

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

NORME DI PROGETTO V1.0.0



**Informazioni sul documento**

<b>Versione</b>	1.0.0
<b>Data Redazione</b>	03/03/2017
<b>Redazione</b>	Bonato Enrico Bonolo Marco Pace Giulio Pezzuto Francesco
<b>Verifica</b>	Sovilla Matteo
<b>Approvazione</b>	Sanna Giovanni
<b>Uso</b>	Interno
<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
0.1.0	07/03/2017	Matteo Sovilla	Verifica e correzione
0.0.8	05/03/2017	Giulio Pace	Stesura sezione Processi organizzativi
0.0.7	05/03/2017	Marco Bonolo	Stesura sezione Processo di fornitura
0.0.6	04/03/2017	Marco Bonolo	Stesura sezioni Progettazione, Codifica in Processo di sviluppo
0.0.5	03/03/2017	Pezzuto Francesco	Stesura sezione Analisi dei requisiti in Processi di sviluppo e creazione template per documento
0.0.4	02/03/2017	Bonato Enrico	Stesura sezione Documentazione in Processi di supporto
0.0.3	01/03/2017	Bonolo Marco	Stesura sezione Riunioni in Processi organizzativi
0.0.2	01/03/2017	Pezzuto Francesco	Stesura Introduzione e sezione Comunicazioni in Processi organizzativi
0.0.1	01/03/2017	Pace Giulio	Creazione scheletro del documento e stesura parziale dei documenti

# 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>Processo di Fornitura</b>	<b>4</b>
2.1	Scopo . . . . .	4
2.2	Risultati . . . . .	4
2.3	Descrizione . . . . .	4
2.4	Identificazione opportunità . . . . .	4
2.5	Accordo contrattuale . . . . .	5
2.6	Esecuzione del contratto . . . . .	5
2.7	Consegna e supporto del prodotto e/o servizio . . . . .	5
2.8	Chiusura . . . . .	5
<b>3</b>	<b>Processo di sviluppo</b>	<b>6</b>
3.1	Scopo . . . . .	6
3.2	Aspettative . . . . .	6
3.3	Descrizione . . . . .	6
3.4	Analisi dei requisiti . . . . .	6
3.4.1	Scopo . . . . .	6
3.4.2	Aspettative . . . . .	6
3.4.3	Descrizione . . . . .	6
3.4.4	Classificazione dei requisiti . . . . .	7
3.4.5	Classificazione dei casi d'uso . . . . .	7
3.5	Progettazione . . . . .	8
3.5.1	Scopo . . . . .	8
3.5.2	Aspettative . . . . .	8
3.5.3	Descrizione . . . . .	9
3.5.4	Diagrammi UML . . . . .	9
3.6	Codifica . . . . .	9
3.6.1	Scopo . . . . .	9
3.6.2	Aspettative . . . . .	9
3.6.3	Descrizione . . . . .	9
3.6.4	Standard di codifica . . . . .	10
<b>4</b>	<b>Processi di supporto</b>	<b>11</b>
4.1	Struttura dei documenti . . . . .	11
4.1.1	Descrizione . . . . .	11
4.1.2	Approvazione dei documenti . . . . .	11
4.2	Template . . . . .	11
4.3	Codifica e convenzioni . . . . .	11

4.4	Intestazione . . . . .	11
4.5	Struttura dei documenti . . . . .	12
4.5.1	Prima pagina . . . . .	12
4.5.2	Diario delle modifiche . . . . .	13
4.5.3	Indici . . . . .	13
4.5.4	Struttura generale delle pagine . . . . .	13
4.6	Norme tipografiche . . . . .	13
4.6.1	Punteggiatura . . . . .	14
4.6.2	Stile di testo . . . . .	14
4.6.3	Composizione del testo . . . . .	14
4.6.4	Formati . . . . .	14
4.6.5	Riferimenti Vari . . . . .	15
4.6.6	Sigle . . . . .	17
4.7	Componenti grafiche . . . . .	17
4.7.1	Tabelle . . . . .	17
4.7.2	Immagini . . . . .	17
4.8	Classificazione documenti . . . . .	17
4.8.1	Documenti formali . . . . .	17
4.8.2	Documenti informali . . . . .	17
4.9	Versionamento . . . . .	18
4.9.1	Variazione indici . . . . .	18
4.10	Verifica . . . . .	18
4.10.1	Scopo del processo . . . . .	18
4.11	Attività . . . . .	18
4.11.1	Analisi Statica . . . . .	18
4.11.2	Analisi Dinamica . . . . .	19
4.12	Validazione . . . . .	19
4.12.1	Scopo del processo . . . . .	19
4.12.2	Attività . . . . .	19
4.13	Issue . . . . .	19
4.13.1	Issue Tracking . . . . .	19
4.13.2	Gestione delle issue . . . . .	19
4.13.3	Strumenti per l'issue tracking . . . . .	20
4.13.4	Verifica ortografica . . . . .	20
4.14	Consigli di utilizzo . . . . .	20
4.14.1	Visualizzazione dei Task . . . . .	20
4.14.2	Visualizzazione conversazioni . . . . .	20
<b>5</b>	<b>Processi organizzativi</b>	<b>21</b>
5.1	Scopo . . . . .	21
5.2	Aspettative . . . . .	21
5.3	Ruoli di progetto . . . . .	21
5.3.1	Responsabile di progetto . . . . .	21
5.3.2	Amministratore . . . . .	21
5.3.3	Analista . . . . .	22
5.3.4	Progettista . . . . .	22

5.3.5	Programmatore . . . . .	22
5.3.6	Verificatore . . . . .	22
5.4	Comunicazioni . . . . .	23
5.4.1	Esterne . . . . .	23
5.4.2	Interne . . . . .	23
5.5	Composizione e-mail e conversazioni . . . . .	23
5.5.1	Destinatario . . . . .	23
5.5.2	Mittente . . . . .	24
5.5.3	Oggetto . . . . .	24
5.5.4	Corpo . . . . .	24
5.5.5	Allegati . . . . .	24
5.6	Riunioni . . . . .	24
5.6.1	Frequenza . . . . .	24
5.6.2	Convocazione riunione . . . . .	24
5.6.3	Svolgimento riunione . . . . .	26
5.6.4	Verbale . . . . .	26
5.7	Strumenti di coordinamento . . . . .	26
5.8	Strumenti di versionamento . . . . .	27
5.9	Rischi . . . . .	27
5.10	Altri strumenti . . . . .	27
5.10.1	Google Drive . . . . .	27
5.10.2	ProjectLibre . . . . .	28
5.10.3	L <sup>A</sup> T <sub>E</sub> X . . . . .	28

# 1 Introduzione

## 1.1 Scopo del documento

Questo documento definisce le norme che i membri del gruppo *KaleidosCode* adotteranno nello svolgimento del progetto *SWEDesigner*.

Tutti i membri del gruppo sono tenuti a leggere il documento e a seguirne le norme descritte per uniformare il materiale prodotto, ridurre il numero di errori e migliorare l'efficienza.

In particolare verranno definite norme riguardanti:

- Interazioni tra i membri del gruppo;
- Stesura di documenti e convenzioni;
- Modalità di lavoro durante le varie fasi del progetto;
- Ambiente di lavoro.

## 1.2 Scopo del prodotto

Lo scopo del progetto è la realizzazione di un software di costruzione di diagrammi UML con la relativa generazione di codice Java e Javascript utilizzando tecnologie web.

## 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

- ISO 12207:  
[https://en.wikipedia.org/wiki/ISO/IEC\\_12207](https://en.wikipedia.org/wiki/ISO/IEC_12207) (03/03/2017);
- Swebok v3: <https://www.computer.org/web/swebok/v3> (03/03/2017).

### 1.4.2 Riferimenti informativi

- Specifiche UTF-8:  
<http://www.unicode.org/versions/Unicode6.1.0/ch03.pdf> (03/03/2017);
- Capitolato d'appalto:  
<http://www.math.unipd.it/~tullio/IS-1/2016/Progetto/C6.pdf> (03/03/2017);
- Glossario: *Glossario v1.0.0*;

- **Piano di Progetto:** *Piano di progetto v1.0.0;*
- **Piano di Qualifica:** *Piano di qualifica v1.0.0.*

## 2 Processo di Fornitura

### 2.1 Scopo

Lo scopo del Processo di Fornitura è quello di consegnare un prodotto e/o un servizio che soddisfi i requisiti concordati.

### 2.2 Risultati

I risultati ottenuti in seguito ad una corretta attuazione del Processo di Fornitura sono:

- stabilire un accordo tra fornitore e proponente in merito allo sviluppo, al mantenimento, al funzionamento, alla consegna e all'installazione del prodotto e/o servizio;
- realizzare un prodotto e/o servizio che soddisfi i requisiti concordati;
- consegnare il prodotto al proponente in conformità con i requisiti concordati;
- installare il prodotto in conformità con i requisiti concordati.

### 2.3 Descrizione

Rispettando lo standard [ISO/IEC 12207], il fornitore deve svolgere le seguenti attività:

- identificazione opportunità (*Studio di fattibilità*);
- accordo contrattuale;
- esecuzione del contratto;
- consegna e supporto del prodotto e/o servizio;
- chiusura.

### 2.4 Identificazione opportunità

Successivamente alla pubblicazione dei capitolati d'appalto, il *Responsabile di progetto* ha il compito di convocare il numero di riunioni necessarie al confronto tra i membri del gruppo sui capitolati proposti. Gli *Analisti* hanno così modo di ricavare sufficienti informazioni riguardanti le conoscenze e preferenze di ogni membro del gruppo. Sulla base delle decisioni prese, gli *Analisti* devono redigere uno *Studio di fattibilità* dei capitolati secondo:

- **Dominio tecnologico:** conoscenze sulle tecnologie impiegate nello sviluppo del progetto in questione;
- **Dominio applicativo:** conoscenze sul dominio di applicazione del prodotto;
- **Individuazione di rischi e criticità:** punti critici ed eventuali rischi percorribili durante lo sviluppo.

Nello *Studio di fattibilità* sono racchiuse le motivazioni che hanno spinto il nostro gruppo a candidarsi come fornitore per il proponente *Zucchetti s.p.a.*



## 2.5 Accordo contrattuale

Il fornitore deve accordarsi con il proponente *Zucchetti s.p.a.* per chiarire, definire e accettare le richieste presenti nel documento di presentazione del capitolato fornito dal proponente.

## 2.6 Esecuzione del contratto

Il fornitore è tenuto a collaborare con il proponente *Zucchetti s.p.a.* per tutta la durata del progetto al fine di raggiungere i seguenti obiettivi:

- Chiarire ogni dubbio riguardante i vincoli sui requisiti;
- Chiarire ogni dubbio riguardante i vincoli di progetto.

Il fornitore è tenuto a procurare al proponente *Zucchetti s.p.a.* e ai committenti *Prof. Vardanega Tullio* e *Prof. Cardin Riccardo* i seguenti documenti:

- *Piano di progetto*
- *Analisi dei requisiti*
- *Piano di qualifica*

## 2.7 Consegna e supporto del prodotto e/o servizio

Dopo aver terminato le fasi di sviluppo, verifica e validazione, il fornitore è tenuto a consegnare al proponente *Zucchetti s.p.a.* il prodotto realizzato in conformità con i requisiti richiesti. Dovrà quindi consegnare:

- CD ROM contenente il prodotto realizzato;
- Manuale di installazione;
- Manuale d'uso.

Il fornitore, dopo la consegna del prodotto, non si occuperà della fase di manutenzione del suddetto.

## 2.8 Chiusura

La chiusura dell'accordo contrattuale tra fornitore e proponente è sancita dalla consegna del prodotto realizzato.

## 3 Processo di sviluppo

### 3.1 Scopo

In questa fase vengono incluse tutte le attività volte a creare il prodotto.

### 3.2 Aspettative

I risultati ottenuti in seguito ad una corretta attuazione del Processo di Sviluppo sono:

- realizzare un prodotto e/o servizio che soddisfi i requisiti concordati;
- realizzare un prodotto e/o servizio che soddisfi le attività di validazione e verifica;
- determinare eventuali vincoli tecnologici e requisiti;
- determinare gli obiettivi di sviluppo.

### 3.3 Descrizione

Nel rispetto dello standard [ISO/IEC 12207], si devono svolgere le seguenti attività:

- *Analisi dei requisiti*;
- Progettazione;
- Codifica;
- Qualificazione.

### 3.4 Analisi dei requisiti

#### 3.4.1 Scopo

Determinare tutti i requisiti del progetto. Il risultato di questa attività è un documento in cui vengono elencati i requisiti e i relativi casi d'uso.

#### 3.4.2 Aspettative

Produrre il documento *Analisi dei requisiti* in conformità ai requisiti richiesti dal proponente.

#### 3.4.3 Descrizione

Vengono analizzati e tracciati tutti i requisiti attraverso l'analisi della specifica del capitolato e la convocazione di riunioni con il proponente *Zucchetti s.p.a.* volte al chiarimento di eventuali dubbi o all'approfondimento di requisiti già noti.

### 3.4.4 Classificazione dei requisiti

Viene stilata una lista dei requisiti individuati nel capitolato e nelle riunioni avvenute con il proponente. La classificazione degli stessi deve avvenire secondo la seguente codifica:

$$R[Importanza][Tipo][Codice]$$

dove:

- **Importanza** può assumere i seguenti valori:
  - **0** se il requisito è obbligatorio;
  - **1** se il requisito è desiderabile;
  - **2** se il requisito è opzionale.
- **Tipo** può assumere i seguenti valori:
  - **F** se il requisito è funzionale;
  - **Q** se il requisito è di qualità;
  - **P** se il requisito è prestazionale;
  - **V** se il requisito è di vincolo.
- **Codice** identifica univocamente ciascun requisito in modo gerarchico.

Inoltre, per ciascun requisito deve essere indicata:

- **Fonte** dell'individuazione del requisito che può essere:
  - **Capitolato**;
  - **Caso d'uso**;
  - **Interno** (discussioni del gruppo);
- **Descrizione** breve e chiara.

### 3.4.5 Classificazione dei casi d'uso

L'analisi e l'identificazione dei casi d'uso, o use case (UC), deve procedere dal generale al particolare.

Ciascun caso d'uso sarà classificato gerarchicamente con la seguente dicitura:

$$UC[Codice\ del\ padre].[Codice\ identificativo]$$

dove:

- **Codice del padre** rappresenta il codice univoco del relativo caso d'uso padre qualora esistesse, altrimenti è omesso;
- **Codice identificativo** rappresenta il codice univoco e progressivo del corrispondente caso d'uso. Il codice può includere diversi livelli di gerarchia che devono essere separati da un punto.

Inoltre, per ciascun caso d'uso deve essere indicato:

- **Nome** del caso d'uso;
- **Attori** coinvolti;
- **Descrizione** chiara e sufficientemente dettagliata;
- **Precondizione**;
- **Scenario principale degli eventi** che descrive la sequenza dei casi d'uso figli;
- **Scenari alternativi** che descrivono la sequenza di eventuali casi d'uso non appartenenti allo scenario principale;
- **Requisiti** ricavati dal caso d'uso;
- Eventuali **Inclusioni**;
- Eventuali **Estensioni**;
- Eventuali **Generalizzazioni**;
- **Postcondizione**.

Infine, ogni caso d'uso deve essere rappresentato da un grafico UML.

## 3.5 Progettazione

### 3.5.1 Scopo

In questa fase viene descritta una soluzione del problema, soddisfacente per tutti gli stakeholder.

### 3.5.2 Aspettative

I risultati ottenuti in seguito ad una corretta attuazione di tale fase sono:

- definire l'architettura logica di sistema che identifica gli elementi del sistema;
- definire l'architettura logica di sistema in conformità ai requisiti definiti;
- mantenere, tracciare e verificare i requisiti di sistema con l'architettura logica di sistema;
- garantire la qualità attraverso la correttezza per costruzione.

### 3.5.3 Descrizione

Partendo dai requisiti sviluppati nella fase precedente, i *Progettisti* devono sviluppare l'architettura logica del sistema attuando i seguenti obbiettivi:

- utilizzare componenti con specifiche chiare e coese;
- realizzare l'architettura rispettando le risorse e i costi prefissati;
- adottare strutture che si adattino al cambiamento;
- utilizzare componenti riusabili;
- suddividere il sistema fino a quando ogni componente ha una complessità trattabile;
- riconoscere le componenti terminali.

Per definire l'architettura logica del sistema i *Progettisti* utilizzeranno diversi tipi di diagrammi UML 2.0.

### 3.5.4 Diagrammi UML

I diagrammi UML 2.0 che verranno utilizzati nella fase di progettazione sono:

- Diagrammi delle attività;
- Diagrammi delle classi;
- Diagrammi dei package;
- Diagrammi di sequenza.

## 3.6 Codifica

### 3.6.1 Scopo

Lo scopo della fase di codifica è quello di implementare il prodotto costruendo unità software eseguibili che riflettano la struttura definita in fase di progettazione.

### 3.6.2 Aspettative

I risultati ottenuti in seguito ad una corretta attuazione di tale fase sono:

- realizzare le unità software in conformità ai requisiti;
- tracciare le unità software e le relativi componenti dell'architettura logica di sistema;
- definire criteri di verifica delle unità software.

### 3.6.3 Descrizione

Partendo dall'architettura logica di sistema definita nella fase precedente, i *Programmatori* devono implementare le unità software definendo anche i criteri di verifica di tali unità. Per svolgere tali attività devono attenersi ai metodi e agli standard di codifica utilizzati.

### **3.6.4 Standard di codifica**

In questa sottosezione devono essere elencate le regole per l'indentazione delle unità software.

## 4 Processi di supporto

### 4.1 Struttura dei documenti

#### 4.1.1 Descrizione

Di seguito sono riportate le regole che si dovranno seguire per la scrittura dei documenti.

#### 4.1.2 Approvazione dei documenti

Ogni documento deve seguire le seguenti fasi:

- Il documento viene redatto dai vari redattori;
- I *Verificatore* procedono al controllo di eventuali errori o imprecisioni;
- in caso di errori i *Verificatore* procedono a segnalarli ai redattori i quali provvedono a correggere gli errori ripartendo dalla prima fase;
- Se il documento passa la fase di verifica viene consegnato al *Responsabile di progetto* che decide se approvarlo o meno.  
Nel caso venga rifiutato il *Responsabile di progetto* deve indicare le criticità ai redattori i quali provvedono a correggere gli errori ripartendo dalla prima fase.

### 4.2 Template

Per rendere omogenea e semplice la stesura dei documenti è stato creato un template in L<sup>A</sup>T<sub>E</sub>X che rispetta le regole stilistiche riportate in questo documento. Questo template è condiviso tramite github nel repository documents\templates.

### 4.3 Codifica e convenzioni

Tutti i file devono seguire la convenzione UTF-8 per la codifica dei caratteri e LF (U+000A) per andare a capo. Per indicare il nome di variabili, classi e funzioni si deve usare la lingua inglese.

### 4.4 Intestazione

Tutti i file di documentazione devono avere la seguente intestazione:

```
%Document-Author: Cognome Nome + Cognome Nome + ...  
%Document-Date: GG/MM/AAAA  
%Document-Description: Descrizione documento
```

indicando Cognome e nome dei vari autori, data di creazione e breve descrizione del documento.

Per quanto riguarda i file contenenti codice si deve:

- Usare le seguente intestazione all'inizio di ogni file:

```
/*  
* File-Name: Nome del file  
* File-Author: Cognome Nome dell'autore  
* File-Date: Data di creazione  
* File-Summary: Breve descrizione del file  
* File-Description: Descrizione dettagliata del file  
**/
```

- Prima di ogni classe scrivere un commento con la seguente struttura:

```
/*  
Class-Name: Nome della classe  
File-Summary: Breve descrizione della classe  
**/
```

- Per ogni metodo si deve scrivere un commento così strutturato:

```
/*  
Method-Name: Nome della classe  
Method-Summary: Breve descrizione della classe  
Method-Input: breve descrizione dei parametri della funzione nel caso ci siano  
Method-Output: breve descrizione dei valori di ritorno nel caso ci siano  
**/
```

## 4.5 Struttura dei documenti

### 4.5.1 Prima pagina

La prima pagina di ogni documento riporta i seguenti campi:

- Nome dell' università
- Nome del gruppo;
- Nome del progetto;
- Descrizione Progetto;
- Titolo documento con versione;
- Logo del gruppo;
- Versione;
- Data redazione
- Cognome e nome dei redattori del documento;
- Cognome e nome dei *Verificatori* del documento;



- Cognome e nome del *Responsabile di progetto* del documento;
- Uso;
- Lista di distribuzione;
- Versione del documento;
- Email del gruppo.

#### 4.5.2 Diario delle modifiche

Subito dopo la prima pagina è presente una tabella che riassume le modifiche apportate al documento ordinate in modo decrescente rispetto alla data. Per ogni riga viene indicato:

- Versione del documento;
- Data della modifica;
- Cognome Nome di chi ha apportato la modifica;
- Descrizione della modifica.

#### 4.5.3 Indici

In ogni documento viene usato un indice per le sezioni e, nel caso siano presenti, anche per grafici e figure.

#### 4.5.4 Struttura generale delle pagine

Nell' intestazione della pagine sono presenti:

- Logo del gruppo;
- Nome del documento.

A piè pagina si possono trovare:

- Nome del gruppo;
- Nome del progetto
- Pagina corrente e pagine totali indicate come "Pagina X di Y" dove X indica la pagina attuale e Y le pagine totali;

### 4.6 Norme tipografiche

In questa sezione vengono descritte le regole di tipografia e ortografia comuni per tutti i documenti

#### 4.6.1 Punteggiatura

- **Parentesi:** il testo racchiuso tra parentesi non deve iniziare o finire con spaziature inoltre alla fine non devono esserci presenti caratteri di punteggiatura;
- **Punteggiatura:** i caratteri di punteggiatura non devono mai essere preceduti da spaziatura;
- **Maiuscole:** l'iniziale maiuscola viene usata per il nome del team, del progetto, dei documenti, dei ruoli, delle varie fasi di lavoro e per le parole Proponente e Committente.  
Inoltre viene usata negli elenchi puntati e nei casi indicati dalla lingua italiana.

#### 4.6.2 Stile di testo

- **Corsivo:** usato nei seguenti casi:
  - **Citazioni:** viene usato il corsivo quando una frase viene citata;
  - **Abbreviazioni:** viene usato per evidenziare abbreviazioni;
  - **Nomi particolari:** i nomi di figure particolari come *Progettista* o *Analista*;
  - **Documenti:** viene usato il corsivo per i nomi dei vari documenti.
- **Grassetto:** usato nei seguenti casi:
  - **Elenchi puntati:** viene usato il grassetto per parole o frasi chiave all'interno di un elenco.

viene inoltre usato per parole o particolari passaggi importanti;

- **Maiuscolo:** usato soltanto per acronimi o eventuali macro  $\text{\LaTeX}$  nei documenti;
- **$\text{\LaTeX}$ :** per ogni riferimento a  $\text{\LaTeX}$  bisogna utilizzare il comando  $\text{\LaTeX}$ .

#### 4.6.3 Composizione del testo

- **Elenchi puntati:** prima lettera minuscola (salvo casi precedenti) e devono terminare con punto e virgola, tranne l'ultimo elemento che terminerà con il punto;
- **Note a piè pagina:** prima lettera maiuscola e devono terminare con il punto.

#### 4.6.4 Formati

- **Numeri:** uso dello standard SI/ISO 31-10;
- **Percorsi:** deve essere usato il comando  $\text{\LaTeX}$   $\text{\url}$  per indirizzi web o mail;
- **Ore:** le ore devono seguire lo standard ISO 8601 quindi espresse come:

HH:MM

dove HH indica l'ora espressa tramite 2 cifre (0-23) e MM indica i minuti espressi sempre con 2 cifre (0-59);

- **Date:** le date devono seguire lo standard ISO 8601 quindi saranno espresse come:

GG/MM/AAAA

dove AAAA indica 4 cifre per l'anno, MM indica 2 cifre per il mese e GG indica 2 cifre per il giorno;

#### 4.6.5 Riferimenti Vari

Per i vari riferimenti si devono usare i seguenti comandi  $\LaTeX$ : (che garantiscono la corretta scrittura, con la prima lettera di ogni parola che non sia una preposizione maiuscola).

- **Ruoli:** per i ruoli si deve usare il comando `\role{Nome del ruolo}` come riportato di seguito:
  - `\responsabileprogetto = Responsabile di progetto;`
  - `\amministratore = Amministratore;`
  - `\analista = Analista;`
  - `\proponente = Analista;`
  - `\progettista`
  - `\programmatore = Programmatore;`
  - `\verificatore = Verificatore;`
  - `\segretario = Segretario;`
  - `\amministratori = Amministratori;`
  - `\analisti = Analisti;`
  - `\progettisti = Progettisti;`
  - `\programmatori = Programmatori;`
  - `\verificatori = Verificatori;`
  - `\segretari = Segretari.`
- **Documenti:** per i documenti si deve usare il comando `\doc{Nome del documento}` come riportato di seguito:
  - `\pianodiprogetto = Piano di progetto;`
  - `\pianodiquifica = Piano di qualifica;`
  - `\normediprogetto = Norme di progetto;`
  - `\studiodifattibilita = Studio di fattibilità;`
  - `\analisierequisiti = Analisi dei requisiti;`
  - `\specificatecnica = Specifica tecnica;`

- \definizionediprodotto=*Definizione di prodotto*;
  - \manualeutente=*Manuale utente*;
  - \glossario=*Glossario*.
- **Documenti con versione:** per i documenti con versione si deve usare il comando \doc{Nome del documentov} come riportato di seguito:
    - \pianodiprogettov =*Piano di progetto v1.0.0*;
    - \pianodiqualficav=*Piano di qualifica v1.0.0*;
    - \normediprogettov=*Norme di progetto v1.0.0*;
    - \studiodifattibilitav=*Studio di fattibilità v1.0.0*;
    - \analisierequisitiv=*Analisi dei requisiti v1.0.0*;
    - \specificatecnicav=*Specifica tecnica v1.0.0*;
    - \definizionediprodotto=*Definizione di prodotto v1.0.0*;
    - \manualeutentev=*Manuale utente*;
    - \glossariov=*Glossario v1.0.0*.
  - **Revisione:** per le revisioni si deve usare il comando \rev{Nome Revisione} come riportato di seguito:
    - \revisionedeirequisiti=**Revisione dei Requisiti**;
    - \revisionediaccettazione=**Revisione di Accettazione**;
    - \revisionediprogettazione=**Revisione di Progettazione**;
    - \revisionediqualfica=**Revisione di Qualifica**.
  - **Nome del gruppo:** per il nome del gruppo definito come "KaleidosCode" si dovrà usare \KaleidosCode;
  - **Nome del Proponente:** per riferirsi al Proponente come "Zucchetti s.p.a." si dovrà usare \proponente;
  - **Nome del Committente:** per riferirsi al Committente come "Prof. Vardanega Tullio" si dovrà usare \vardanega;
  - **Nome del progetto:** per riferirsi al proponente come "SWEDesigner". si dovrà usare \progetto.

Inoltre i nomi di file senza percorso completo si devono scrivere usando il formato monospace e per scrivere nomi dei componenti si deve usare il formato "Cognome Nome".

#### 4.6.6 Sigle

per rendere più accessibili tabelle e diagrammi si devono usare (dove necessario) le seguenti sigle:

- **AdR:***Analisi dei requisiti*;
- **GL:***Glossario*;
- **NdP:***Norme di progetto*;
- **PdP:***Piano di progetto*;
- **PdQ:***Piano di qualifica*;
- **SdF:***Norme di progetto*;
- **ST:***Specifiche tecniche*;
- **RA:***Revisione di Accettazione*;
- **RP:***Revisione di Progettazione*;
- **RQ:***Revisione di Qualifica*;
- **RR:***Revisione dei Requisiti*.

### 4.7 Componenti grafiche

#### 4.7.1 Tabelle

Ogni tabella deve essere accompagnata da una descrizione che specifichi anche l'indice ad essa associata per renderla tracciabile all'interno del documento.

#### 4.7.2 Immagini

Le immagini devono essere convertite in pdf prima di essere incorporate nei documenti e ciascuna deve essere accompagnata da una descrizione e l'indice ad essa associata.

### 4.8 Classificazione documenti

#### 4.8.1 Documenti formali

Un documento è ritenuto formale solo dopo essere stato approvato dal *Responsabile di progetto*, dopo l'approvazione viene considerato pronto per la revisione da parte del Committente.

#### 4.8.2 Documenti informali

Un documento è ritenuto informale fino all'approvazione del *Responsabile di progetto* di Progetto, fino a quel momento è considerato ad uso interno.

## 4.9 Versionamento

Ogni documento deve essere accompagnato dal numero della versione attuale così formato:

vX.Y.Z

dove:

- **X**: numero di uscite formali del documento e aumenta a ogni approvazione da parte del *Responsabile di progetto*;
- **Y**: usato per modifiche sostanziali e verifiche;
- **Z**: usato per indicare modifiche minori.

Per riferirsi a una specifica versione del documento si deve usare il seguente formato:

*Nome Documento vX.Y.Z*

mentre il nome da applicare ai file è:

NomeDocumento\_vX.Y.Z.pdf

### 4.9.1 Variazione indici

Le variazione degli indici avvengono da parte di:

- **X**: deve essere aggiornata dal *Responsabile di progetto* dopo la sua approvazione;
- **Y**: deve essere incrementato da chi esegue la modifica o la verifica;
- **Z**: deve essere incrementato da chi esegue la modifica.

queste modifiche sono mostrate dal diario delle modifiche.

## 4.10 Verifica

### 4.10.1 Scopo del processo

Scopo della verifica è quello di accertare che i documenti prodotti non contengano errori e che rispettino le regole riportate in questo documento.

## 4.11 Attività

### 4.11.1 Analisi Statica

Una tecnica di verifica che si applica al codice sorgente e alla documentazione e non richiede l'esecuzione del prodotto software. Le principali strategie sono due:

**Inspection:** questa tecnica si applica quando si conoscono le problematiche oggetto di verifica. Consiste in una ricerca mirata attraverso il documento per l'identificazione dei problemi, spesso basandosi su una lista di errori precedentemente stilata.

**Walkthrough:** questa tecnica consiste nello scorrere l'intero documento alla ricerca di errori generici. Viene applicata soprattutto nelle prime fasi di stesura del documento o di codifica a causa dell'impegno necessario per attuarla, che aumenta notevolmente con l'aumentare delle dimensioni del documento da verificare.

#### 4.11.2 Analisi Dinamica

Questa tecnica può essere applicata solo al software, nella sua interezza o solo in parte. Consiste nell'usare una serie di test automatici creati dal gruppo che mettano alla prova il software prodotto e restituiscano risultati relativi a eventuali problemi rilevati. I test devono essere ripetibili durante l'intero ciclo di vita e devono coprire un insieme finito di casi, con valori di ingresso, uno stato iniziale e esito decidibile.

### 4.12 Validazione

#### 4.12.1 Scopo del processo

La validazione si occupa di accertare che il prodotto realizzato sia conforme alle attese.

#### 4.12.2 Attività

Di questa fase si occupa il *Responsabile di progetto* il quale deve verificare che i documenti e il codice sorgente rispettino a pieno i requisiti imposti dal Committente e dal proponente. Il *Responsabile di progetto* una volta che i documenti ottengono l'approvazione dei *Verificatori* controlla eventuali errori o mancanze. Nel caso se ne identifichino, si deve segnalare le criticità ai redattori che provvederanno a correggerle.

### 4.13 Issue

#### 4.13.1 Issue Tracking

L'issue tracking è un'attività che permette di tenere traccia di tutti gli errori riscontrati durante la verifica dei documenti e del software da parte dei *Verificatori*.

#### 4.13.2 Gestione delle issue

Nel caso un *Verificatore* riscontri delle anomalie deve seguire la seguente procedura:

- il *Verificatore* deve aprire una nuova issue con la descrizione del problema;
- il *Responsabile di progetto* deve verificare questa issue e provvede ad assegnarla a uno dei redattori del documento o al programmatore in caso di un errore relativo al codice;
- una volta risolta e verificata viene segnata come risolta da parte del *Responsabile di progetto*.

### 4.13.3 Strumenti per l'issue tracking

Per tracciare gli issue si deve usare il servizio Issues messo a disposizione da GitHub.

### 4.13.4 Verifica ortografica

Per eseguire una prima verifica automatica viene usato il controllo ortografico Language-Tool versione 3.6, integrabile in TexStudio.

## 4.14 Consigli di utilizzo

### 4.14.1 Visualizzazione dei Task

Per vedere i Task assegnati basta spostarsi nella pagina “MY TASKS” presente tra i link in alto a sinistra.

Per avere una visione d'insieme sui Task dell'intero gruppo basta spostarsi nella pagina “Team Calendar” presente nella tendina laterale. Da qui è possibile vedere le scadenze indicative disposte in un calendario.

### 4.14.2 Visualizzazione conversazioni

Per vedere le conversazioni all'interno del gruppo, è sufficiente andare in “Team Conversation” dalla tendina laterale. Da qui è possibile anche iniziare una nuova conversazione, che verrà notificata a tutti i membri del gruppo.



## 5 Processi organizzativi

### 5.1 Scopo

Lo scopo del processo è quello di produrre il *Piano di progetto v1.0.0*, al fine di pianificare e gestire i ruoli che i membri dovranno assumere nella dinamica della progettazione di *SWEDesigner*.

### 5.2 Aspettative

I risultati ottenuti in seguito ad una corretta attuazione di tale processo sono:

- definire i ruoli dei membri del gruppo;
- pianificare e calendarizzare l'esecuzione dei compiti programmati;
- produrre il *Piano di progetto*.

### 5.3 Ruoli di progetto

L'assegnazione dei ruoli viene pianificata all'interno del *Piano di progetto*. Le ore di lavoro devono essere distribuite in modo più possibile omogeneo tra i membri del gruppo. Ogni membro deve inoltre ricoprire ciascun ruolo almeno una volta

#### 5.3.1 Responsabile di progetto

Il *Responsabile di progetto* è il portavoce nonché il responsabile delle scelte del gruppo. In particolare si occupa di:

- approvare l'offerta economica;
- gestire le risorse;
- pianificare e coordinare le attività;
- analizzare e gestire i rischi;
- approvare i documenti;
- assicurarsi che vengano rispettate le *Norme di progetto*;
- assicurarsi che vengano rispettate le pianificazioni definite nel *Piano di progetto*.

#### 5.3.2 Amministratore

L'*Amministratore* è colui che si occupa di gestire l'efficienza dell'ambiente di lavoro. In particolare si occupa di:

- studiare e fornire strumenti che migliorano l'ambiente di lavoro;
- gestire l'archiviazione, il versionamento e la configurazione dei documenti e del software;

- eliminare o ridurre per quanto possibile le difficoltà nella gestione di processi e di risorse;
- automatizzare il lavoro dove possibile.

### 5.3.3 Analista

L'*Analista* deve identificare e tracciare il dominio del problema. In particolare si occupa di:

- trasformare le richieste del cliente in specifiche per il prodotto;
- elaborare le specifiche nell'*Analisi dei requisiti* e nello *Studio di fattibilità*.

### 5.3.4 Progettista

Il principale ambito di lavoro del *Progettista* è lo stack tecnologico. In particolare si occupa di:

- indicare le tecnologie più idonee allo sviluppo del progetto;
- descrivere il funzionamento del sistema e progettare l'architettura;
- produrre una soluzione ammissibile in termini di risorse impiegate.

### 5.3.5 Programmatore

Il *Programmatore* è l'addetto alla codifica. In particolare si occupa di:

- implementare le soluzioni indicate dal *Progettista*;
- scrivere codice opportunamente commentato, versionato e mantenibile in accordo al documento *Norme di progetto*;
- stilare la documentazione del codice;
- realizzare e fornire gli strumenti per la verifica e la validazione del prodotto.

### 5.3.6 Verificatore

Il *Verificatore* è l'addetto a tutte le attività di verifica. In particolare si occupa di:

- controllare che le regole stabilite dalle *Norme di progetto* siano rispettate durante ogni attività di progetto.

## 5.4 Comunicazioni

### 5.4.1 Esterne

Per le comunicazioni esterne è stata creata la casella di posta elettronica:

[kaleidos.codec6@gmail.com](mailto:kaleidos.codec6@gmail.com)

Tale indirizzo deve essere l'unico canale di comunicazione tra il gruppo di lavoro e l'esterno. Il *Responsabile di progetto* è l'unico ad accedere all'indirizzo ed è quindi l'unico a poter comunicare con il committente del progetto. È compito del *Responsabile di progetto* informare i membri del gruppo delle discussioni avvenute e, qualora fosse necessario, inoltrare loro il messaggio attraverso una mailing list.

### 5.4.2 Interne

Per le comunicazioni interne viene utilizzato il sistema di comunicazione offerto in Asana. Tale sistema deve essere utilizzato dai membri del gruppo per comunicare tra loro. Tutte le conversazioni devono avere come destinatario l'indirizzo "Kaleidos Code". In questo modo, ogni componente è costantemente informato sullo scambio di informazioni interne. Qualora fosse necessario l'uso di e-mail, come ad esempio nel caso di un inoltro di messaggio da parte del *Responsabile di progetto*, è stata creata una mailing list:

[kaleidos.code@group.com](mailto:kaleidos.code@group.com)

Per facilitare le comunicazioni tra i membri del gruppo, viene utilizzato anche il sistema di messaggistica e videoconferenza Google Hangout. L'uso di quest'ultimo, nel caso in cui vengano prese decisioni o emergano contenuti utili allo sviluppo del progetto, comporta l'obbligo di redigere un verbale da parte di un membro del gruppo, che pubblicherà attraverso il sistema di comunicazione di Asana e ne salverà una copia in Google Drive una volta terminata la conversazione. La verbalizzazione ha l'obiettivo di tenere traccia di ogni argomento discusso, in quanto una comunicazione verbale non documentata non è accettabile per il corretto svolgimento del progetto.

Per una comunicazione istantanea è utilizzato anche il sistema di messaggistica Telegram. Si richiede che la conversazione venga documentata come sopra descritto.

## 5.5 Composizione e-mail e conversazioni

In questo paragrafo viene descritta la struttura che deve avere un messaggio sia per una comunicazione esterna che per una conversazione interna attraverso il servizio offerto in Asana e mailing list.

### 5.5.1 Destinatario

- **Interno - Asana:** l'unico indirizzo utilizzabile è il nome del gruppo: Kaleidos Code;
- **Interno - e-mail:** l'unico indirizzo utilizzabile è [kaleidos.codec6@gmail.com](mailto:kaleidos.codec6@gmail.com);
- **Esterno:** varia a seconda che ci si debba rivolgere al proponente, al *Prof. Vardanega Tullio* o al *Prof. Cardin Riccardo*.

### 5.5.2 Mittente

- **Interno - Asana:** è rappresentato automaticamente dallo username del creatore della conversazione;
- **Interno - e-mail:** l'indirizzo di chi scrive il messaggio;
- **Esterno:** l'unico indirizzo utilizzabile è [kaleidos.codec6@gmail.com](mailto:kaleidos.codec6@gmail.com) e deve essere usato solamente dal *Responsabile di progetto*.

### 5.5.3 Oggetto

L'oggetto deve essere chiaro ed esaustivo, possibilmente non confondibile con altri preesistenti.

L'oggetto di una comunicazione, una volta avviata, non deve mai essere cambiato.

Solamente per le e-mail, nel caso si debba comporre un messaggio di risposta vi è l'obbligo di aggiungere la particella "Re:" all'inizio dell'oggetto per poter distinguere i livelli di risposta; se si dovesse trattare di un inoltro, si deve usare invece la particella "I:".

### 5.5.4 Corpo

Il corpo di un messaggio deve contenere tutte le informazioni necessarie alla piena comprensione della comunicazione.

Nel caso di e-mail, se alcune parti del messaggio hanno uno o più destinatari specifici, sarà necessario aggiungere il loro nome prima del relativo paragrafo attraverso la segnatura *@destinatario*; in Asana invece, si dovrà menzionare lo specifico destinatario attraverso l'apposita funzionalità alla creazione del messaggio.

Solamente per le e-mail, in caso di risposta o inoltro del messaggio, il contenuto aggiunto deve essere sempre collocato in testa. Si consiglia di non cancellare il resto del messaggio, per consentire una visione completa della discussione.

### 5.5.5 Allegati

Qualora fosse necessario, è permesso l'invio di allegati.

## 5.6 Riunioni

### 5.6.1 Frequenza

Le riunioni interne del gruppo di lavoro avranno una cadenza settimanale.

### 5.6.2 Convocazione riunione

#### Interna

Il *Responsabile di progetto* ha il compito di convocare le riunioni generali interne, ossia le riunioni a cui sono tenuti a partecipare tutti i membri del gruppo. Se un componente qualsiasi ritiene necessaria la convocazione di una riunione, deve inoltrare la richiesta al responsabile il quale decide se respingerla o accettarla. Il *Responsabile di progetto* deve convocare l'assemblea con almeno quattro giorni di preavviso attraverso l'invio di una richiesta nella sezione Comunicazioni di Asana, il cui corpo è formato da:

- Oggetto: Convocazione riunione n. X (dove X indica il numero progressivo di riunioni effettuate)
- Corpo:
  - Data e ora prevista
  - Luogo previsto
  - Ordine del giorno

Ogni componente deve rispondere al messaggio nel minor tempo possibile confermando la presenza o in caso contrario motivando l'eventuale assenza. In caso di manca risposta il *Responsabile di progetto* è obbligato a contattare direttamente il componente che non ha fornito una risposta. Una volta ricevute tutte le risposte, il *Responsabile di progetto* può decidere se confermare, rinviare o annullare la riunione, basandosi sul numero di presenze e/o assenze, per garantire il maggior numero possibile di presenti. La conferma, il rinvio, l'annullamento devono essere notificati tramite messaggio nella sezione Comunicazioni di cui sopra.

### **Esterna**

Il *Responsabile di progetto* ha il compito di convocare le riunioni generali esterne, ossia le riunioni a cui sono convocati tutti i membri del gruppo e il proponente e/o committente. Se un componente qualsiasi ritiene necessaria la convocazione di una riunione, deve inoltrare la richiesta al *Responsabile di progetto* il quale decide se respingere o accettare tale richiesta. Il *Responsabile di progetto* deve prima di tutto concordare con il proponente e/o committente delle date e dei luoghi possibili in cui svolgere la riunione attraverso l'invio di un'email contenente:

- Oggetto: Richiesta riunione
- Corpo
  - Motivazione
  - Eventuali date e/o luoghi possibili

Dopo aver concordato delle date possibili, il responsabile deve inviare un messaggio a i membri del gruppo nella sezione Comunicazioni in Asana in cui sono specificate:

- Oggetto: Richiesta riunione col proponente
- Corpo
  - Date e/o luoghi possibili

Ogni membro del gruppo deve rispondere alla comunicazione confermando la presenza o in caso contrario motivando l'eventuale assenza. In caso di mancata risposta, il *Responsabile di progetto* deve contattare direttamente il componente che non ha fornito una risposta. Una volta ricevute tutte le risposte, il responsabile può decidere se confermare, rinviare, annullare la riunione con il proponente basandosi sul numero di presenze e /o assenze. Il *Responsabile di progetto* deve poi, in caso di:

- conferma: comunicare a tutti i membri e al proponente e/o committente la data e il luogo definitivi;
- rinvio: comunicare a tutti i membri la decisione presa e ripetere il procedimento dall'inizio;
- annullamento: comunicare a tutti i membri la decisione presa e ripetere il procedimento dall'inizio.

### 5.6.3 Svolgimento riunione

#### Esterna

All'inizio di ogni riunione, verificata la presenza dei membri previsti, viene scelto un segretario che avrà il compito di annotare gli argomenti trattati e di redigere il verbale della riunione, che dovrà poi essere inviato ai restanti membri del gruppo. Tutti i partecipanti devono tenere un comportamento consono al miglior svolgimento dell'assemblea e al raggiungimento degli obiettivi della stessa. Il segretario deve inoltre assicurarsi che venga seguito l'ordine del giorno in modo da affrontare ogni argomento previsto.

### 5.6.4 Verbale

#### Riunione interna

Il verbale di Riunione interna è un documento informale che permette di tenere traccia degli argomenti discussi in ogni riunione. Il segretario, scelto a rotazione, ha il compito di redigere questo documento. Il verbale prodotto deve poi essere condiviso con tutti i membri del gruppo mediante Google Drive, in modo da rendere sempre disponibile e accessibile il contenuto dello stesso.

#### Riunione esterna

In caso di riunione esterna con il proponente e/o committente, il verbale è un documento ufficiale che può avere valore normativo e quindi deve essere redatto seguendo criteri specifici. Per agevolare la scrittura di tale documento è stato preparato un template L<sup>A</sup>T<sub>E</sub>X che ne definisce la struttura e ne organizza i contenuti. È obbligatorio quindi seguire il sopraccitato schema per creare il verbale, inviare il verbale prodotto come allegato al proponente e/o committente in risposta all'email della riunione e condividere il verbale prodotto con tutti i membri del gruppo attraverso Google Drive.

## 5.7 Strumenti di coordinamento

Come piattaforma di gestione del gruppo è stato scelto **Asana**. Asana fornisce:

- Un sistema di gestione dei task;
- Un calendario per organizzare i compiti;
- La visualizzazione del repository associato al progetto;
- Un sistema di rendicontazione del tempo;

- La possibilità di integrare Google Drive, GitHub e web app come Instagant.

Sono state valutate diverse alternative ma, dopo un'attenta fase di test, nessuna di queste è stata ritenuta all'altezza di Asana per quantità e qualità delle caratteristiche proposte. Sono stati provati i seguenti software:

- **Wrike:** scartato perché la versione free gestisce solo fino a cinque utenti;
- **Trello:** scartato perché carente in funzionalità rispetto alle alternative;
- **Teamwork:** scartato perché pur raggiungendo la completezza di Asana in quanto a caratteristiche non fornisce un'interfaccia altrettanto immediata, aumentando il tempo speso dal team nell'apprendimento dell'uso degli strumenti.

## 5.8 Strumenti di versionamento

Sono stati presi in considerazione diversi software di versionamento (Git, SVN, Mercurial) prima di decidere di usare **Git**. I motivi principali della scelta sono:

- **Flessibilità:** Git è un repository distribuito con la possibilità di *commit* e *revert* locali;
- **Esperienza del team:** Git è già stato usato da alcuni componenti del gruppo KaleidosCode.

## 5.9 Rischi

Il *Responsabile di progetto* deve monitorare i rischi indicati nel *Piano di progetto*. Qualora ne identifichi di nuovi deve:

- mettere al corrente il team;
- elaborare una nuova strategia per la gestione dei nuovi rischi;
- aggiornare il documento *Piano di progetto*.

## 5.10 Altri strumenti

### 5.10.1 Google Drive

In questo repository vengono messi solo file che non necessitano di controllo di versione. Questo repository conterrà:

- Documenti che sono recuperabili da altre fonti (es. internet);
- File di installazione dei software utilizzati dal gruppo. In questo modo si punta a garantire l'uso della stessa versione software per ogni componente del gruppo;
- I manuali (software, libri, PDF G ) di consultazione, così da avere uniformità di versione e di informazione;

La possibilità di installare Drive sul proprio PC dà modo ad ogni componente del gruppo di avere a disposizione documentazione e software anche offline. Google Drive viene utilizzato come strumento di supporto allo sviluppo della documentazione e del software presente su Git.

### 5.10.2 ProjectLibre

Per pianificare le attività legate allo sviluppo del progetto e la gestione delle risorse si è scelto di utilizzare **ProjectLibre**. ProjectLibre è un programma open source per il project management. È basato su Java ed è quindi eseguibile su ogni sistema operativo. È il successore ufficiale di OpenProj. Tale software è stato scelto in quanto:

- È portabile, essendo basato su Java;
- È open-source;
- Genera diagrammi Gantt;
- Genera automaticamente diagrammi Program Evaluation and Review Technique (PERT) a partire dal Gantt;
- Genera automaticamente la Work Breakdown Structure (WBS), a partire dal Gantt con allocazione di risorse;
- Calcola i parametri Schedule Variance(SV) e Budget Variance (BV);
- Permette il salvataggio su file XML: essendo un file testuale è possibile fare il merge dei file in caso di conflitti su repository.

### 5.10.3 L<sup>A</sup>T<sub>E</sub>X

Per la stesura dei documenti è stato scelto di utilizzare il linguaggio L<sup>A</sup>T<sub>E</sub>X. La scelta è stata quasi obbligata in quanto L<sup>A</sup>T<sub>E</sub>X permette di separare il contenuto dalla formattazione, permettendo un versionamento più semplice e consentendo di definire template condivisi per uniformare i documenti. Altre soluzioni come Microsoft Office, LibreOffice o Google Docs non avrebbero consentito un livello altrettanto elevato di separazione risultando in un aumento del lavoro da parte dei membri del gruppo e una maggior difficoltà di uniformazione dei contenuti. Per la scrittura di documenti L<sup>A</sup>T<sub>E</sub>X l'editor consigliato è **TeXstudio**.