



Piano di Qualifica

Versione: 0.4.0

23/11/2024

Redattori

Malik Giafar Mohamed

Verifica

Ion Cainareanu, Maria Fuensanta Trigueros Hernandez

Approvazione

Ion Cainareanu, Maria Fuensanta Trigueros Hernandez

Uso

Esterno

nextsoftpadova@gmail.com

Registro dei cambiamenti

| Versione | Data | Autore | Descrizione | Verifica | Approvazione |
|----------|------------|----------------------|--|---|-------------------------------------|
| 0.4.0 | 15/01/2025 | Stefano Baso | Aggiunta test documenti | | |
| 0.3.1 | 14/01/2025 | Stefano Baso | Continuo aggiunta schema sezioni e tabelle | Marco Perazzolo | |
| 0.2.0 | 13/12/2024 | Stefano Baso | Aggiunta schema sezioni | Ion Cainareanu | Ion Cainareanu |
| 0.1.0 | 23/11/2024 | Malik Giafar Mohamed | Creazione Documento | Ion Cainareanu, Maria Fuensanta Trigueros Hernandez | Maria Fuensanta Trigueros Hernandez |

Indice

| | | |
|-------|---|----|
| 1 | Elenco dei grafici | 4 |
| 2 | Introduzione e scopo | 4 |
| 2.1 | Glossario | 4 |
| 2.2 | Maturità e miglioramenti | 4 |
| 2.3 | Riferimenti | 4 |
| 2.3.1 | Riferimenti normativi | 4 |
| 2.3.2 | Riferimenti informativi | 5 |
| 3 | Qualità di processo | 5 |
| 3.1 | Scopo ed obiettivi | 5 |
| 3.2 | Processi primari | 5 |
| 3.2.1 | Fornitura | 5 |
| 3.2.2 | Sviluppo | 6 |
| 4 | Qualità del prodotto | 7 |
| 4.1 | Efficienza | 7 |
| 4.2 | Usabilità | 7 |
| 4.2.1 | Facilità di utilizzo | 7 |
| 4.3 | Manutenibilità | 7 |
| 4.3.1 | Metriche | 8 |
| 4.4 | Affidabilità | 8 |
| 4.4.1 | Metriche | 9 |
| 4.5 | Funzionalità | 9 |
| 4.5.1 | Metriche | 9 |
| 4.6 | Documentazione | 10 |
| 4.6.1 | Indice di leggibilità di Gulpease | 10 |
| 5 | Test e specifiche | 10 |
| 5.1 | Tipologie di test | 10 |
| 5.1.1 | Test di Unità | 10 |

| | |
|--|----|
| 5.1.2 Test di Integrazione | 10 |
| 5.1.3 Test di Sistema | 11 |
| 5.1.4 Test di Accettazione | 11 |
| 5.1.5 Test di Regressione | 11 |
| 6 Resoconto delle attività di verifica | 11 |
| 6.1 Fornitura | 11 |
| 6.2 MPDD01 | 11 |
| 7 Valutazioni per il miglioramento | 11 |

Elenco delle immagini

Elenco delle tabelle

| | |
|---|----|
| Table 1: Metriche di fornitura | 6 |
| Table 2: Metriche di sviluppo | 6 |
| Table 3: Metriche di progettazione di dettaglio | 6 |
| Table 4: Metriche di codifica | 7 |
| Table 5: Metriche di tempo medio | 7 |
| Table 6: Obiettivo di funzionalità | 7 |
| Table 7: Obiettivo di usabilità | 7 |
| Table 8: Obiettivo di manutenibilità | 7 |
| Table 9: Metriche di manutenibilità | 8 |
| Table 10: Obiettivo di affidabilità | 9 |
| Table 11: Metriche di affidabilità | 9 |
| Table 12: Obiettivo di funzionalità | 9 |
| Table 13: Obiettivo di usabilità | 10 |
| Table 14: Obiettivo di qualità della documentazione | 10 |
| Table 15: Obiettivo di leggibilità | 10 |
| Table 16: Valutazione documenti fase 1 | 11 |

1 Elenco dei grafici

2 Introduzione e scopo

Il Piano di Qualifica è un documento soggetto a modifiche incrementalì, finalizzate principalmente alla definizione delle metriche di valutazione del prodotto. Tali metriche saranno stabilite in conformità ai requisiti e alle aspettative del proponente, con l'obiettivo di determinare correttamente la qualità del prodotto attraverso un processo di miglioramento continuo. Questo approccio tende ad evolversi nel tempo, in particolare una volta stabilita una linea guida.

Il presente documento si propone di:

- Definire le metriche e le metodologie di controllo e misurazione.
- Stabilire quantità, qualità dei test e relative metriche.
- Descrivere l'applicazione dei test e documentarne i risultati, valutando la conformità rispetto alle attese e alle metriche definite.

2.1 Glossario

Con lo scopo di rendere chiara la terminologia usata all'interno del documento, viene allegato un Glossario nel file dedicato che servirà a chiarire le terminologie usate dandone una definizione. Ogni termine la cui definizione sarà presente nel Glossario, verrà identificato con una lettera G come pedice, nel formato parola_G.

2.2 Maturità e miglioramenti

Il documento in oggetto è stilato seguendo un approccio Agile, un approccio flessibile allo sviluppo software, che privilegia la collaborazione, il feedback continuo e il rilascio rapido di valore adattandosi ai cambiamenti. Con lo scopo quindi di essere aggiornato nel corso degli sviluppi e in relazione tra membri del gruppo e proponente, ad oggi non può essere considerato una versione completa.

2.3 Riferimenti

2.3.1 Riferimenti normativi

- Norme di Progetto

2.3.2 Riferimenti informativi

Materiale didattico del corso

- Qualità di prodotto: <https://www.math.unipd.it/~tullio/IS-1/2021/Dispense/T12.pdf>;
- Qualità di processo: <https://www.math.unipd.it/~tullio/IS-1/2021/Dispense/T13.pdf>;
- Indice di Gulpease: <https://www.ilc.cnr.it/dylanlab/apps/texttools/>

3 Qualità di processo

3.1 Scopo ed obiettivi

La qualità di un sistema è determinata dai processi che lo costituiscono e viene misurata attraverso l'uso di metriche specifiche, atte a valutare tali processi e verificarne il raggiungimento degli obiettivi di qualità stabiliti. Il modello di riferimento è il Ciclo PDCA (Plan - Do - Check - Act), il quale consente di avere un miglioramento continuo tramite una gestione strutturata delle attività. Questo approccio si basa su una pianificazione accurata, il monitoraggio mediante metriche definite e l'integrazione dei risultati ottenuti nella fase di produzione operativa.

Di seguito, vengono presentati i processi identificati e i corrispondenti livelli di qualità prefissati. Per ciascuna metrica è fornita una descrizione che ne illustra le modalità di applicazione e definisce i valori considerati accettabili nel contesto delle verifiche di qualità.

3.2 Processi primari

3.2.1 Fornitura

In questa fase del processo vengono analizzate tutte le scelte effettuate durante lo sviluppo, verificandone la conformità con gli obiettivi stabiliti nelle diverse fasi del progetto. Vengono definite le misure da implementare, assicurando il rispetto dei termini e delle condizioni prestabiliti. L'obiettivo principale è garantire che la fornitura sia allineata alle aspettative, sia in termini di risorse impiegate che di risultati ottenuti.

Un concetto chiave in questo contesto è l'MPC (Minimum Predictive Capability), una metrica fondamentale per valutare l'affidabilità di un modello di apprendimento automatico nel generare risultati accurati. L'MPC rappresenta il livello minimo di precisione che un modello deve raggiungere per essere considerato accettabile, contribuendo a garantire che le previsioni siano coerenti con gli standard richiesti.

Di seguito sono descritte le principali metriche e calcoli associati che verranno riportati nella tabella sottostante mettendo in relazione il valore plausibile e il valore ottimale:

- BAC (Budget At Completion): Costo totale preventivato per il completamento del progetto.
- EAC (Estimated At Completion): Valore stimato per i compiti rimanenti.
- CPI (Cost Performance Index): Indice di prestazione dei costi, misura l'efficienza con cui il budget viene utilizzato.
- ETC (Estimated To Completion): Stima del costo finale aggiornato alla data di misurazione.
- EV (Earned Value): Valore ottenuto fino al momento attuale, calcolato come percentuale del lavoro svolto \times EAC.
- PV (Planned Value): Valore pianificato fino al momento attuale, calcolato come percentuale del lavoro pianificato \times BAC.
- AC (Actual Cost): Budget effettivamente speso fino al momento attuale.
- CV (Cost Variance): Differenza tra il valore ottenuto (EV) e il costo effettivo (AC), calcolato come $EV - AC$.

- **SV (Schedule Variance)**: Differenza tra il valore ottenuto (EV) e quello pianificato (PV), calcolato come $EV - PV$. Un valore negativo indica un ritardo rispetto alla pianificazione.
- **BV (Budget Variance)**: Differenza rispetto al budget preventivato, calcolato come $AC - CV$.

| Codice | Descrizione | Soglia accettabile | Ottimo |
|---------|-------------------------|-------------------------------------|------------------------------|
| MPC-EAC | Estimated at Completion | $\pm 5\%$ rispetto al preventivo | Corrispondente al preventivo |
| MPC-ETC | Estimated to Completion | ≥ 0 | $\leq EAC$ |
| MPC-EV | Earned Value | ≥ 0 | $\leq EAC$ |
| MPC-PV | Planned Value | ≥ 0 | $\leq BAC$ |
| MPC-AC | Actual Cost | ≥ 0 | $\leq EAC$ |
| MPC-CV | Cost Variance | $\geq -5\%$ | $\geq 0\%$ |
| MPC-SV | Schedule Variance | $\geq -10\%$ | $\geq 0\%$ |
| MPC-BV | Budget Variance | $\pm 10\%$ | $\leq 0\%$ |

Table 1: Metriche di fornitura

Questi indicatori consentono di monitorare l'andamento del progetto in termini di costi, tempi e precisione delle previsioni, supportando una gestione efficiente e mirata.

3.2.2 Sviluppo

3.2.2.1 Progettazione architeturale

- **SFIN (Structural Fan-In)** Indice di utilità = indice di quantità di componenti che utilizzano un determinato modulo, un indice alto indica un maggior utilizzo del componente.
- **SFOUT (Structural Fan-Out)** Indice di dipendenza = indica il numero di componenti utilizzati dal modulo in oggetto, un indice alto indica un alto accoppiamento.

| Codice | Descrizione | Soglia accettabile | Ottimo |
|-----------|--------------------|--------------------|--------|
| MPC-SFIN | Structural Fan-in | / | / |
| MPC-SFOUT | Structural Fan-out | / | / |

Table 2: Metriche di sviluppo

3.2.2.2 Progettazione di dettaglio

Indice per la media del numero di metodi presenti in ogni package, un indice alto potrebbe comportare il refactoring.

| Codice | Descrizione | Soglia accettabile | Ottimo |
|--------|-------------------|--------------------|--------|
| MPC-NM | Number of Methods | 3-11 | 3-8 |

Table 3: Metriche di progettazione di dettaglio

3.2.2.3 Codifica

- **BLC (Bugs for Line of Code)** = indice per il numero di righe di codice che possono contenere bug o errori.
- **VNUD (Variabili Non Utilizzate o non Definite)** = indice per il numero di variabili utilizzate o non definite, queste sono a tutti gli effetti errori di programmazione che possono comportare bug. Variabili non utilizzate occupano spazio inutilmente in memoria e creano confusione all'interno del codice.

| Codice | Descrizione | Soglia accettabile | Ottimo |
|---------|---|--------------------|--------|
| MPC-BLC | Bugs for Line of Code | 0-70 | 0-25 |
| MPC-VND | Variabili non utilizzate e non definite | 0 | 0 |

Table 4: Metriche di codifica

4 Qualità del prodotto

Per mantenere ed assicurare la qualità del prodotto software il gruppo ha adottato il modello di qualità stabilito dallo standard ISO/IEC 9126, adattandolo alle esigenze e requisiti del progetto. Tale standard propone una serie di metriche e regole per migliorare l'organizzazione dei processi e di conseguenza la qualità del prodotto. Di seguito verranno elencate e descritte le metriche che verranno utilizzate.

4.1 Efficienza

L'efficienza indica il tempo di elaborazione della richiesta da parte del software per raggiungere il risultato.

| Codice | Descrizione | Soglia accettabile | Ottimo |
|--------|-------------------------|--------------------|-----------|
| MPDS01 | Tempo di risposta medio | 3 secondi | 2 secondi |

Table 5: Metriche di tempo medio

4.2 Usabilità

L'usabilità riguarda l'esperienza dell'utente nell'interagire con il nostro prodotto, capirne il suo funzionamento e apprezzarne le sue funzioni.

| Codice | Nome | Descrizione | Metriche associate |
|--------|------------------------|---|--------------------|
| OPDS01 | Usabilità del prodotto | Il prodotto deve essere facilmente usabile dall'utente in modo da raggiungere il più velocemente possibile quello che cerca | MPDS01 |

Table 6: Obiettivo di funzionalità

4.2.1 Facilità di utilizzo

Questa rappresenta la velocità con cui l'utente trova quello che sta cercando, calcolata in base al numero di click minimo che si deve effettuare per arrivare all'obiettivo.

| Codice | Descrizione | Soglia accettabile | Ottimo |
|--------|----------------------|--------------------|-------------|
| MPDS01 | Facilità di utilizzo | $FU \leq 3$ | $FU \leq 5$ |

Table 7: Obiettivo di usabilità

4.3 Manutenibilità

La manutenibilità del software è la facilità con cui può essere modificato, corretto, adattato o aggiornato.

| Codice | Nome | Descrizione | Metriche associate |
|--------|-----------------------------|---|----------------------------|
| OPDS02 | Analizzabilità del prodotto | Una facile analisi del codice permette di localizzare in tempi minimi il blocco di codice che riguarda l'errore o l'aggiornamento | MPDS03 MPDS04 MPDS06 |
| OPDS03 | Modificabilità del prodotto | Permette una manutenzione più agevolata per la correzione | MPDS05 MPDS02 |

Table 8: Obiettivo di manutenibilità

4.3.1 Metriche

- **Complessità ciclomatica:** misura la complessità strutturale di un programma basandosi sul grafo di controllo del flusso del codice. In particolare, rappresenta il numero di cammini linearmente indipendenti attraverso il codice sorgente. Viene calcolata con la seguente formula:

$$V(G) = E - N + 2P$$

in cui:

- E è il numero di archi (transizioni tra i nodi),
 - N è il numero di nodi (blocchi di codice o decision points),
 - P è il numero di componenti connesse (tipicamente $P = 1$ per un singolo metodo o funzione)
- **Profondità della gerarchia:** indica il numero massimo di livelli di ereditarietà in una gerarchia di classi. Una gerarchia più profonda può favorire il riuso del codice ma aumenta la complessità e il rischio di propagazione degli errori. Per un design più manutenibile, è preferibile mantenere la profondità entro limiti ragionevoli (3-4 livelli), favorendo la composizione rispetto a una struttura gerarchica troppo profonda.
 - **Parametri per metodo:** indica il numero di parametri per metodo. Un indice basso rappresenta un numero basso di parametri richiesti dal metodo, di conseguenza risulta di più facile comprensione e utilizzo.
 - **Code Smell:** indice che rappresenta indicatori qualitativi di potenziali problemi nel codice. E' utile per valutare la leggibilità, la modificabilità, e la testabilità del codice. Si dividono in:
 - Duplicated Code: frammenti di codice identici o simili in più punti, che aumentano i costi di manutenzione perché le modifiche devono essere replicate ovunque.
 - Long Methods: metodi eccessivamente lunghi, che riducono la leggibilità e la comprensione del codice.
 - God Class: una classe con troppe responsabilità (violazione del principio di Single Responsibility), difficile da testare e modificare.
 - High Coupling: una forte dipendenza tra componenti, che rende il sistema rigido e suscettibile a errori quando una parte viene modificata.
 - Low Cohesion: componenti con funzionalità eterogenee che non si relazionano strettamente, rendendo il codice più complesso da comprendere.
 - **Facilità di comprensione:** rappresenta il rapporto tra commenti presenti e codice totale per capirne il suo funzionamento.

| Codice | Descrizione | Soglia accettabile | Ottimo |
|--------|--------------------------|--------------------|-----------|
| MPDS02 | Profondità di gerarchia | PG ≤ 3 | PG ≤ 2 |
| MPDS03 | Parametri per metodo | PPM ≤ 8 | PPM ≤ 4 |
| MPDS04 | Complessità ciclomatica | CC ≤ 20 | CC ≤ 10 |
| MPDS05 | Code smell | CS ≤ 50 | CS ≤ 10 |
| MPDS06 | Facilità di comprensione | FC ≥ 0.10 | FC ≥ 0.20 |

Table 9: Metriche di manutenibilità

4.4 Affidabilità

L'affidabilità riguarda il livello minimo di prestazioni da mantenere durante l'uso in determinate situazioni.

| Codice | Nome | Descrizione | Metriche associate |
|--------|------------------------|---|----------------------------|
| OPDS04 | Prodotto maturo | Evita errori o malfunzionamenti durante l'utilizzo | MPDS07 MPDS10 |
| OPDS05 | Tolleranza agli errori | Mantiene il livello di prestazioni anche durante un uso scorretto o in presenza di errori | MPDS11 MPDS08 MPDS09 |

Table 10: Obiettivo di affidabilità

4.4.1 Metriche

- **Code Coverage:** percentuale di codice eseguito nei test. Un indice di copertura del codice alto significa che è stato testato più codice, riducendo quindi la presenza di bug.
- **Branch Coverage:** percentuale di copertura di tutti i branch all'esecuzione del codice. Il compito dei test è anche quello di verificare tutti i rami esistenti per verificarne la correttezza.
- **Presenza di vulnerabilità:** indice per il numero di vulnerabilità ancora presenti nel codice.
- **Presenza di bug:** indice per il numero di bug ancora presenti nel codice.
- **Successo dei test:** indice in percentuale relativo al successo dei test definiti dai programmatori.

| Codice | Descrizione | Soglia accettabile | Ottimo |
|--------|---------------------------|--------------------|---------------|
| MPDS07 | Code Coverage | CC \geq 75% | 100% |
| MPDS08 | Presenza di vulnerabilità | VLN \leq 2 | 0 |
| MPDS09 | Presenza di bug | BUG \leq 20% | BUG \leq 5% |
| MPDS10 | Branch Coverage | BC \geq 75% | 100% |
| MPDS11 | Successo dei test | \geq 75% | 100% |

Table 11: Metriche di affidabilità

4.5 Funzionalità

La funzionalità è la capacità di fornire funzioni / azioni per ogni esigenza stabilita.

| Codice | Nome | Descrizione | Metriche associate |
|--------|-----------------------------|---|--------------------|
| OPDS06 | Appropriatezza del prodotto | Fornire le funzioni richieste ed essere in linea con i requisiti fissati nell'Analisi dei Requisiti | MPDS12 MPDS13 |

Table 12: Obiettivo di funzionalità

4.5.1 Metriche

- **Requirement coverage:** indice della copertura dei requisiti descritti nell'Analisi dei Requisiti. Viene calcolato con il rapporto percentuale tra numero di requisiti rispettati e numero di requisiti totali, con la formula:

$$RC = \frac{R_{RISP}}{R_{TOT}} 100$$

- **Requisiti obbligatori soddisfatti:** indice della copertura dei requisiti obbligatori descritti nell'Analisi dei Requisiti. Viene calcolato con il rapporto percentuale tra numero di requisiti rispettati e numero di requisiti totali, con la formula:

$$RC = \frac{R_{ROS}}{R_{ROT}} 100$$

| Codice | Descrizione | Soglia accettabile | Ottimo |
|--------|-----------------------------------|--------------------|--------|
| MPDS12 | Requirement coverage | RC <= 75% | 100% |
| MPDS13 | Requisiti obbligatori soddisfatti | 100% | 100% |

Table 13: Obiettivo di usabilità

4.6 Documentazione

La documentazione ha ruolo di supporto, in particolare definisce le norme da seguire durante lo sviluppo, la divisione delle risorse e responsabilità. E' necessario quindi definire una linea guida anche per la redazione dei documenti per evitare ambiguità e renderli chiari

| Codice | Nome | Descrizione | Metriche associate |
|--------|---------------------------|--|--------------------|
| OPDD01 | Leggibilità dei documenti | Per mantenere una buona comprensione, il documento deve essere leggibile | MPDD01 |

Table 14: Obiettivo di qualità della documentazione

4.6.1 Indice di leggibilità di Gulpease

L'indice di leggibilità di Gulpease è una metrica che valuta la semplicità di un testo in italiano, ideata per stimare quanto sia comprensibile da lettori con livelli diversi di istruzione. L'indice si basa su tre parametri: il numero di lettere, il numero di parole e il numero di frasi. La formula è:

$$\text{Gulpease} = 89 - \frac{\text{N}^\circ \text{ lettere}}{\text{N}^\circ \text{ parole}} 10 + \frac{\text{N}^\circ \text{ frasi}}{\text{N}^\circ \text{ parole}} 30$$

Il punteggio varia da 0 a 100, dove valori alti indicano maggiore leggibilità. Tipicamente, un testo comprensibile per chi ha una licenza elementare ha un punteggio sopra 80, mentre per chi possiede una licenza media è sufficiente un punteggio superiore a 60. Il metodo è particolarmente utile per valutare documenti destinati a un pubblico ampio, come testi scolastici o burocratici.

| Codice | Descrizione | Soglia accettabile | Ottimo |
|--------|-----------------------------------|--------------------|-----------|
| MPDD01 | Indice di leggibilità di Gulpease | GULP ≥ 40 | GULP ≥ 60 |

Table 15: Obiettivo di leggibilità

5 Test e specifiche

Il seguente capitolo presenta in maniera dettagliata le strategie e scelte di testing, atte a garantire la correttezza del prodotto e facilitarne la validazione. Viene quindi scelto l'uso del Modello a V, il quale definisce test in parallelo allo sviluppo.

5.1 Tipologie di test

5.1.1 Test di Unità

I test di unità valutano il corretto funzionamento delle singole unità di codice all'interno del software. Un'unità di codice è una funzione, una classe o qualsiasi componente che svolge un'attività specifica in modo indipendente rispetto al resto del sistema. Attualmente, nella prima versione del Piano di Qualifica, né le unità né i test corrispondenti sono stati definiti. La definizione delle unità avverrà con l'avvio del processo di progettazione e sviluppo software.

5.1.2 Test di Integrazione

I test di integrazione valutano il corretto funzionamento delle diverse componenti del software e il modo in cui vengono integrate tra loro, evidenziandone eventuali problemi. In questa stesura iniziale del Piano di Qualifica non sono state ancora individuate le componenti del prodotto, di conseguenza i test di integrazione non sono ancora stati definiti.

5.1.3 Test di Sistema

I test di sistema verificano il sistema completo del prodotto software, prendendo in considerazione tutti i componenti e interfacce con altri sistemi. Questi test controllano che il software rispetti tutti i requisiti prestabiliti e che sia adatto all'uso in produzione.

5.1.4 Test di Accettazione

I test di accettazione assicurano che il software soddisfi i requisiti e parametri stabiliti dal capitolato. Sono svolti in presenza del committente e verificano che il prodotto possa essere consegnato al committente o messo in produzione.

5.1.5 Test di Regressione

I test di regressione servono a testare che le aggiornamenti / modifiche rilasciati nel software non incidano negativamente sulle funzioni già presenti. Ciò consiste nella ripetizione di test di unità, integrazione e sistema.

6 Resoconto delle attività di verifica

6.1 Fornitura

6.2 MPDD01

Di seguito la tabella con i risultati ottenuti dai documenti secondo l'indice di Gulpease. Come metro di valutazione del documento viene esclusa la prima pagina che, trattandosi dell'intestazione, potrebbe portare ad un risultato inesatto.

| Documento | Risultato | Esito |
|-----------------------|-----------|----------|
| Analisi dei Requisiti | | |
| Piano di qualifica | | |
| Piano di Progetto | | |
| Norme di Progetto | | |
| 2024-11-15 | 68 | Superato |
| 2024-11-24 | 66 | Superato |
| 2024-12-09 | 75 | Superato |
| 2024-12-18 | 66 | Superato |
| 2025-01-03 | 73 | Superato |
| 2024-11-25 | 67 | Superato |
| 2024-12-24 | 64 | Superato |

Table 16: Valutazione documenti fase 1

7 Valutazioni per il miglioramento

Nel seguente capitolo vengono riportate delle osservazioni sulle criticità incontrate con lo scopo di adottare dei miglioramenti.