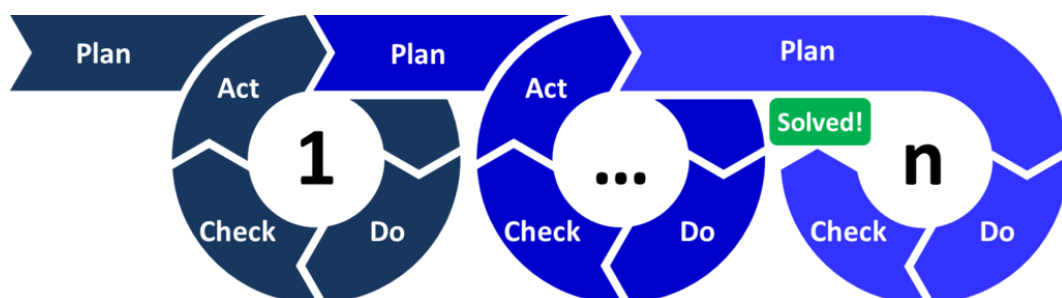


FIGURA 2 - RAPPRESENTAZIONE PDCA (FONTE: WIKIPEDIA)



## B Resoconto attività di verifica

In questa sezione dell'Appendice vengono posti gli esiti delle attività di verifica svolte sui documenti consegnati alle revisioni di avanzamento.

### B.1 Revisione dei Requisiti

#### B.1.1 Analisi statica dei documenti

I documenti analizzati mediante la tecnica WALKTHROUGHGE ha permesso l'individuazione di errori frequenti, che sono stati inseriti in una check list, che permetta una correzione più rapida e dunque un'attività di verifica più breve. Tramite questa lista sarà possibile l'uso della tecnica INSPECTIONGE per le attività di verifica future.

#### B.1.2 Esiti verifiche automatizzate

La Tabella 1 riporta gli INDICI DI GULPEASEGE dei documenti realizzati fino ad ora.

Documento	Data Inizio	Data Fine	Indice Gulpease	Stato
<i>Norme Di Progetto v.1.0.0</i>	2019/12/04	2020/01/08	54	Superato
<i>Analisi Dei Requisiti v.1.0.0</i>	2019/12/19	2020/01/08	63	Superato
<i>Studio Di Fattibilità v.1.0.0</i>	2019/12/10	2019/12/15	47	Superato
<i>Piano Di Progetto v.1.0.0</i>	2019/12/05	2020/01/12	50	Superato
<i>Piano Di Qualifica v.1.0.0</i>	2019/12/22	2020/01/11	46	Superato
<i>Glossario Interno v.1.0.0</i>	2019/12/04	2020/01/11	48	Superato
<i>Glossario Esterno v.1.0.0</i>	2019/12/05	2020/01/11	52	Superato
<i>Lettera Di Presentazione</i>	2019/01/13	2019/01/13	73	Superato
<i>VII_2019-11-20</i>	2019/11/20	2019/11/20	51	Superato
<i>VI2_2019-12-04</i>	2019/12/04	2019/12/04	49	Superato
<i>VI3_2019-12-16</i>	2019/12/16	2019/12/16	55	Superato
<i>VI4_2019-12-19</i>	2019/12/19	2019/12/19	56	Superato
<i>VI5_2020-01-10</i>	2020/01/10	2020/01/10	54	Superato
<i>VE_2019-12-18</i>	2019/12/18	2019/12/18	52	Superato

**TABELLA 1 - ESITI INDICI DI GULPEASE**



## C Valutazioni per il miglioramento

In questa sezione dell'Appendice si riporta l'autovalutazione del gruppo Carbon12 in merito al processo produttivo svolto nel primo periodo. Si è discusso dei problemi sorti e valutato le modalità di risoluzione in modo efficiente onde evitare la ripetizione dei medesimi errori e che tali problematiche si ripresentino in futuro. L'esito dell'autovalutazione è riportato nei paragrafi seguenti.

I membri del gruppo hanno esternato le problematiche riscontrate durante il lavoro svolto e il confronto collegiale ha estrapolato tre ambiti principali:

- **Organizzazione:** problemi inerenti all'organizzazione del gruppo;
- **Ruoli:** problemi inerenti allo svolgimento di un ruolo;
- **Strumenti:** problemi inerenti all'uso degli strumenti impiegati.

Essendo questa un'autovalutazione non potrà essere pienamente accurata, come lo sarebbe un'analisi compiuta da un esperto esterno e nemmeno obiettiva o sufficientemente EFFICACE<sup>2</sup>; nonostante ciò porterà sicuramente dei contributi positivi e costruttivi nei processi lavorativi rendendo il team più SISTEMATICO<sup>2</sup>, DISCIPLINATO<sup>2</sup>, collaborativo, QUANTIFICANDO<sup>2</sup> il lavoro svolto.

Per far sì che questo sistema funzioni, i membri del gruppo necessariamente devono essere onesti, esponendo i propri problemi e pensieri riguardanti le attività e metodologie lavorative applicate, al fine di solidificare la coesione tra tutti i componenti del gruppo.

Questa sezione verrà aggiornata se si dovessero riscontrare altri parametri di miglioramento necessari alla maturazione del team.

### C.1 Valutazione sull'organizzazione

Il problema riscontrato riguarda l'organizzazione degli incontri con tutti i membri del gruppo. Essendo il team composto da studenti e non da lavoratori non tutti sono sempre disponibili in caso fosse necessaria una riunione, sia per motivi di studio, per seguire un corso, per lavoro o anche per motivazioni personali.

Questa difficoltà si è presentata nel periodo iniziale dello svolgimento del lavoro, ma il gruppo si è adoperato in modo tale che gli incontri di persona fossero limitati alle occasioni in cui era strettamente necessario, utilizzando strumenti software di organizzazione e comunicazione, come Windows Teams. Tale software ha permesso di comunicare lo svolgimento delle attività, la pianificazione tramite un Planner e la possibilità di effettuare chiamate nelle quali era possibile anche la condivisione dello schermo.

Si è poi deciso che, nel qual caso il team ritenesse necessario un incontro con tutti i membri, la pianificazione di tale incontro deve essere antecedente di almeno due settimane, per permettere ai membri del team di organizzarsi in modo tale da essere liberi nel giorno stabilito.

Per quanto concerne gli incontri con membri esterni al gruppo, si è deciso in primis di valutare l'importanza degli argomenti da trattare e a seconda di ciò misurare un lasso di tempo sufficiente per il gruppo di elaborare tali problemi e per l'individuo esterno per organizzare una data per l'incontro. Il team ha poi stabilito che al momento dell'organizzazione dell'incontro, gli argomenti che verranno trattati (argomenti all'Ordine del Giorno, esposti nei verbali) vengono decisi in anticipo facendo uso di sondaggi, che permettano di classificare quali siano gli argomenti di maggior rilevanza assegnando

<sup>2</sup> Vedi *Quantificabile* nel Glossario Esterno

delle priorità. Inoltre, questi sondaggi possono aiutare a definire anche la durata possibile dell'incontro, che sia virtuale o fisico.

## **C.2 Valutazione sui ruoli**

Attualmente i membri del team hanno svolto le attività nei panni di Amministratore, Responsabile, Analista e Verificatore.

Vengono successivamente elencati difetti e problematiche riscontrate nella prima FASEGE di analisi.

### **C.2.1 Amministratore**

Il difetto riscontrato in questa figura sono i compiti che possiede, in quanto spesso è marginale, poiché il Responsabile quasi istintivamente assumeva i compiti dell'Amministratore. L'Amministratore è stato sicuramente utile nelle fasi iniziali per la scelta degli strumenti, ma il suo ruolo è stato quasi sovrastato dal Responsabile di progetto. Se necessario per motivi di budget la figura dell'Amministratore in alcune fasi del progetto potrebbe non comparire ed essere sostituito dal Responsabile.

### **C.2.2 Responsabile**

La problematica emersa per il Responsabile riguarda il sovraccaricamento del lavoro assunto, come accennato nel paragrafo precedente il Responsabile ha inglobato il ruolo dell'Amministratore in questo periodo di analisi iniziale. A causa della carenza di esperienza, si è deciso di limitare il ruolo del Responsabile, suddividendo più accuratamente i compiti che gli concernono e in caso di farsi assistere da un Amministratore.

Il Responsabile si è spesso trovato in difficoltà nella gestione delle diverse opinioni del team, ciò però ha permesso una maturazione da parte di tutti i membri, in quanto i soggetti in contrasto si sono adoperati per gestire il problema collaborativamente, supervisionati, ove necessario, dal Responsabile per gestire la situazione in modo costruttivo senza far sentire a disagio nessuna delle due parti.

Infine, il Responsabile ha avuto difficoltà, dovute all'inesperienza, nella gestione del lavoro altrui, in quanto doveva assegnare le attività rispettando gli impegni che potevano avere i suoi collaboratori. Ciò ha portato una maturazione nei responsabili nella gestione collaborativa di un gruppo numeroso di persone.

### **C.2.3 Analista**

Si sono riscontrate difficoltà nel decomporre e classificare i requisiti per lo studio svolto nell'Analisi dei Requisiti. Questa complicazione è dovuta alle poche conoscenze sull'argomento e sulla difficoltà nella sua analisi. Per ovviare a ciò tutto il team ha collaborato per ottenere la qualità di analisi desiderata.

### **C.2.4 Verificatore**

Per i verificatori è sorta una complicazione nell'analisi dei documenti per la verifica della correttezza e completezza, per motivi di poca conoscenza degli argomenti trattati, ma anche nello stabilire le correzioni da apportare.

I Verificatori hanno perciò dedicato molta attenzione e studio degli argomenti trattati, ottenendo così un'analisi più veritiera.

### C.3 Valutazione sull'uso degli strumenti

Le problematiche riguardanti gli strumenti usati sono di tipologia generale.

Un problema emerso riguarda il prestare attenzione alla numerazione del versionamento dei documenti Word. In uno degli ultimi incontri fatti dal team si è discusso di questa problematica stabilendo che i membri che fanno modifiche a tali documenti devono porre molta più attenzione ad aggiornare il numero di versione.

I componenti del team hanno poi riscontrato una certa difficoltà nella gestione dei commenti in Word e Drive, principalmente dovuti all'inesperienza. Un esempio di ciò che si è riscontrato su Drive è l'invio di e-mail a tutti i membri del team, anche se rivolti ad un unico individuo. Per quanto riguarda Windows Teams, i commenti non vengono notificati al soggetto interessato, per ovviare a ciò si è deciso di commentare senza specificare il nome e notificare al membro interessato tramite la chat dei canali appositamente creati a tal scopo.

Alcune volte le comunicazioni su *Microsoft Teams* venivano perse e dunque qualche membro rimaneva poco aggiornato: si è pertanto ricordato a tutti i membri di verificare frequentemente le novità presenti sulla piattaforma, in quanto per comunicazioni generali non viene menzionata una specifica persona ma viene scritto un messaggio normale.

## Registro delle modifiche

Versione	Data	Descrizione	Nominativo	Ruolo
1.0.0	2020/01/13	Approvazione del documento	Giacomo Callegari	Responsabile
0.3.0	2020/01/12	Verifica del documento	Andrea Longo	Verificatore
0.2.2	2020/01/11	Redazione Appendice B	Nicolò Fassina	Verificatore
0.2.1	2020/01/10	Correzioni e integrazioni al documento	Francesco Gobbo	Amministratore
0.2.0	2020/01/09	Revisione del documento	Manuel De Franceschi	Verificatore
0.1.2	2020/01/08	Redazione Appendice C	Francesco Gobbo	Amministratore
0.1.1	2020/01/04	Correzioni alle metriche	Alessandro Lovo	Amministratore
0.1.0	2020/01/04	Revisione del documento	Manuel De Franceschi	Verificatore
0.0.9	2020/01/03	Redazione della sezione 3.3 – Processi Organizzativi	Alessandro Lovo	Amministratore
0.0.8	2020/01/02	Redazione della sezione 3.1 – Processi Primari, 3.2 – Processi di Supporto	Alessandro Lovo	Amministratore
0.0.7	2019/12/30	Redazione della sezione 2.4 – Usabilità, 2.5 – Manutenibilità, 2.6 - Portabilità	Nicolò Fassina	Amministratore
0.0.6	2019/12/29	Redazione della sezione 2.1 – Funzionalità, 2.2 – Affidabilità, 2.3 – Efficienza	Nicolò Fassina	Amministratore
0.0.5	2019/12/26	Revisione struttura	Nicolò Fassina	Amministratore
0.0.4	2019/12/24	Redazione Appendice A	Francesco Gobbo	Amministratore
0.0.3	2019/12/22	Redazione capitolo 1 – Introduzione	Francesco Gobbo	Amministratore
0.0.2	2019/12/04	Redazione struttura documento	Francesco Gobbo	Amministratore
0.0.1	2019/11/29	Creazione del documento	Andrea Longo	Responsabile