Piano di Qualifica

2025-03-20

Responsabile	Nicolò Bovo
Redattori	Nicolò Bovo
	Giovanni Battista Matteazzi
	Manuel Cinnirella
Verificatori	Manuel Cinnirella
	Elia Leonetti

AlphaCode
Università Degli Studi di Padova
Versione 0.2.0

Registro delle modifiche

Vers.	Data	Descrizione	Autore	Verificatore
0.2.0	2025-03-29	Aggiunta Qualità di Prodotto e Specifiche dei Test	Alessandro Di Pasquale, Massimo Chioru, Elia Leonetti	Manuel Cinnirella
0.1.0	2025-03-20	Introduzione del documento e Qualità di Processo	Giovanni Battista Matteazzi, Nicolò Bovo, Manuel Cinnirella	Elia Leonetti

Indice

I. Introduzione

I - 1. Scopo del documento

Il Piano di Qualifica è un documento essenziale per garantire il controllo della qualità del prodotto software e dei processi impiegati nel suo sviluppo. Definisce le strategie di verifica e validazione adottate dal gruppo, con l'obiettivo di assicurare che il progetto rispetti i requisiti, raggiunga gli obiettivi stabiliti e mantenga elevati standard qualitativi. Il documento si articola in tre componenti principali:

1. Pianificazione della qualità: stabilisce obiettivi, standard e strategie;
2. Controllo della qualità: applica metriche e verifiche per valutare la conformità;
3. Miglioramento continuo: analizza i risultati per ottimizzare metodi e processi.

Essendo un documento in continua evoluzione, verrà aggiornato nel tempo per riflettere miglioramenti o modifiche organizzative. Le sezioni di monitoraggio documenteranno l'andamento delle metriche, consentendo interventi tempestivi in caso di criticità.

I - 2. Scopo del prodotto

Lo scopo di NearYou è progettare, realizzare e validare una piattaforma di advertising personalizzato in tempo reale, capace di unire flussi GPS, profilazione utente e AI generativa per produrre annunci altamente contestuali.

- **Implementazione tecnica:** Il team AlphaCode svilupperà simulatori di posizionamento, pipeline Kafka⟶ ClickHouse⟶PostGIS e modelli LLM per garantire un flusso dati integrato e performante.
- **Misurazione e validazione:** Verranno condotte misurazioni mirate per valutare throughput e latenza delle componenti di streaming (Kafka) e orchestrazione (Airflow), l'efficacia e i tempi di risposta delle query geospaziali in PostGIS, e la velocità di generazione e consegna degli annunci via FastAPI. Analizzeremo inoltre l'impatto delle operazioni AI su CPU e memoria.
- **Test di resilienza e performance:** Il team testerà la resilienza dei microservizi (producer, consumer, webapp) in caso di picchi di traffico o failure di rete. Per confermare scalabilità, robustezza delle logiche di prossimità e manutenibilità del codice, AlphaCode eseguirà stress test ad alto carico su tutte le pipeline, verificando la correttezza funzionale e le performance sotto condizioni estreme.

I - 3. Riferimenti

I - 3.1. Riferimenti informativi

- Norme di Progetto v1.0.0
- Capitolato d'appalto C4: NearYou: sistema di advertising
- Standard ISO/IEC 12207:1995

I - 3.2. Riferimenti normativi

- I processi di ciclo di vita del software
- Glossario

II. Qualità di processo

La qualità di processo è nota essere un fattore di fondamentale importanza per qualsiasi produzione di software che punti all'eccellenza qualitativa. Essa, infatti, influenza con un evidente rapporto di causa effetto la qualità del prodotto finale. Di seguito elenchiamo gli obiettivi di qualità che il gruppo si prefigge di raggiungere nell'ambito della qualità di processo, suddivisi per le tre categorie di processi individuate dallo standard ISO/IEC 12207:1995 (primari, di supporto e organizzativi).

II - 1. Fornitura

Insieme delle azioni e responsabilità del fornitore, con l'obiettivo di definire formalmente gli accordi con il committente in merito a vincoli e requisiti del progetto.

1. **Planned value (MPC-PV)**: Valore stimato del lavoro da eseguire entro una data prefissata, secondo la pianificazione.
2. **Earned value (MPC-EV)**: Quantifica il valore del lavoro realmente completato fino a un certo momento, in relazione al budget pianificato.
3. **Actual cost (MPC-AC)**: Spesa reale sostenuta per le attività concluse fino alla data considerata.
4. **Cost performance index (MPC-CPI)**: Indice che esprime l'efficienza della spesa, indicando quanto valore si è ottenuto per ogni unità di costo impiegata.
5. **Estimated at completion (MPC-EAC)**: Proiezione del costo finale totale del progetto basata sull'andamento attuale.
6. **Schedule variance (MPC-SV)**: Differenza tra il valore ottenuto (EV) e quello previsto (PV); un risultato negativo segnala un ritardo.
7. **Budget variance (MPC-BV)**: Scostamento tra il valore previsto (PV) e la spesa effettiva (AC); valori negativi indicano costi superiori al preventivo.
8. **Estimate to completion (MPC-ETC)**: Previsione della spesa necessaria per terminare le attività residue del progetto.

Metrica	Nome	Valore Accettabile	Valore ottimo
MPC-EV	Earned Value	>= 0	<= EAC
MPC-PV	Planned Value	>= 0	<= BAC
MPC-AC	Actual Cost	>= 0	<= EAC
MPC-EAC	Estimate At Completion	>= 0	<= BAC
MPC-ETC	Estimate To Complete	>= 0	<= BAC

II - 2. Sviluppo

Insieme delle operazioni tecniche volte alla costruzione e manutenzione del sistema software, con l'obiettivo di rispettare i requisiti contrattuali.

1. **Indice di stabilità dei requisiti (MPC-RSI)**: Misura quanto i requisiti siano rimasti costanti nel tempo, segnalando modifiche o instabilità progettuali.

Metrica	Nome	Valore Accettabile	Valore ottimo
MPC-RSI	Indice di stabilità dei requisiti	>= 80%	100%

II - 3. Documentazione

Processo necessario per il tracciamento di tutte le attività relative al progetto.

1. **Indice Gulpease (MPC-GP)**: Misura la facilità di lettura di un testo italiano in base alla lunghezza media delle parole e delle frasi. Il punteggio va da 0 (poco leggibile) a 100 (molto leggibile).
2. **Correttezza Ortografica (MPC-CO)**: Valuta la presenza di errori ortografici e grammaticali nei testi prodotti.

Metrica	Nome	Valore Accettabile	Valore ottimo
MPC-GP	Indice di Gulpease	>= 40	>= 80
MPC-CO	Correttezza ortografica	0	0

II - 4. Verifica

1. **Code Coverage (MPC-CC)**: Percentuale di codice eseguita nei test; un valore più alto implica una copertura più ampia e quindi maggiore affidabilità.
2. **Test Success Rate (MPC-TSR)**: Proporzione di test superati rispetto al totale eseguito. Un alto tasso di successo indica che il software soddisfa i requisiti richiesti.

Metrica	Nome	Valore Accettabile	Valore ottimo
MPC-CC	Code coverage	>= 80%	100%
MPC-TSR	Test success rate	100%	100%

II - 5. Gestione della qualità

1. **Percentuale di metriche soddisfatte (MPC-PMS)**:

Metrica	Nome	Valore Accettabile	Valore ottimo

MPC-PMS	Percentuale Metriche Soddisfatte	>= 80%	100%

1. **Efficienza temporale (MPC-ET)**:

Metrica	Nome	Valore Accettabile	Valore ottimo
MPC-ET	Efficienza temporale	<=3	<=1

III. Qualità di prodotto

La qualità del prodotto rappresenta un aspetto fondamentale nello sviluppo software, poiché misura quanto il sistema realizzato sia in grado di soddisfare i requisiti stabiliti, le esigenze esplicite e implicite degli stakeholder e gli standard di riferimento. Essa è strettamente correlata alla qualità dei processi impiegati, poiché un prodotto di valore nasce da pratiche progettuali e tecniche adeguate e controllate lungo tutto il ciclo di vita. Un prodotto software si considera di elevata qualità quando dimostra le seguenti caratteristiche:

1. **Funzionalità**: risponde pienamente ai requisiti obbligatori, desiderabili e opzionali delineati nell'Analisi dei Requisiti, offrendo le funzionalità attese senza deviazioni.
2. **Affidabilità**: è stabile nel tempo e resistente a guasti, garantendo un comportamento prevedibile anche in condizioni anomale.
3. **Usabilità**: consente all'utente finale un'interazione chiara, semplice ed efficace, riducendo il tempo di apprendimento e minimizzando gli errori d'uso.
4. **Efficienza**: utilizza in modo ottimale le risorse disponibili, rispondendo rapidamente alle operazioni richieste, anche sotto carico.
5. **Manutenibilità**: è progettato con una struttura che facilita interventi futuri, come aggiornamenti, correzioni o estensioni, mantenendo la coerenza e la stabilità del sistema.

III - 1. Funzionalità

La funzionalità esprime quanto il software sia in grado di rispondere alle esigenze espresse nei requisiti, distinguendo tra quelli essenziali, preferibili e accessori.

1. **Requisiti obbligatori soddisfatti (MDP-01)**: Indica la percentuale dei requisiti fondamentali che sono stati effettivamente realizzati nel software. Deve essere sempre pari al 100% per garantire l'aderenza completa alle specifiche minime previste.
2. **Requisiti desiderabili soddisfatti (MDP-02)**: Misura quanto il software integra i requisiti considerati apprezzabili. Un valore elevato di questo indicatore contribuisce ad aumentare il livello di soddisfazione degli utenti.
3. **Requisiti opzionali soddisfatti (MDP-03)**: Esprime la percentuale di requisiti opzionali effettivamente implementati. Una maggiore presenza di tali funzionalità può arricchire il prodotto e aumentarne il valore complessivo.

Metrica	Nome	Valore Accettabile	Valore ottimo
MDP-01	Requisiti obbligatori soddisfatti	100%	100%
MDP-02	Requisiti desiderabili soddisfatti	0%	100%
MDP-03	Requisiti opzionali soddisfatti	0%	100%

III - 2. Affidabilità

L'affidabilità valuta la capacità del software di funzionare correttamente sotto condizioni specifiche.

1. **Code Coverage (MDP-04)**: Misura la percentuale di codice eseguita durante i test. Valori più alti indicano una migliore copertura del codice. Per questo progetto è richiesta una copertura pari o superiore all'80%.
2. **Branch coverage (MDP-05)**: Calcola la percentuale di rami decisionali del codice eseguiti durante test. Aiuta a identificare scenari non testati.
3. **Statement coverage (MDP-06)**: Rappresenta la percentuale di istruzioni eseguite durante i test. Un valore alto garantisce un'analisi più approfondita del codice.
4. **Passed test cases percentage (MDP-07)**: Misura la percentuale di casi di test superati rispetto al totale dei test eseguiti. Un alto valore indica che il software soddisfa i requisiti funzionali e non funzionali previsti.
5. **Failure density (MDP-08)** Indica il numero di fallimenti correttamente riscontrati per unità di dimensione del software, spesso misurata in linee di codice (LOC) o punti funzione. Valori più bassi denotano un software di qualità superiore, con meno difetti rispetto alla sua complessità o dimensione

Metrica	Nome	Valore Accettabile	Valore ottimo
MDP-04	Code Coverage	>= 80%	100%
MDP-05	Branch Coverage	>= 80%	100%
MDP-06	Statement Coverage	>= 60%	100%
MDP-07	Passed test cases percentage	>= 80%	100%
MDP-08	Failure Density	<= 5%	0%

III - 3. Usabilità

1. **Facilità di utilizzo (MDP-09)**: Misura il numero di errori commessi dagli utenti durante l'interazione. Un valore minimo indica un'interfaccia intuitiva.
2. **Tempo di apprendimento (MDP-10)**: Valuta il tempo necessario a un utente per imparare a utilizzare il software. Tempi più brevi migliorano l'esperienza utente.

Metrica	Nome	Valore Accettabile	Valore ottimo
MDP-09	Facilità di utilizzo	<= 2 errori	0 errori
MDP-10	Tempo di apprendimento	<= 5 minuti	<= 1 minuto

III - 4. Efficienza

1. **Utilizzo delle risorse (MDP-11)**: Valuta quanto efficacemente il sistema sfrutta le risorse hardware, come la CPU, la memoria e altri componenti. Un impiego ottimale di queste risorse assicura il corretto funzionamento del sistema, evitando un utilizzo eccessivo o inefficiente delle capacità disponibili.

Metrica	Nome	Valore Accettabile	Valore ottimo
MDP-11	Utilizzo delle risorse	>= 80%	100%

III - 5. Manutenibilità

1. **Code smell (MDP-12)**: Rileva potenziali problemi di progettazione o codice che potrebbero richiedere manutenzione. Segnala parti del codice che potrebbero non essere ottimali e che potrebbero causare difficoltà nel futuro, come un'architettura poco chiara o sezioni di codice ripetitive.
2. **Coefficient of coupling (MDP-13)**: Il Coefficient of Coupling misura il grado di dipendenza tra i moduli o le componenti di un sistema. Un alto COC implica che i moduli siano strettamente interconnessi, il che può rendere difficile apportare modifiche a un modulo senza influenzare altri.
3. **Cyclomatic complexity (MDP-14)** La complessità ciclomatica valuta la complessità del codice sorgente attraverso la misurazione del numero di cammini indipendenti attraverso il grafo di controllo del flusso. Una complessità ciclomatica più alta indica che il codice è più difficile da comprendere e manutenere.

Metrica	Nome	Valore Accettabile	Valore ottimo
MDP-12	Code smells	0	0
MDP-13	Coefficient of coupling	<= 30%	<= 10%
MDP-14	Cyclomatic complexity	<= 5	<= 3

IV. Specifiche dei test

- Questa parte illustra le procedure di test realizzate per assicurare che i vincoli stabiliti nei requisiti vengano completamente rispettati. Conformemente a quanto indicato nel documento Norme di Progetto, il piano di test adotta una metodologia sistematica e si suddivide nelle seguenti tipologie.

IV - 1. Test di unità

- Hanno l'obiettivo di controllare il corretto comportamento dei componenti software più piccoli e indipendenti, realizzati prevalentemente durante la fase progettuale.

IV - 2. Test di integrazione

- Seguono i test di unità e mirano a controllare la comunicazione tra differenti moduli software per assicurare che operino congiuntamente per funzionalità specifiche.

IV - 3. Test di sistema

- Riguardano l'intero sistema, garantendo che tutti i requisiti operativi, di performance e qualitativi stabiliti vengano soddisfatti.

IV - 4. Test di accettazione

- Condotti insieme al committente, servono a garantire che il prodotto finale sia conforme alle aspettative e ai requisiti contrattuali, permettendone il rilascio definitivo.