

Piano di qualifica

v0.3.0



<🔗>Farmacode

farmacode.swe.unipd@gmail.com

Registro delle modifiche

Versione	Data	Scrittori	Revisori	Descrizione
0.3.0	2023-11-27	Carraro Alessandro		Stesura sezione qualità di architettura
0.2.0	2023-11-24	Carraro Alessandro		Stesura sezione introduzione
0.1.0	2023-11-24	Carraro Alessandro		Struttura iniziale del documento

Indice

1) Introduzione	4
1.1) Scopo del documento	4
1.2) Glossario	4
1.3) Maturità e miglioramenti	4
1.4) Riferimenti	4
1.4.1) Riferimenti informativi	4
2) Piano di qualifica	4
2.1) Qualità dell'architettura	4
2.1.1) Modularità	5
2.1.2) Disponibilità	5
2.1.3) Semplicità	5
2.1.4) Incapsulazione	5
2.1.5) Coesione	5
2.1.6) Accoppiamento	5
2.2) Qualità del software	6
2.2.1) Visione orizzontale	6
2.2.2) Visione verticale	6
2.3) Qualità di processo	6
3) Controllo di qualità	6
3.1) Test	6
3.2) Grafici	6
3.3) Tabelle	6
4) Miglioramenti	6

1) Introduzione

Il documento riguardante il piano di qualifica, un elemento di fondamentale importanza per i progetti di sviluppo software che vogliono rispettare i massimi standard di qualità definiti dall'insegnamento dell'ingegneria del software.

La qualità e la valutazione del prodotto sono due concetti fondamentali per fare confronti o determinare il grado di conformità alle attese.

Il documento sarà utile a chi si occupa della creazione del prodotto, a chi userà il prodotto e a chi lo valuterà.

1.1) Scopo del documento

Il presente documento ha lo scopo di fornire una descrizione dettagliata e il più precisa possibile sulle metriche e sulle metodologie di controllo e misurazione della qualità nelle varie parti del prodotto software.

Verranno definiti gli obiettivi di qualità e i relativi processi e risorse necessarie per conseguirli, oltre alla definizione dei test con la loro documentazione: metodologie ed esito.

1.2) Glossario

Al fine di evitare eventuali equivoci o incomprensioni riguardo la terminologia utilizzata all'interno di questo documento, si è deciso di adottare un Glossario, con file apposito, in cui vengono riportate tutte le definizioni rigogliose delle parole ambigue utilizzate in ambito di questo progetto. Nel documento appena descritto verranno riportati tutti i termini definiti nel loro ambiente di utilizzo con annessa descrizione del loro significato.

La presenza di un termine all'interno del Glossario è evidenziata dal *colore blu*.

1.3) Maturità e miglioramenti

Questo documento è stato realizzato utilizzando un approccio incrementale, con lo scopo di semplificare i cambiamenti nel tempo in base alle reciproche esigenze decise da entrambi le parti, ovvero membri del gruppo di progetto e azienda proponente. Pertanto questo documento non può essere considerato esaustivo e completo.

1.4) Riferimenti

1.4.1) Riferimenti informativi

- T6 - Progettazione software (slide del corso di Ingegneria del Software);
- T7 - Qualità del software (slide del corso di Ingegneria del Software);
- T8 - Qualità di processo (slide del corso di Ingegneria del Software).

2) Piano di qualifica

2.1) Qualità dell'architettura

La qualità di una buona architettura è definita da questi aspetti:

Aspetti auto-esplicativi o non qualificabili

- Sufficienza;
- Robustezza;

Aspetti considerati per qualificazione architettura

- Modularità;
- Disponibilità;

- Flessibilità;
- Riusabilità;
- Efficienza;
- Affidabilità;
- Sicurezza rispetto a malfunzionamenti;
- Sicurezza rispetto a intrusioni.
- Semplicità;
- Incapsulazione;
- Coesione;
- Basso accoppiamento;

2.1.1) Modularità

Si intende la suddivisione dell'architettura in parti chiare e ben distinte.

I passaggi di qualità sono due:

- Determinare le parti che compongono l'interfaccia utente e quelle che compongono l'implementazione;
- Particolare attenzione va posta nell'evitare l'effetto domino, ovvero quando la modifica interna di una parte comporta modifiche anche al suo esterno.

Questi aspetti andranno verificati in fase di design e conseguentemente documentati. La strategia scelta per modularizzare è ricercando information hiding.

2.1.2) Disponibilità

Anche detta availability, indica il grado di indisponibilità causata dalla manutenzione.

Questo aspetto andrà verificato in fase di design e conseguentemente documentato.

2.1.3) Semplicità

Questo aspetto andrà verificato in fase di design e conseguentemente documentato.

In particolare andranno verificati:

- la presenza di elementi strettamente necessari;
- ogni elemento sarà il più semplice possibile, senza renderlo banale.

2.1.4) Incapsulazione

Questo aspetto andrà verificato in fase di design e conseguentemente documentato.

In particolare andranno verificati:

- Rendere invisibile all'esterno le componenti architetture;
- Esporre solo l'interfaccia, nascondendo algoritmi e strutture dati usate per realizzarla;
- La manutenzione all'interfaccia non andrà a danneggiare l'implementazione.

2.1.5) Coesione

Questo aspetto andrà verificato in fase di design e conseguentemente documentato.

In particolare andranno verificati:

- Funzionalità vicine stanno nella stessa componente;
- Minore interdipendenza fra componenti;
- Architettura più comprensibile;
- Giusto equilibrio fra modularità e coesione stessa.

La strategia scelta per modularizzare è ricercando information hiding.

2.1.6) Accoppiamento

Questo aspetto andrà verificato in fase di design e conseguentemente documentato. In particolare andranno verificati:

- Dipendenze fra parti, esempio parti interne ed esterne;
- Accoppiamento sia minimizzato:
 - il grado U di utilizzo reciproco di M componenti.
- Metriche fan-in e fan-out:
 - SFIN, indice di utilità;
 - SFOUT, indice di dipendenza;
 - Una buona progettazione ha componenti con SFIN elevato.

2.2) Qualità del software

2.2.1) Visione orizzontale

2.2.2) Visione verticale

2.3) Qualità di processo

3) Controllo di qualità

3.1) Test

3.2) Grafici

3.3) Tabelle

4) Miglioramenti