

**KALEIDOSCODE**  
**SWEDesigner**  
SOFTWARE PER DIAGRAMMI UML

PIANO DI QUALIFICA V1.0.0



Informazioni sul documento

<b>Versione</b>	1.0.0
<b>Data Redazione</b>	09/03/2017
<b>Redazione</b>	Bonato Enrico Bonolo Marco Pace Giulio Sovilla Matteo
<b>Verifica</b>	Pezzuto Francesco
<b>Approvazione</b>	Sanna Giovanni
<b>Uso</b>	Esterno
<b>Distribuzione</b>	<i>Prof. Vardanega Tullio</i> <i>Prof. Cardin Riccardo</i> <i>Zucchetti s.p.a.</i>

---

## Diario delle Modifiche

Versione	Data	Autore	Descrizione
0.0.1	09/03/2017	Bonolo Marco	Creazione scheletro del documento e stesura della sezione Introduzione

---

# Indice

<b>1</b>	<b>Introduzione</b>	<b>2</b>
1.1	Scopo del documento . . . . .	2
1.2	Scopo del prodotto . . . . .	2
1.3	Glossario . . . . .	2
1.4	Riferimenti utili . . . . .	2
1.4.1	Riferimenti normativi . . . . .	2
1.4.2	Riferimenti informativi . . . . .	2
<b>2</b>	<b>Definizione obiettivi di qualità</b>	<b>3</b>
2.1	Funzionalità . . . . .	3
2.2	Affidabilità . . . . .	3
2.3	Usabilità . . . . .	3
2.4	Efficienza . . . . .	4
2.5	Manutenibilità . . . . .	4
2.6	Portabilità . . . . .	4
2.7	Altre qualità . . . . .	4
<b>3</b>	<b>La strategia di gestione della qualità nel dettaglio</b>	<b>5</b>
3.1	Risorse . . . . .	5
3.1.1	Necessarie . . . . .	5
3.1.2	Disponibili . . . . .	5
3.2	Misure e metriche . . . . .	5
3.2.1	Metriche per i processi . . . . .	6
3.2.2	Metriche per i documenti . . . . .	6
3.2.3	Metriche per il codice . . . . .	7

# 1 Introduzione

## 1.1 Scopo del documento

Questo documento definisce gli obiettivi e le metodologie che ogni membro del gruppo *KaleidosCode* adotterà per garantire un determinato livello di qualità del prodotto. A tal proposito ogni membro del gruppo è tenuto a leggere, perseguire e raggiungere gli obiettivi definiti in esso.

## 1.2 Scopo del prodotto

Lo scopo del progetto è la realizzazione di un software di costruzione di diagrammi UML<sub>G</sub> con la relativa generazione di codice Java<sub>G</sub> e Javascript<sub>G</sub> utilizzando tecnologie web. Il prodotto deve essere conforme ai vincoli qualitativi richiesti dal committente.

## 1.3 Glossario

Al fine di evitare ogni ambiguità di linguaggio e massimizzare la comprensione dei documenti, i termini tecnici, di dominio, gli acronimi e le parole che necessitano di essere chiarite, sono riportate nel documento *Glossario v1.0.0*.

Ogni occorrenza di vocaboli presenti nel *Glossario* è marcata da una “G” maiuscola in pedice.

## 1.4 Riferimenti utili

### 1.4.1 Riferimenti normativi

- Capitolato d'appalto:  
<http://www.math.unipd.it/~tullio/IS-1/2016/Progetto/C6.pdf> (09/03/2017).

### 1.4.2 Riferimenti informativi

- Qualità del software (Slide del Corso di Ingegneria del Software):  
<http://www.math.unipd.it/~tullio/IS-1/2016/Dispense/L10.pdf> (09/03/2017);
- Qualità di Processo (Slide del Corso di Ingegneria del Software):  
<http://www.math.unipd.it/~tullio/IS-1/2016/Dispense/L11.pdf> (09/03/2017);
- Glossario: *Glossario v1.0.0*.

## 2 Definizione obiettivi di qualità

Prendendo come riferimento lo standard [ISO/IEC 9126] e lo standard [ISO/IEC 12207] il team si impegna a garantire che SWEDesigner abbia le seguenti qualità:

### 2.1 Funzionalità

Si garantisce che il sistema prodotto abbia tutte le funzionalità che il documento *Analisi dei requisiti v1.0.0* indica. L'implementazione di ogni requisito deve essere quanto più completa ed economica.

- **Misura:** l'unità di misura utilizzata sarà la quantità di requisiti mappati in componenti del sistema create e funzionanti;
- **Metrica:** la sufficienza è raggiunta quando vengono soddisfatti tutti i requisiti obbligatori;
- **Strumenti:** il sistema deve superare tutti i test previsti dalla documentazione prodotta e consegnata in sede di Revisione dei Requisiti.

### 2.2 Affidabilità

Il sistema deve essere quanto più possibile robusto. Nel caso di eventuali errori deve essere di facile ripristino.

- **Misura:** l'unità di misura utilizzata sarà la quantità di esecuzioni che vanno a buon fine;
- **Metrica:** visto che non è possibile valutare a monte tutte le possibili casistiche di utilizzo le esecuzioni dovranno il più possibile coprire la possibile gamma di possibilità. Per questo motivo è impossibile stabilire oggettivamente una esatta soglia che corrisponda alla sufficienza;
- **Strumenti:** ancora da definire.

### 2.3 Usabilità

Il sistema deve risultare per quanto possibile intuitivo e di facile utilizzo. Deve coniugare una facilità di apprendimento e utilizzo con il soddisfacimento di tutte le necessità dell'utente.

- **Misura:** poichè non esiste una metrica oggettiva che riguarda questo ambito l'unità di misura utilizzata sarà una valutazione soggettiva dell'usabilità;
- **Metrica:** non esistendo una metrica oggettiva è impossibile determinare con certezza quale sia la sufficienza. In ogni caso i membri del gruppo si impegneranno a garantire un'usabilità più alta possibile;
- **Strumenti:** si vedano le *Norme di progetto v1.0.0*.

## 2.4 Efficienza

Il sistema deve ridurre al minimo l'utilizzo delle risorse impiegate e deve fornire le funzionalità richieste nel minor tempo possibile.

- **Misura:**
- **Metrica:**
- **Strumenti:**

## 2.5 Manutenibilità

Il sistema deve essere più possibile estensibile e comprensibile.

- **Misura:** l'unità di misura utilizzata sarà quella descritta nella sezione "Metriche per il codice"
- **Metrica:** il prodotto deve avere la sufficienza in tutte le metriche descritte nella sezione "Metriche per il codice".
- **Strumenti:** si vedano le *Norme di progetto v1.0.0* .

## 2.6 Portabilità

Il sistema deve essere più portabile possibile. Il front end<sub>G</sub> dovrà funzionare correttamente su più browser possibile. Inoltre dovrà essere supportato da più sistemi operativi possibili.

- **Misura:** il front end deve rispettare gli standard W3C<sub>G</sub>;
- **Metrica:** Il software dovrà avere le caratteristiche di portabilità descritte. Per questo motivo sarà necessario raggiungere la sufficienza in tutte le metriche descritte nella sezione "Metriche per il codice";
- **Strumenti:** si vedano le *Norme di progetto v1.0.0* .

## 2.7 Altre qualità

Saranno importanti per la qualità del progetto anche i seguenti aspetti:

- **incapsulamento:** un buon livello di incapsulamento è preferibile in quanto aumenta la riusabilità e la manutenibilità del codice. A questo scopo saranno quindi utilizzate interfacce dove possibile
- **coesione:** le funzionalità che concorrono a uno stesso obiettivo devono risiedere nello stesso componente in modo da favorire semplicità e manutenibilità. In questo modo viene inoltre ridotto l'indice di dipendenza.

## 3 La strategia di gestione della qualità nel dettaglio

### 3.1 Risorse

#### 3.1.1 Necessarie

Per la realizzazione del prodotto sono necessarie le risorse umane e tecnologiche citate di seguito.

- **Risorse umane:** sono descritte dettagliatamente nel *Piano di progetto*.
  - *Responsabile di progetto*;
  - *Amministratore*;
  - *Analista*;
  - *Progettista*;
  - *Programmatore*;
  - *Verificatore*.
- **Risorse software:** sono descritte dettagliatamente nelle *Norme di progetto*. Si tratta di software che permettano:
  - la comunicazione e la condivisione del lavoro tra gli elementi del team;
  - la stesura della documentazione in formato LaTeX;
  - la creazione di diagrammi UML;
  - la codifica nei linguaggi di programmazione scelti;
  - la semplificazione delle attività di verifica;
  - la gestione dei test sul codice.
- **Risorse hardware:** ciascun componente del gruppo ha bisogno di un computer con tutti i software necessari. È necessario avere a disposizione almeno un luogo dove poter effettuare le riunioni del team.

#### 3.1.2 Disponibili

Ogni membro del team ha a disposizione uno o più computer personali dotati degli strumenti necessari.

Le riunioni interne si svolgono presso le aule del dipartimento di Matematica dell'Università degli Studi di Padova.

### 3.2 Misure e metriche

Il processo di verifica deve essere quantificabile per fornire informazioni utili, bisogna quindi stabilire le metriche da adottare per le misurazioni. Si definiranno due intervalli di misure:

- **Range di accettazione:** intervallo di valori vincolante per l'accettazione del prodotto;

- **Range ottimale:** intervallo di valori entro cui è consigliabile rientri la misurazione. Il mancato rispetto di questa condizione non pregiudica l'accettazione del prodotto, ma richiede verifiche più approfondite in merito.

### 3.2.1 Metriche per i processi

#### Schedule Variance

È una metrica di progetto standard, indica se si è in linea, in anticipo o in ritardo rispetto alla schedulazione pianificata delle attività di progetto. È pari alla differenza tra il valore delle attività pianificate e il valore delle attività svolte alla data corrente.

##### Parametri utilizzati

- **Range di accettazione:**  $\geq -(\text{preventivo} * 5\%)$ ;
- **Range ottimale:**  $\geq 0$ .

#### Budget Variance

È una metrica di progetto standard, indica se si spende di più o di meno rispetto a quanto preventivato alla data corrente. È pari alla differenza tra costo pianificato e costo effettivamente sostenuto alla data corrente.

##### Parametri utilizzati

- **Range di accettazione:**  $\geq -(\text{preventivo} * 10\%)$ ;
- **Range ottimale:**  $\geq 0$ .

### 3.2.2 Metriche per i documenti

#### Indice Gulpease

Definito nel 1988 all'Università degli Studi di Roma "La Sapienza" per valutare la leggibilità di un documento redatto in lingua italiana, l'indice Gulpease<sub>G</sub> si basa sul calcolo del numero di caratteri contenuto in una parola rapportato con altri fattori quali il numero di parole e di frasi. La formula per il calcolo dell'indice Gulpease è la seguente:

$$89 + \frac{300 (\text{numero di frasi}) - 10 (\text{numero di lettere})}{\text{numero di parole}}$$

Il risultato indica quindi la complessità del documento con un valore compreso tra 0 e 100, dove 100 indica la più alta leggibilità. Attraverso gli studi condotti, risulta che testi con un indice:

- **inferiore a 80** sono difficili da leggere per chi ha la licenza elementare;
- **inferiore a 60** sono difficili da leggere per chi ha licenza media;
- **inferiore a 40** sono difficili da leggere per chi ha un diploma superiore.



Tale indice, però, non indica la comprensibilità del testo. Il documento potrebbe contenere frasi incomprensibili ed avere comunque un alto indice Gulpease. Per la tipologia dei documenti redatti, la formalità nella scrittura e gli argomenti trattati risulta difficile adeguare la stesura del testo ad un indice Gulpease ottimale. Per questo motivo, ogni documento sarà valutato anche da un individuo che avrà il compito di valutare se parti di testo dovranno essere semplificate o meno. Inoltre, i limiti imposti da tale indice saranno sufficientemente rilassati per accettare anche frasi poco più complesse.

#### **Parametri utilizzati**

- **Range di accettazione:** 40 - 100;
- **Range ottimale:** 50 - 100.

### **3.2.3 Metriche per il codice**

#### **Rapporto linee di commento su linee di codice**

Indica il rapporto tra linee di commento e linee di codice in un file (linee vuote escluse). Ritenendo importante la rapidità di comprensione del codice, questa metrica è utile per stimare la manutenibilità.

#### **Parametri utilizzati**

- **Range di accettazione:**  $\geq 0.25$ ;
- **Range ottimale:**  $\geq 0.30$ .

#### **Numero di parametri**

Indica il numero di parametri formali di un metodo. Più alto è il numero dei parametri formali, più aumenta la quantità di memoria occupata nella pila dei processi.

#### **Parametri utilizzati**

- **Range di accettazione:** 0 - 8;
- **Range ottimale:** 0 - 5.

#### **Numero di campi dati**

Indica il numero di campi dati interni ad una classe. Un numero elevato può rendere difficile la manutenibilità del codice della classe, oltre ad essere indice di cattiva programmazione.

È possibile ridurre il numero di campi dati attraverso l'incapsulamento di ulteriori classi.

#### **Parametri utilizzati**

- **Range di accettazione:** 0 - 16;
- **Range ottimale:** 0 - 10.

**Complessità ciclomatica**

Indica il numero di cammini linearmente indipendenti attraverso il grafo di controllo di flusso del metodo/funzione: i nodi del grafo corrispondono a gruppi indivisibili di istruzioni, mentre gli archi connettono due nodi se il secondo gruppo può essere eseguito immediatamente dopo il primo.

È possibile ridurre l'indice di complessità attraverso la suddivisione del metodo/funzione in più parti.

**Parametri utilizzati**

- **Range di accettazione:** 0 - 10;
- **Range ottimale:** 0 - 6.

È accettato anche un valore più elevato, qualora dovesse influire positivamente sulla velocità di esecuzione.