



Piano di Qualifica

Gruppo SWEetBIT — Progetto SWEDesigner

Informazioni sul documento

Versione	1.0.0
Redazione	Sebastiano Bertolin
Verifica	Da inserire
Approvazione	Da inserire
Uso	Interno
Distribuzione	Prof. Tullio Vardanega Prof. Riccardo Cardin Gruppo SWEetBIT

Descrizione

Questo documento descrive le strategie di verifica, adottate dal gruppo SWEet BIT, atte a garantire gli obiettivi qualitativi riguardanti il prodotto SWEDesigner.

Indice

1	Introduzione	3
1.1	Scopo del documento	3
1.2	Scopo del Prodotto	3
1.3	Glossario	3
1.4	Riferimenti	3
1.4.1	Informativi	3
1.4.2	Normativi	4
2	Visione generale delle strategie di verifica	5
2.1	Definizione obiettivi di qualità	5
2.1.1	Qualità di prodotto	5
2.1.2	Qualità di processo	5
2.2	Organizzazione	6
2.3	Pianificazione strategica e temporale	6
2.4	Gestione delle responsabilità	6
3	Strategia di verifica nel dettaglio	7
3.1	Risorse	7
3.2	Misure e metriche	7
3.3	Tecniche di analisi	7
4	Bibliografia	8

Versioni del documento

Versione	Data	Persone coinvolte	Descrizione
0.0.1	AAAA/MM/GG	Sebastiano Bertolin	Stesura Introduzione e Definizione Obbiettivi di Qualità

1 Introduzione

1.1 Scopo del documento

Questo documento descrive le strategie adottate dal gruppo SWEet BIT per il conseguimento degli obiettivi di qualità riguardanti il prodotto. Il raggiungimento di tali obiettivi è possibile solo tramite una precisa e continua verifica delle attività svolte; così facendo è possibile rilevare e correggere tempestivamente eventuali anomalie, minimizzando lo spreco di risorse.

1.2 Scopo del Prodotto

Lo scopo del progetto è la realizzazione di una *Web App_G* che fornisca all'utente un *UML_G Designer_G* con il quale riuscire a disegnare correttamente diagrammi delle classi e descrivere il comportamento dei metodi interni alle stesse attraverso l'utilizzo di -da decidere il tipo di schema-. La *Web App_G* permetterà all'utente di generare codice Java o Javascript dal diagramma disegnato ed eventualmente andare a ritoccarne il risultato al fine di ottenere un codice eseguibile, funzionante e funzionale.

1.3 Glossario

Con lo scopo di evitare ambiguità di linguaggio e di massimizzare la comprensione dei documenti, il gruppo ha steso un documento interno che è il *Glossario v1.0.0*. In esso saranno definiti, in modo chiaro e conciso i termini che possono causare ambiguità o incomprensione del testo.

1.4 Riferimenti

1.4.1 Informativi

- **Analisi dei requisiti:** *Analisi dei requisiti v1.0.0*
- **Piano di progetto:** *Piano di progetto v1.0.0*
- **Lucidi dell'insegnamento di Ingegneria del Software:**
<http://www.math.unipd.it/~tullio/IS-1/2016/>
- **Libro SWEBOK v3.0: Chapter 10: Software Quality:**
<https://www.computer.org/web/swebok/>
- **ISO_G/IEC_G 9126 Software engineering — Product quality:**
https://en.wikipedia.org/wiki/ISO/IEC_9126

- ***ISO_G/IEC_G 15504 Information technology – Process assessment:***
https://en.wikipedia.org/wiki/ISO/IEC_15504

1.4.2 Normativi

- **Norme di progetto:** *Norme di progetto v1.0.0*
- **Capitolato di appalto SWEDesigner (C6):**
<http://www.math.unipd.it/~tullio/IS-1/2016/Progetto/C6.pdf>

2 Visione generale delle strategie di verifica

2.1 Definizione obiettivi di qualità

Questa sezione si presta a descrivere sia gli obiettivi di qualità, richiesti dal committente, riguardanti il prodotto che quelli relativi ai processi necessari alla sua produzione.

2.1.1 Qualità di prodotto

Il gruppo SWEet BIT si è impegnato ad aderire allo standard ISO_G/IEC_G 9126, in modo garantire le seguenti caratteristiche di qualità per il prodotto:

Funzionalità: il prodotto deve soddisfare i requisiti indicati nel documento *Analisi dei Requisiti*.

Affidabilità: il software prodotto deve essere robusto, deve essere prevista una completa gestione degli errori ed eccezioni, in modo da prevenire perdite di dati e facilitarne il ripristino.

Usabilità: il software prodotto deve risultare semplice e di facile utilizzo da parte dell'utente; esso deve essere di facile apprendimento, l'interfaccia utente deve risultare familiare e intuitiva. Il prodotto deve soddisfare al meglio le esigenze dell'utente finale.

Efficienza: Il software deve essere capace di fornire tutti i servizi attesi con il minimo dispendio di risorse.

Manutenibilità: Il software prodotto deve essere facilmente modificabile, in caso di correzione di errori e migliorie da apportare; esso deve essere facilmente adattabile in caso di cambio di ambiente operativo e/o cambio di requisiti.

Portabilità: Il software prodotto deve, per la parte $front\ end_G$, offrire gli stessi servizi nella maggior parte dei $browser_G$; la parte di $back\ end_G$ deve funzionare su vari sistemi operativi.

Per ognuna delle caratteristiche principali appena elencate esistono diverse sotto-caratteristiche, le quali sono approfondite nella sotto-sezione Standard e metodi per la gestione della Qualità dove è descritto lo standard ISO_G/IEC_G 9126

.

2.1.2 Qualità di processo

La qualità dei processi che portano allo sviluppo di un software hanno un ruolo fondamentale sulla qualità del prodotto. Il gruppo SWEet BIT ha deciso di adottare il

ciclo di Deming, o ciclo $PDCA_G$, come modello per il continuo miglioramento dei processi produttivi; come complemento al ciclo di Deming è stato scelto il modello descritto dallo standard ISO_G/IEC_G 15504 detto $SPICE_G$, il quale fornisce gli strumenti per la valutazione dei processi produttivi.

2.2 Organizzazione

Per ognuna delle fasi descritte dal *Piano di progetto* è necessario seguire una verifica mirata al tipo di processo e relativo prodotto:

- **Analisi:** in questa fase vengono controllati i processi e la documentazione prodotta, la verifica dovrà essere eseguita secondo i metodi descritti nel documento *Norme di progetto*.
- **Analisi nel dettaglio:** durante questa fase sono verificati i processi che portano all'incremento dei documenti, e i documenti stessi, prodotti nella precedente fase di Analisi. Le procedure di verifica sono descritte nel documento *Norme di progetto*.
- **Progettazione architetturale:** in questa fase vengono verificati i processi, e relativi prodotti, che portano all'incremento dei documenti redatti nella precedente fase; sempre in questa fase sono verificati i processi e prodotti riguardanti la progettazione architetturale. Le procedure di verifica sono descritte in dettaglio nel documento *Norme di progetto*.

2.3 Pianificazione strategica e temporale

Intendendo rispettare le scadenze specificate nel *Piano di progetto* è necessario un approccio organico e ben pianificato alle attività di verifica. Deve essere pianificato ogni incremento su codice e documentazione, di conseguenza lo dovranno essere tutte le attività di verifica in modo da avere il tempo di effettuare le dovute correzioni. Ogni processo di verifica dovrà essere automatizzato il più possibile, questo per poter, dove possibile, risparmiare su risorse umane. L'utilizzo di software apposito, e i metodi, per la verifica sono descritti nel documento *Norme di progetto*.

2.4 Gestione delle responsabilità

Il *Responsabile ei progetto* e i *Verificatori* hanno la responsabilità delle attività di verifica; i compiti di queste figure sono argomento trattato nel documento *Piano di progetto*.

3 Strategia di verifica nel dettaglio

3.1 Risorse

3.2 Misure e metriche

3.3 Tecniche di analisi

4 Bibliografia

Riferimenti bibliografici

- [1] Per approfondimenti sullo standard ISO_G/IEC_G 9126 vedi
https://en.wikipedia.org/wiki/ISO/IEC_9126