

1 Qualità di processo

Da processi scadenti derivano prodotti scadenti. La qualità di processo è quindi 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.

Per garantire la qualità di processo abbiamo deciso di adottare il Capability Maturity Model (CMM) che definisce una scala, suddivisa in cinque livelli, per misurarne la maturità.

Le misurazioni ottenute vengono utilizzate all'interno della strategia di miglioramento continuo della qualità, realizzata tramite Plan-Do-Check-Act (PDCA).

Per maggiori informazioni su CMM e PDCA consultare le rispettive appendici A e B.

Inoltre, sono stati individuati dallo standard ISO/IEC 12207:2008 i processi ritenuti più importanti nel ciclo di vita del prodotto. Per ciascuno di essi sono stati individuati gli obiettivi di qualità e i rispettivi intervalli di accettabilità e ottimalità.

Per quantificare gli obiettivi di qualità vengono utilizzate delle metriche, descritte nel documento “*Norme di Progetto v2.0.0*”.

Viene assegnato un codice identificativo ad ogni obiettivo al fine di semplificarne il tracciamento. Il metodo di denominazione è descritto nel documento “*Norme di Progetto v2.0.0*”.

1.1 Infrastructure Management Process

Questo processo si occupa di mantenere, monitorare e modificare l'infrastruttura per assicurare che essa continui ad eseguire i servizi necessari allo svolgimento del progetto. Con infrastruttura si intendono elementi hardware, software, metodi, strumenti, tecniche e standard impiegati nello sviluppo del prodotto.

1.1.1 Strategie

Per tutto l'arco del progetto l'infrastruttura dovrà possedere le seguenti caratteristiche:

- tutti gli strumenti e le procedure per utilizzarli verranno descritti esaurientemente nel documento “*Norme di Progetto v2.0.0*”;
- la piattaforma PragmaDB sarà disponibile ogni qual volta un componente del gruppo ne richiederà l'accesso;
- i dati contenuti in PragmaDB saranno sempre coerenti e aggiornati;
- la piattaforma PragmaDB estenderà le proprie funzionalità in base alle esigenze del progetto;

1.1.2 Obiettivi di qualità

1.1.2.1 Disponibilità PragmaDB - OPC1

Indica la disponibilità di utilizzo della piattaforma di PragmaDB rispetto alla richiesta. Questo valore viene espresso in percentuale.

- **Metrica utilizzata:** percentuale di accessi avvenuti correttamente a PragmaDB (MPC1);
- **Soglia di accettabilità:** $80\% \leq X \leq 100\%$;
- **Soglia di ottimalità:** $90\% \leq X \leq 100\%$;

1.2 Project Planning, Assessment & Control Process

Questo processo (derivante dall'unione dei processi Project Planning Process e Project Assessment & Control Process) si occupa del necessario per la pianificazione del progetto il quale deve contenere la scelta del modello del ciclo di vita del prodotto, la pianificazione dei tempi e costi da sostenere, la descrizione dei compiti e delle attività associate, l'allocazione dei compiti e delle responsabilità e le metriche atte a misurare lo stato del progetto rispetto la pianificazione.

1.2.1 Strategie

Lo sviluppo del progetto dovrà seguire la pianificazione, in particolare:

- il progetto dovrebbe rispettare i tempi indicati nel documento “*Piano di Progetto v2.0.0*”;
- il progetto dovrebbe rispettare i costi indicati nel documento “*Piano di Progetto v2.0.0*”;
- ruoli e compiti verranno descritti esaurientemente nel documento “*Piano di Progetto v2.0.0*”.

1.2.2 Obiettivi di qualità

1.2.2.1 Rispetto dei tempi - OPC2

Indica se i tempi pianificati sono stati rispettati.

- **Metrica utilizzata:** Schedule Variance (MPC2);
- **Soglia di accettabilità:** $\geq -5\%$;
- **Soglia di ottimalità:** $\geq 0\%$.

1.2.2.2 Rispetto dei costi - OPC3

Indica se i costi pianificati, in data corrente, sono rispettati.

- **Metrica utilizzata:** Cost Variance percentuale (MPC3);
- **Soglia di accettabilità:** $\geq -10\%$;
- **Soglia di ottimalità:** $\geq 0\%$.

Eventuali costi non accettabili dovranno essere compensati entro la fine dell'attività di progetto in quanto non è permesso eccedere i costi preventivati oltre la soglia definita.

1.3 Risk Management Process

Questo processo identifica, analizza, tratta e monitora continuamente i rischi che possono sorgere nel ciclo di vita del progetto.

1.3.1 Strategie

Il gruppo dovrà gestire correttamente i rischi, in particolare:

- all'inizio della fase di progetto, i rischi devono essere individuati e descritti nel documento “*Piano di Progetto v2.0.0*”;
- per ogni rischio, devono essere definite strategie per il riconoscimento e il trattamento;
- all'inizio di ogni periodo, l'analisi dei rischi permetterà l'individuazione di eventuali nuovi rischi;

- il livello di probabilità che i rischi si presentino dovrà sempre essere tenuto sotto controllo.

1.3.2 Obiettivi di qualità

1.3.2.1 Rischi non preventivati - OPC4

Evidenzia il numero dei rischi presentati e non preventivati nell'arco del progetto; un valore molto alto potrebbe indicare una povera analisi dei rischi.

- **Metrica utilizzata:** numero dei rischi non preventivati (MPC4);
- **Soglia di accettabilità:** 0 - 3;
- **Soglia di ottimalità:** 0.

1.4 System/Software Requirements Analysis Process

Questo processo si occupa di trasformare le idee del proponente in un insieme di requisiti tecnici atti a guidare la progettazione del sistema.

1.4.1 Strategie

I requisiti identificati dal gruppo e successivamente inseriti nel documento “*Analisi dei Requisiti v2.0.0*” dovranno possedere le seguenti caratteristiche:

- dovranno essere inseriti nella piattaforma PragmaDB;
- si dovrà tenere traccia delle fonti da cui sono stati ricavati;
- si dovrà tenere traccia della loro implementazione;
- dovranno essere approvati dal proponente;
- nessun requisito dovrà risultare superfluo o ambiguo.

1.4.2 Obiettivi di qualità

1.4.2.1 Requisiti obbligatori soddisfatti - OPC5

Indica il numero dei requisiti obbligatori soddisfatti, espresso in percentuale.

- **Metrica utilizzata:** Numero dei requisiti obbligatori soddisfatti (MPC5);
- **Soglia di accettabilità:** 100%;
- **Soglia di ottimalità:** 100%.

1.4.2.2 Requisiti desiderabili soddisfatti - OPC6

Indica il numero dei requisiti desiderabili soddisfatti, espresso in percentuale.

- **Metrica utilizzata:** Numero dei requisiti desiderabili soddisfatti (MPC6);
- **Soglia di accettabilità:** 70%;
- **Soglia di ottimalità:** 100%.

1.4.2.3 requisiti facoltativi soddisfatti - OPC7

Indica il numero dei requisiti facoltativi soddisfatti, espresso in percentuale.

- **Metrica utilizzata:** Numero dei requisiti facoltativi soddisfatti (MPC7);
- **Soglia di accettabilità:** 0%;
- **Soglia di ottimalità:** 100%.

1.5 System/Software Architectural Design Process

Questo processo si occupa di identificare la corrispondenza tra requisiti di sistema ed elementi del sistema.

1.5.1 Strategie

L'architettura ottenuta svolgendo le attività di questo processo dovrà possedere le seguenti caratteristiche:

- il sistema dovrà presentare basso accoppiamento ed alta coesione;
- ogni componente dovrà essere progettato puntando su incapsulamento, modularizzazione e riuso di codice;
- per ogni elemento del sistema dovrà essere possibile tracciare il requisito associato.

1.5.2 Obiettivi di qualità

1.5.2.1 Structural Fan-In - OPC8

In riferimento ad un modulo del software, indica quanti altri moduli lo utilizzano durante la loro esecuzione; tale indicazione permette di stabilire il livello di riuso implementato.

- **Metrica utilizzata:** SF-IN (MPC8);
- **Soglia di accettabilità:** ≥ 0 ;
- **Soglia di ottimalità:** ≥ 2 ;

1.5.2.2 Structural Fan-Out - OPC9

In riferimento ad un modulo del software, indica quanti moduli vengono utilizzati durante la sua esecuzione; tale indicazione permette di stabilire il livello di accoppiamento implementato.

- **Metrica utilizzata:** SF-OUT (MPC9);
- **Soglia di accettabilità:** 0 - 6;
- **Soglia di ottimalità:** 0 - 2.

1.6 Software Detailed Design Process

Questo processo si occupa di fornire, dato un sistema, una sua progettazione di dettaglio che permetta codifica ed esecuzione di test.

1.6.1 Strategie

Le attività di questo processo dovranno possedere le seguenti caratteristiche:

- il livello di dettaglio della progettazione dovrà guidare la codifica e l'esecuzione dei test senza bisogno di informazioni aggiuntive;
- la progettazione di dettaglio, che include architettura di basso livello, relazioni fra le varie unità software concepite e la definizione dettagliata delle interfacce dovrà essere esposta chiaramente nel documento “*Definizione di Prodotto v1.0.0*”;
- per ogni elemento dell'architettura a basso livello dovrà essere possibile tracciare il requisito associato.

1.6.2 Obiettivi di qualità

1.6.2.1 Numero di metodi per classe - OPC10

Indica il numero di metodi definiti in una classe; un valore molto alto potrebbe indicare una cattiva decomposizione delle funzionalità a livello di progettazione.

- **Metrica utilizzata:** numero di metodi per classe (MPC10);
- **Soglia di accettabilità:** 1 - 10;
- **Soglia di ottimalità:** 1 - 8.

1.6.2.2 Numero di parametri per metodo - OPC11

Indica il numero di parametri passati ad un metodo; un valore molto alto potrebbe indicare un metodo troppo complesso e ulteriormente scomponibile in metodi più semplici.

- **Metrica utilizzata:** numero di parametri per metodo (MPC11);
- **Soglia di accettabilità:** 0 - 6;
- **Soglia di ottimalità:** 0 - 4.

1.7 Software Construction Process

Questo processo si occupa di produrre unità di software eseguibili che riflettono la progettazione effettuata.

1.7.1 Strategie

Le unità software prodotte dovranno rispettare le seguenti caratteristiche:

- il codice dovrà risultare facilmente manutenibile;
- il codice prodotto dovrà essere facilmente comprensibile e testabile;
- per ogni unità di software dovrà essere possibile tracciare il requisito e l'elemento architeturale associato.

1.7.2 Obiettivi di qualità

1.7.2.1 Complessità ciclomatica - OPC12

Indica la complessità di funzioni, moduli, metodi o classi di un programma. Alti valori di complessità ciclomatica implicano una ridotta manutenibilità del codice.

- **Metrica utilizzata:** indice di complessità ciclomatica (MPC12);
- **Soglia di accettabilità:** 1 - 20;
- **Soglia di ottimalità:** 1 - 10.

1.7.2.2 Livelli di annidamento - OPC13

Indica il numero di procedure e funzioni annidate, ovvero richiamate all'interno di altre procedure o funzioni. Più livelli di annidamento potrebbero rendere il codice di difficile comprensione e potrebbero portare a commettere errori logici durante la realizzazione del codice. In caso di troppi livelli di annidamento sarebbe opportuno riscrivere il metodo, o la funzione, affinché sia facilmente comprensibile.

- **Metrica utilizzata:** numero di livelli di annidamento (MPC13);
- **Soglia di accettabilità:** 1 - ;
- **Soglia di ottimalità:** 1 - 2.

1.7.2.3 Linee di commento per linee di codice - OPC14

Indica il numero di linee di commento rispetto alle linee totali del codice. Un valore basso indica un codice poco comprensibile e, di conseguenza, difficilmente manutenibile. Un valore troppo alto indica un eccesso di commenti e un appesantimento dei file.

- **Metrica utilizzata:** percentuale linee di commento per linee di codice (MPC14);
- **Soglia di accettabilità:** 10% - 40%;
- **Soglia di ottimalità:** 20% - 30%.

1.7.2.4 Manutenibilità - OPC15

Permette di stabilire il grado di manutenibilità del codice prodotto.

- **Metrica utilizzata:** indice di manutenibilità (MPC15);
- **Soglia di accettabilità:** 10 - 100;
- **Soglia di ottimalità:** 20 - 100.

1.8 System/Software Integration Process

Questo processo si occupa di integrare le unità software tra loro, produrre software coerente con la progettazione e dimostrare che il prodotto soddisfi i requisiti identificati.

1.8.1 Strategie

Le attività previste da questo processo dovranno puntare a raggiungere un alto livello di automazione, in particolare:

- l'integrazione delle varie parti del sistema sarà completamente automatizzata utilizzando lo strumento di continuous integration Jenkins, come definito nel documento “*Norme di Progetto v2.0.0*”;
- il livello di integrazione raggiunto del sistema sarà sempre consultabile grazie all'utilizzo dello strumento di continuous integration Jenkins, come definito nel documento “*Norme di Progetto v2.0.0*”.

1.8.2 Obiettivi di qualità

1.8.2.1 Componenti integrate - OPC16

Indica il numero di componenti definite in progettazione che sono attualmente implementate e integrate nel sistema. Nel nostro caso, tutte le componenti progettate andranno a costituire il sistema.

- **Metrica utilizzata:** percentuale di componenti integrate nel sistema (MPC16);
- **Soglia di accettabilità:** 100%;
- **Soglia di ottimalità:** 100%.

1.9 System/Software Qualification Testing Process

Questo processo si occupa di assicurare che ogni requisito individuato sia stato implementato nel prodotto.

1.9.1 Strategie

Questo processo deve possedere le seguenti caratteristiche:

- le attività di test previste dal processo verranno svolte su un sistema le cui componenti sono verificate e correttamente integrate fra loro;
- le attività di test dovranno raggiungere il maggior livello di automazione nell'esecuzione tramite lo strumento di continuous integration Jenkins;
- le attività di test dovranno essere eseguite in numero sufficiente in modo tale da garantire un'ottima copertura dei requisiti;
- il software dovrà implementare tutti i requisiti obbligatori.

1.9.2 Obiettivi di qualità

1.9.2.1 Test di unità eseguiti - OPC17

Indica il numero di test di unità eseguiti tra quelli definiti dal gruppo.

- **Metrica utilizzata:** percentuale di test di unità eseguiti (MPC17);
- **Soglia di accettabilità:** 90% - 100%;
- **Soglia di ottimalità:** 100%.

1.9.2.2 Test di integrazione eseguiti - OPC18

Indica il numero di test di integrazione eseguiti tra quelli definiti dal gruppo.

- **Metrica utilizzata:** percentuale di test di integrazione eseguiti (MPC18);
- **Soglia di accettabilità:** 70% - 100%;
- **Soglia di ottimalità:** 80% - 100%.

1.9.2.3 Test di sistema eseguiti - OPC19

Indica il numero di test di sistema eseguiti in modo automatico tra quelli definiti dal gruppo.

- **Metrica utilizzata:** percentuale di test di sistema eseguiti (MPC19);
- **Soglia di accettabilità:** 70% - 100%;
- **Soglia di ottimalità:** 80% - 100%.

1.9.2.4 Test di validazione eseguiti - OPC20

Indica il numero di test di validazione eseguiti manualmente tra quelli definiti dal gruppo.

- **Metrica utilizzata:** percentuale di test di validazione eseguiti (MPC20);
- **Soglia di accettabilità:** 100%;
- **Soglia di ottimalità:** 100%.

1.9.2.5 Test superati - OPC21

Indica il numero di test superati.

- **Metrica utilizzata:** percentuale dei test superati (MPC21);
- **Soglia di accettabilità:** 90% - 100%;
- **Soglia di ottimalità:** 100%.

1.10 Software Verification Process

Questo processo si occupa di controllare che il software prodotto soddisfi correttamente i requisiti identificati.

1.10.1 Strategie

Questo processo deve possedere le seguenti caratteristiche:

- la documentazione verrà verificata mediante inspection;
- i test dinamici sui vari elementi saranno al più possibile automatizzabili;
- i test dinamici sui vari elementi del software copriranno gran parte delle possibili casistiche di utilizzo;
- l'esito di ogni test deve essere tracciabile.

1.10.2 Obiettivi di qualità**1.10.2.1 Branch coverage - OPC22**

Indica il numero di rami decisionali percorsi nei test utilizzati. Questo obiettivo assicura che i branch derivanti da una condizione siano eseguiti da almeno un test.

- **Metrica utilizzata:** percentuale di rami decisionali percorsi (MPC22);
- **Soglia di accettabilità:** 75% - 100%;
- **Soglia di ottimalità:** 85% - 100%.

1.10.2.2 Function coverage - OPC23

Indica il numero di funzioni che sono state chiamate nei test utilizzati.

- **Metrica utilizzata:** numero di funzioni chiamate nei test (MPC23);
- **Soglia di accettabilità:** 70% - 100%;
- **Soglia di ottimalità:** 80% - 100%.

1.10.2.3 Statement coverage - OPC24

Indica il numero di istruzioni che sono state eseguite nei test utilizzati. Maggiore è il valore maggiore sarà il numero di statement eseguiti almeno una volta dai test.

- **Metrica utilizzata:** numero di istruzioni nei test (MPC24);
- **Soglia di accettabilità:** 75% - 100%;
- **Soglia di ottimalità:** 85% - 100%.